Design and Performance Evaluation of Full-Duplex Relay Node in LoRaWAN-Based System

Van-Truyen Phan[1,2], Xuan-Tung Truong Minh[2,3], and Van-Truong Truong[1,2](✉)

[1] Faculty of Electrical-Electronic Engineering, Duy Tan University,
Da Nang 550000, Vietnam
{phanvantruyen,truongmxuantung,truongvantruong}@dtu.edu.vn
[2] Institute of Research and Development, Duy Tan University,
Da Nang 550000, Vietnam
[3] Mechanical Engineering Faculty, Duy Tan University,
Da Nang 550000, Vietnam

Abstract. In recent years, wireless sensor network (WSN) has attracted much attention from researchers and the industry. In particular, low power wide-area wireless network (LPWAN) protocols are widely applied in many fields thanks to their effective cost, long range, and energy efficiency. This study proposed a hardware design for a full-duplex relay node (FDRN) for relaying data from sensor nodes (SN) to the LoRaWAN gateway. Accordingly, we proposed the LLC-RTOS algorithm based on FreeRTOS to simultaneously receive, convert, and transmit LoRaWAN signals at FDRN. We proposed a novel LPWAN model based on LoRaWAN for application in WSNs by joining FDRN to the traditional 4-layer LoRaWAN model. Finally, we use the practical method to evaluate the system performance based on the packet loss percentage (PLP), and received signal strength indicator (RSSI) under three scenario: (a) single SN communicated with gateway via FDRN, (b) multiple SNs communicated with gateway via FDRN, and (c) multiple SNs communicated with gateway via multiple FDRNs. The obtained results show the efficiency of the FDRN design and the proposed model when the system can achieve a PLP less than 0.1% and RSSI 100 dBm within a 4 km communication range.

Keywords: LoRa · LoRaWAN · Full-duplex relay node · Wireless sensor network · LoRaWAN server · Application server

1 Introduction

In recent years, the outstanding development of Industry 4.0 has created the impetus for the development of intelligent systems in almost all fields of industry, agriculture, healthcare, transportation, and civil [1]. The trend of the Internet

N.-S. Vo et al. (Eds.): INISCOM 2022, LNICST 444, pp. 77–90, 2022.
https://doi.org/10.1007/978-3-031-08878-0_6

of Things (IoT) has opened up many advantages for the research, construction, and deployment of intelligent networks globally, in which wireless sensor network (WSN) plays a critical role [2].

Several low power wide-area network (LPWAN) communication technologies such as LoRa and Sigfox are being applied in WSNs due to their effective cost, long-range, and energy efficiency [3,4]. LoRa enables low data rate transmission over a covered area of up to 10 km with several mA peak broadcast currents [5]. Excellent customization at the physical layer allows LoRa to support communication between sensor nodes (SNs). However, because of customization, node-to-node (N2N) communication in LoRa networks is not integrated into the global pattern, so LoRa communication is not accessible to standardized and maintained LoRaWAN Gateways from LoRa Alliance.

Meanwhile, LoRaWAN is an open network protocol that provides connections between LPWAN gateways and IoT end-devices according to LoRa Alliance standards. The study [6] by Erturk *et al.* present the 4-layers standard architecture of LoRaWAN networks, as Fig. 1. In particular, the end node (EN) supporting LoRaWAN is a sensor or actuator that is wirelessly connected to the LoRaWAN network through gateways using LoRa modulation technology. ENs are mostly battery operated and perform functions to digitize physical or environmental information [7]. The LoRaWAN gateway connects to the ENs via the IP backbone to receive the devices LoRa modulated RF data and forward to the server in the LoRaWAN network. The network server plays the role of managing the entire network and establishes a secure 128-bit AES connection for data transmission and control. The network server ensures the authenticity of every sensor on the network and the integrity of the messages. Finally, the application servers are responsible for securely processing, managing, and interpreting data received from the sensors and generating downlink payloads to the ENs.

