



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

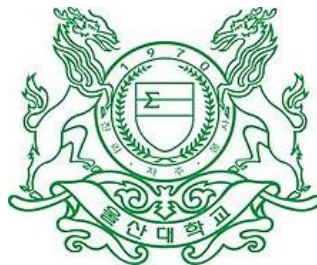
저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

**DOCTOR OF PHILOSOPHY**

**DESIGN AND IMPLEMENTATION OF  
MULTIHOP REAL-TIME LORA PROTOCOL**



**The Graduate School  
of the University of Ulsan  
Department of Electrical, Electronic,  
and Computer Engineering**

**TRAN HUU PHI**

**DESIGN AND IMPLEMENTATION OF MULTIHOP  
REAL-TIME LORA PROTOCOL**

**Supervisor: Professor OH HOON**

**A Dissertation**

**Submitted to  
the Graduate School of the University of Ulsan  
In Partial Fulfillment of the Requirements  
for the Degree of**

**DOCTOR OF PHILOSOPHY**

**by**

**TRAN HUU PHI**

**Department Electrical, Electronic and Computer Engineering  
University of Ulsan, Korea  
July 2022**

# Design and Implementation of Multihop Real-Time Lora Protocol

This certifies that  
the dissertation of TRAN HUU PHI is approved

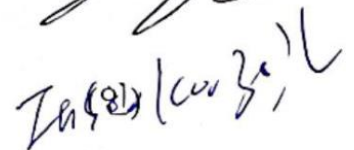
Committee Chair Prof. Yoon, Seokhoon




Committee Member Prof. Oh, Hoon (Advisor)



Committee Member Prof. Koo, In-Soo



Committee Member Prof. Chung, Jin-Ho



Committee Member Dr. Jung, Woo-Sung



**Department Electrical, Electronic and Computer Engineering**  
**University of Ulsan, Korea**  
**July 2022**

## Abstract

Recently, LoRa (Long Range) technology has been attracting attention in industrial monitoring applications due to its long communication distance and high link reliability. This dissertation aims at designing a new multi-hop LoRa protocol to provide real-time and reliable data transmission as well as cover a sufficient large area in industrial applications. The protocol constructs and maintains a two-hop tree topology to extend the network coverage to *wireless unfriendly zones* such as underground tunnels, enclosed spaces, and harsh construction houses with steel or concrete obstructions. Then, based on the two-hop network topology, a TDMA-based real-time slot schedule is applied to remove data collision as well as guarantee the time constraint of every data transmission. Furthermore, the proposed protocol also addresses technical aspects of dynamic networks such as the auto-configuration of multi-hop LoRa networks, dynamic topology management, and updating of slot schedules.

The protocol was evaluated through analysis, simulation, and various real site experiment scenarios. It can be seen by analysis that the proposed protocol can support a densely deployed network of hundreds to thousands of nodes. Furthermore, the simulation and experiment results show that the protocol could achieve high reliability on data transmission as well as generate low maintaining overhead against the node mobility.

## **Acknowledgments**

This doctoral dissertation cannot be done without help from others. Therefore, I would like to express my gratitude to a number of people who have made all of the work becomes possible.

First and foremost, I would like to thank my supervisor, Professor OH HOON, who inspired and encouraged me. During my PhD period in Korea, he has provided me with a lot of support both in research and in life. I have gained a lot of experience in both doing research and real projects under his instructions.

Furthermore, I would like to thank all the respected professors in the Department of Electrical, Electronic, and Computer Engineering who gave me a lot of invaluable knowledge through their lectures.

Also, I would also like to thank the office staffs of BK21, our department, and graduate school who gave me a lot of help and support during my studying period.

Last, but not least, I would like to thank my family who are always by my side encouraging and trusting me unconditionally. Especially, thank my wife who took care of my life, and my little girl who was born recently and gave me a lot of motivation and happiness.

**Tran Huu Phi**

University of Ulsan, Ulsan, South Korea

July 2022

# Contents

Abstract.....	i
Acknowledgments.....	ii
Contents .....	iii
List of figures.....	vi
List of tables.....	viii
List of abbreviations .....	ix
Chapter 1. INTRODUCTION .....	1
1.1. LoRa and LoRaWAN .....	1
1.1.1. LoRa Physical Layer .....	1
1.1.2. LoRaWAN .....	3
1.2. Multi-hop Real-time LoRa protocol .....	6
1.2.1. Problems and Issues .....	6
1.2.2. Previous approaches.....	9
1.3. Our Approach .....	13
1.4. Dissertation Contribution and Structure .....	15
Chapter 2. PREVIOUS WORKS.....	17
2.1. Industrial Wireless Sensor Networks.....	17
2.2. Industrial LoRa Protocols .....	19
2.2.1. I-LoRa and RT-LoRa .....	19
2.2.2. RT-LoRa-LSI.....	21
2.3. Multi-hop LoRa Protocols .....	23
2.3.1. LoRaBlink.....	24
2.3.2. CT-LoRa .....	25
2.3.3. LoRa-Mesh.....	26
2.3.4. Syn-LoRa-Mesh .....	27
Chapter 3. MULTIHOP REAL-TIME LORA PROTOCOL .....	30

3.1. Network model .....	30
3.2. Motivation.....	31
3.3. Notation and Message.....	34
3.4. Protocol Design .....	35
3.4.1. Protocol Structure and Frame-Slot Architecture.....	35
3.4.2. Tree Construction and Maintenance .....	36
3.4.3. Time Synchronization .....	38
3.4.4. Slot Scheduling .....	39
3.4.5. Data transmission .....	42
Chapter 4. MULTI-HOP REAL-TIME LORA PROTOCOL FOR DYNAMIC NETWORKS .....	47
4.1. Problem Identification .....	47
4.2. Notations and Messages .....	50
4.3. Protocol Design .....	51
4.3.1. Protocol Structure.....	51
4.3.2. Network Initialization .....	52
4.3.3. Scheduling Information Dissemination and Slot Scheduling .....	54
4.3.4. Data transmission and Topology Maintenance .....	56
Chapter 5. PERFORMANCE EVALUATION.....	61
5.1. Analysis .....	61
5.1.1. Comparison of Characteristics for multi-hop LoRa Protocols.....	61
5.1.2. Number of Supportable Nodes.....	64
5.1.3. Energy Consumption.....	67
5.2. Simulation.....	68
5.2.1. Channel model .....	68
5.2.2. Simulation setup.....	69
5.2.3. Results and Discussions .....	70
5.3. Experiment.....	72
5.3.1. The effectiveness of two-hop LoRa network.....	72
5.3.2. Dynamic experiment scenarios .....	75
Chapter 6. CONCLUSIONS AND FUTURE WORKS.....	80



6.1. Conclusions .....	80
6.2. Future Works .....	81
PUBLICATIONS.....	82
REFERENCES .....	83

## List of figures

Figure 1.1. LoRaWAN network topology .....	4
Figure 1.2. LoRaWAN Class A .....	5
Figure 1.3. LoRaWAN Class B .....	5
Figure 1.4. LoRaWAN Class C .....	6
Figure 2.1. IEEE 802.15.4 MAC super-frame structure .....	18
Figure 2.2. Super-frame structure .....	20
Figure 2.3. A frame-slot architecture of $2^3$ slots with the logical slot indices assigned and the slot scheduling of two tasks A and B .....	23
Figure 2.4. The operation of CT flooding protocol .....	25
Figure 2.5. A node joining and data querying processes .....	27
Figure 2.6. Synchronous LoRa mesh topology .....	28
Figure 3.1. Network Topology.....	30
Figure 3.2. A small LoRa network deployed across multiple buildings on the campus of the University of Ulsan.....	32
Figure 3.3. Protocol Operation.....	35
Figure 3.4. Multi-channel frame-slot architecture .....	36
Figure 3.5. Time synchronization .....	38
Figure 3.6. Two-hop tree slots scheduling using logical slot indices .....	41
Figure 4.1. Link failure and inconsistency of slot schedule due to node mobility .....	47
Figure 4.2. The operational structure of the proposed protocol.....	51
Figure 4.3. Node registration and tree construction process.....	52
Figure 4.4. Distribution of group scheduling information.....	54
Figure 5.1. Number of supportable nodes with varying $\alpha$ value (BW=125, CR=1, SF=7) .....	66

Figure 5.2. Number of supportable nodes for different SFs (BW=125, CR=1, N = 8, $\alpha$ = 0.7).....	66
Figure 5.3. Comparison of energy consumption for one-hop and two-hop transmissions with varying payload sizes (BW=125 kHz, CR=1 and Ptx=13 dBm).....	68
Figure 5.4. Example of spatial distribution with 500 end nodes in an area of 800x800(m <sup>2</sup> ) .....	70
Figure 5.5. The average and PDR and PDR_NoOrphan with different $\lambda$ .....	71
Figure 5.6. The average MobileOH with different $\lambda$ .....	71
Figure 5.7. The testbed deployed on the University of Ulsan .....	73
Figure 5.8. PRRs with 2-hop nodes using different SFs.....	74
Figure 5.9. Photos of GW and some nodes installed .....	75
Figure 5.10. Static node deployment scenario and the change of topology .....	76
Figure 5.11. The change of connections according to node movement (The red-colored dashed lines indicate the unreliable connection) .....	78

## List of tables

Table 1.1. LoRa bit rate and transmission range with different SF settings .....	3
Table 3.1. Comparison of PRRs for one-hop and two-hop transmission in Figure 3.2 .....	32
Table 3.2. Energy consumption subject to the modes of a LoRa SX1276 transceiver	42
Table 4.1. Network Information Table, NIT(i).....	60
Table 5.1. Comparison of the features of multi-hop LoRa protocols .....	62
Table 5.2. The lower bound of the UL slot based on payload (SF7, BW125) .....	65
Table 5.3. Supply current values for a transmitter .....	67
Table 5.4. Simulation parameters .....	70
Table 5.5. Experiment Parameters .....	72
Table 5.6. Experiment results from using SF7 .....	73
Table 5.7. Experiment Parameters and Values .....	75
Table 5.8. Experiment results .....	77
Table 5.9. Experiment results .....	79

## List of abbreviations

ACK	Acknowledgement
CAD	Channel Activity Detection
CAP	Contention Access Period
CE	Capture Effect
CFP	Contention Free Period
CRC	Cyclic Redundancy Check
CS	Children Set
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CT	Concurrent Transmission
CTS	Clear to Send
GSSI	Group Slot Scheduling Information
IMOCS	Industrial Monitoring and Control System
IoT	Internet of Things
JREQ	Join Request
JRES	Join Response
LBT	Listen Before Talk
LOS	Line Of Sight
LPWAN	Low Power Wide Area Network
LSI	Logical Slot Indexing
LSS	Local Slot Schedule
LSSI	Local Slot Scheduling Information
MAC	Medium Access Control
MDPU	MAC Protocol data unit

NSSI	Network Slot Scheduling Information
PDR	Packet Delivery Rate
PER	Packet Error Rate
PF	Task Profile
PRR	Packet Reception Ratio
QoS	Quality of Service
RR	Registration Request
RSSI	Received Signal Strength Indicator
RTS	Request to Send
SD	Slot Demand
SMC-SSMA	Smart Multi-channel SSMA
SNR	Signal to Noise Ratio
SSMA	Slotted Sense Multiple Access
TCR	Tree Construction Request
TDMA	Time Division Multiple Access
TIT	Tree Information Table
TOA	Time On Air
TP	Transmission Period
TSD	Total Slot Demand
uPF	updated Profile
uSSI	updated Slot Scheduling Information
WSN	Wireless Sensor Network
WUZ	Wireless Unfriendly Zone

# Chapter 1.

## INTRODUCTION

This chapter presents the fundamentals of our studies. First, we briefly introduce the LoRa technology and LoRa Wide Area Network (LoRaWAN) protocol. Then, we discuss the problems and issues in designing multi-hop LoRa protocol for industrial monitoring and control systems. After that, we summarize our approach on addressing the problems. The dissertation contribution and structure are given in the last section.

### 1.1. LoRa and LoRaWAN

#### 1.1.1. LoRa Physical Layer

LoRa [1, 2] is developed by Semtech to standardize LPWAN technologies for Internet of Thing (IoT) applications such as smart metering, smart cities, smart agriculture, etc. In general, those applications often require collecting small amounts of data and operate under a low-power since the devices are often battery-powered. LoRa has higher link budget compared to the other wireless technologies such as WiFi, Zigbee, Cellular network, etc., and thus it provides a longer transmission range by using much lower data rate. The LoRa modulation uses Chirp Spread Spectrum (CSS) [3] which varies the frequency linearly over time in order to encode information to achieve relatively low transmission power and robustness from channel degradation mechanisms such as channel noise, multi-path fading, and Doppler effect. A LoRa receiver can decode a signal with 19.5 dB below the noise floor and thus, it enables a very long transmission range. As provided in LoRa specification, a LoRa gateway can cover up to three kilometers in urban areas and fifteen kilometers or more in rural areas with line of sight. LoRa also provide very low power on data communication which

allows the end devices can run for up to 10 years with a battery powered. The current available LoRa transceivers operate on free license radio frequency bands from 137 MHz to 1020 MHz; however, they are often operated in ISM bands (EU: 868 MHz, USA: 915 MHz, and ASIA: 169 MHz and 433 MHz).

LoRa uses spreading factors (SFs) to control the bit rate and allows a tradeoff between the data rate and transmission range. The use of higher SFs lowers the data rate but improves the receiving sensitivity of the LoRa receiver and thus, it improves the transmission range. In the LoRa modulation, the SFs are orthogonal meaning that multiple SFs creates multiple virtual channels, and a GW can demodulate multiple packets that are transmitted using different SFs simultaneously. A LoRa transmission is configured by the following parameters:

- Transmission Power (TP): LoRa allows the setting of TP from -4 to 20 (dBm). However, the hardware implementation often limits the TP from 2 to 20 dBm.
- Channel Frequency (CF): CF is the center frequency that is allowed based on the region of deployment and limited by the hardware.
- Coding Rate (CR): LoRa provides signal protection against interference with integrated Forward Error Correction (FEC) [4]. The CR value can be set from 1 to 4 corresponding to the FEC rate equal to  $\frac{4}{4+CR}$  from 4/5 to 4/8. The use of higher CR makes the signal more robust, but it increases the payload and thus increases the energy consumption.
- Spreading Factor (SF): The value of SF can be selected from 6 to 12 that defines the number of bits per a LoRa symbol. Once the SF is selected, the number of chirps per symbol is  $2^{SF}$ . The increase in the SF by one doubles the symbol time and transmission duration; however, it increases the Signal to Noise Ratio (SNR), thus increases the receiving sensitivity and transmission range.



- Bandwidth (BW): A LoRa network often operates at the frequency bandwidth of 500 kHz, 250 kHz, or 125 kHz. The higher BW provides a higher data rate and thus shorter the packet time on-air; however, it lowers the receiving sensitivity by integrating additional noise.

The relationship among the symbol time ( $T_s$ ), spreading factor and bandwidth can be expressed as follows:

$$T_s = 2^{SF} / BW \quad (1.1)$$

Taking into account the implementation of FEC, the wanted bit rate ( $R_B$ ) can be calculated using the following equation:

$$R_B = SF * \frac{BW}{2^{SF}} * \frac{4}{4 + CR} \quad (1.2)$$

The following Table 1.1 shows the different SF settings (from 7 to 12) on BW = 125 kHz and CR = 1 with the corresponding bit rates, estimated ranges, and the time on air (ToA) for an 11-byte payload packet.

*Table 1.1. LoRa bit rate and transmission range with different SF settings*

<b>SF</b>	<b>Bit Rate</b>	<b>Transmission Range</b>	<b>Time on Air</b>
7	5470 bps	2 km	61 ms
8	3125 bps	4 km	103 ms
9	1760 bps	6 km	185 ms
10	980 bps	8 km	371 ms
11	440 bps	11 km	740 ms
12	290 bps	14 km	1400 ms

### 1.1.2. LoRaWAN

LoRaWAN [2, 5, 6] is MAC standard for LoRa networks that are based on star network topology. It defines the network architecture and the way of channel access for

LoRa end devices based on the LoRa physical layer that enable long range and low power communication. The LoRaWAN has a great influence in determining network performance such as energy consumption, network capacity, the quality of service and the applications served by the network.

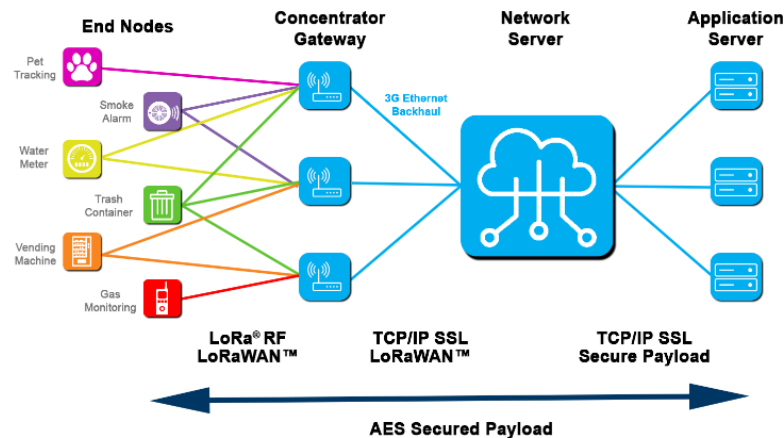


Figure 1.1. LoRaWAN network topology [5]

Many wireless low-power networks utilize a mesh network topology [7] in which the data travels through a multi-hop path to reach the destination to extend the network coverage. However, it also adds complexity and reduces the network capacity and battery lifetime. Thanks to the LoRa modulation that provides long-range transmission, LoRaWAN uses a simple star of star network topology that allows gateways to relay data from sensor nodes to a network server. Figure 1.1 shows the LoRaWAN network topology that includes a network server, application servers, gateways (GW), and end devices (or nodes). GWs and a network server are wall-powered, and they are connected to backhaul infrastructure for high-speed communication while nodes are generally battery-powered, and they communicate to GWs through LoRa wireless links. A node can associate with multiple GWs. During the network operation, the network server collects sensor data from end nodes through GWs and provides the services that are provided by application servers.

For different types of applications, LoRaWAN defines three types of end device: Class A, Class B, and Class C. Class A is required to be implemented in all end devices, while class B and class C are optional implemented for the applications that require low control latency and high quality of service.

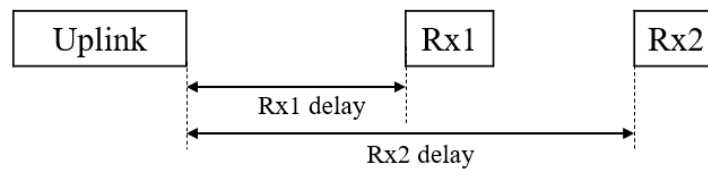


Figure 1.2. LoRaWAN Class A

The LoRaWAN **Class A** supports bi-directional communication, and the communication is always initiated by the end device. A device can send an uplink data at any time using pure Aloha channel access. Once the uplink data is transmitted, the device opens two receiving slots: Rx1 and Rx2 as illustrated in Figure 1.2. Upon receiving the uplink data from the end device, GW sends the acknowledgement or downlink command in either Rx1 or Rx2 slot. If the end device does not receive a downlink message during the Rx1, it will open the Rx2, otherwise, it goes to sleep for saving energy. The end device operated LoRaWAN class A is the most energy-efficient since it spends most of the time in sleep mode. However, it has a high downlink latency since the downlink is always initiated by the uplink message.

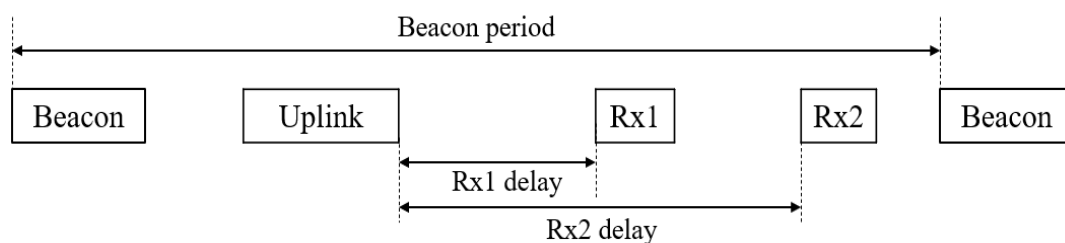
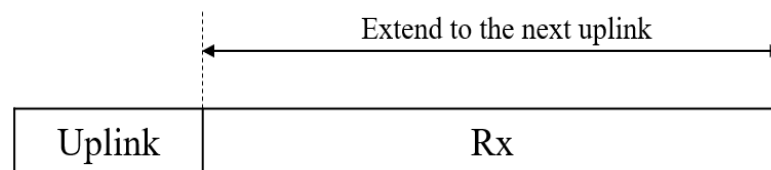


Figure 1.3. LoRaWAN Class B

To reduce the downlink latency, the LoRaWAN **Class B** allows GW to transmit a downlink beacon periodically by using scheduled receiving slots as illustrated in Figure

1.3. Upon receiving the beacon message from the GW, the end devices synchronize the time and open the receiving slots periodically. This type of device consumes more energy compared to class A since the end device additionally wakes up every receiving slot and thus, it spends more time in active mode. However, it provides a deterministic downlink latency, and the server can control the beacon period to balance the energy consumption and the required downlink latency.