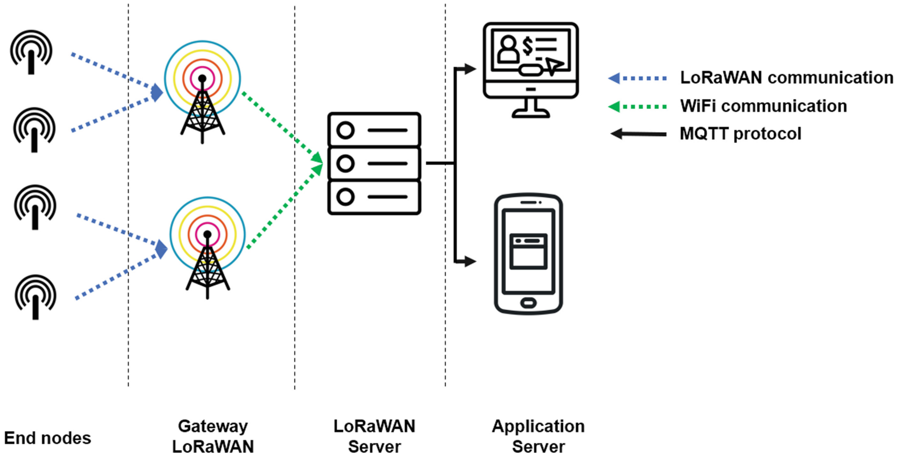

Fig. 1. The LoRaWAN network architecture

Data is formatted in a unique standard that makes it easy to synchronize with the LoRaWAN network server and application Server. However, the standard LoRaWAN model is only suitable for star network architectures, where SNs connect directly to one or more LoRaWAN gateways and cannot send messages to each other. It poses significant practical deployment challenges for WSN applications, where multiple LoRaWAN gateways must be designed to manage large numbers of nodes distributed over a wide range. The system's scalability is limited when the gateway must be placed at a location with backhaul access to the network server [8]. Moreover, the information forwarding mechanism between nodes in the WSN serves for data routing, cluster head selection, leading to reduce the information traffic transmitted to the gateway, saving energy, and increasing the network's lifetime. Therefore, ensuring device-to-device communication in the LoRaWAN network is interested in researchers learning and proposing solutions [9–12].

For instance, Daniel *et al.* in [9] proposes to solve the problem of scaling a LoRaWAN network by using multiple gateways that coordinate data forwarding. The gateways communicate over a standard LoRaWAN network and have a built-in routing table, which updates through each data relay loop. The authors propose a protocol that works based on the tunneling technique. The gateways deployed the Hybrid Wireless Mesh Protocol and Ad-hoc OnDemand Distance Vector Routing for data forwarding. The results show that the larger the number of data hops in the system, the higher the delay time, but it can still respond in real-time with a delay of approximately 1.58 s for the 3-step routing process. Truong *et al.*'s solution to utilize the Zigbee multi-hop network with LoRa in [11] is also worth noting. The authors propose a hybrid Zigbee and LoRa network to leverage Zigbee's N2N communication to complement the LoRaWAN network disadvantage. The proposed model can be applied in 3 case studies: air quality monitoring, agricultural monitoring, and Internet of underwater Thing monitoring.

Even so, securing N2N communication in LoRaWAN is still a matter of research, thus prompted us to carry out this study. In this paper, we aim at a circuit that relays data between ENs in a real-time response LoRaWAN network. The main contributions of this paper are as follows:

- We designed the LoRa – LoRaWAN full-duplex relay node (FDRN) hardware, ensuring N2N communication in the LoRaWAN network. Accordingly, we proposed a novel LoRaWAN network model consisting of 5 device layers, in which the SN layer communicating with the relay node is added to expand the data collection area of the WSN system.
- We proposed to use the FreeRTOS in the FDRN board, namely LLC-RTOS, for LoRa data acquisition and processing.
- We evaluated system performance based on the packet loss percentage (PLP), and the received signal strength indicator (RSSI) index following the distance of devices and the data rate.

The remainder of this paper is organized as follows. Section 2 presents the system model for N2N communication and the novel extended LoRaWAN model, which outlines the hardware and software design of FDRN. The experimental results under actual conditions and discussion are presented in Sect. 3. Section 4 concludes the paper and presents the future work.

2 System Design

2.1 The Hardware of FDRN

The FDRN designed is highlighted in Fig. 2, which includes two LoRa Ra–02 SX1278 modules operating in the 433 MHz band that communicates with the Atmega328P MCU via Serial Peripheral Bus (SPI) protocol. SPI is a full-duplex synchronous communication standard in the architecture of one master (MCU) and many slaves (Ra-02), using connections Serial Clock (SCK), Master Input Slave Output (MISO), Master Output Slave Input (MOSI), and Slave Select (SS). SCK is the synchronous clock pin for the communication generated by the master; each SCK clock indicates 1 bit of incoming or outgoing data. MISO transmits the data from the slaves to the MCU, while the MOSI operates in the opposite direction. Finally, SS is used to select the slave to communicate with; each slave is communicated with the master via a separate SS pin. If the MCU pulls the particular Ra-02 SS into low, communication will occur.

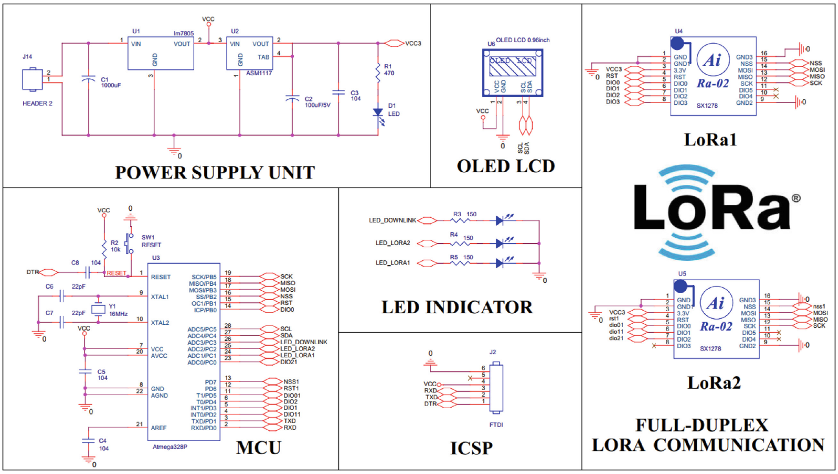

Fig. 2. The FDRN hardware design

The OLED module is connected to MCU via Inter-Integrated Circuit (I2C) protocols, displaying receiving and transmitting LoRa packet information. I2C is a simple synchronous serial communication protocol developed by Philips Semiconductors for data transfer between an MCU and multiple salves. I2C devices use only two signaling lines, Serial Clock Line (SCL) and Serial Data Line (SDA). The transmitted data is sent over the SDA wire and synchronized with the clock signal from the SCL. To prevent short-circuit occurring when devices simultaneously pull signals to high and low logic level, both I2C bus lines act as open drain drivers. That is, to put it more simply, any device connected to the I2C network can drive SDA and SCL low logic levels but cannot drive them high one. That's said, a 10 kΩ pull-up resistor deployed in the OLED module is used for each bus line to keep them high logic level by default.

Three LEDs indicate the transmit, receive, and error signals. The circuit works under 5 V and 3.3 V voltage regulation, powered by two 5500 mAh LiPo batteries. Ra-02 modules operate with 3.3 V power, while the rest are powered with 5 V. Thus, the MCU operating at 5 V TTL logic can be compatible with other 5 V TTL devices or 3.3 V TTL/CMOS devices without any logic level converter. The 433 MHz antennas with 10cm long Ipex connectors are distributed around the edges of the circuit to amplify with the 3 dBi gain signal. Figure 3 is FDRN prototype, which is manufactured follow industrial printed circuit board standards.

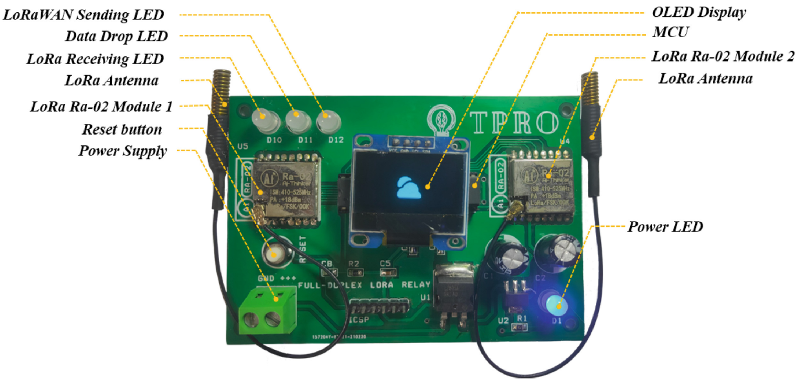

Fig. 3. The FDRN prototype

2.2 The Software of FDRN

We propose using the LoRa to LoRaWAN conversion based on FreeRTOS, namely LLC-RTOS, for FDRN to forward data from the SN to the LoRaWAN gateway, as Algorithm 1.