*Figure 1.4. LoRaWAN Class C*

Another LoRaWAN class, **Class C**, is purposed for no latency, no energy constraints applications such as streetlights, electrical meters etc. As illustrated in Figure 1.4, the end device keeps receiving state all time except when it transmits an uplink message so that it can receive the downlink message from the network server at any time. However, this type of device consumes many times more energy than class A devices and thus it is often wall powered.

## **1.2. Multi-hop Real-time LoRa protocol**

### **1.2.1. Problems and Issues**

To secure the safety of workers and utilize resources efficiently in industry fields, a context-aware computing server collects and analyzes context data such as the location and health status of a worker, the location and operation states of equipment, the oxygen density, and the poisonous gas, etc. in workplaces. For this purpose, sensor communication devices or end nodes with sensor modules are installed in those places

or embedded into the objects such as workers and equipment, and required to send sensor data to the server periodically using a wireless network that is formed by those nodes. So far, many studies have been focused on designing communication protocols for industrial wireless sensor networks (WSNs) based on IEEE 802.15.4 standard [8]. However, they have difficulties in securing reliable data transmission due to their vulnerability to obstacles or external interference and have limitations in covering a wide area or supporting densely deployed nodes.

Recently, LoRa technology despite its low spectrum efficiency and low data rate, has been drawing attention as an alternative or a complement to wireless sensor networks owing to its provision of long-range communications and a reliable link that enables a simple star topology. However, the use of a LoRa network in industrial monitoring and control applications is still a challenge since the network has to cover wireless-unfriendly zones (WUZs) such as underground tunnels, enclosed spaces, and harsh construction houses with steel or concrete obstructions, but still has to deliver data in real-time and reliable manner. It is read in the LoRa specification [9] that LoRa can provide a transmission range up to two kilometers using the smallest spreading factor; however, the transmission range can be reduced to within hundreds or tens of meters due to signal attenuation by obstructions and the installations of nodes in WUZs. Therefore, it is necessary to design an efficient multi-hop LoRa protocol for LoRa networks that can satisfy industrial requirements.

The design of a multi-hop LoRa protocol for *industrial monitoring and control systems* (IMOCSs) should be able to address the following issues.

- First, industrial monitoring applications often require data collection from the densely distributed sensor nodes at intervals of a few seconds or tens of seconds, thus increasing the network traffic. The high traffic makes it difficult to achieve

good reliability due to high possibility of data collision. Especially in LoRa networks with relatively low data rates compared to the traditional WSNs based on the IEEE 802.15.4 standard. Furthermore, every node can have its own data transmission period during which data is generated and it is required to be delivered to the GW by the end of its period to prevent erroneous operations from the temporal inaccuracy of sensor data. To solve this problem, a TDMA-based slot schedule approach should be considered for data transmission in designing the communication protocol.

- Second, the use of a multi-hop communication further increases the network traffic since data generated at a  $k$ -hop data source from the GW has to be transmitted  $k$  times to reach GW. It not only increases the possibility of data collision due to the long communication range but also reduces the energy efficiency of intermediate nodes since they have to relay data from their higher hop counts data sources. In addition, it may cause a lot of control overhead during the construction and maintenance of a multi-hop network topology. We have to note that the overhead from control messages is very expensive in LoRa technology since it uses relatively low data rates. Considering that LoRa has a relatively long transmission range, the protocol should allow a sufficient number of hops count for balancing between network coverage and control overhead.
- Finally, in an industrial deployment, the wireless signals are time-varying due to external interference and/or the change of state of manufacturing or obstruction. Furthermore, the sensor nodes can be attached to working machines or workers that move around the working area during the operation time. Thus, designing a multi-hop LoRa protocol should be able to construct and maintain a reliable multi-

hop network topology as well as provide an efficient solution in dealing with slot scheduling and rescheduling of the network.

### **1.2.2. Previous approaches**

There have been many studies on industrial WSNs for reliable data transmission in dynamic networks over the past decade. In WSNs based on IEEE 502.15.4, nodes are generally considered fixed send data to the sink via multiple radio hops. For reliable data transmission, several communication protocols [10-15] using slot scheduling have been proposed. However, in industrial sites, even though nodes send data according to a slot schedule, the reliability of data transmission may not be secured due to various communication obstacles and external interference. Furthermore, node mobility not only increases control overhead but also causes data loss temporarily due to link failure and frequent updates of a slot schedule, making it more difficult to secure data transmission reliability. Thus, WSNs in industrial applications often suffers from reliability problem in high traffic network and has limited network coverage due to signal interference and/or obstructions.

Recently, LoRa has been attracting attention for industrial applications due to its long-range and robust communication link. The LoRaWAN [5] was proposed by Semtech Alliance as a standard for medium access control of LoRa networks that uses a star-of-star network topology [6] and employs the concept of the Aloha protocol [16] in which node transmits data freely as long as it has data. This protocol targets applications with long data transmission intervals of several minutes or more because its spectrum efficiency is low. According to previous studies, it is known that the network reliability of LoRaWAN reduces significantly as the number of nodes increases [17-20]. Even though many studies [21-25] have been conducted to overcome

the reliability problem, the modification still did not achieve notable improvement over the legacy LoRaWAN. For industrial use cases, some studies [26-29] have focused on designing communication protocols for industrial LoRa network based on star network topology. The authors in [26] proposed the industrial LoRa (I-LoRa) protocol that supports both real-time and non-real-time data transmissions. After that, an improved version of I-LoRa, named RT-LoRa, was proposed in [27]. They use a super-frame, as a data collection period, that is divided into a contention access period (CAP) for aperiodic data transmission and a contention-free period (CFP) for periodic data, during which the Aloha protocol and the slot scheduling approaches are used, respectively. In [28], a real-time LoRa protocol was proposed for IMOCSSs. The protocol is based on a frame-slot architecture that consists of an uplink (UL) and a downlink (DL) period. The UL period is further sliced into a number of data transmission slots. The protocol employs a slot schedule approach that schedule UL slots for data transmission in the way such that the real-time property of every data transmission is guaranteed. Another slot scheduling-based protocol, TS-LoRa, allows nodes to determine a slot in a frame autonomously in order to reduce slot scheduling overhead. Through simulations and experiments, these protocols achieve high reliability in data transmission. However, they only support star network topology that limits the network coverage. Even though, multiple spreading factors can be supported to balance network coverage and data rate, it reduces the efficiency of channel access due to the impact of imperfect orthogonality [30] in LoRa technology. Furthermore, in industrial deployment, LoRa nodes can be deployed in WUZs in which the line of sight (LOS) of the LoRa signal is blocked. Many studies [31-33] have been conducted to investigate the LoRa performance in indoor industrial deployment scenarios. The results show that in such harsh environments, the LoRa transmission range is reduced significantly [31], and the use of higher SF might



not improve the reliability of data transmission while consuming much more energy [33]. Thus, the above-mentioned protocols suffer from the limitation of network coverage when they are deployed in industrial fields that include a lot of obstructions or/and WUZs.

To solve the coverage problem, recent studies have focused on developing multi-hop LoRa protocols [34-37]. In [34] and [35], the authors proposed Glossy-based [38] multi-hop LoRa protocols that take advantage of concurrent transmission (CT) to improve the network reliability. In LoRaBlink [34], the protocol repeats a time frame that consists of a network construction (NC) period and a data transmission (DT) period, both being logically divided into a number of slots. In the NC period, the sink floods the network with a beacon message to construct the network, in which every node determines its hop count from the sink based on the received beacon. In the DT period, a node that has data to send selects the next available slot and broadcasts the data. Upon receiving data, every node that is located at one level lower than the sender rebroadcasts the data in the slot immediately following, so that the receiver can demodulate the data correctly. This process allows multiple nodes to transmit their own data simultaneously, and the receiver can demodulate the transmission with the strongest signal while it suppresses the other signals. In addition, the process can consume high amounts of energy due to the nature of flooding. In the concurrent transmission LoRa (CT-LoRa) protocol [35], the authors dealt with how a multi-hop node can send data reliably to a sink by making use of CT. Instead of using network construction (like the previous approach), a source broadcasts data freely within the assigned slot. Upon receiving data, a node assigns a time offset before rebroadcasting, so that the receiver can better demodulate the data. Basically, this approach improves the concept of the Glossy approach by utilizing the characteristics of LoRa technology.

In a typical multiple-building area network using 18 end nodes, CT-LoRa could achieve a packet delivery rate of almost 100%. However, if many nodes are involved in transmitting data, it becomes difficult to assign appropriate offsets. Furthermore, flooding-based transmission is not free from the efficiency of energy management. In [36], the authors proposed the LoRa-Mesh protocol in which the GW maintains a whole tree topology for the participating nodes by receiving the link state from every node, and sends a query message to a specified node to get data along a tree path. This approach could improve the packet delivery rate significantly, compared to LoRaWAN in network scenarios deployed in both campus and indoor environments. However, it causes high overhead, since it has to query the node whenever it wants to get data from the node. The authors in [37] proposed the synchronous LoRa-Mesh (Sync-LoRa-Mesh) protocol to collect data reliably from WUZs like underground or indoor areas. It defines a repeater node (RN) that has reliable LoRaWAN connectivity to the GW. The RN forms a tree with underground nodes and collects data in a synchronous manner through scheduling, and it then transmits aggregated data to the GW using the LoRaWAN. Through experiment, the results show that the protocol improves the packet error rate significantly for underground nodes compared to reference nodes that transmit data using LoRaWAN. However, the RN still has to rely on the LoRaWAN, which is not suitable for high-traffic networks.

Despite remarkable progress in multi-hop LoRa network, the application of LoRa in industrial fields still has difficulty in dealing with high traffic networks. Furthermore, node mobility can further increase the network overhead since it has to maintain a dynamic topology. To the best of our knowledge, no one has ever conducted research on a multi-hop LoRa protocol that can provide real-time and reliable data transmission as well as efficiently deal with node mobility.

### 1.3. Our Approach

This dissertation aims at designing a multi-hop real-time LoRa protocol, named as Two-hop RT-LoRa protocol, for industrial monitoring applications that not only require real-time and reliable data transmission but also support node mobility. The protocol uses a two-hop network topology to provide the connectivity to nodes that are deployed in WUZs or far away from GW. Every node is required to send its real-time data to a network server. The nodes that have direct connections to their nearby GWs are said to be *1-hop nodes* while some nodes make use of other nodes to relay their data to GW are known as *2-hop nodes*. Considering that LoRa provides a long communication range and low data rate, a two-hop network topology originating from a GW can cover a sufficient large industrial area and generate a low control overhead during the construction and maintenance of a network topology. The use of a higher number of hop-count can significantly increase network overhead and make it difficult to design a communication protocol since it increases the possibility of data collision. To further extend the network coverage, multiple GWs can be considered in the network deployment. The use of two GWs can extend the coverage of the network up to eight hops such that two GWs covers three nodes (four hops) arranged linearly between them, and each GW additionally covers two nodes (two hops) arranged linearly on opposite sides.

For real-time and reliable data transmission, the Two-hop RT-LoRa protocol uses a TDMA-based slot scheduling approach. The slot scheduling works based on a frame-slot architecture and a logical slot indexing (LSI) algorithm that was proposed recently in [28]. A frame or data collection cycle consists of a downlink (DL) and an uplink (UL) period that are used for GWs to broadcast DL messages and nodes to transmit their UL real-time data, respectively. A UL period is further divided into  $2^N$  data slots, where  $N$

is defined as a *frame factor*, that are indexed from 1 to  $2^N$  based on their physical positions. The LSI algorithm assigns a *logical slot index* to each of  $2^N$  UL slots in a way such that if  $2^C$  ( $0 \leq C < N$ ) UL slots are selected sequentially starting from any logical slot index, each of  $2^C$  slots will be distributed in one of  $2^C$  equally divided parts of a frame. Then, if a node transmits  $2^C$  data in the selected slots, it can meet the real-time property since every transmission is completed within its period of  $2^N/2^C$ . By using LSI, the slot schedule becomes simple since every node is assigned a number of UL slots as its demand with sequential logical indices except the slots that are scheduled for the other nodes in the network, then it can transmit data without data collision while satisfying the time constraint of every data transmission.

Considering industrial deployment, the LoRa signal is changing dynamically due to external interference and/or node mobility, constructing and maintaining a reliable network topology plays an important role in the protocol operation. Furthermore, since the protocol uses a slot scheduling approach for data transmission, the change of a connection state has to be reflected in the network topology quickly and then requires an updated slot schedule. The protocol also addresses the technical aspects of dynamic networks such as auto-configuration of a two-hop LoRa network, slot scheduling, topology maintenance, and updating of a slot schedule. First, in autonomously configuring a two-hop network, it is important to ensure that the tree topology can be gradually and stably built in the process of individual node registration. Therefore, a GW includes the list of registered nodes in the tree construction message before broadcasting so that a node can confirm whether it is registered or not. Considering that the one-hop relay nodes play an important role in the stability of tree topology, a node decides whether it can become a relay by itself based on its link quality. As a server generates and broadcasts slot scheduling information based on the tree topology, each

node can easily and autonomously create a slot schedule that does not conflict with the slot schedules of other nodes. In this process, the server provides the retransmission slot schedule for relay nodes so that all relay nodes can safely rebroadcast the slot scheduling information to their children without collision. During data transmission, when a node detects a link failure, it immediately reports the change of link state to the GW using an unscheduled slot to prevent collision with other data transmission. Upon receiving this, the GW completes topology maintenance by transmitting only updated slot scheduling information using the DL message. As a result, the topology and slot schedule are updated quickly without causing a high scheduling overhead.

#### **1.4. Dissertation Contribution and Structure**

The contributions of this dissertation are summarized as follows:

- We identify the problems in designing multi-hop real-time LoRa protocol for industrial monitoring and control applications that not only requires real-time and reliable data collection but also covers a sufficient large industrial area and supports node mobility.
- We propose a multi-hop network topology to extend the network coverage to WUZs. Then, based on the proposed network topology, a real-time scheduling approach is derived based on a frame-slot architecture and a logical slot indexing algorithm.
- For the application in dynamic networks and supporting node mobility, we propose solutions to deal with technical aspects in a dynamic network such as auto-configuration of a network topology construction, topology maintenance, slot scheduling and rescheduling of the network.

The remainder of this dissertation is organized as follows:

Chapter 2 presents the previous works that are related to the research topic of this dissertation. First, we introduce some researches on WSN and LoRa networks that were conducted for industrial applications. Then, we briefly introduce some multi-hop LoRa protocols.

In chapter 3, we present a detailed description of a multi-hop real-time LoRa protocol that is designed to address the coverage problem of the LoRa network and provide real-time, reliable data transmission.

In chapter 4, we redesign the protocol for its application in dynamic networks and supporting node mobility. This chapter presents the solution of constructing and maintaining multi-hop network topology as well as slot scheduling and rescheduling of a network against node mobility.

Chapter 5 presents comprehensive performance evaluation works of the proposed protocol. The protocol is evaluated through analysis, simulation, and various experiment scenarios.

Finally, the conclusions and future research directions of our works are given in Chapter 6.

## Chapter 2.

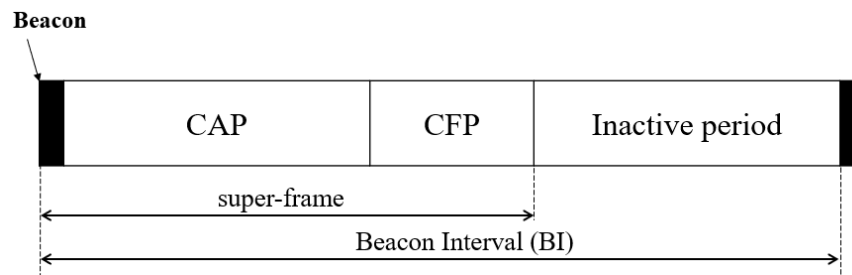
### PREVIOUS WORKS

This chapter gives a discussion about the previous studies that are related to our works. The first Section 2.1 briefly reviews communication protocols for industrial wireless sensor networks that are based on IEEE 802.15.4. Then, it is followed by similar works using LoRa networks in Section 2.2. Finally, recent studies on multi-hop LoRa networks are given in Section 2.3.

#### 2.1. Industrial Wireless Sensor Networks

The traditional WSNs are based on the physical layer specified in the IEEE 802.15.4 that enables low-cost, low-power data transmission. For reliable data transmission, the research in industrial WSNs often employs a TDMA approach in designing communication protocols. In [13], the authors proposed an industrial MAC (I-MAC) protocol that employs a slot schedule approach for a multi-hop tree topology to remove data collision for both downlink control and uplink data transmissions. Furthermore, the protocol uses control messages such as RTS, CTS, and ACK for reliable data transmission. It was proved by experiment that the protocol achieves high reliability on data transmission. However, this approach causes high scheduling overhead and thus it is only suitable for a small network with less than 50 nodes. In [14], the slotted sense multiple access (SSMA) protocol using a tree topology and sharable slots addressed the issue of data transmission reliability under various signal interferences in industrial environments and the change of network topology by node mobility. In this method, instead of allocating a tiny slot to each link, a shareable slot is allocated to each tree level of the tree topology, and nodes at each level send data to nodes at one lower level

through contention using CSMA/CA [39] within their sharable slot. In this way, data transmission is done level by level from the highest level to the lowest level with the sink. This method eliminates scheduling overhead by making slot scheduling topology independent and greatly improves the reliability of data transmission by limiting channel contention to nodes in the same tree level. However, since this approach still allows channel contention, it is not free from data collision. To further improve the reliability, the authors in [15] proposed a smart multi-channel SSMA (SMC-SSMA) protocol. In this approach, in the process of securing a common channel, each node includes its control message transmission delay in the control message it transmits. Then, every node learns the delay times of its neighboring nodes by overhearing the control messages and then adjusts its own control message transmission delay so that its control messages avoid collision with those of its neighboring and hidden nodes. After securing a channel, a node transmits data using a data channel not used by the neighbors, enabling parallel transmission.



*Figure 2.1. IEEE 802.15.4 MAC super-frame structure*

The IEEE 802.15.4 standard also provides MAC layer specifications. It defines two types of devices: Full Function Device (FFD) and Reduced Function Device (RFD), and supports two basic types of network topology: star and peer-to-peer topologies. For large scale WSN, cluster-tree topology [40] is often employed as a peer-to-peer network since it is suitable for most applications. The 802.15.4 MAC is operated based on a super-frame structure as presented in Figure 2.1. The super-frame starts with a beacon



period and defines the period in which the sensor nodes exchange data with the network coordinators. A super-frame consists of a Contention Access Period (CAP) and Contention Free Period (CFP) in which the sensor node accesses the channel using the CSMA/CA and a slot schedule, respectively. The CFP allows reliable data transmission by allocating Guaranteed Time Slots (GTS) [41] to nodes so that the channel access is contention-free. For industry automation, the WirelessHART [42] and ISA100.11a [11] were derived as industrial standards that modified IEEE 802.15.4 to allow a full TDMA channel access combined with frequency channel hopping. They also use clear channel assessment (CCA) and channel blacklisting to overcome external interference. However, these standards do not specify the slot scheduling mechanism. Thus, it is still open research that raises a lot of challenges for the application in dynamic networks.

## **2.2. Industrial LoRa Protocols**

Recently, research using the LoRa for industrial application has been actively conducted due to its robust link and long transmission distance with low power. In this section, we discuss some industrial LoRa protocols including Industrial LoRa (I-LoRa), real-time LoRa (RT-LoRa) and real-time LoRa with logical slot indexing (RT-LoRa-LSI).

### **2.2.1. I-LoRa and RT-LoRa**

I-LoRa protocol [26] was proposed to support both real-time and non-real-time data transmission in the industrial Internet of Things. The protocol operates based on a star network topology in which a LoRa gateway acts as a sink that collects uplink data from and transmits the downlink control commands to the end devices. The I-LoRa defines

a super-frame structure as shown in Figure 2.2. A super-frame consists of five sections: Beacon, CAP, CFP, Downlink, and CFP Acknowledgement (CFP Ack).



*Figure 2.2. Super-frame structure*

A super-frame starts with a beacon period that is used for time synchronization of the network and it is followed by a CAP. During CAP, nodes send non-real-time data to the sink using pure Aloha channel access. Every node that has data to send generates a random transmission parameter (channel and SF) and then transmits its data after a random delay. Every data transmission in CAP is generated under the local duty cycle restriction and completed before the end of CAP. Then, CAP is followed by a CFP that is used for real-time confirmed uplink data transmission. This period is further divided into a number of time slots. Each time slot is sufficient to transmit one uplink data packet with the maximum payload length that is defined by the application. The multi-channels and multi-SFs are also employed to support more uplink traffic. During the CFP period, every node is assigned a channel, SF, and time slot for its data transmission. Considering that the transmissions using different SFs are orthogonal, the sink can receive multiple data packets over the same channel at the same time. The super-frame is finished by a Downlink and a CFP Ack that are used for GW to transmit downlink commands and acknowledgement for the confirmed uplink messages, respectively.

The I-LoRa does not fully exploit some technical aspects such as supporting node mobility and improving the Quality of Service (QoS) of the mobile nodes. Furthermore, it uses pure Aloha for data transmission during CAP which cannot provide sufficient reliability. As a result, an improved version of the I-LoRa protocol, RT-LoRa protocol,

was proposed in [27]. RT-LoRa supports node mobility by transmitting multiple Beacon messages on different SFs during the Beacon period. This approach can help a node update its possible transmission parameters as a recommended set of SFs based on the received beacons and thus, it supports node mobility. Then during CAP, a node selects a random SF on the recommended SF values and transmits data using a slotted Aloha mechanism. To support different levels of QoS for real-time data follows, RT-LoRa defines three QoS classes (Normal, Reliable, and Most Reliable) that use three different timeslots assignment strategies and thus, it enables real-time data transmission.