We use the Sandeep LoRa [13] library for SN data acquisition and the LoRaWAN-MAC-in-C (LMIC) provided by IBM for LoRaWAN connectivity [14]. The MCU sets up the appropriate IO, SPI, and I2C ports and sends commands to start the Ra-02 modules when the system is powered up. The parameters specific to LoRa and LoRaWAN networks are set, including operation frequency (f), spreading factor (SF), code rate (CR), and bandwidth (BW). FDRNs start their operation by sending a *join request* message with 8 bytes unique application identifier, 8 bytes unique device identifier, and 2 bytes random attract replay to the LoRaWAN server in the *join procedure*. This

Algorithm 1. LoRa to LoRaWAN conversion based on FreeRTOS (LLC-RTOS)

```
Setup: Library LoRa and LoRaWAN
Setup: IO port, SPI port, I2C port
Setup: LoRa and LoRaWAN parameter: $f = 433$ MHz, $SF = 7$, $CR = 1$, and $BW = 125$ KHz.
Setup: RTOS Preemptive scheduling and Tasks
Send join request to LoRaWAN server
Wait until get join accept from LoRaWAN server
Joint LoRaWAN network with secure connection
Setting up FreeRTOS tasks and Preemptive scheduling
while True do
    if SN ID available then
        Run algorithm for LoRa SN data acquisition
        Decompress the LoRa data
        Convert data to LoRaWAN format
        Run algorithm for sending data to LoRaWAN gateway
        LED indicator and OLED display
    else
        Drop message
```

process ensures the security of the communication process, as only authorized devices can participate in the network. After using secure *matching keys* to check the legitimacy of the FDRNs, the LoRaWAN server initializes the *session keys*, i.e., Network Session Key and Application Session Key, in response to the *join accept* message to the FDRN via normal downlink. Next, FDRN decrypts the *join accept* message and receives the session key to join the LoRaWAN network. Next, the FDRN establishes a LoRa network to communicate with the SNs, based on the previously agreed IDs allocated to the SNs of the declared communication frequency band. Due to the highly customizable characteristics of using LoRa raw data, security setup operations are not used.

Accordingly, the first Ra-02 module holds the role of communicating with SNs using LoRa, while the rest module communicates with the Dragino gateway according to the LoRaWAN standard. FreeRTOS integrated on the MCU is responsible for initiating and executing tasks, including LoRa data acquisition (Task 1) as Algorithm 2, saving LoRa raw data into 64 bytes memory (Task 2), LoRa to LoRaWAN packet format conversion (Task 3), display packet information on OLED (Task 4), and LoRaWAN packet delivery (Task 5).

Algorithm 2 describes the process by which the FDRN receives data from SNs. Each LoRa module always has two methods of receiving information: single receive mode (SRM) and Continuous receive mode (CRM). However, due to the use of battery power at FDRN, we use SRM with alternate Standby mode, ensuring maximum energy savings. In this mode, the Ra-02 module continuously searches for the preamble, a particular signal used to detect incoming LoRa signals, during the time slot with a length from 4 to 1023 LoRa symbols. Otherwise, the LoRa signal is fully detected and received, the RxDone interrupt is initiated after the payload block, and Cyclic Redundancy Check (CRC) is performed. The Ra-02 then returns to the Standby state and waits for the next slot. In the event that a preamble signal is not detected at the end of the time slot cycle, the Ra-02 initiates the RxTimeout interrupt and returns to Standby mode. Furthermore, since the communication period of the SNs to the FDRN is fixed, the SRM can still ensure a shallow packet loss rate.

Algorithm 2. LoRa single receive mode for SN data acquisition algorithm

```
Start
Wait for Interrupt request (IRQ)
if RxTimeout IRQ then
    Go to Standby mode
    Go to End
if RxDone IRQ then
    Go to Standby mode
    if Payload CRC detect error then
        Goto End
    else
        Read SN data
End
```

After the data from the SNs is saved into the FIFO data buffer, preprocessing is performed. For the convenience of processing and decompression, data from sensors will be formated with JavaScript Object Notation (JSON), where the *key* being the sensor information and the *value* is the value that the sensor read. The FDRN checks the sensor information against the pre-conventional ID. If the SN's ID belongs to the device group managed by the FDRN, it will decompress the data. This process is indicated by *LoRa receiving* LED. Information about the packet such as packet sequence, and sensing data, is displayed on the OLED, and saved to memory. If the ID belongs to another device group, FDRN automatically drops that packet and continues to consider the next packet. If the data counter is not continuous, the FDRN detects the lost packet and indicates by the *data drop* LED.

We continue to describe the process of sending data to the Dragino gateway in Algorithm 3. Sensor data is stored in the FIFO buffer of the second Ra-02 module. Note that the data in the FIFO buffer cannot be erased when the transmission ends unless the device goes into an Sleep state or a new message sequence arrives. TxDone interrupt signals occurred when each successful packet was sent to the gateway. This process is indicated by *LoRaWAN sending* LED. Ra-02 continues to check if there is still data to send; if there is, the whole process is performed again; if not, the module enters the Standby state.

Algorithm 3. Sending data to LoRaWAN server algorithm

```
Start
Write data in FIFO buffer
Wait for TxDone IRQ
if New transmit data is available then
    Go to Start
else
    Go to End
End
```

Under The Thing Network Fair Use Policy, we limit the uplink airtime to 30 s per day per FDRN. In other words, the real-time constraints in LoRaWAN are not too strict. However, the data collected from the SNs have higher requirements when each environmental data update occurs in a 1 s cycle. Based on that, we apply Preemptive scheduling to the system and set the priority levels in the execution of tasks are as follows: Task 1 has the highest priority, followed by Task 2, Task 3 and Task 4 have the same priority at the 3rd level, and Task 5 has the lowest priority. The CPU of MCU always controls the tasks with the highest priority; when an Interrupt service routine (ISR) is generated, the system will pause the executing task, complete the ISR, then the system executes the task with the highest priority at the time. The system then resumes the interrupted tasks. So, in preemptive mode, the system can respond to urgent tasks such as collecting IoT data promptly.

2.3 The Novel LoRaWAN Architecture

In this section, we propose the novel LoRaWAN architecture, in which five layers of devices are coordinated to work in a comprehensive system, as shown in Fig. 4. Specifically, four LoRa32 modules [15] act as SNs for receiving signals from the surrounding environment and communicate with the Dragino LoRaWAN gateway through the help of two FDRNs. We use The Things Network (TTN) as the LoRaWAN Server and TagoIO as the Application Server.

The first layer contains SNs, i.e., LoRa32 modules, which are battery operated and responsible for digitizing environmental parameters thanks to the sensor system. LaRa32 is accompanied by unique identifiers for activation when participating in the LoRa network, and at the same time, ensuring the safety of packets when transmitted in the network. The second layer includes FDRNs, which are responsible for receiving signals from two SNs, encrypting data according to an ID of SNs, and formatting data following to LoRaWAN standard. Next, the data is packed and forwarded to the LoRaWAN Gateway.

The third layer is LoRaWAN Gateway, i.e., Dragino LG01N. It is an open-source LoRaWAN Gateway, which allows converting LoRaWAN wireless data from FDRNs to LoRaWAN Server through an IP backbone, such as WiFi, Ethernet, 3G, or 4G connection [16]. In fact, LG01N operates entirely at the physical layer and is also a LoRa data relay. It checks the data integrity of each incoming