### **2.2.2. RT-LoRa-LSI**

In [28], the authors proposed a real-time LoRa protocol for industrial monitoring and control applications. The protocol is operated based on a star network topology in which end nodes connect with their nearby GWs for data communication. GWs receive sensor data from end nodes and forward the received data to a network server for analyzing and providing necessary services. The protocol uses a frame as a data collection cycle that consists of a DL period and an UL period. In the DL period, the network server broadcasts a DL message through GWs. The DL message is used for the time synchronization of nodes in the network, and it can contain commands for the network or a specific node. The UL period is divided into  $2^N$  time slots. During this period, nodes transmit their real-time data using a slot schedule from the network server.

The protocol defines a real-time *task* that modelled the data transmission from a node. A task is specified by its data transmission period during which data is generated, and it has to be transmitted by the end of the transmission period to guarantee the real-time property. Tasks can belong to different classes according to their data transmission

period, such that for a frame with  $2^N$  slots, if a task has as its transmission period  $TP = 2^N/2^c$ , this task belongs to class  $c$  ( $0 \leq c \leq N$ ). For example, a class 0 of task ( $c = 0$ ) has  $TP = 2^N$  and thus it transmits one data segment per frame.

The protocol uses a real-time slot scheduling approach to assign UL slots for tasks in the network. The slot scheduling is based on an LSI algorithm that assigns a *logical slot index* to every slot in an array of slots along with its *physical slot index*. Since the LSI algorithm is also used in our proposed Two-hop RT-LoRa protocol, it is briefly introduced. The LSI algorithm uses the notion of  $2^k$ -*Constraint*. It is said that  $2^k$  logical indices satisfy the  $2^k$ -*Constraint* if each of the  $2^k$  logically indexed slots appears in only one of the  $2^k$  equally divided parts of a frame. Then, the LSI algorithm is briefly given as in Algorithm 2.1.

*Algorithm 2.1. The logical slot indexing (LSI)*

---

```

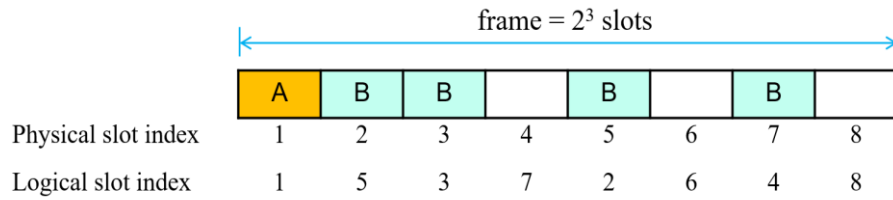
// Assume that a frame has  $2^k$  physical slots
1  $2^1$  logical indices are assigned such that they satisfy  $2^1$ -Constraint ;
2 for  $i = 2$  to  $N$  do
3   for  $j = 2^{i-1} + 1$  to  $2^i$  do
4     assign  $j$  s.t.  $\forall k = 1 \dots i$ , the logical indices from  $j$  to  $j - 2^k + 1$  satisfy
        $2^k$ -Constraint ;

```

---

In Algorithm 2.1, the statement in line 1 guarantees that the first two logical indices are assigned such that they appear in each of two equally divided parts. When  $i = 2$  (line 2), the logical indices 3 and 4 are assigned (line 3) such that when 3 is assigned, (2, 3) satisfies  $2^1$ -*Constraint* and when 4 is assigned, (4, 3) satisfies  $2^1$ -*Constraint*, and also (1, 2, 3, 4) satisfies  $2^2$ -*Constraint* (line 4) in a recursive manner. When  $i = 3$ , the logical indices 5, 6, 7, and 8 are assigned, similarly satisfying the corresponding constraints. For a detailed description of the LSI algorithm, refer to paper [28].

Employing the LSI algorithm, the slot schedule of a task becomes simple since it is assigned a number of slots as its requirement with sequential logical slot indices without duplication with the other tasks in the network. This can be done easily in a centralized network since the network server manages tasks information of all nodes in the network. Then, the network server shares this information with all the nodes in the network by using DL messages so that every node can determine its slot assignment in a distributed manner.



*Figure 2.3. A frame-slot architecture of  $2^3$  slots with the logical slot indices assigned and the slot scheduling of two tasks A and B*

Let us give an example of a task scheduling in a frame with  $2^3$  slots. Figure 2.3 shows a frame-slot architecture with the logical slot indices and the slot schedule of two tasks of nodes A and B that belong to classes 0 and 2, respectively. Task A takes logical slot index 1 corresponding to the physical slot index 1. After that, task B takes logical slot indices 2, 3, 4, and 5 corresponding to physical slots 2, 3, 5, and 7. Then, tasks A and B transmit one packet per the period of  $2^3$  slots and the period of  $2^1$  slots, respectively.

### 2.3. Multi-hop LoRa Protocols

This section introduces multi-hop LoRa protocols that were proposed recently including LoRaBlink, concurrent transmission LoRa (CT-LoRa), LoRa mesh network (LoRa-Mesh), and synchronous LoRa mesh (Syn-LoRa-Mesh).

### 2.3.1. LoRaBlink

The concept of a multi-hop LoRa network was proposed firstly in [34] with the LoRaBlink protocol. The protocol is designed to support reliable and energy-efficient in multi-hop LoRa communication. The protocol repeats an epoch as an operation cycle that consists of  $N_B$  DL beacon slots and  $N_D$  UL data slots. The GW starts broadcasting a beacon message at the first DL slot. Upon receiving the first beacon message, every node synchronizes the time and recognizes its hop count from GW. Then, it rebroadcasts the beacon at the immediately following slot. During the UL period, every node generates data and starts its data transmission after performing channel activity detection (CAD) [43]. If a data source found that the channel is free, it triggers data broadcasting in the next UL slot. Upon receiving a data message, every node that is located at one level lower than the data source rebroadcasts the data message in the immediately following UL slot. This process is performed until the data is reached the GW. The protocol enjoys high reliability in data transmission by utilizing the concurrent transmission in which multiple nodes transmit data at the same time and a receiver can receive one of them successfully with high probability. Since the protocol uses directed flooding for DL beacon and UL data transmissions, it reduces the number of nodes attending to the broadcasting process and thus improves the energy efficiency of the network. Furthermore, the use of CAD at the beginning of every slot can further improve the energy consumption. After performing CAD, a node goes to sleep until the next slot for saving energy if it finds that there is no data transmission happened during current slot.

The protocol was evaluated with a small network of 6 end nodes. One node acts as a sink that receives the data packets from the other nodes. Experiment results show that the LoRaBlink protocol can provide high reliability of over 80% for all nodes.

### 2.3.2. CT-LoRa

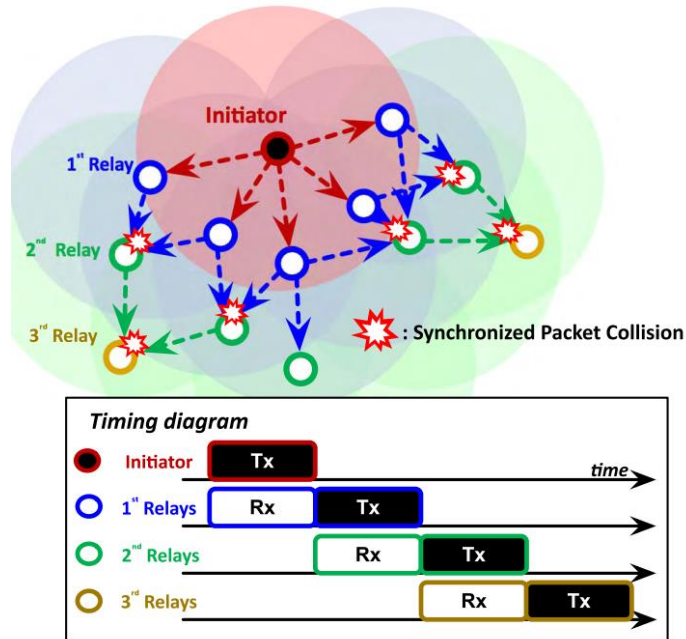


Figure 2.4. The operation of CT flooding protocol [35]

Another flooding-based multi-hop LoRa protocol, CT-LoRa, was proposed in [35] that takes the advantage of LoRa technology under CT to improve the reliability of data transmission. The flooding protocol has been investigated [44] firstly in a multi-hop WSN based on the IEEE 802.15.4 physical layer. Instead of avoiding data collision like the traditional approach, this approach allows the synchronized packet collision by multiple immediate nodes relaying the same payload packets at the same time. As shown in Figure 2.4, a data source initiates a data flooding process. Then, every node rebroadcasts the data packet immediately after its first receiving. This process is performed until the packet is flooded to the whole network. The advantages of the flooding approach have two folds. First, it removes the complexity of constructing and maintaining a multi-hop network topology. Second, this approach minimizes the end-to-end latency since it is free from exchanging control packets for a collision avoidance mechanism. To accommodate multiple data sources, the CT-LoRa employs a slot

schedule in which the time slots are assigned to all nodes in the network in a round-robin manner. A time slot is sufficient large to complete one flooding process.

For evaluation, the authors provide a comprehensive study on the performance of LoRa under CT. The results show that the LoRa is robust to CT thanks to the capture effect (CE) if two or more signal has a power offset of greater than 3 dB. In non-CE conditions, the receiver can still receive one data packet with high probability by introducing carrier frequency offset and timing offset. As a result, an offset-CT mechanism is proposed in which every node inserts a random delay time before retransmitting the packet. The protocol was evaluated through various experiment scenarios in multiple-building area networks. The results show that the protocol achieves high PRR of 98% on average, and it can further be improved by using offset-CT.

### **2.3.3. LoRa-Mesh**

In [36], the LoRa-Mesh protocol was proposed based on mesh network topology to extend the network coverage in an indoor campus deployment. This approach removes the requirement of additional GWs in covering a sufficiently large area and providing high reliability in data transmission. At the beginning of protocol operation, a GW triggers a topology construction by broadcasting beacon messages periodically and thus, its nearby nodes can join GW by considering the link quality of the received beacon. Whenever receiving a joining request from a node, GW recognizes the path to this node so that it maintains the connection paths to all connected nodes in the networks. During the data collection, GW sends a querying message to a specific data source along its path to request the UL data. In this process, an orphan node can join



any node considering the link quality of overhearing messages. An example of node joining, and data querying processes are illustrated in Figure 2.5.

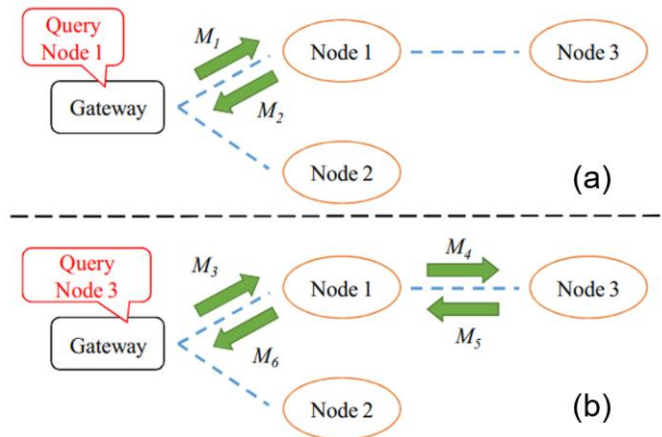


Figure 2.5. A node joining and data querying processes [36]

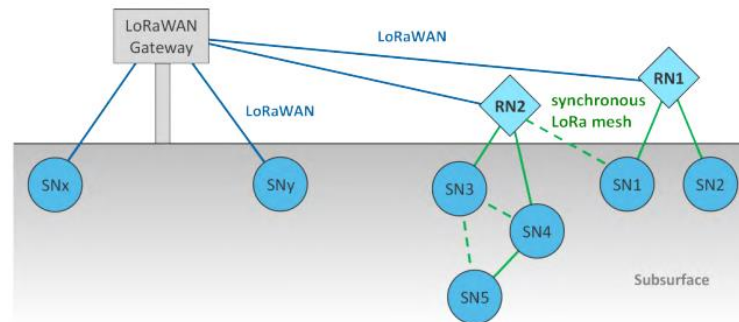
In Figure 2.5-(a), GW sends  $M_1$  message to request data from node 1. After receiving  $M_1$ , node 1 responds with data message  $M_2$  so that node 3 can join node 1 by overhearing  $M_2$  message. Then, the querying process of GW to node 3 is illustrated in Figure 2.5-(b).

Since a node joins the network by considering link quality to its parent, the network maintains reliable connections from GW to all nodes. The protocol was evaluated with a network of one GW and 19 end nodes that are deployed in an 800m x 600m area of the university campus. The experiments were performed with different maximum allowable hop-distance from 1 to 3. The results show that the mesh topology can increase the network performance significantly in terms of reliability. However, it suffers from high energy consumption since nodes listen to the channel all time except when it is in data transmission and thus, they are wall powered.

### 2.3.4. Syn-LoRa-Mesh

The study in [37] presented the limitations and difficulties of the LoRaWAN network in covering underground or indoor deployments. According to the results, the

packet error rate (PER) increases significantly for underground nodes compared to those deployed above-ground, and the coverage range of a GW is reduced to 500m at maximum for reliable data communication. To overcome these limitations, a synchronous LoRa mesh (Syn-LoRa-Mesh) protocol is proposed to provide a reliable network connection to difficult-to-access locations.



*Figure 2.6. Synchronous LoRa mesh topology [37]*

As illustrated in Figure 2.6, Syn-LoRa-Mesh extends the coverage of LoRaWAN by introducing repeater nodes (RN) that have a reliable LoRaWAN connection to GW. An RN acts as the root of a sub-network (sub-net) and constructs a mesh topology with its nearby underground sensor nodes (SNs) that are beyond the LoRaWAN coverage. The time synchronization of RNs is made by a global timing source or using an inexpensive GPS receiver. For reliable data transmission, the protocol manages a communication cycle and allows a TDMA-based slot schedule for data transmission within a sub-net. The slot schedule is made by RN for its sub-net without overlapping with other sub-nets. After receiving data from all SNs, RN aggregates the data and transmits a single LoRaWAN packet to GW for energy efficiency. This way of data aggregation limits the number of SNs that an RN can support since the aggregated packet cannot exceed the LoRaWAN maximum payload. The performance of the Syn-LoRa-Mesh protocol was evaluated by experiment scenarios that consist of 5 RNs and 11 SNs. The experiment recognized the PERs of SNs and compared their value with

the LoRaWAN reference nodes that are located at the same locations as those SNs. The results show that the protocol improved the PERs of SNs significantly compared to LoRaWAN nodes and it achieves a very low PER of 2.2% on average.

# Chapter 3.

## MULTIHOP REAL-TIME LORA

### PROTOCOL

This chapter presents the detailed design of the Two-hop RT-LoRa protocol for industrial monitoring and control applications. First, we discuss the construction of a reliable two-hop network topology. After that, a real-time task scheduling approach for a two-hop network is derived based on frame-slot architecture and the logical slot indexing algorithm. For better energy management in relay nodes, a data aggregation approach is also discussed.

#### 3.1. Network model

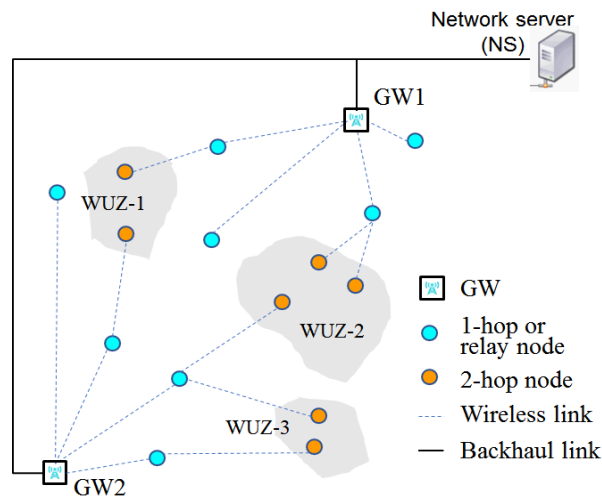


Figure 3.1. Network Topology

The considered LoRa network consists of a network server (NS), multiple gateways, and a number of end devices (end nodes or nodes) where the NS and GWs are interconnected by a high-speed backhaul network. The NS collects data from many participating nodes through the GWs, and provides the necessary services based on

analysis of the collected data. Since one GW has to deal with many nodes, in general, it is wall-powered, whereas end nodes are battery-powered.

Every node is required to send data to the server during its own transmission period, specified during network initialization. Therefore, every node can have a different data transmission period. Even though a wireless link in the LoRa network has a relatively long range, if a node is deployed in a WUZ, it may not directly connect to the GW in a reliable manner. Thus, nodes form a two-hop tree topology originating from the GW, where each tree consists of *1-hop nodes* that can connect to the GW directly, and zero or more *2-hop nodes* that can connect to the GW via an 1-hop node or a *relay node*. Then, the 2-hop node needs a relay node that can forward its data to the GW. Every node can act as a relay node. A node that belongs to a tree is said to be a *tree-node*, and a link between that node and its parent is a *tree-link*.

Figure 3.1 shows one typical two-hop network that can be deployed in industrial areas. The network consists of two GWs and 15 end nodes where eight 1-hop nodes (cyan circles) are located in open spaces, and seven 2-hop nodes (brown circles) are located in WUZs. Once each tree is constructed, the proposed protocol works in the same way for different trees. Therefore, in the rest of this chapter, the protocol design considered will be for only one tree.

### **3.2. Motivation**

LoRa technology, characterized by the provision of a long-range and reliable link, enables a star topology LoRa network in which a LoRa GW collects data directly from end nodes against node mobility. The star topology makes it easy to design a data transmission protocol. However, data transmission in the LoRa network suffers from data collisions due to a lengthy packet time on air (ToA). Hence, a simple protocol like

LoRaWAN is not suitable for industrial monitoring and control systems with relatively high data traffic. Furthermore, the stability of a LoRa link is vulnerable to any failure to secure the LOS. Hence, the GW often fails to cover the nodes deployed in WUZs, thereby requiring a relay node that receives data from those nodes and forwards them to the GW.

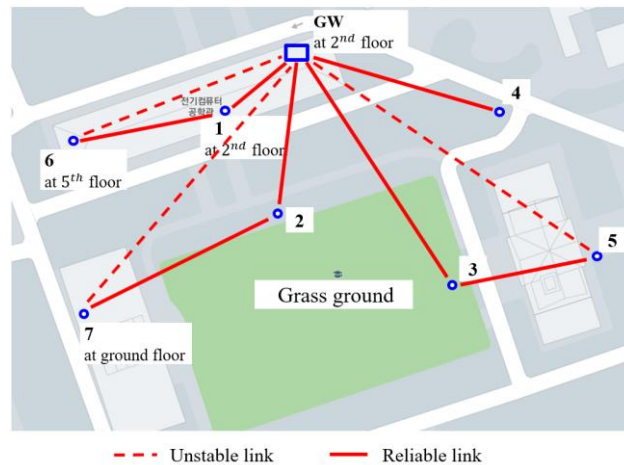


Figure 3.2. A small LoRa network deployed across multiple buildings on the campus of the University of Ulsan

Table 3.1. Comparison of PRRs for one-hop and two-hop transmission in Figure 3.2

Node	5	6	7
One-hop transmission	25%	53%	0%
Two-hop transmission	99%	98%	97%

Experiments were set up in a university campus testbed that included buildings and closed rooms, as shown in Figure 3.2, where the GW was placed in the laboratory, three nodes (1, 6, and 7) were inside the buildings, and four nodes (2, 3, 4, and 5) were outside. Every node was programmed to send a packet with a payload of 20 bytes at regular intervals of five seconds with  $SF = 7$  on a single channel for 500 seconds. A comparison of packet reception ratios (PRRs) was made between nodes 5, 6, and 7

sending data directly to the GW and sending data via nodes 3, 1, and 2, respectively. The results are summarized in Table 3.1. Two-hop transmission shows clear improvement, whereas node 7 experienced connection failure with the direct one-hop transmission.

Meanwhile, some problems were derived from the two-hop LoRa network. First, two-hop transmission can increase the possibility of collision, since the data generated by 2-hop nodes have to be transmitted twice, and thus, the increased traffic can increase interference due to the long transmission range. The interference problem is often resolved by employing time division multiple access. Second, 2-hop nodes with different transmission periods also have to send data within their respective periods or time constraints. A slot schedule for a two-hop tree requires efficient slot scheduling and schedule distribution. Third, a balanced two-hop tree topology has to be constructed to distribute the processing overhead among the 1-hop relay nodes under the constraint that every tree-link should be reliable. Lastly, an 1-hop relay node may consume more energy, since it has to forward the received data. In the LoRa technology, a node consumes much more energy in sending mode [45]. Aggregated data transmission may alleviate this problem.

The above problems are addressed with the following design principles. Upon receiving a tree construction request, every node can evaluate the link quality by examining the received signal strength indicator (RSSI) and the signal-to-noise ratio (SNR), and then judge whether they are adequate for it to become an 1-hop node. Furthermore, for energy balancing, an 1-hop relay can limit the number of nodes it can handle. Once the two-hop tree is constructed, every node can share its task profile with other nodes via the GW, and can generate an identical slot schedule for all the participating nodes based on the frame-slot architecture and the logical slot indices. A

2-hop node needs two slots within its data transmission period for its own transmission and its parent's relay. This can be done using the logical slot indices. After slot scheduling, the 1-hop node can locally rearrange the transmission sequence for itself and its children and determine which slots are to be used for transmission (or skipped) to maximize data aggregation. In this chapter, a two-hop real-time LoRa protocol is designed based on the above design principles.

### 3.3. Notation and Message

Some notations and messages are defined for convenience in explanations. A data collection cycle, or a *frame*, is divided into a number of *time slots (slots)* such that the length of a slot is sufficient to send one data packet. Since each node is required to send data, it can be modeled as a *real-time task* (or *task*) that generates data periodically according to its *transmission period (TP)*. We assume that each node has only one task, thus task and node are used interchangeably. A task is specified by its *class* that is defined based on its TP such that in a frame of  $2^N$  slots, where N is defined as frame factor, a task with period  $TP = 2^N / 2^c$  is given class c ( $0 \leq c \leq N$ ). This implies that a task of class c has a *slot demand (SD)* of  $2^c$  slots and transmits  $2^c$  packets during one frame period. A task of node  $x$  can be expressed by its *profile, PF(x)*, as follows:

$$PF(x) = (x, class(x)) \quad (3.1)$$

where  $x$  and  $class(x)$  indicate the address and the class of node  $x$ , respectively.

Some of the messages that are used to build a two-hop tree are defined as follows.

Message	Description
JREQ = ( <i>forced-flag</i> )	A node sends a <i>join request (JREQ)</i> to a tree-node in order to become a child. If <i>forced-flag</i> = 1, the receiving node must accept the sender as a child; otherwise, it may reject the join request based on the number of its children.



JRES = ( <i>accept-flag</i> )	A tree-node sends a <i>join response</i> (JRES) to the JREQ. It responds with <i>accept-flag</i> = 1 to indicate that it accepts the join request.
TCR = ( <i>ndaddr</i> , <i>ndlevel</i> )	A node, either a tree-node or a GW, broadcasts a <i>tree construction request</i> (TCR) to construct a tree topology, where <i>ndaddr</i> and <i>ndlevel</i> indicate the address and the tree level, respectively, of sending node <i>nd</i> .

### 3.4. Protocol Design

#### 3.4.1. Protocol Structure and Frame-Slot Architecture

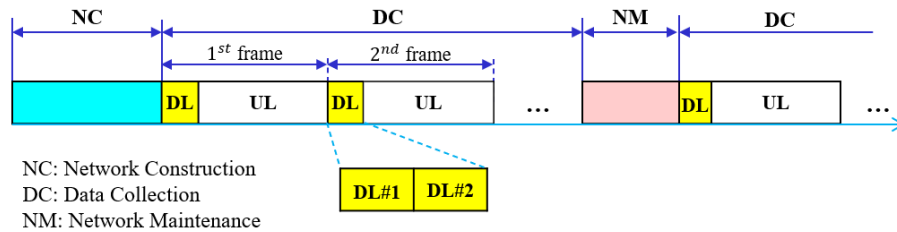


Figure 3.3. Protocol Operation

As shown in Figure 3.3, the protocol structure consists of the network construction (NC) period, the data collection (DC) period, and an optional network maintenance (NM) period. During the NC period, nodes form a two-hop tree originating at GW. Then, during the DC period, GW repeats the DC cycle that corresponds to the frame. After a series of frames, GW may start an optional NM period if broken links need to be fixed.

With  $m$  channel,  $m$  overlapping frames can be defined during one frame period. A frame using channel  $i$ ,  $Ch_i$ , is denoted by  $Ch_i$ -frame. A frame is divided into a downlink (DL) period and an uplink (UL) period. The DL period is further divided into two slots: DL#1 and DL#2, one for transmission of a command to 1-hop nodes, and another for

rebroadcasting the command toward 2-hop nodes. At the beginning of the DL period, all nodes listen to the common channel, say  $Ch_1$ , to receive a command from the GW. Upon receiving the command, every node synchronizes the start time of the UL period.

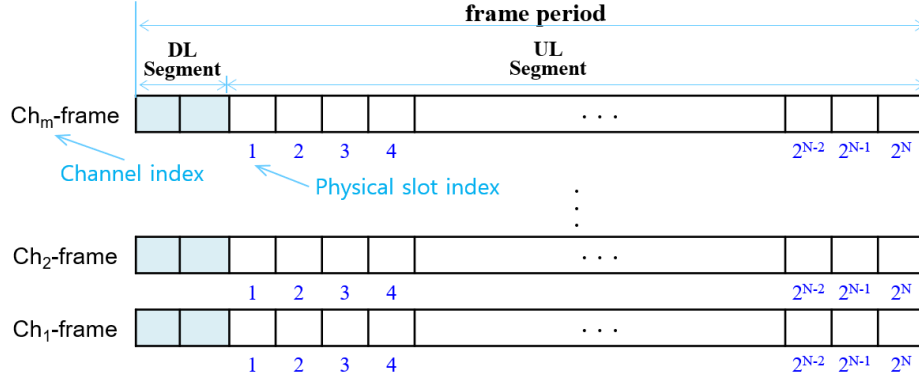


Figure 3.4. Multi-channel frame-slot architecture

The UL period is further sliced into  $2^N$  data slots where  $N$  as a frame factor is an integer constant. A slot is sufficiently large to send one data packet in which consists of three parts: guard time, CAD time, and transmission (Tx) time. The guard time is used to make up for time synchronization errors. Tx time is also referred to as the packet ToA. The data slots in the UL period are identified by physical slot indices, numbered sequentially from 1 to  $2^N$ . A frame-slot architecture using  $m$  channels is illustrated in Figure 3.4.

### 3.4.2. Tree Construction and Maintenance

A tree is constructed such that every node has a good tree-link quality, and an 1-hop relay node will limit the number of children to ensure energy balancing. When a node, say  $x$ , receives a TCR from node  $y$ , it evaluates the quality of the link  $(x, y)$  using evaluation function  $linkq(x, y)$ :

$$linkq(x, y) = \begin{cases} good & \text{if } RSSI \geq RSSI_{th} \text{ and } SNR \geq SNR_{th} \\ fair & \text{if } RSSI \geq RSSI_{th} \text{ or } SNR \geq SNR_{th} \\ bad & \text{if } RSSI < RSSI_{th} \text{ and } SNR < SNR_{th} \end{cases} \quad (3.2)$$

where  $RSSITh$  and  $SNRTh$  are the threshold values for RSSI and SNR, respectively, which indicate the minimum values for good link quality. Every node  $x$  maintains a tree information table,  $TIT(x)$ , as follows:

$$TIT(x) = (P(x), level(x), linkq(x, P(x)), CS(x)) \quad (3.3)$$

where  $P(x)$  is node  $x$ 's parent,  $level(x)$  is the level of node  $x$ , and  $CS(x)$  indicates the set of node  $x$ 's children.

Tree construction is performed as follows. The GW broadcasts  $TCR = (GW, 0)$  to initiate tree construction. Upon receiving the TCR from  $y$ , node  $x$  becomes an 1-hop node only if the  $ndlevel$  in the TCR is zero and  $linkq(x, y)$  is good. Then, it updates  $TIT(x)$  with  $level(x) = level(y) + 1$  and rebroadcasts the TCR after a random delay only if  $level(x) = 1$ . After the delay, an orphan node, say  $x$ , selects a node that provides a good link quality in  $TIT(x)$  as a candidate for its parent. Then, it tries to join the selected node, say  $y$ , by sending  $JREQ = (0)$ . Upon receiving the JREQ, node  $y$  responds with  $JRES = (1)$  only if  $|CS(y)|$  is less than or equal to the specified maximum limit,  $maxChildren$ , for energy balancing. If node  $x$  receives  $JRES = (1)$ , it determines node  $y$  as its parent. If it does not receive JRES, it launches the join process one more time after a random delay. After two failures, node  $x$  executes the join process by excluding the candidate parents that were tried previously. If it again receives  $JRES = (0)$ , it tries the join process with another parent candidate. In this process, every node  $x$  can maintain its  $CS(x)$ . Some remaining orphan nodes can join the 1-hop nodes that provide a fair link quality by sending  $JREQ = (1)$ . If necessary, some additional GWs can be installed.

If a node does not receive a DL command from its parent for three consecutive frames, it judges that it has lost its parent. If an 1-hop node fails to receive data from a specific child for three consecutive frames, it removes the child from its children list

and releases the corresponding slot schedule. A disconnected node or new node waits for the next NM period to try the join process. In the NM period, a node informs its child of a bad link if it finds the PRR from the child is less than a specified value, so it can find a new parent.

### 3.4.3. Time Synchronization

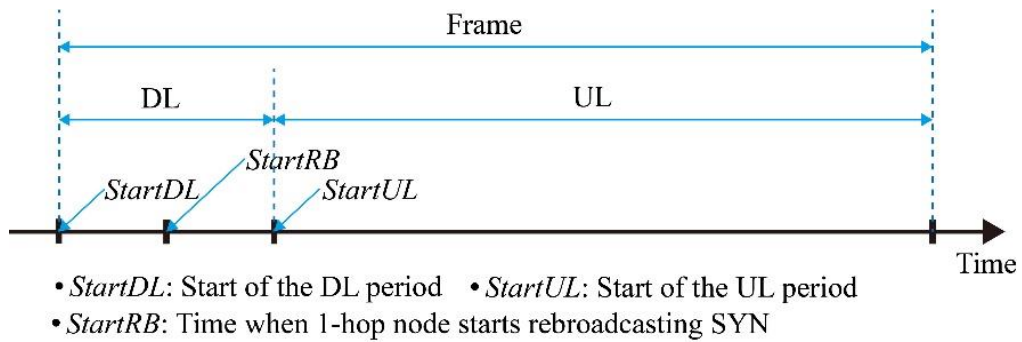


Figure 3.5. Time synchronization

Simple time synchronization is performed considering the low data rate of LoRa technology. Suppose that GW broadcasts a DL message at  $StartDL$  given in Figure 3.5. Upon receiving the DL message, every 1-hop relay node can calculate its *rebroadcast time*,  $StartRB$ , that corresponds to the start of DL#2, by taking into account the transmission delay (i.e., ToA: packet time on air) of the DL message as follows:

$$StartRB = sysTime() - ToA + \frac{DL}{2} \quad (3.4)$$

where  $sysTime()$  is the local time when the 1-hop node finishes receiving the DL message. Then, 1-hop and 2-hop nodes can calculate their *uplink start times*  $StartUL(1-hop)$  and  $StartUL(2-hop)$  as follows:

$$\begin{aligned} StartUL(1-hop) &= sysTime() - ToA + DL \\ StartUL(2-hop) &= sysTime() - ToA + \frac{DL}{2} \end{aligned} \quad (3.5)$$

By using (3.5), all 1-hop and 2-hop nodes can start their UL periods at the same time.

### 3.4.4. Slot Scheduling

After the construction of a two-hop network, the network server divides 1-hop nodes into multiple groups for the use of multiple channels on UL data transmission. The grouping is made such that the network traffic of groups are balanced for better utilizing the channel. Since the slot scheduling for different groups is made independently and similarly, this section only presents the slot scheduling mechanism for a single channel.

For slot scheduling, every node is required to report its profile to a server. A 1-hop relay node collects the profiles of its children and merges them with its own task profile before reporting. Suppose that a network has a list of  $k$  1-hop nodes as  $(n_1, n_2, \dots, n_i, \dots, n_k)$ . Then, the *integrated profile*,  $PF(n_i)$ , of 1-hop node  $n_i$  is expressed as follows:

$$PF(n_i) = (n_i, class(n_i), PF(n_{i1}), PF(n_{i2}), \dots, PF(n_{ij}), \dots, PF(n_{ic_i})) \quad (3.6)$$

where  $PF(n_{ij})$  indicates the profile of the  $i^{th}$  child of node  $n_i$ , and  $c_i$  is the number of  $n_i$ 's children. Let  $SD(x)$  and  $TSD(x)$  denote the *slot demand* of node  $x$  and the *total slot demand* of  $x$  and  $x$ 's children, respectively. Then,  $TSD(n_i)$  is expressed as follows:

$$TSD(n_i) = SD(n_i) + \sum_{j=1}^{c_i} SD(n_{ij}) \quad (3.7)$$

where  $SD(n_i) = 2^{class(n_i)}$  and  $SD(n_{ij}) = 2 * 2^{class(n_{ij})}$  since 2-hop node needs twice as many slots as its demand. Then, every 1-hop node  $n_i$  reports  $PF(n_{ij})$  to a server so that the server can manage the *task profile* ( $PF$ ) for all nodes in the network as follows:

$$PF = (PF(n_1), PF(n_2), \dots, PF(n_k)) \quad (3.8)$$

If tasks are scheduled in the order of the elements in  $PF$ , the *start logical slot index*,  $startLSI(n_i)$ , of node  $n_i$  in slot schedule is calculated as follows:

$$startLSI(n_i) = \sum_{j=1}^{i-1} TSD(n_j) + 1 \quad (3.9)$$

A sever calculates total slot demands for all 1-hop nodes according to (3.7), and distributes the *network slot scheduling information* (*NSSI*) using a DL message:

$$NSSI = ((n_1, TSD(n_1)), (n_2, TSD(n_2)), \dots, (n_k, TSD(n_k))) \quad (3.10)$$

Upon receiving *NSSI*, every 1-hop node  $x$  gets its  $startLSI(x)$  according to (3.9) and generates a *local slot schedule*,  $LSS(x)$ , using Algorithm 3.1 with  $startLSI(x)$  and  $TSD(x)$ . The  $LSS(x)$  consists of its *receiving slots*,  $RxSlots(x)$ , used to receive data from its children and its *transmitting slots*,  $TxSlots(x)$ , used to transmit its own data and relay data received from its children.

*Algorithm 3.1. Slot scheduling of 1-hop relay node*

---

```

//Alloc(x): Slot Allocation of node x is a list of SD(x) sequential logical
slot numbers starting with startLSI(x) according to the LSI algorithm
1: At node x that receives NSSI:
2: calculates startLSI(x) using (3.9);
   //get the ascending sorted (ascSort) list of physical slot indices (psi)
   corresponding to Alloc(x)
3: TxSlots(x) = ascSort{psi(y) | y ∈ Alloc(x)};
4: startLSI = startLSI(x) + SD(x);
5: RxSlots(x) = {};
6: for each y ∈ CS(x) do
7:   Alloc(y) = a list of SD(y) sequential logical slot numbers starting
   with startLSI;
9:   psiAlloc(y) = ascSort{psi(v) | v ∈ Alloc(y)};
10:  RxSlots(x) = RxSlots(x) ∪ {v | v ∈ psiAlloc(y), v is in odd position};
11:  TxSlots(x) = TxSlots(x) ∪ {v | v ∈ psiAlloc(y), v is in even position};
12:  startLSI = startLSI + SD(y);
13: endFor

```

---

Furthermore, 1-hop relay node  $x$  generates its *local slot scheduling information*,  $LSSI(x)$ , that is required for its children to perform slot scheduling:

$$LSSI(x) = (startLSI, PF(x_1), PF(x_2), \dots, PF(x_{k_x})) \quad (3.11)$$

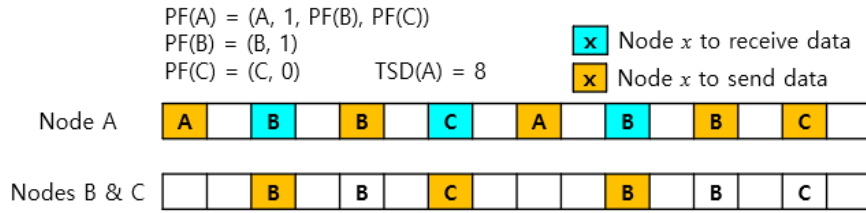
where  $startLSI = startLSI(x) + SD(x)$ , and  $k_x$  indicates the number of node  $x$ 's children. Then, node  $x$  broadcasts  $LSSI(x)$ , and its child generates a slot schedule that includes  $TxSlots$  using Algorithm 3.2.

*Algorithm 3.2. Slot scheduling of 2-hop node*

- 
- 1: At node  $x_i$  that receives  $LSSI(x)$ :
  - 2:  $startLSI = startLSI + \sum_{j=1}^{i-1} SD(x_j)$ ;
  - 3: get  $Alloc(x_i)$  starting with  $startLSI$ ;
  - 4:  $psiAlloc(x_i) = ascSort\{psi(v) \mid v \in Alloc(x_i)\}$ ;
  - 5:  $TxSlots(x_i) = \{x \mid x \in psiAlloc(x_i), x \text{ is in odd position}\}$ ;
- 

Logical index	1	9	5	13	3	11	7	15	2	10	6	14	4	12	8	16
Physical index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

(a) Logical slot indexing with 16 slots



(b) Local slot schedule

*Figure 3.6. Two-hop tree slots scheduling using logical slot indices*

Let us give an example to generate the local slot schedule for a network that consists of one 1-hop node A and two 2-hop nodes (B and C) that are the children of node A, with a frame of  $2^4 = 16$  UL slots. The LSI algorithm assigns logical slot index to each of 16 UL slot as illustrated in Figure 3.6-(a). Suppose that  $PF(A) = (A, 1, (B, 1), (C, 0))$  and  $startLSI(A) = 1$ . Then,  $TSD(A) = 8$  and  $Alloc(A) = (1, 2)$ ,  $Alloc(B) = (3, 4, 5,$

6), and  $Alloc(C) = (7, 8)$ . Then, we get the ascending-sorted physical slot indices:  $psiAlloc(A) = (1, 9)$ ,  $psiAlloc(B) = (3, \underline{5}, 11, \underline{13})$ , and  $psiAlloc(C) = (7, \underline{15})$ . The underlined numbers in even positions are transmission slot numbers:  $TxSlots(A) = (1, 5, 9, 13, 15)$ , and  $RxSlots(A) = (3, 7, 11)$  that corresponds to  $TxSlots(B) \cup TxSlots(C)$  from Algorithm 3.2. The slot schedule is illustrated in Figure 3.6-(b).

### 3.4.5. Data transmission

#### a. Data aggregation

Table 3.2. Energy consumption subject to the modes of a LoRa SX1276 transceiver

Node mode	Supply current in LoRa SX1276
Sending	28 (mA) at $P_{tx} = 13dBm$
Receiving	10.3 (mA) at BW = 125 kHz
Idle	1.5 ( $\mu A$ )
Sleep	0.2 ( $\mu A$ )

If every node transmits data according to the scheduled transmission slots, a relay node can consume much more energy than 2-hop node, since it has to receive all the data from its children and forward them to the GW. According to the datasheet of the LoRa SX1276 [45], shown in Table 3.2, a node consumes 28 mA in sending mode, much higher than the 10.3 mA in receiving mode.

In this section, an optimal slot selection (OSS) algorithm is designed to select the minimal number of transmission slots, thereby maximizing data aggregation at a relay node. The design of the OSS algorithm is based on the basic principle that every node delays its transmission as late as possible under the condition that it still should be able to transmit its packet in any of the scheduled slots to complete its transmission no later



than the end of its transmission period. For this, every node needs to know the latest slot among the allocated slots within its transmission period.

Given the slot schedule, every relay node can select the physical slots in which it must transmit based on its data transmission periods. For the design of the OSS algorithm, we define the deadline list ( $DL$ ) for a relay node. Suppose that node  $s$  has the shortest transmission period among relay  $r$  and its children in  $C(r)$ . Let  $TP_x$  denote the period of node  $x$ . Then, deadline list for relay node  $r$ ,  $DL(r)$ , can be obtained as follows:

$$DL(r) = \{p \mid p = a \times TP_s, p \leq UL, a = 1 \dots k, k \text{ is an integer}\} \quad (3.12)$$

where  $UL$  is the uplink period in the slots.  $DL(r)$  indicates the list of the last physical slot numbers in all periods of task  $s$  within one frame.

For example, both nodes A and B have the shortest period of 8 in Figure 3.6. Then, for relay node A,  $DL(A) = (8, 16)$ . For relay node  $r$  and  $DL(r)$ , the OSS algorithm is detailed in Algorithm 3.3.

*Algorithm 3.3. Optimal Slot Selection*

---

```

// x.TxFlag: if this is True, a relay must transmit data in slot x
// r: a relay node
1: for each  $x \in TxSlots(r)$  do
2:    $x.TxFlag = False$ ;
3: endFor
4: for each  $k \in DL(r)$  do
6:    $x = \max\{y \mid y \in TxSlots(r), y \leq k\}$ 
7:    $x.TxFlag = True$ ;
8: endFor

```

---

This algorithm is quite simple, since relay node  $r$  selects the latest transmission slot before missing each deadline in  $DL(r)$ . For node A in Figure 3.6,  $TxSlots(A) = (1, \underline{5}, 9,$

13, 15), where the underlined physical slot indices the slots in which relay A must transmit aggregated data. However, the aggregated data may exceed the MAC protocol data unit (MPDU) if a relay node waits until it meets the selected transmission slot. Thus, a relay node has to judge whether it has to transmit partially aggregated data or not *on-the-fly* by checking the size of the aggregated data. If the size of the currently aggregated data exceeds MPDU, it must transmit the aggregated data in the current slot to start queuing again.

In addition, a filtering technique can be employed. A relay may discard some data by analyzing its data or the received data before aggregation. This can further reduce the number of transmissions, or can reduce the size of aggregated data.

### ***b. Data transmission***

Based on *TxSlots* and *RxSlots*, Algorithm 3.4 explains how 1-hop (relay) nodes and 2-hop nodes perform data transmission with data aggregation.