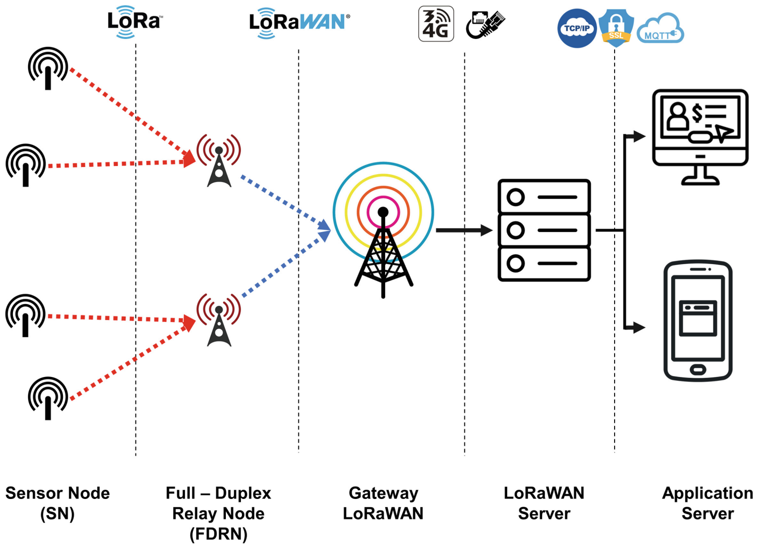

Fig. 4. The novel LoRaWAN architecture

LoRa RF message. The LoRa message is dropped if the CRC is incorrect. If CRC is true, the LG01N forwards them to the LoRaWAN Server, along with some metadata such as the received message RSSI, and the timestamp. LG01N is equipped with Dragino HE Linux module with OpenWrt operating system, communicating with LoRa module via Atmega328P processor. Using Message Queuing Telemetry Transport (MQTT) communication protocol, the LG01N is suitable for low-bandwidth IoT devices.

The fourth layer is LoRaWAN Server, i.e., TTN [17]. It is an infrastructure to store the data of the LoRaWAN network globally, with an ecosystem consisting of 21.5 thousand LoRaWAN Gateways are operating at the same time, covering over 151 countries and territories. TTN establishes a secure 128-bit AES connection for end-to-end data transfer, i.e., from FDRN to Applications in the Cloud and vice versa. TTN ensures the authenticity of every sensor and the integrity of all messages. While the TTN cannot see or access application data, it performs essential roles, including checking device addresses, validating frames, and managing frame counters. Furthermore, TTN decodes data through the Payload Decoder algorithm to obtain Raw Payload packets and forwards them to the appropriate application servers. In the case of downlink LoRa setup, TTN queues payloads coming from the Application Server to SN connected to the network.

The application server TagoIO is the last layer in our proposed LoRaWAN architecture. TagoIO is an IoT cloud platform, translates raw value from TTN into parameters users can connect and interact with [18]. The data on TagoIO will be displayed in the form of a Dashboard, supported on desktop and smartphone to help users easily monitor data anytime, anywhere.

3 Experiment Results and Discussion

This section presents experiment scenarios for the proposed LoRaWAN system and evaluates the system performance using the PLP, and RSSI. PLP is a parameter commonly used to check the bit error rate in the communication process and is measured by ratio of the total number of incorrectly received bits and total number of transmitted bits. RSSI is an index to measure the strength of the signal at the receiver; in theory, the larger this parameter, the better.

In the first experiment, we investigated the system performance with the different number of SNs and FDRNs. We carried out the following three scenario examinations as Fig. 5. Specifically, in the first experiment (a), the LoRaWAN communication network consisting of one SN and one FDRN was investigated. FDRN is fixed at a location that has line-of-sight (LoS) connection to gateway is available for easy and secure connection. We set up the SN at 10 m away from the FDRN and increased the distance until 6 km. At each location, the SN sends 1000 packets with a sequence number to the FDRN. The process of data statistics is handled by the TagoIO Dashboard with direct link to TTN. In the experiment (b), the system performance parameters are evaluated when the FDRN serves two SNs, placed in symmetrical positions across the FDRN. And in the last experiment (c), the network of two FDRNs and four SNs communicating with each other was investigated.

Figure 6 describes the interface for statistics data from Dashboard TagoIO. The interface includes function blocks that allow observation of transmission distance parameters, packet sequence number, RSSI per packet, packet status, average RSSI statistics, and the number of packets lost during communication.