*Algorithm 3.4. Data transmission*

At 1-hop (relay) node $r$	At 2-hop node $x$
1: <u>Upon receiving SYN from GW:</u> // initialization	1: <u>Upon receiving SYN from a relay:</u> //initialization
2: set $startULTimer = StartUL - sysTime$ ;	2: set $startULTimer = StartUL - sysTime$ ;
3: <u>When <math>startDLtimer</math> expires:</u>	3: <u>When <math>startDLtimer</math> expires:</u>
4: $StartUL = StartUL + UL + DL$ ;	4: $StartUL = StartUL + UL + DL$ ;
5: set $startULTimer = StartUL - sysTime$ ;	5: set $startULTimer = StartUL - sysTime$ ;
6: wait for command;	6: wait for command;
7: <u>Upon receiving a command or SYN from GW:</u>	7: <u>Upon receiving a command or SYN from a relay:</u>
8: process command;	8: process command;
9: sleep;	9: sleep;
10: <u>When <math>startULTimer</math> expires:</u>	10: <u>When <math>startULTimer</math> expires:</u>
11: $txseq = 1$ ; $rxseq = 1$ ;	11: $txseq = 1$ ;
	12: $txsn = TxSlots(x)[txseq]$ ;
	13: set $TxTimer = (txsn - 1) * slotLen$ ;
	14: sleep;

<pre> 12: <math>txsn = TxSlots(r)[txseq]</math>; <math>rxsn = RxSlots(r)[rxseq]</math>; 13: set <math>TxTimer = (txsn - 1) * slotLen</math>; 14: set <math>RxTimer = (rxsn - 1) * slotLen</math>; 15: sleep; 16: <u>When <math>TxTimer</math> expires:</u> 17: <b>if</b> <math>txsn \in psiAlloc(r)</math> <b>then</b> enqueue <math>myPkt</math>; 18: aggregate all queued packets into <math>AggPkt</math>; 19: <b>if</b> (<math>txsn.TxFlag = True</math>) or (<math>MPDU - size(AggPkt) &lt; size(Pkt)</math>) <b>then</b> 20: send <math>AggPkt</math>; clear queue; 21: <b>endIf</b> 22: <b>if</b> <math>txseq \leq  TxSlots(r) </math> <b>then</b> 23: <math>txseq = txseq + 1</math>; <math>txsn = TxSlots(r)[txseq]</math>; 24: set <math>TxTimer = StartUL + (txsn - 1) * slotLen - sysTime</math>; 25: <b>else</b> 26: <math>StartDL = StartUL + UL</math>; 27: set <math>startDLTimer = StartDL - sysTime</math>; 28: <b>endIf</b> 29: sleep; 30: <u>When <math>RxTimer</math> expires:</u> 31: wait for a packet; 32: <b>if</b> <math>r</math> receives <math>Pkt</math> <b>then</b> enqueue <math>Pkt</math>; 33: <b>if</b> <math>rxseq \leq  RxSlot(r) </math> <b>then</b> 34: <math>rxseq = rxseq + 1</math>; <math>rxsn = TxSlots(r)[rxseq]</math>; 35: set <math>RxTimer = StartUL + (rxsn - 1) * slotLen - sysTime</math>; 36: <b>endIf</b>; 37: sleep; </pre>	<pre> 15: <u>When <math>TxTimer</math> expires:</u> 16: send <math>myPkt</math>; 17: <b>if</b> <math>txseq \leq  TxSlots(x) </math> <b>then</b> 18: <math>txseq = txseq + 1</math>; 19: <math>txsn = TxSlots(x)[txseq]</math>; 20: set <math>TxTimer = StartUL + (txsn - 1) * slotLen - sysTime</math>; 21: <b>else</b> 22: <math>StartDL = StartUL + UL</math>; 23: set <math>startDLTimer = StartDL - sysTime</math>; 24: <b>endIf</b> 25: sleep; </pre> <p> <math>MPDU</math> MAC protocol data unit  <math>Pkt</math> A packet  <math>myPkt</math> A packet the node generates itself  <math>AggPkt</math> An aggregated packet  <math>rxseq</math> Used as an index for the <math>RxSlots</math> list  <math>txseq</math> Used as an index for the <math>TxSlots</math> list  <math>rxsn</math> A variable for <math>Rx</math> slot number  <math>txsn</math> A variable for <math>Tx</math> slot number  <math>X[i]</math> The <math>i^{th}</math> element of array or list <math>X</math>  <math>slotLen</math> The time length of a slot  <math>sysTime</math> System time at the processing point </p>
---	--

Upon receiving SYN from the GW or a relay, every node performs time synchronization, calculates  $StartUL$ , and goes to sleep. Then, every node wakes up at  $StartUL$  to start data transmission according to the slot schedule. For 1-hop nodes, as

soon as a node enters the UL period, it calculates the next transmission time from the first slot in  $TxSlots(r)$  in line 13, and calculates the next receiving time from the first slot in  $RxSlots(r)$  in line 14. Then, every node wakes at the next scheduled slot to transmit (line 16) or receive a packet (line 30). A relay node performs data aggregation until it must transmit a packet by checking  $TxFlag$  or the size of an aggregated packet that may reach the maximum possible size (line 19). Furthermore, every 1-hop node always waits for a packet from its children (line 30). For 2-hop nodes, as soon as a node enters the UL period, it calculates the next transmission time from the first slot in  $TxSlots(x)$  (line 13). Then, whenever a node finishes sending a packet, if it has more slots, it calculates the next transmission slot (line 19); otherwise, it goes into the next DL period (line 23).

To prevent external interference, the proposed protocol employs CAD to implement the LBT mechanism. Thus, a node always senses the channel before its data transmission, and sets a random delay if the channel is busy. In general, every node may assume that its signal will be stronger than any external signal. Thus, even with the existence of external interference, a node can receive a packet with a high probability of success by exploiting the capture effect [46].

The GW evaluates the condition of the network periodically by analyzing the number of successfully received data packets from the participating nodes. If it fails to receive data from a certain portion of the nodes, it can initiate network maintenance by broadcasting a request network maintenance command during the DL period. Then, at the end of the next UL period, every node starts the NM period.

# Chapter 4.

## MULTI-HOP REAL-TIME LORA PROTOCOL FOR DYNAMIC NETWORKS

This chapter aims at redesigning the Two-hop RT-LoRa protocol for supporting the applications in dynamic networks. Based on the network topology and the data transmission mechanisms that were presented in the previous chapter, this chapter focuses on providing solutions for constructing and maintaining a robust network topology as well as slot scheduling and rescheduling of the network against the topology change in dynamic networks.

### 4.1. Problem Identification

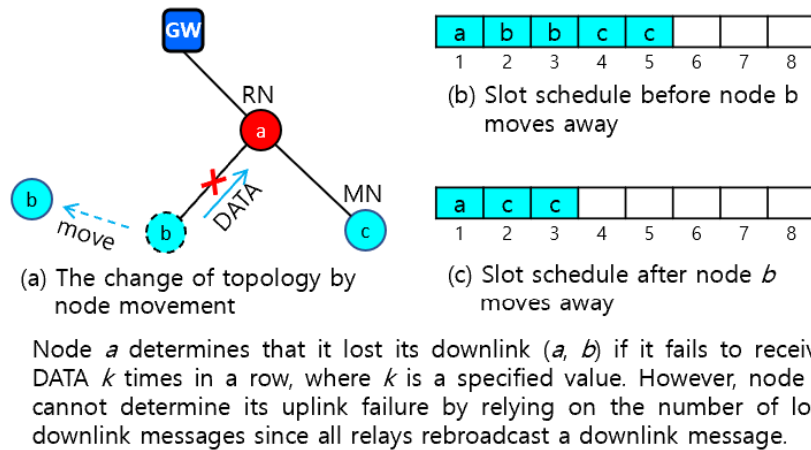


Figure 4.1. Link failure and inconsistency of slot schedule due to node mobility

Chapter 3 presented the design of a multi-hop real-time LoRa protocol that uses a two-hop topology for extension of network coverage and a two-hop slot scheduling for reliable data transmission. Every node uses the slot scheduling information transmitted by a server to create its own slot schedule such that it satisfies its own transmission

interval if it transmits data according to the slot schedule, and does not cause any collision in data transmission with other nodes. However, the protocol does not respond to the change of topology in dynamic networks.

There are some issues to consider in designing a real-time protocol for dynamic two-hop LoRa networks. First, the protocol should be able to detect the uplink or downlink failure of a node quickly. A node, either a GW or a relay node, can detect its downlink failure by counting the number of missing data's from its child. Furthermore, a 1-hop (relay) node can detect its uplink failure easily by counting the number of missing downlink messages from a GW. However, it is not possible for a 2-hop node to use the downlink message to detect uplink failure. The reason is that all relay nodes rebroadcast an identical downlink message. Therefore, this requires a different method than making use of downlink messages. One way is that a 2-hop node can detect its uplink failure by using the downlink failure information of its parent. For example, consider a simple two-hop network topology in Figure 4.1-(a) that consists of relay node  $a$  and 2-hop nodes  $b$  and  $c$ . Let a downlink state of node  $x$  with  $k$  children,  $x_1, x_2, \dots$ , and  $x_k$ , be represented as  $(x(x_1, x_2, \dots, x_k))$ . Suppose that node  $a$  detected the failure of its downlink  $(a, b)$  and reported its changed downlink state  $(a(c))$  to GW. If GW broadcasts  $(a(c))$ , node  $b$  can be aware of the failure of its uplink  $(b, a)$ .

Second, the protocol should be able to update a slot schedule quickly for the change of topology. If GW detects the downlink failure of a 1-hop relay or leaf node, the GW simply releases the slots allocated to that node and its children, and then broadcasts a downlink message to request the relevant nodes to switch to orphan nodes. Then, each orphan node needs to individually rejoin the network and be allocated slots. However, when a relay node detects a downlink failure for a child, the slot rescheduling becomes a bit more complicated. In this case, the relay node will first report its changed downlink

state to GW so that the GW can generate a new slot schedule for the relay node. The child can know its uplink failure if receives the changed link state of its parent from GW. For example, referring to Figure 4.1, suppose that GW has a slot schedule (SS) for the downlink state ( $a(b, c)$ ) of node  $a$ , as  $SS(a) = (a = (1), b = (2, 3), c = (4, 5))$  as shown in Figure 4.1-(b). Note that every 2-hop node need two slots, one for its own transmission and another for its parent to forward the received data. If node  $a$  detects its downlink failure due to the movement of node  $b$  and reports a changed downlink state ( $a(c)$ ) to GW, the new slot schedule becomes  $SS(a) = (a = (1), c = (2, 3))$  as in Figure 4.1-(c).

Third, the protocol has to handle a *slot schedule conflict* problem that occurs when multiple nodes use the same slot temporarily. Suppose that node  $a$  has a new slot schedule shown in Figure 4.1-(c) after it detects the failure of its downlink ( $a, b$ ). Unfortunately, node  $b$  fails to receive the changed downlink state ( $a(c)$ ) that was broadcasted by GW and thus, it still uses the previous slot number 2 until its uplink failure is detected.

Fourth, as GWs and relay nodes form the mobile backbone of the LoRa network, it is of great importance to build and maintain the network topology in such a way that the relay nodes have stable uplink. If a 1-hop relay node loses an uplink, a significant amount of overhead may occur in the process of topology change and slot rescheduling.

Finally, one important question is whether it is appropriate to have three or more radio hops in a low data rate LoRa network, even at the cost of the increased control overhead and the increased likelihood of collisions due to increased traffic. For example, if  $k$ -hop is allowed in multi-hop LoRa networks, the data generated at  $(k+1)^{th}$  tree level has to be transmitted  $k$  times before reaching a GW. This may increase interference severely due to the long transmission distance of LoRa. However, even

though  $k$  is limited to 2, the use of two GWs can extend the coverage of a LoRa network up to eight hops such that two GWs cover three nodes arranged linearly between them, and each GW additionally covers two nodes arranged linearly on opposite sides as  $(x_1 - x_2 - G_1 - x_3 - x_4 - x_5 - G_2 - x_6 - x_7)$  where  $G_1$  and  $G_2$  are gateways and  $x_i, i = 1..7$ , indicates an end node.

In conclusion, this chapter aims at redesigning the Two-hop RT-LoRa protocol in consideration of the issues discussed so far.

## 4.2. Notations and Messages

Some notations used in the following sections of this chapter are summarized as follows:

Notation	Meaning
RNL	indicates a <i>registered node list</i> in which every node has registered with a server.
$P(x)$	indicates the parent of node $x$ .
$TCRInt$	indicates the interval that GW uses to broadcast a tree construction request (TCR) message during initialization period.
$MaxChildren$	indicates the <i>maximum number of children</i> that a 1-hop relay node can have.
$uPF(x)$	indicates the <i>updated profile</i> that node $x$ generates if it detects link breakage to any of its children.
$uSSI(x)$	indicates an <i>updated slot scheduling information</i> that a server generates for node $x$ that has reported $uPF(x)$

Some messages used in constructing and maintaining a dynamic two-hop network topology are summarized as follows:

Message	Description
---------	-------------



$TCR = (level, RNL)$	is a <i>tree construction request</i> (TCR) message that a server broadcasts at the intervals of $TCRInt$ during initialization period and <i>level</i> indicates the tree level of the node that broadcasts this message.
$RR(x)=(x,P(x),PF(x))$	is the <i>registration request</i> (RR) message that node $x$ sends to its parent $P(x)$ to register with a server.

### 4.3. Protocol Design

#### 4.3.1. Protocol Structure

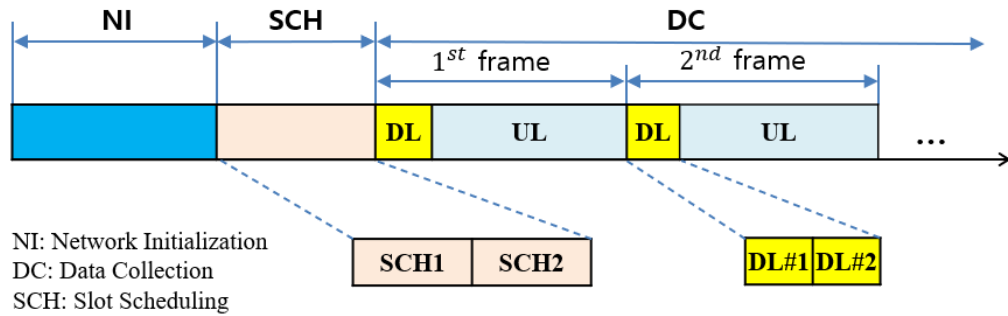


Figure 4.2. The operational structure of the proposed protocol

The protocol operation starts with network initialization (NI). As each node is installed, it starts registering with a server via GWs immediately, and the installed nodes progressively form a two-hop tree originating from a GW. When a specified percentage of end nodes are registered, the server starts a slot scheduling (SCH) period. Unregistered nodes are registered later during the data collection (DC) period. The SCH period is divided into two subperiods, SCH1 and SCH2, for the slot scheduling of 1-hop nodes and that of 2-hop nodes, respectively. Then, the server initiates a DC period that repeats a frame or data collection cycle. The maintenance of network topology is made continuously during data collection. The operational structure of the proposed protocol for dynamic LoRa networks is illustrated in Figure 4.2.

### 4.3.2. Network Initialization

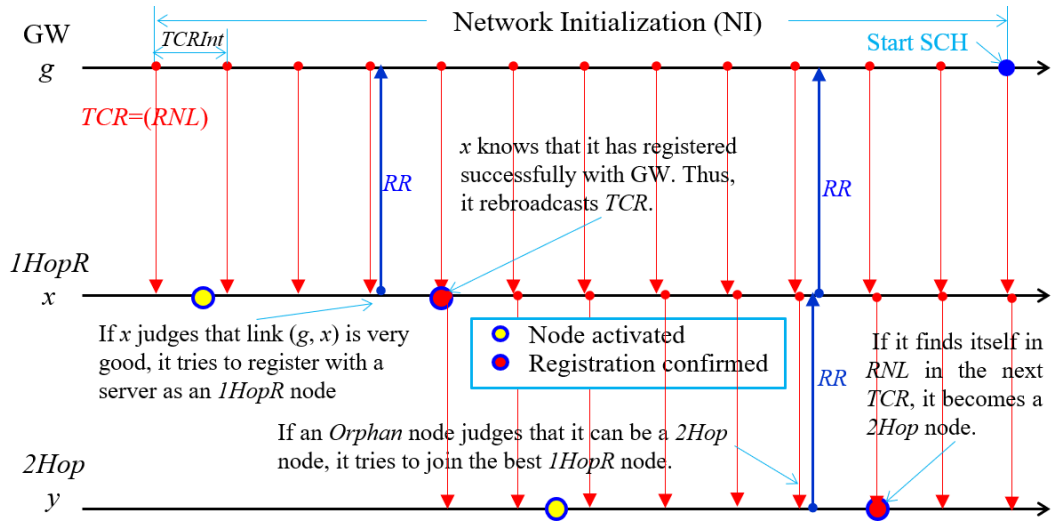


Figure 4.3. Node registration and tree construction process

During the NI period, nodes register with a server and form a two-hop tree topology by considering link quality. A server starts constructing a tree by having a GW broadcast a tree construction request (TCR) message at regular intervals. The TCR includes a current registered node list (RNL) which is empty at the beginning and is updated whenever a server receives a registration request (RR) message from a new node.

A node determines its node type (*nodeType*) using a received signal strength indicator and a signal-to-noise ratio after receiving multiple TCRs and comparing them with the specified threshold values,  $RSSI\_Th1$ ,  $RSSI\_Th2$ ,  $SNR\_Th1$ , and  $SNR\_Th2$  such that  $RSSI\_Th1 > RSSI\_Th2$  and  $SNR\_Th1 > SNR\_Th2$ . An orphan (*Orphan*) node turns into a 1-hop (*1Hop*) node or a 1-hop relay (*1HopR*) node if the uplink quality is good; otherwise, it remains a 2-hop candidate (*2HopCan*) node temporarily, as detailed in Algorithm 4.1. If a *2HopCan* node determines its parent, it becomes a 2-hop (*2Hop*) node. Since a *1HopR* node determines the stability of the network, its uplink has to be highly robust.

*Algorithm 4.1. Node type determination*

---

```
// avg_rssi indicates the average of RSSIs
// avr_snr indicates the average of SNRs
1: At an Orphan node that receives  $k$  TCRs,  $k > 1$ :
2: calculate  $avg\_rssi$  and  $avg\_snr$  with multiple TCRs;
3: determine  $nodeType$  using  $avg\_rssi$  and  $avg\_snr$  as follows:
4: if  $avg\_rssi \geq RSSI\_Th1$  and  $avg\_snr \geq SNR\_Th1$  then
5:    $nodeType = 1HopR$ ;
6: else if  $avg\_rssi \geq RSSI\_Th2$  and  $avg\_snr \geq SNR\_Th2$  then
7:    $nodeType = 1Hop$ ;
8: else
9:    $nodeType = 2HopCan$ ;
```

---

A *2HopCan* node turns to a *2Hop* node only if it can join any *1HopR* node. The joining process is as follows. A *2HopCan* node waits to receive more TCRs to find a good *1HopR* node. Suppose that it received TCRs from multiple *1HopR* nodes. Then, it first selects all *1HopR* nodes with *RSSI* and *SNR* greater than and equal to  $RSSI\_Th2$  and  $SNR\_Th2$ , respectively. Then, a *2HopCan* node turns to a *2Hop* node if it can connect to any *2HopR* node with the minimum link quality that can maintain connectivity. This is quite reasonable in LoRa networks since the data transmitted from any *2Hop* node can reach GW directly or indirectly via *1HopR* node, thereby increasing the probability of data delivery to GW. The effect of increasing the reliability of transmission due to data being transmitted additionally along another path without wasting spectrum is referred to as an *augmented transmission effect*. The *2HopCan* node  $x$  selects a *1HopR* node with the largest *RSSI* value as a candidate parent, say  $y$ , among the selected *1HopR* nodes to send RR. Upon receiving RR, node  $y$  forwards RR to GW only if it has children less than or equal to *MaxChildren*. In this process, since node  $x$  can register with the server without *1HopR* node  $y$ 's knowledge, the GW must

discard the RR sent directly by node  $x$ . When node  $x$  finds itself in RNL the next time it receives TCR, it becomes a  $2Hop$  node. Otherwise, node  $x$  must hold off joining the network until the start of the DC period. Tree construction and node registration process of a  $1HopR$  node  $x$  and a  $2Hop$  node  $y$  with GW  $g$  is illustrated in Figure 4.3.

In this process, every node, say  $x$ , maintains a *node information table*,  $NodeIT(x)$ , as follows:

$$NodeIT(x) = (P(x), level(x), RSSI(P(x)), SNR(P(x)), CS(x)) \quad (4.1)$$

where  $P(x)$  is the parent of node  $x$ ,  $level(x)$  is the tree level of node  $x$ ,  $RSSI(P(x))$  and  $SNR(P(x))$  indicate  $RSSI$  and  $SNR$  for a link  $(x, P(x))$ , respectively, and  $CS(x)$  indicates the set of node  $x$ 's children.

If a server finds a specified portion of nodes registered, it initiates SCH period by broadcasting slot scheduling information.