Figure 7 shows the impact of the number of SN and FDRN on the PLP over different distances. Observing the left side of Fig. 7, when the distance between devices is between 10 m and 300 m, scenario (c) performs the worst when the PLP is approximately 17%, followed by scenario's PLP (b) reaching approx 5%, while scenario (a) has no packet loss. It shows that when LoRa devices are distributed too close together, they will affect the received signal at FDRN. The more complex the system with more SNs and FDRNs, the more significant the impact. When the devices are far away enough, the PLP drops to approximately zero. However, when the transmission distance increases more than 5 km, the SN is entirely unable to communicate with the FDRN causing the PLP to increase to 100%.

Figure 8 shows the impact of SN and FDRN on the RSSI of FDRN over different distances. We use the built-in function provided by [13] to measure the RSSI. We use scenarios (a) and (b) in this experiment. The results show that RSSI decreases with distance, and case (b) have a higher RSSI than case (a) by 5 to 7 dBm, that is, the received signal strength at FDRN in case (b) is better. It shows that when using multiple SNs at the same frequency resource, the received signal strength at FDRN can be enhanced. Therefore, if there is a requirement for packet accuracy, we can designate multiple SNs in the same

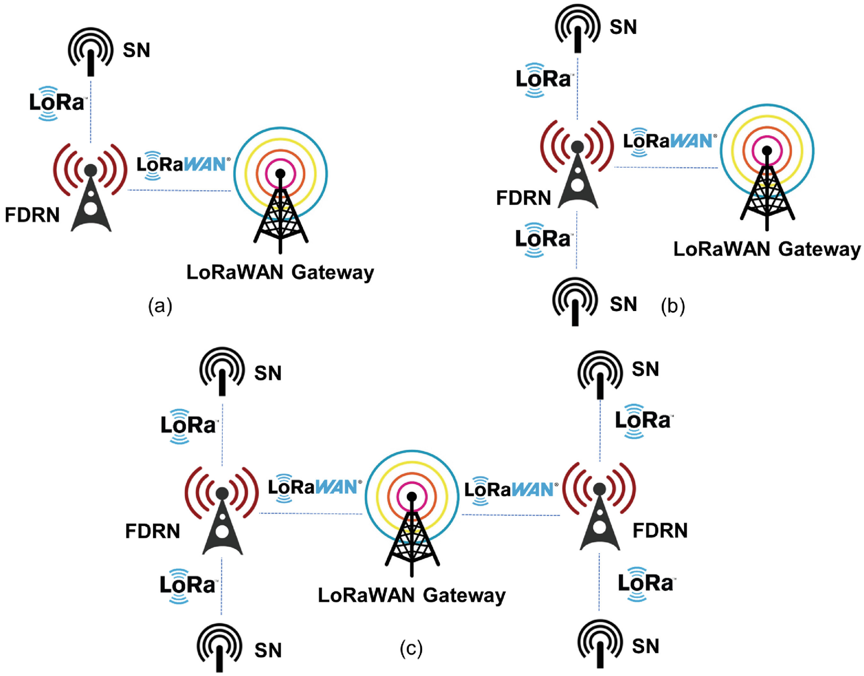

Fig. 5. Scenario examinations (a), (b), and (c)

monitored area, and these SNs simultaneously send signals to the FDRN to enhance RSSI. However, Fig. 7 shows that the PLP of the case (b) is higher than case (a) when the distance is from 0 to 300 m. It can be explained as when the distance between the SNs is relatively close; the SNs will affect each other when transmitted on the same frequency. It is the biggest drawback in the LoRa communication network compared to LoRaWAN when the self-frequency hopping mechanisms of LoRaWAN cannot be applied in LoRa communication.

When the distance between the SNs is large enough, the RSSI gradually decreases with the distance but still ensures a reasonable level, i.e., from −30 to −70 dBm, and the signal can be received and appropriately decoded at the FDRN. When the distance is approximately 5 km, the RSSI measured in case (a) reaches saturation with an inferior performance of −138 dBm. It means that the received signal is purely noise in the frequency 433 MHz, and the necessary LoRa data cannot be suitably decoded. While in (b), the saturation RSSI event reached a value of approximately −130 dBm and occurred at a distance of 4.5 km.