### 4.3.3. Scheduling Information Dissemination and Slot Scheduling

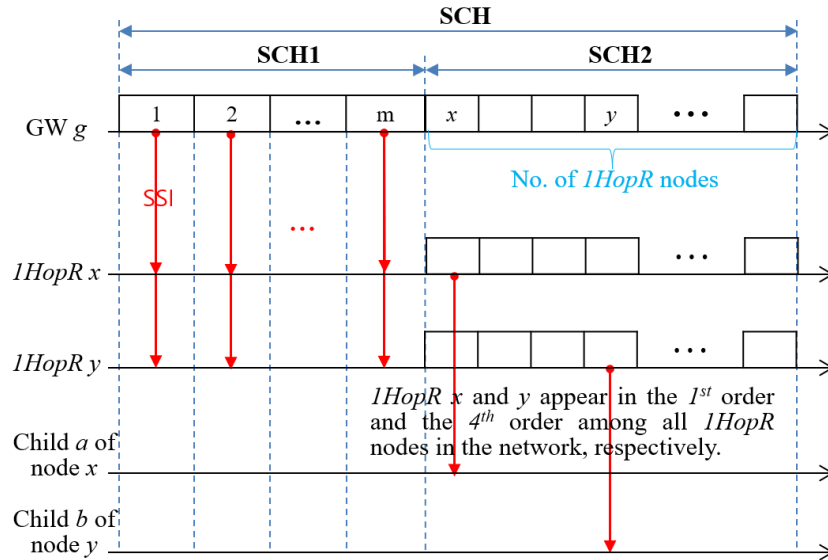


Figure 4.4. Distribution of group scheduling information

During the NI period, a server manages the full tree topology and the profile of all nodes with the received RRs. For the convenience of management, it divides all nodes

into  $m$  groups corresponding to  $m$  channels and distributes scheduling information in groups.

Suppose that group  $i$  has a list of  $k_i$  1-hop nodes as  $(n_{i1}, n_{i1}, \dots, n_{ik_i})$ . Then, a server generates *group slot scheduling information*,  $GSSI(i)$  for group  $i$  as follows:

$$GSSI(i) = (i, (n_{i1}, TSD(n_{i1})), (n_{i2}, TSD(n_{i2})), \dots, (n_{ik_i}, TSD(n_{ik_i}))) \quad (4.2)$$

where  $k_i$  is the number of 1-hop nodes in group  $i$ , and  $n_{ij}$  indicates the  $j^{th}$  1-hop node.

Then, the *network slot scheduling information* ( $NSSI$ ) can be represented in terms of groups as follows:

$$NSSI = (GSSI(1), GSSI(2), \dots, GSSI(m)) \quad (4.3)$$

Then, the server broadcasts the entire  $NSSI$  to all 1-hop nodes by broadcasting *slot scheduling information* ( $SSI$ ) messages  $m$  times in such a way that it broadcasts the  $SSI$ , including  $GSSI(i)$  in the  $i^{th}$  slot among the  $m$  slots of SCH1. If the size of the  $SSI$  is too large, the server segments it and transmits each successively. Then, each  $1HopR$  node,  $x$ , generates  $LSSI(x)$  as follows:

$$LSSI(x) = (g(x), startLSI, PF(x_1), PF(x_2), \dots, PF(x_{k_x})) \quad (4.4)$$

where  $g(x)$  indicates the group to which node  $x$  belongs, and  $startLSI = startLSI(x) + SD(x)$ , and  $k_x$  indicates the number of children of node  $x$ .

This implies that node  $x$  obtains its slot demand starting at  $startLSI(x)$  and its children obtain their slot demands starting at  $startLSI$  in the frame,  $Ch_{g(x)}$ -frame.

To prevent collision during the distribution of  $LSSI$ , every  $1HopR$  node  $x$  broadcasts  $SSI = (LSSI(x))$  in the  $i^{th}$  slot of the SCH2 period if it appears in the  $i^{th}$  order among  $1HopR$  nodes belonging to the  $NSSI$  as shown in Figure 4.4. The slot scheduling for  $1HopR$  and  $2Hop$  nodes follows Algorithm 3.1 and Algorithm 3.2, respectively.

#### 4.3.4. Data transmission and Topology Maintenance

Every node transmits data to its parent according to the slot schedule while a server broadcasts a DL message at the beginning of every frame. This section describes the methods to deal with link failure against node mobility, node joining and leaving, and the update of a slot schedule during data transmission.

##### *a. Link Failure*

A link has a time-varying condition due to node movement, signal interference, or the intervention of obstacles. Thus, a node and its parent should be able to detect and repair link failure and update a slot schedule accordingly.

A node detects downlink failure by making use of data transmission. If node  $x$  has not received data from any child  $y$  for three consecutive frames, it decides that link  $(x, y)$  is broken and updates  $NodeIT(x)$  with  $CS(x) = CS(x) - y$ . If node  $x$  is GW, it releases the slots allocated to node  $y$ . If node  $x$  is of  $IHopR$ , it creates an updated profile,  $uPF(x)$ , for its new link state as follows:

$$uPF(x) = (x, class(x), \{PF(c) \mid c \in CS(x)\}) \quad (4.5)$$

Then,  $IHopR$  node  $x$  sends  $RR = (x, P(x), uPF(x))$  to GW.

To avoid collision, a node transmits RR using an unscheduled slot in its frame. Upon receiving RR, a server generates an *updated slot scheduling information*,  $uSSI(x)$ , for  $IHopR$  node  $x$  as follows:

$$uSSI(x) = (g(x), startLSI(x), uPF(x)) \quad (4.6)$$

Then, the server includes  $uSSI(x)$  in the next DL message so that node  $x$  and node  $x$ 's children can update their slot schedules.

However, it is not possible for a node to detect that the uplink to its parent is down using a DL message. The reason is that since all  $IHopR$  nodes broadcast the same DL

message at the same time, the node does not always receive a DL message its parent node broadcasts. In fact, a *2Hop* node can receive a DL message through various routes, such as a GW, its parent *1HopR* node, or other *1HopR* nodes. So, very conservatively, if a node does not receive a DL message for three consecutive frame periods, it determines that the uplink is down and changes its type to *Orphan*. An additional way for *2Hop* node  $x$  to detect uplink failure is to analyze  $uSSI(P(x))$  in the DL message. If *2Hop* node  $x$  finds itself removed from  $uSSI(P(x))$ , it decides that its uplink  $(x, P(x))$  is down and changes its type to *Orphan*.

### ***b. Node Joining***

To maintain the reliability of data transmission, an *Orphan* node should be able to join a GW or *1HopR* node without interfering with other data transmissions. So, every *1HopR* node  $r$  always includes two parameters:  $jFlag$  and  $jSlotNo$  before sending data as follows:

$$DATA = (r, P(r), jFlag, jSlotNo, payload) \quad (4.7)$$

An *Orphan* node, say  $x$ , that overhears  $DATA$  judges the quality of link  $(x, r)$ , uses  $jFlag$  to determine whether it is possible to join node  $r$ , and if possible, sends a join message using an unscheduled slot,  $jSlotNo$ , to node  $r$  to avoid collision. In this case,  $jFlag = 1$  indicates that node  $r$  can have a new child, and  $jSlotNo$  is an unscheduled slot number in the frame that node  $r$  uses for slot scheduling. The joining process of an *Orphan* node is very similar to the node registration process, except that it uses a collision avoidance technique and has to explore a channel or group to join because they do not use a common channel.

Considering that there are multiple groups using different channels, an *Orphan* node shuffles  $m$  channels to have a *channel explore list* (ChXpList) and tries to overhear  $DATA$  to find a *1HopR* node with good link quality and  $jFlag = 1$  during  $k$  frame

periods, sequentially selecting a channel from ChXpList every UL frame period. Note that ChXpList helps distribute *Orphan* nodes evenly into different groups. An *Orphan* node also receives DL messages using a common channel at the beginning of every frame period for the same  $k$  frame periods. An *Orphan* node,  $z$ , after receiving DL messages, calculates link quality and decides whether or not it can become a *1Hop* node. If node  $z$  can become *1Hop* or *1HopR* node, it transmits RR including  $uPF(z)$  to GW using a randomly selected channel and waits for  $uSSI(z)$  in the next DL message. Otherwise, node  $z$  checks to see if it can be a *2Hop* node based on the *DATA* that it has overheard. Based on the overheard *DATA*, it selects a *1HopR* node with the best link quality and  $jFlag = 1$ , and joins the selected *1HopR* node by sending RR using  $jSlotNo$  on the channel that the *1HopR* node uses. Upon receiving RR, *1HopR* node  $x$  updates  $CS(x) = CS(x) \cup \{z\}$ , generates  $uPF(x)$ , and sends  $RR = (x, P(x), uPF(x))$  to GW. Then, a server registers a new node  $z$ , generates  $uSSI(x)$ , and broadcasts the DL message, including  $uSSI(x)$ .

### c. Slot Scheduling Information Management

Given a list of  $k$  1-hop nodes for group  $i$  as  $(n_{i1}, n_{i2}, \dots, n_{ik})$ , a server maintains a *network information table*,  $NIT(i)$ , as follows:

$$NIT(i) = (n_{ij}, TSD(n_{ij}), PF(n_{ij}), valid) \quad (4.8)$$

where *valid* indicates whether the corresponding entry is valid or not. Then, the start logical slot index  $startLSI(n_{ij})$  of node  $n_{ij}$  in group  $i$  can be calculated easily by (3.9).

A server updates the table whenever it receives an updated profile from any 1-hop node during the data collection period. As mentioned in the subsection of problem identification, the update of the schedule can cause a slot schedule conflict problem. Two solutions to this problem can be considered. First, upon receiving  $uPF(x)$  from a



*IHopR* node  $x$ , a server can produce  $uSSI(x)$  using a list of slots that do not include the slots allocated previously to node  $x$ . In this case, even though any child, say  $y$ , disconnected from node  $x$  sends data continuously, its transmission slot will never conflict with the new slot schedule of node  $x$ . As another method, upon receiving  $uPF(x)$ , a server includes  $Removed(y)$  in the next DL message to indicate that node  $x$  has removed its child  $y$ . If node  $y$  receives it, fortunately, it changes its state to *Orphan* and attempts to join GW or any of relay nodes. In this case, the server may have to broadcast  $Removed(y)$  continuously until it receives  $uPF(y)$  or  $uPF(P(y))$  from node  $y$  or node  $P(y)$  (if node  $y$  found a new parent), respectively. Upon receiving  $uPF(y)$  or  $uPF(P(y))$ , the server generates and broadcasts  $uSSI(x)$  for  $y$ 's previous parent  $x$ . This implies that the new scheduling of node  $x$  waits until node  $y$  finds its new parent, GW or  $P(y)$ . The first method may not work well unless node  $x$  finds a sufficient number of unscheduled slots except for the slots allocated to itself previously. If this happens, it may have to migrate to another group. In the second method, if node  $y$  does not receive the DL message, it may have to wait long by broadcasting the  $Removed(y)$  continuously. One solution is to let node  $y$  change its type to *Orphan* if it misses DL messages for the specified number of frame periods.

In our protocol, the first solution was implemented as follows. A *IHopR* node reports an updated profile to a server if it has a new child joined or loses any of its children, and then turns to a *virtual node* immediately. Upon receiving the updated profile, the server sets the validity of the corresponding entry to *zero* and changes the node to a *virtual node* ( $v$ ) immediately in the *NIT*. Then, the *IHopR* node remains a virtual node, waiting for a new slot scheduling information from the server.

Suppose that a server receives  $RR = (n_{ij}, P(n_{ij}), uPF(n_{ij}))$  from node  $n_{ij}$ . The server changes node  $n_{ij}$  to a virtual node in  $NIT(i)$  and schedules  $TSD(n_{ij})$  either using the slots

occupied by the other virtual nodes that have a sufficient number of slots or starting with  $nextLSI(i)$  as follows:

$$nextLSI(i) = \sum_{x \in G(i)^+} TSD(x) + 1 \quad (4.9)$$

where  $G(i)^+$  is a set of all nodes and virtual nodes that belong to group  $i$ .

Table 4.1. Network Information Table,  $NIT(i)$

(a) Before the update of $n_{i2}$				(a) After the update of $n_{i2}$			
<b><i>1Hop</i></b>	<b><i>TSD</i></b>	<b><i>PF</i></b>	<b>valid</b>	<b><i>1Hop</i></b>	<b><i>TSD</i></b>	<b><i>PF</i></b>	<b>valid</b>
$n_{i1}$	3	$PF(n_{i1})$	1	$n_{i1}$	3	$PF(n_{i1})$	1
$n_{i2}$	5	$PF(n_{i2})$	1	$n_{i2} \rightarrow v_2$	5	-	0
...				...			
$n_{i(j-1)}$	1	$PF(n_{i(j-1)})$	1	$n_{i(j-1)}$	1	$PF(n_{i(j-1)})$	1
$n_{ij} \rightarrow v_1$	5	-	0	$n_{ij} \rightarrow n_{i2}$	3	$PF(n_{i2})$	1
$n_{i(j+1)}$	1	$PF(n_{i(j+1)})$		$n_{ij} \rightarrow v_3$	2	-	0
...				$n_{i(j+1)}$	1	$PF(n_{i(j+1)})$	
$n_{ik}$	1	$PF(n_{ik})$	1	...			
...				$n_{ik}$	1	$PF(n_{ik})$	1
				...			

For example, consider Table 4.1 in which  $NIT(i)$  has one virtual node  $v_1$ . Suppose that  $n_{i2}$  has reported  $uPF(n_{i2})$  with  $TSD(n_{i2}) = 3$  after losing one child. Then, the server allocates  $TSD(n_{i2})$  to a new virtual node  $v_2$  immediately. Then, node  $n_{i2}$  obtains 3 slots from the slots occupied by  $v_1$ , instead of its previous slots that are now occupied by  $v_2$ . Then, the remaining 2 slots out of 5 slots are given to a new virtual node  $v_3$ . A server broadcasts  $uSSI(n_{i2})$  for node  $n_{i2}$  and its children as follows:

$$uSSI(n_{i2}) = (i, startLSI(n_{i2}), uPF(n_{i2})) \quad (4.10)$$

# Chapter 5.

## PERFORMANCE EVALUATION

In this chapter, we present the evaluation works for the Two-hop RT-LoRa protocol. First, we exploit the advantages of the protocol by comparing its characteristics with other multi-hop LoRa protocols. Then, the number of supportable nodes that the protocol can support is calculated through mathematical equations. Finally, the performance of the Two-hop RT-LoRa protocol is evaluated by resorting to simulation and various experiment scenarios.

### 5.1. Analysis

#### 5.1.1. Comparison of Characteristics for multi-hop LoRa Protocols

Table 5.1 compares the features of different multi-hop LoRa protocols. Two-hop RT-LoRa allocates distinct transmission slots to each node by considering signal attenuation of participating nodes, and it allows every node to transmit data within the allocated transmission slots, thereby enabling reliable data transmission. LoRaBlink allows slotted channel access by flooding the network with a beacon message for time synchronization, and it also enables a node to transmit data only at the boundary of the slots. It exploits the notion of concurrent transmission, since multiple nodes can broadcast a beacon message or data simultaneously. However, this approach can still experience collisions if a receiver receives multiple data from two or more nodes simultaneously, where the difference in receiving power is less than a certain threshold. CT-LoRa tries to improve the reliability of data transmissions by giving different time offsets to different concurrent transmitters. However, these two approaches cause high overhead, since they involve the flooding of messages and data. Thus, this may not be

suitable for industrial monitoring and control applications with relatively heavy data traffic. LoRa-Mesh issues a query message whenever a gateway wants to get data from a particular node. In sensor networks, every node tends to send one datum to a gateway periodically. Thus, this way of data acquisition can cause the transmission of too many control messages for data acquisition. In Sync-LoRa-Mesh, a relay node establishes a reliable tree with underground nodes and acquires data from those nodes synchronously by scheduling slots, and then acts as one of the LoRaWAN nodes to forward the collected data to a gateway.

*Table 5.1. Comparison of the features of multi-hop LoRa protocols*

Features	LoRaBlink	CT-LoRa	LoRa-Mesh	Sync-LoRa-Mesh	Two-hop RT-LoRa
Reliability	Data acquisition using directed flooding that exploits the notion of concurrent non-destructive transmission	A flooding-based data acquisition that exploits the notion of timing offset-based concurrent transmission	Data acquisition by constructing a reliable tree and issuing a query to individual nodes	Data acquisition using slot scheduling for underground nodes, but using LoRaWAN for forwarding data to a gateway	Data acquisition using slot scheduling that takes into account signal attenuation by distance and obstruction
	High	High	High	High only for underground nodes	High
Real-time support	Not mentioned	Real-time scheduler can be used easily	Not mentioned	Not supported	Uses a real-time schedule for all nodes
	Not mentioned	Yes	No	No	Yes
Supported number of wireless hops	No limitation	No limitation	No limitation	No limitation	Two hops, but can support up to eight hops with two gateways
Energy consumption	Every node has a low duty cycle, but remains active for one flooding-based data transmission	Every node has a low duty cycle, but remains active during each scheduled slot for one flooding-based data transmission	All nodes remain active during query to set up a path, and the nodes on the path remain active during data transmission	All subnet nodes remain active during the scheduled receiving slots and transmitting slots	Every node remains active only during the scheduled slots
	Mid, but high in high traffic	Mid, but high in high traffic	High	Mid	Low and optimized
Overhead	Transmission of beacon and data	Transmission of slot schedule and data by flooding	Query based data acquisition	Subnet slot scheduling for	One-time distribution of minimal task

	transmission by flooding			synchronous data acquisition	profile information
	High	High	High	Mid	Low
No. of supporting nodes	Not mentioned, but suitable for low traffic	Not mentioned, but suitable for low traffic	Not mentioned, but suitable for low traffic	Limited of number of nodes in a subnet subject to the maximum packet size of LoRaWAN	Maximal support subject to one transmission slot length
Supporting dynamic network	Network construction before data collection	Not required topology construction	Not mentioned	Not mentioned	Network maintenance during data collection
	Yes	Yes	No	No	Yes

In Two-hop RT-LoRa, nodes share their data transmission periods with other nodes via a server so that each node can be allocated transmission slots such that it meets its time constraint. In CT-LoRa, a gateway can generate a real-time schedule for the nodes, and every node can send data by means of flooding without interference within the allocated slot. Other approaches do not consider real-time data transmission. These protocols (except Two-hop RT-LoRa) do not limit the number of supportable wireless hops and nodes, whereas Sync-LoRa-Mesh limits the number of nodes that a relay can support under a constraint on the data size that LoRaWAN allows in a single transmission. In Two-hop RT-LoRa, one gateway allows only two wireless hops, since more wireless hops can lead to higher amounts of data traffic. Note that data in a 2-hop node has to be transmitted twice to reach the gateway, and each transmission can easily interfere with other transmissions in long-range LoRa networks. In CT-LoRa, data are transmitted to the gateway using flooding. All nodes have to remain active during data transmission, and they retransmit data in a concurrent manner. In LoRaBlink, nodes control data retransmissions using the hop distance to the gateway such that a node retransmits data only if it has a shorter hop distance to the gateway than the sender. Two-hop RT-LoRa optimizes energy consumption, since it allows a node to remain

active during the assigned slots, and it uses the optimal aggregated transmission. One most important feature of Two-hop RT-LoRa is using slot scheduling-based data transmission that does not allow data collisions. Therefore, if a slot size is optimized, the optimal number of nodes can be supported. In LoRaBlink and CT-LoRa, flooding time becomes a duty cycle that is optimized by exploiting concurrent transmission. Nevertheless, a single datum is retransmitted many times by many nodes until it arrives at the gateway. Among those protocols, only Two-hop RT-LoRa considers the applications in dynamic networks and supports node mobility. Even though LoRaBlink and CT-LoRa can easily support dynamic networks since they are based on the flooding approach. However, they cause a lot of network overhead that reduces the network performance significantly.

According to the discussion so far, Two-hop RT-LoRa enables energy-efficient, reliable, real-time transmission, and can support a considerable number of nodes by completely removing the possibility of collision. In consequence, it can satisfy the requirements of industrial applications.

### 5.1.2. Number of Supportable Nodes

The UL data format consists of a preamble, the payload, and a 16-bit payload cyclic redundancy check (CRC). Suppose that  $symbols(X)$  and  $bytes(X)$  denote the number of symbols and bytes of message  $X$ , respectively. Then, the preamble duration,  $T_{preamble}$ , and the payload duration,  $T_{payload}$ , are given as follows:

$$\begin{aligned} T_{preamble} &= (symbols(preamble) + 4.25) * T_{sym}, \text{ and} \\ T_{payload} &= (symbols(payload + CRC)) * T_{sym} \end{aligned} \quad (5.1)$$

where  $T_{sym} = 2^{SF} / BW$ , and 4.25 indicates a fixed number of symbols including two up-chirps, two down-chirps, and a 1/4 up-chirp that are transmitted in the preamble on-air

according to the LoRa modulation specifications [45], and  $symbols(payload + CRC)$  is given as follows:

$$symbols(payload + CRC) = 8 + \max(ceil\left[\frac{8PL - 4SF + 28 + 16 - 20H}{4(SF - 2DE)}\right], (CR + 4), 0) \quad (5.2)$$

where  $PL$  is the payload in *bytes*,  $SF$  is a spreading factor,  $H = 0$  and  $H = 1$  when a header is enabled and disabled, respectively,  $DE = 0$  and  $DE = 1$  when low data rate optimization is enabled and disabled, respectively, and  $CR$  is the coding rate in [1, 4].

The lower bound of the UL slot length,  $slotLen_{LB}$ , is given as follows:

$$slotLen_{LB} \geq T_{preamble} + T_{payload} \quad (5.3)$$

Then, the lower bounds of the UL slot length for different payloads with the parameters  $symbols(preamble) = 8$ ,  $SF = 7$ ,  $BW = 125$ , and  $CR = 1$  are summarized in Table 5.2, where  $T_{slot} \geq 107.78$  ms when  $PL = 60$  bytes.

Table 5.2. The lower bound of the UL slot based on payload (SF7, BW125)

PL (bytes)	30	60	90	120
$slotLen_{LB}(ms)$	66.82	107.78	153.86	199.94

Let  $\alpha$  be the proportion of 1-hop nodes in the network ( $0 < \alpha \leq 1$ ). For simplicity, assume that every node generates one packet during a frame period. Then, the *network slot demand*,  $NetSD(n)$ , that  $n$  nodes require during a frame period is given as follows:

$$NetSD(n) = \alpha * n + 2(1 - \alpha) * n = (2 - \alpha) * n \leq 2^N \quad (5.4)$$

In other words,  $nSupportableNodes$  is the number of nodes that the network can support on a single channel, given as follows:

$$nSupportableNodes = n \leq \frac{2^N}{(2 - \alpha)} \quad (5.5)$$

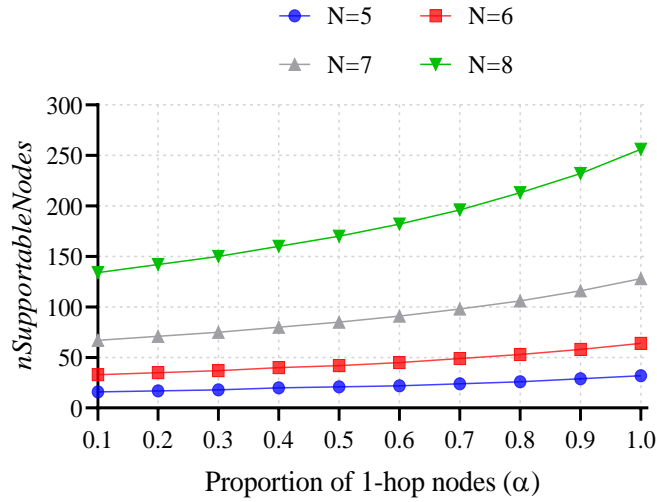


Figure 5.1. Number of supportable nodes with varying  $\alpha$  value ( $BW=125$ ,  $CR=1$ ,  $SF=7$ )

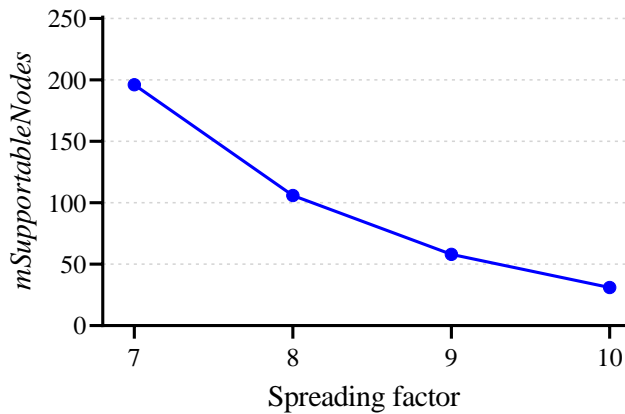


Figure 5.2. Number of supportable nodes for different SFs ( $BW=125$ ,  $CR=1$ ,  $N = 8$ ,  $\alpha = 0.7$ )

Based on (5.5), Figure 5.1 illustrates the number of nodes that a single-channel GW can support with varying  $N$  and  $\alpha$  when  $PL = 30$  bytes and  $SF = 7$ . It shows that  $nSupportableNodes$  increases linearly with  $\alpha$ . If  $N = 8$  and  $\alpha = 0.7$ , a single-channel GW can support 200 nodes. Note that the minimum frame size with  $N = 8$  (= 256 slots) will become 17,105 ms. In Figure 5.2,  $nSupportableNodes$  decreases logarithmically according to the increase in SF with  $N = 8$  and  $\alpha = 0.7$ . This is because the increase in SF by unity doubles the slot length.



### 5.1.3. Energy Consumption

In Two-hop RT-LoRa, data can be transmitted in two hops, rather than one hop, to a gateway to improve reliability. We analyze how this way of data transmission can affect energy consumption by comparing energy consumption in a two-hop transmission and in a one-hop transmission.  $E_{twohop}$  and  $E_{onehop}$ , as energy consumptions for two-hop transmission and one-hop transmission, respectively, can be expressed as follows:

$$E_{twohop} = (2 * P_{Tx} + P_{Rx}) * PktToA(SF_i), \text{ and} \quad (5.6)$$

$$E_{onehop} = P_{Tx} * PktToA(SF_i) \quad (5.7)$$

where  $P_{Tx}$  and  $P_{Rx}$  indicate transmitting power and receiving power of a packet, respectively, and  $PktToA(SF_i)$  indicates the time on air of a transmitted packet when  $SF_i$  is used.

In this analysis, the Semtech SX1276 working at 868 MHz is used as an energy reference model [45] where, at 25° C with input voltage is 3.3 V, the supply current values for a transmitter vary according to transmission power, and those for the receiver vary according to the bandwidth, as seen in Table 5.3.

*Table 5.3. Supply current values for a transmitter*

Transmitter	$P_{Tx}$ (dBm)	7	13	17	
	$I_{Tx}$ (mA)	20	28	90	
Receiver	BW (kHz)	≤62.5	125	250	500
	$I_{Rx}$ (mA)	9.9	10.3	11.1	12.6

Based on (5.6) and (5.7), Figure 5.3 compares energy consumption for two-hop(SF7), which transmits a packet in two hops with SF7, and for one-hop(SF8), one-hop(SF9), and one-hop(SF10), which transmit packets directly to the gateway using

SF8, SF9, and SF10, respectively, in terms of energy consumption when varying the size of the payload. The energy consumption of two-hop(SF7) is higher than that of one-hop(SF8) to some degree, but the difference is not significant; however, two-hop(SF7) consumes far less energy than both one-hop(SF9) and one-hop(SF10). Thus, we conclude that it is desirable to use two-hop(SF7) for industrial zones in which nodes can experience high signal attenuation.

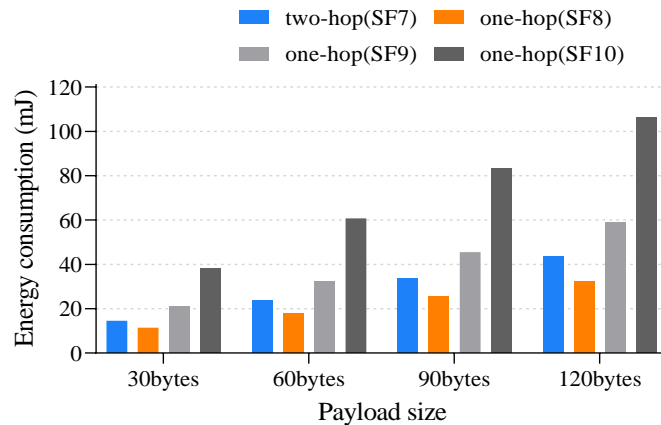


Figure 5.3. Comparison of energy consumption for one-hop and two-hop transmissions with varying payload sizes ( $BW=125$  kHz,  $CR=1$  and  $P_{tx}=13$  dBm)

## 5.2. Simulation

### 5.2.1. Channel model

The simulation uses the log-distance path loss model with shadowing [47], which is widely used to model wireless channels in built-up and densely populated areas. Using this model, the path loss at communication distance  $d$  is described as follows:

$$PL(d) = PL(d_0) + 10\gamma \log_{10} \frac{d}{d_0} + X_{\sigma} \quad (5.8)$$

where  $PL(d_0)$  is the path loss at reference distance  $d_0$ ,  $\gamma$  is the pass loss exponent,  $d$  is the transmission distance ( $d > d_0$ ), and  $X_{\sigma}$  is the zero-mean Gaussian distributed variable with standard deviation  $\sigma$ .

$$p(X_\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{X_\sigma^2}{2\sigma^2}\right) \quad (5.9)$$

The received power at distance  $d$ ,  $P_r(d)$ , is calculated based on transmission power  $P_t$  and path loss at distance  $d$  as follows:

$$P_r(d) = P_t - PL(d) \quad (5.10)$$

Assuming that the LoRa signal can be demodulated if the received power is greater than or equal to receiving sensitivity of the receiver,  $P_{\min}$ . The probability of receiving a packet at distance  $d$ ,  $p_{\text{receiving}}(d)$ , can be calculated as follows:

$$p_{\text{receiving}}(d) = p(P_r(d) \geq P_{\min}) \quad (5.11)$$

Refer to section 2.7.2 in [47]:

$$p_{\text{receiving}}(d) = Q\left(\frac{P_{\min} - P_r(d)}{\sigma}\right) \quad (5.12)$$

where the Q function is defined as the probability that a Gaussian random variable  $x$  with mean zero variance one is greater than  $z$ :

$$Q(z) = p(x > z) = \int_z^\infty \frac{1}{\sqrt{2\pi}} e^{-y^2/2} dy \quad (5.13)$$

### 5.2.2. Simulation setup

For simulation, 500 nodes are randomly distributed over a rectangular area of 800x800 m<sup>2</sup>, and one gateway is placed in the center of the area, as illustrated in Figure 5.4. The blue-filled circle indicates the GW transmission range. The two-hop LoRa network is constructed under the assumption that the connectivity between two nodes exists only if the link provides a receiving probability greater than 95%. The channel parameters are referred to the experiment results in [48], with  $d_0 = 1$  m,  $PL(d_0) = 40.7$  dB,  $\gamma = 3.54$ , and  $\sigma = 5.34$ . The scenario has additional 20 end nodes that move around

the deployment area. Every mobile node moves arbitrarily at the speed of 2 m/s that models the movement of workers carrying sensor nodes, and takes the pausing time that follows a Poisson process with an expected value  $\lambda$  in minutes.

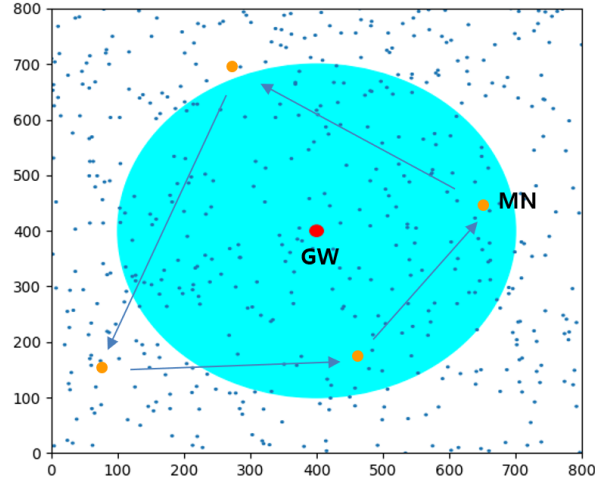


Figure 5.4. Example of spatial distribution with 500 end nodes in an area of  $800 \times 800 (m^2)$

The key parameters and values are listed in Table 5.4.

Table 5.4. Simulation parameters

Parameter	Value	Parameter	Value
No. GWs	1	Data rate (SF, BW)	7, 125
No. static nodes	500	Frame factor	7
No. mobile nodes	20	UL slot size	100 (ms)
UL payload size	50 (bytes)	DL slot size	200 (ms)
TP, class	13.2 s, 0	Frame size	13.2 (s)
Expected pausing time ( $\lambda$ )	5 to 25	No. Data channels	8

### 5.2.3. Results and Discussions

Some performance evaluation metrics are used as follows. The *packet delivery rate* (*PDR*) is the ratio of the number of packets received successfully by GW to the number of packets generated by all end nodes during simulation. Additionally, it may be

meaningful to evaluate the quality of links. Thus, the packet delivery rate for the nodes except orphan nodes is evaluated under the premise that *Orphan* nodes do not transmit data, referred to as *PDR\_NoOrphan* that is defined as the ratio of the number of packets received successfully at GW to the number of packets transmitted by all end nodes.

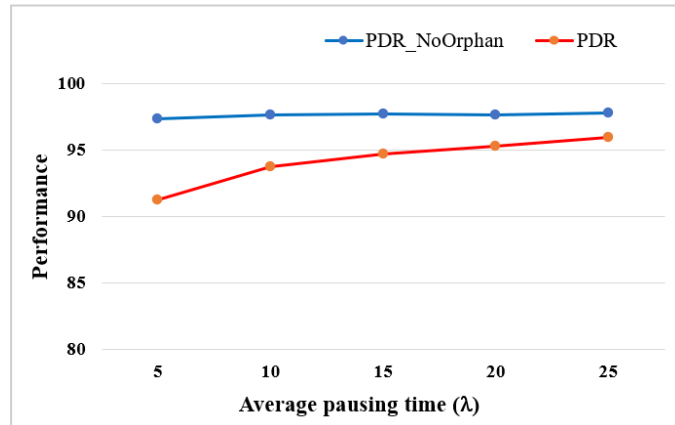


Figure 5.5. The average and PDR and PDR\_NoOrphan with different  $\lambda$

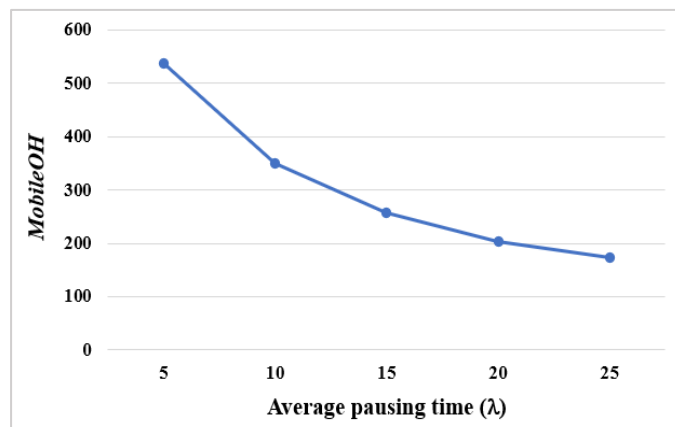


Figure 5.6. The average MobileOH with different  $\lambda$

Simulations were performed over a period of 10,000 frames (= 132,000 s), changing the value of  $\lambda$  from 5 to 25 mins. Figure 5.5 shows the average *PDR* and *PDR\_NoOrphan* of 20 mobile nodes. It is seen that the average *PDR\_NoOrphan* remains high above 97% for all values of  $\lambda$ , while the average *PDR* increases from 91.2% to 95.9% as the value of  $\lambda$  increases from 5 to 25. The gap between two graphs implies that packet loss caused by the transmission of orphan nodes cannot be ignored.

Additionally, control overheads of the mobile node (*MobileOH*), which is defined as the number of control packets transmitted by the mobile node and its parent, was measured during simulation. Figure 5.6 shows the average *MobileOH* of 20 mobile nodes for different  $\lambda$  values. It is shown that *MobileOH* decreases sharply as  $\lambda$  increases up to 25. When  $\lambda = 5$ , the mobile node needs more than 500 control packets during simulation, which is 5% of the number of packets generated.

### 5.3. Experiment

#### 5.3.1. The effectiveness of two-hop LoRa network

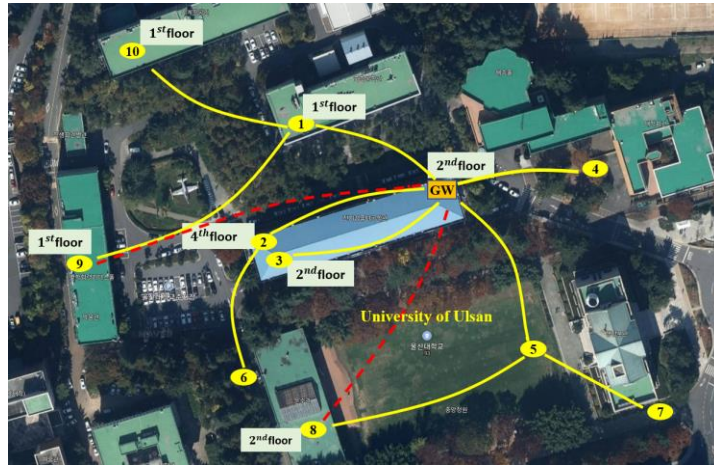
##### a. Experiment Setup

A testbed that consisted of one GW and 10 end nodes was constructed for an experiment on the campus of the University of Ulsan, where the GW was placed inside the laboratory of the computer engineering building, and the end nodes were positioned manually, such that some radio signals were highly attenuated by multiple layers of concrete walls and buildings. Every node transmitted a packet of 30 bytes per data collection cycle of 6.8 seconds. The experiment was performed over 500 data collection cycles, recognizing the PRR of every node. The key parameters and values are summarized in Table 5.5.

Table 5.5. Experiment Parameters

Parameter	Value	Parameter	Value
No. nodes	10	UL slot size	100 ms
Payload size	30 bytes	DL slot size	200 ms
No. UL channels	1	Frame size	6.8 s
Frame factor	7	Data rate	SF7, BW125
TP, class	6.8 s, 0	CR	1

*b. Results and Discussions*



*Figure 5.7. The testbed deployed on the University of Ulsan*

Figure 5.7 shows a map of the end nodes deployed on campus after network construction, where the GW was located inside our laboratory, nodes 1, 2, 3, 8, 9, and 10 were inside buildings, and nodes 4, 5, 6, and 7 were in open areas of the campus. The solid lines and the dashed lines indicate stable, unstable (or bad) links to the GW, respectively. The nodes with bad link quality became 2-hop nodes that connected to an 1-hop relay node. For example, nodes 7 and 8 became 2-hop nodes that used node 5 as an 1-hop relay.

*Table 5.6. Experiment results from using SF7*

<b>Node ID</b>	<b>Hop distance</b>	<b>PRR (%)</b>
1	1	99
2	1	99
3	1	95
4	1	100
5	1	98
6	2	95 (0)
7	2	94 (0)
8	2	97 (28)
9	2	98 (11)
10	2	97 (0)

---

Note: The value in parenthesis indicates PRR when the node transmitted a packet directly to the GW

Experiments were performed by assigning SF7 to every node, and the results are shown in Table 5.6. With construction of a two-hop network, the gateway could achieve PRRs above 95% for 1-hop nodes, and above 94% for 2-hop nodes. The PRRs for 2-hop nodes are clearly compared with those for the same nodes that transmitted packets directly to the GW (given in parentheses).

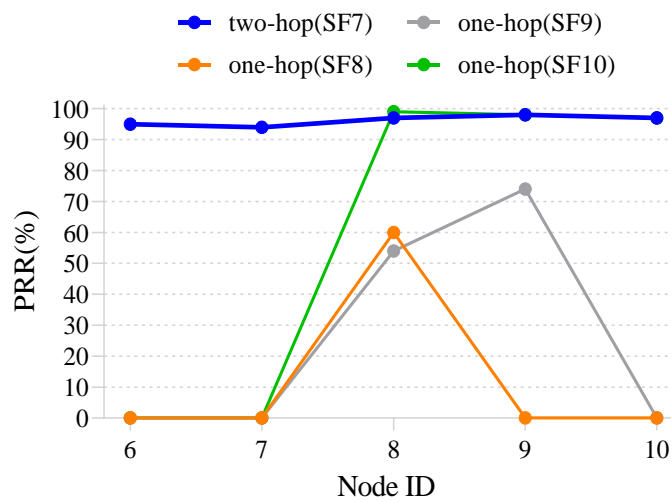


Figure 5.8. PRRs with 2-hop nodes using different SFs

More experiments were performed by having the 2-hop nodes transmit packets directly (i.e., in a one-hop transmission) to the GW by increasing the SF, as shown in Figure 5.8. In that figure, two-hop(SF7) and one-hop(SF $x$ ) indicate the PRRs when a two-hop transmission was executed with SF7 and when a one-hop direct transmission was executed by using SF $x$ . Even though the 2-hop nodes used the higher SF, say SF10, the improvement in PRR was limited, showing that nodes 6 and 7 still failed to send packets. This implies that two-hop transmission is much more effective in terms of energy consumption and transmission reliability.



### 5.3.2. Dynamic experiment scenarios

#### a. Experiment I with static node scenario



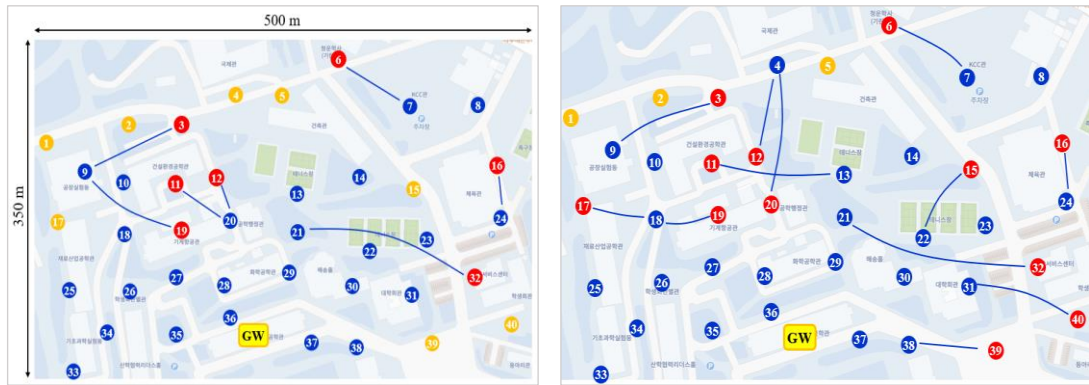
Figure 5.9. Photos of GW and some nodes installed

In the static node scenario, 1 GW and 40 nodes were deployed on a 500 x 350 (m<sup>2</sup>) area of a university campus. Each node receives GPS data, wind speed, wind direction, temperature and humidity from the weather detection system and sends those data to the GW periodically. Since some nodes were installed in communication shadow areas such as valleys on campus, inside buildings, and behind buildings, many nodes could not be directly connected to the GW. As shown in the upper left photo of Figure 5.9, the GW was installed in the lecture room on the 6<sup>th</sup> floor of the Computer Science Building, with the antenna exposed to the outside. The key experiment parameters and values are summarized in Table 5.7.

Table 5.7. Experiment Parameters and Values

Parameter	Value	Parameter	Value
UL slot size	100 ms	Data rate	SF7, BW125
DL slot size	200 ms	Frame Size	13.2 s
Frame factor	7	TP, class	13.2 s, 0
No. UL channel	1	Payload size	24 bytes

Figure 5.10-(a) shows the two-hop topology constructed with  $(RSSI\_Th1, SNR\_Th1) = (-110 \text{ dBm}, -3.5 \text{ dB})$  and  $(RSSI\_Th2, SNR\_Th2) = (-115 \text{ dBm}, -5.5 \text{ dB})$  immediately after network construction. The blue colored, brown-colored, and red-colored circles indicate  $1HopR$ ,  $1Hop$ , and  $2Hop$  nodes, respectively.



(a) Node connections after network initialization

(b) Node connections after 8 hours of operation



(c) Node connections after 16 hours of operation

(d) Node connections after 24 hours of operation

Figure 5.10. Static node deployment scenario and the change of topology

The experiment was conducted for several days, recording network connection status every 8 hours as shown in Figure 5.10-(b), (c), and (d). As shown in the figures, nodes 4, 5, 7, 15, and 20 change connections frequently due to their link instability. The reason is that node 20 was placed low behind the Chemical Engineering Building that blocks the building having the GW, and nodes 4, 5, and 15 were placed quite far from the GW and intercepted by several buildings and trees. Node 7 was placed in an open

area but blocked by the hills because it is placed low. Nodes 20 and 7 started out as *1HopR* nodes and turned into *2Hop* nodes after 8 and 16 hours, respectively, and then node 20 turned into *1Hop* nodes after 24 hours.

Table 5.8. Experiment results

	After initialization	After 8 hours	After 16 hours	After 24 hours
No. 1-hop nodes	33	28	25	26
No. 2-hop nodes	7	12	15	14
AVG. <i>PDR_NoOrphan</i> (1-hop)	N.A.	99.4%	99.4%	99.2%
Worst <i>PDR_NoOrphan</i> (1-hop)	N.A.	97.9%	97.7%	95.9%
AVG. <i>PDR_NoOrphan</i> (2-hop)	N.A.	97.5%	97.7%	98.5%
Worst <i>PDR_NoOrphan</i> (2-hop)	N.A.	92.2%	93.4%	95.2%

Table 5.8 summarizes the number of 1-hop nodes, the number of 2-hop nodes, and the lowest and average *PDR\_NoOrphan* values among the *PDR\_NoOrphans* of 1-hop and 2-hop nodes. The proposed protocol achieves a high *PDR\_NoOrphan* of over 99% on average for 1-hop nodes and over 97% for 2-hop nodes. Node 15 had a low *PDR\_NoOrphan* of 92.2% because its one-hop connection was unstable during the first 8 hours. However, it can be seen that the *PDR\_NoOrphan* continued to increase after node 15 became a 2-hop node or a 1-hop node with a stable connection. A similar situation occurred on node 7. After 8 hours, node 7 connected to node 8 to become a *2Hop* node. After 24 hours, it is shown that all 1-hop or 2-hop nodes could achieve a high *PDR\_NoOrphan* of 95% or more.

### b. Experiment II with mobile node scenario

In this experiment, two additional mobile nodes, 41 and 42, were allowed to travel along their respective predetermined paths over the previous static node scenario. At the starting point of the experiment, node 41 is installed inside the vehicle and travels along a green curved path starting at position L1 at a speed of 6 km/h ( $\approx 1.7$  m/s) as shown in Figure 5.11-(a). After about 30 minutes, one of our lab members carried node 42 and moved from the starting point L2 along the blue curved path at about 4 km/h ( $\approx 1.1$  m/s), which corresponds to the normal walking speed, as shown in Figure 5.11-(b).

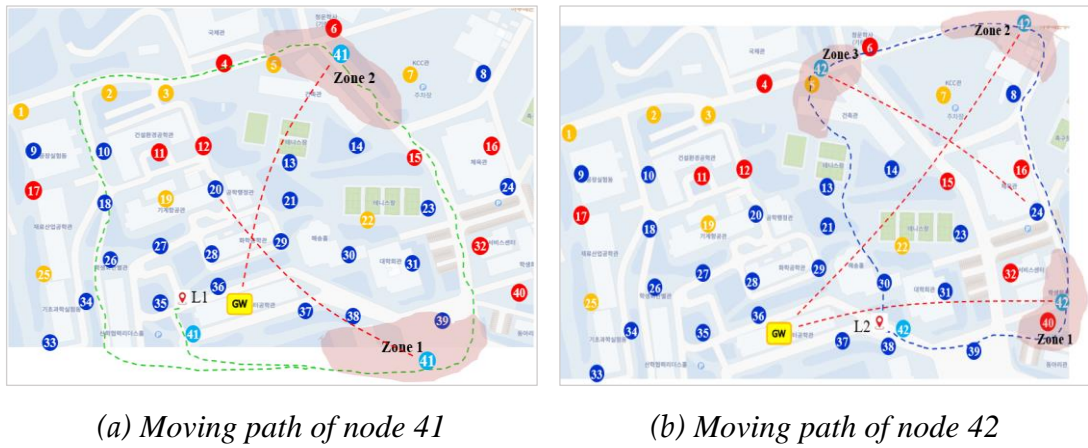


Figure 5.11. The change of connections according to node movement (The red-colored dashed lines indicate the unreliable connection)

Two mobile nodes moving along their respective paths performed data transmission while repeating the process of joining and leaving the already established and operating fixed LoRa network as in experiment I. Mobile nodes, 41 and 42, move 4 laps and 1 lap, respectively, along their respective paths.

Table 5.9 summarizes the total experiment time, number of parent changes,  $PDR\_NoOrphan$ , and  $PDR$  for two mobile nodes. It is shown that two mobile nodes achieve a high  $PDR\_NoOrphan$  of over 90%. Taking advantage of the *augmented transmission effect*, the  $PDR\_NoOrphan$  is improved to over 95%, the value in parentheses. This means that mobile nodes can transmit data quite reliably, even though

they are constantly on the move. Note that two mobile nodes take a large difference of nearly 10% in PDR. The reason is that the slowly moving node 42 takes link disconnection time longer than mobile node 41.

*Table 5.9. Experiment results*

<b>Node ID</b>	<b>41</b>	<b>42</b>
Experiment time	1 hour	25 minutes
Number of parent changes	1	2
<i>PDR_NoOrphan</i>	93.3 (97.4)	90.2 (96.5)
<i>PDR</i>	90.3 (94.3)	80.8 (86.4)

Referring to Figure 5.11-(a), let us explain how mobile node 41 changes its node type. Starting as a *1Hop* type at point L1, mobile node 41 maintains a *1Hop* node during its first and second laps even though it has experienced link instability temporarily during these laps. However, in the third lap, when node 41 enters zone 2 pass zone 1 and its link becomes unstable, it became an orphan since it failed to receive the DL message three times in a row. Then, it gets node 20 as a new parent to be a *2Hop* type. In the fourth lap, when node 41 enters zone 1, it keeps the link to its parent despite that it is unstable temporarily. Then, it remains a *2Hop* type by the end of experiment.

Take a look at Figure 5.11-(b) in which node 42 takes only one lap at a walking speed. Node 42 starts as an *1Hop* type at point L2. When it enters zone 1, its link becomes unstable, but remains same. Upon entering zone 2, node 42 changed its parent to node 24 since it failed to receive the DL message three times in a row. However, when it enters zone 3, it changes its parent back to GW due to its link instability. It is natural that node 42 changes its parent more since it maintains the state of link disconnection longer by moving more slowly than node 41.

## Chapter 6.

# CONCLUSIONS AND FUTURE WORKS

### 6.1. Conclusions

This dissertation presented the design and implementation of a multi-hop real-time LoRa protocol, Two-hop RT-LoRa, for industrial monitoring and control systems that not only require real-time, reliable, and energy-efficient data transmission but also be able to deal with topology change in a dynamic network. The protocol is based on a two-hop tree topology and a real-time slot schedule of a two-hop tree topology. In real-time IoT applications, the proposed protocol can allocate slots easily to 1-hop and 2-hop end nodes such that every node meets its time constraint if it transmits packets in the allocated slots. Furthermore, the protocol can overcome the problem of packet loss from signal attenuation by enabling two-hop transmission. For energy saving, a 1-hop relay node can minimize the number of transmissions by using data aggregation. It was shown by analysis that our protocol can support hundreds of nodes on a single channel. Since the protocol uses a slot-scheduled approach, it can achieve high reliability in data transmissions, regardless of the number of deployed nodes.

Furthermore, the protocol also addressed the technical aspects of dynamic multi-hop networks, such as automatic configuration of multi-hop LoRa networks, dynamic topology management, and updating of real-time slot schedules and dealt with some technical issues related to node mobility. By resorting to both simulation and experiment, it was verified that the proposed protocol could achieve high reliability against signal attenuation and node mobility, but with low control overhead in topology management and schedule updates. According to experimental results with university

campus deployment scenarios with forty nodes, the proposed protocol could achieve packet delivery rates of over 95% against node mobility.

## **6.2. Future Works**

In the proposed protocol, we suggest using a single data rate setting and constructing a two-hop network topology that enables relaying mechanisms. It was proved that the use of a high data rate with a two-hop network topology not only extends the network coverage effectively but also provides more efficient energy management compared to the use of single-hop with lower data rates. However, this approach did not fully exploit the SF orthogonality property, one of the advantages of LoRa technology that enable the use of multiple SFs to create virtual channels and thus, it reduces the network throughput. A future direction of our work is to enable multiple data rate settings in the protocol operation to improve the network throughput.

Furthermore, the protocol constructs a tree topology that is limited to a two-hop connection from GW. In some special situations, the protocol should allow a higher number of hop count and thus, the network coverage can be further extended without the deployment of additional GWs. Since the protocol uses a slot schedule for data transmission in which time slots are assigned to data transmissions to remove the data collision. This slot scheduling approach does not allow slot reuse mechanism and thus it limits the supported network traffic. A slot reuse solution among multiple trees or even in a single tree should be an interesting research topic in our future works.

## PUBLICATIONS

- [1] H. P. Tran, W. Jung, T. Yoon, D. Yoo, and H. Oh, “A twohop real-time LoRa protocol for industrial monitoring and control systems”, IEEE Access, vol. 8, pp. 126239-126252, 2020.
- [2] Huu Phi Tran and Hoon Oh, “Implementation and Evaluation of Multi-hop LoRa Networks”, Journal of Communications vol. 17, no. 4, pp. 273-279, April 2022.  
Doi: 10.12720/jcm.17.4.273-279
- [3] Tran, H.P.; Jung, W.-S.; Yoo, D.-S.; Oh, H. “Design and Implementation of a Multi-Hop Real-Time LoRa Protocol for Dynamic LoRa Networks”, Sensors 2022, 22, 3518. <https://doi.org/10.3390/s22093518>
- [4] Quy Lam Hoang, Huu Phi Tran, Woo-Sung Jung, Si Hong Hoang, Hoon Oh, “A Slotted Transmission with Collision Avoidance for LoRa Networks”, Procedia Computer Science, Volume 177, 2020, Pages 94-101, ISSN 1877-0509, <https://doi.org/10.1016/j.procs.2020.10.016>.



## REFERENCES

- [1] S. Cop., "SX1276/77/78/79 - 137 MHz to 1020 MHz Low Power Long Range Transceiver," *Datasheet*, 2020.
- [2] S. Corporation, "LoRa and LoRaWAN: A Technical Overview," 2019.
- [3] B. Reynders and S. Pollin, "Chirp spread spectrum as a modulation technique for long range communication," in *2016 Symposium on Communications and Vehicular Technologies (SCVT)*, 2016, pp. 1-5.
- [4] C. Wang, D. Sklar, and D. Johnson, "Forward Error-Correction Coding," *Crosslink*, vol. 3, pp. 26-29, 2001.
- [5] L. Alliance, "LoRaWAN TM 1.1 Specification," 2017.
- [6] 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*, vol. 16, no. 9, 2016.
- [7] C. Cordeiro and D. Agrawal, "Ad Hoc and Sensor Networks Theory and Applications 2nd Edition," 2011.
- [8] "IEEE Standard for Low-Rate Wireless Networks," *IEEE Std 802.15.4-2020 (Revision of IEEE Std 802.15.4-2015)*, pp. 1-800, 2020.
- [9] Semtech, "AN1200.22 LoRa Modulation Basics," 2015.
- [10] J. Song *et al.*, "WirelessHART: Applying Wireless Technology in Real-Time Industrial Process Control," in *2008 IEEE Real-Time and Embedded Technology and Applications Symposium*, 2008, pp. 377-386.
- [11] *ISA100.11a Standard, Wireless Systems for Industrial Automation: Process and Related Applications*, 2009.
- [12] W. Song, R. Huang, B. Shirazi, and R. LaHusen, "TreeMAC: Localized TDMA MAC protocol for real-time high-data-rate sensor networks," in *2009 IEEE*

- International Conference on Pervasive Computing and Communications*, 2009, pp. 1-10.
- [13] H. Oh and P. Van Vinh, "Design and implementation of a MAC protocol for timely and reliable delivery of command and data in dynamic wireless sensor networks," (in eng), *Sensors (Basel, Switzerland)*, vol. 13, no. 10, pp. 13228-13257, 2013.
- [14] H. Oh and C. T. Ngo, "A Slotted Sense Multiple Access Protocol for Timely and Reliable Data Transmission in Dynamic Wireless Sensor Networks," *IEEE Sensors Journal*, vol. 18, no. 5, pp. 2184-2194, 2018.
- [15] T. Nguyen and H. Oh, "A Smart Multi-Channel Slotted Sense Multiple Access Protocol for Industrial Wireless Sensor Networks," *IEEE Internet of Things Journal*, vol. PP, pp. 1-1, 12/21 2021.
- [16] N. Abramson, "THE ALOHA SYSTEM: another alternative for computer communications," *Fall Joint Computer Conference*, vol. 37, pp. 281-285, 01/01 1977.
- [17] M. Bor, U. Roedig, T. Voigt, and J. Alonso, *Do LoRa Low-Power Wide-Area Networks Scale?* 2016.
- [18] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne, "Understanding the Limits of LoRaWAN," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 34-40, 2017.
- [19] F. V. d. Abeele, J. Haxhibeqiri, I. Moerman, and J. Hoebeke, "Scalability Analysis of Large-Scale LoRaWAN Networks in ns-3," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 2186-2198, 2017.
- [20] A. Mahmood, E. Sisinni, L. Guntupalli, R. Rondón, S. A. Hassan, and M. Gidlund, "Scalability Analysis of a LoRa Network Under Imperfect

- Orthogonality," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 3, pp. 1425-1436, 2019.
- [21] B. R. e. al, "Improving Reliability and Scalability of LoRaWANs Through Lightweight Scheduling," *IEEE INTERNET OF THINGS JOURNAL*, vol. 5, 2018.
- [22] M. Rizzi, P. Ferrari, A. Flammini, E. Sisinni, and M. Gidlund, "Using LoRa for industrial wireless networks," in *2017 IEEE 13th International Workshop on Factory Communication Systems (WFCS)*, 2017, pp. 1-4.
- [23] T. Polonelli, D. Brunelli, A. Marzocchi, and L. Benini, "Slotted ALOHA on LoRaWAN-Design, Analysis, and Deployment," *Sensors*, vol. 19, p. 838, 02/18 2019.
- [24] R. Piyare, A. L. Murphy, M. Magno, and L. Benini, "On-Demand LoRa: Asynchronous TDMA for Energy Efficient and Low Latency Communication in IoT," *Sensors*, vol. 18, no. 11, 2018.
- [25] J. Haxhibeqiri, I. Moerman, and J. Hoebeke, "Low Overhead Scheduling of LoRa Transmissions for Improved Scalability," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3097-3109, 2019.
- [26] L. Leonardi, F. Battaglia, G. Patti, and L. L. Bello, "Industrial LoRa: A Novel Medium Access Strategy for LoRa in Industry 4.0 Applications," in *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, 2018, pp. 4141-4146.
- [27] L. Leonardi, F. Battaglia, and L. L. Bello, "RT-LoRa: A Medium Access Strategy to Support Real-Time Flows Over LoRa-Based Networks for Industrial IoT Applications," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10812-10823, 2019.

- [28] Q. L. Hoang, W. S. Jung, T. Yoon, D. Yoo, and H. Oh, "A Real-Time LoRa Protocol for Industrial Monitoring and Control Systems," *IEEE Access*, vol. 8, pp. 44727-44738, 2020.
- [29] D. Zorbas, K. Abdelfadeel, P. Kotzanikolaou, and D. Pesch, "TS-LoRa: Time-slotted LoRaWAN for the Industrial Internet of Things," *Computer Communications*, vol. 153, pp. 1-10, 2020/03/01/ 2020.
- [30] D. Croce, M. Gucciardo, S. Mangione, G. Santaromita, and I. Tinnirello, "Impact of LoRa Imperfect Orthogonality: Analysis of Link-Level Performance," *IEEE Communications Letters*, vol. 22, no. 4, pp. 796-799, 2018.
- [31] J. Haxhibeqiri, A. Karaagac, F. V. d. Abeele, W. Joseph, I. Moerman, and J. Hoebeke, "LoRa indoor coverage and performance in an industrial environment: Case study," in *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2017, pp. 1-8.
- [32] J. Petäjälä, K. Mikhaylov, M. Hämäläinen, and J. Iinatti, "Evaluation of LoRa LPWAN technology for remote health and wellbeing monitoring," in *2016 10th International Symposium on Medical Information and Communication Technology (ISMICT)*, 2016, pp. 1-5.
- [33] R. Sanchez-Iborra, J. Sanchez-Gomez, J. Ballesta-Viñas, M.-D. Cano, and A. F. Skarmeta, "Performance Evaluation of LoRa Considering Scenario Conditions," *Sensors*, vol. 18, no. 3, 2018.
- [34] M. Bor, J. Vidler, and U. Roedig, "LoRa for the Internet of Things," *EWSN*, pp. 361-366, 02/15 2016.
- [35] C. Liao, G. Zhu, D. Kuwabara, M. Suzuki, and H. Morikawa, "Multi-Hop LoRa Networks Enabled by Concurrent Transmission," *IEEE Access*, vol. 5, pp. 21430-21446, 2017.

- [36] H. Lee and K. Ke, "Monitoring of Large-Area IoT Sensors Using a LoRa Wireless Mesh Network System: Design and Evaluation," *IEEE Transactions on Instrumentation and Measurement*, vol. 67, no. 9, pp. 2177-2187, 2018.
- [37] C. Ebi, F. Schaltegger, A. Rüst, and F. Blumensaat, "Synchronous LoRa Mesh Network to Monitor Processes in Underground Infrastructure," *IEEE Access*, vol. 7, pp. 57663-57677, 2019.
- [38] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh, *Efficient network flooding and time synchronization with Glossy*. 2011, pp. 73-84.
- [39] M. Oliver and A. I. R. Escudero, "Study of different CSMA/CA IEEE 802.11-based implementations," 1999.
- [40] F. Cuomo, S. Luna, E. Cipollone, P. Todorova, and T. Suihko, *Topology Formation in IEEE 802.15.4: Cluster-Tree Characterization*. 2008, pp. 276-281.
- [41] P. Park, C. Fischione, and K. H. Johansson, "Performance Analysis of GTS Allocation in Beacon Enabled IEEE 802.15.4," in *2009 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, 2009, pp. 1-9.
- [42] *Industrial Communication Networks - Wireless Communication Network and Communication Profiles - WirelessHART*, 2010.
- [43] Semtech, "AN1200.21 Reading channel RSSI during a CAD," 2014.
- [44] M. Z. F. Ferrari, L. Thiele, and O. Saukh, "Efficient network flooding and time synchronization with Gloosy," *10th Int. Conf. Inf. Process. Sensor Network*, 2011.
- [45] Semtech, "SX1276/77/78/79 - 137 MHz to 1020 MHz Low Power Long Range Transceiver - DATASHEET," 2019.

- [46] U. Noreen, L. Clavier, and A. Bounceur, "LoRa-like CSS-based PHY layer, Capture Effect and Serial Interference Cancellation," in *European Wireless 2018; 24th European Wireless Conference*, 2018, pp. 1-6.
- [47] A. Goldsmith, *Wireless Communications*. Cambridge: Cambridge University Press, 2005.
- [48] W. Xu, J. Y. Kim, W. Huang, S. S. Kanhere, S. K. Jha, and W. Hu, "Measurement, Characterization, and Modeling of LoRa Technology in Multifloor Buildings," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 298-310, 2020.