Distance (m)	stt	RSSI (dBm)	Message status	Controls
1400	474	-102	successful	✎ 🗑
1400	473	-100	successful	✎ 🗑
1400	472	-100	successful	✎ 🗑
1400	471	-104	drop	✎ 🗑
1400	470	-103	successful	✎ 🗑
1400	469	-105	drop	✎ 🗑
1400	468	-103	successful	✎ 🗑
1400	467	-103	successful	✎ 🗑
1400	466	-103	successful	✎ 🗑
1400	465	-102	successful	✎ 🗑
1400	464	-101	successful	✎ 🗑
1400	463	-102	successful	✎ 🗑
1400	462	-102	successful	✎ 🗑
1400	461	-103	drop	✎ 🗑
1400	461	-104	successful	✎ 🗑
1400	460	-101	successful	✎ 🗑
1400	459	-102	successful	✎ 🗑
1400	458	-102	successful	✎ 🗑
1400	457	-102	successful	✎ 🗑
1400	456	-102	successful	✎ 🗑

Fig. 6. TagoIO dashboard interface

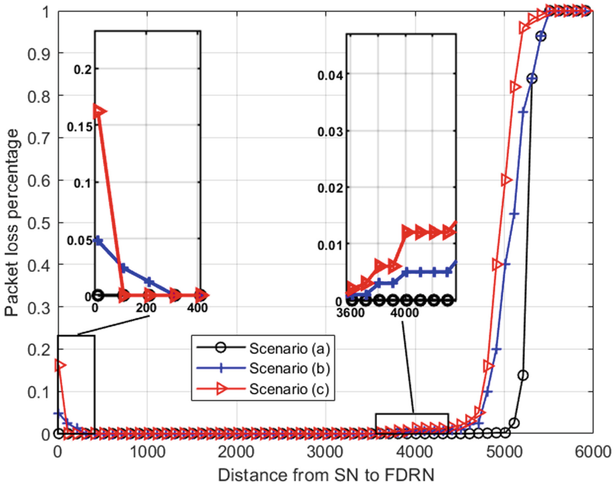

Fig. 7. The impact of the distance and number of SN and FDRN to the packet loss percentage

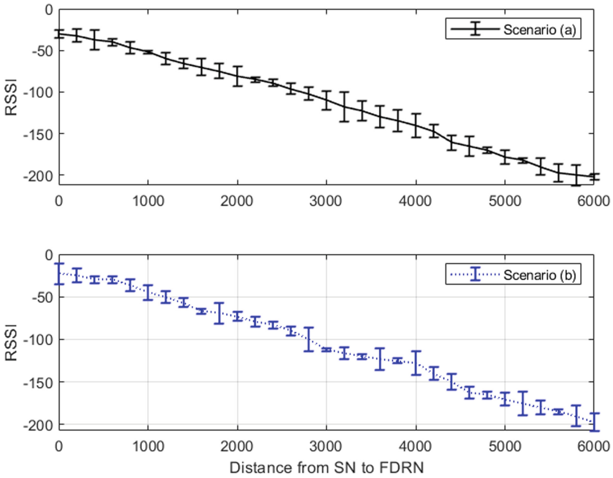

Fig. 8. The impact of the number of SN and FDRN to the RSSI

4 Conclusion and Future Work

This paper designs the hardware and software for the LoRa - LoRaWAN FDRN in the WSN network. In which, the MCU communicates with two LoRa Ra02 modules, integrated with FreeRTOS to simultaneously perform the tasks of acquiring data from the SNs, converting LoRa data format to LoRaWAN, storing and displaying data, and transmitting data to the LoRaWAN gateway. Accordingly, we propose a new LoRaWAN model with five layers operating synchronously, adding D2D communication between SN and FDRN layers to the traditional LoRaWAN model. Finally, we evaluate system performance in terms of RSSI and PLP according to the number of SNs, the number of FDRNs, and the distance between devices in the network by experimental method. The proposed system can work well within the communication range of 4 km and achieve a PLP of less than 0.1%.

In future studies, we will design SNs that integrate listen before talk capabilities to ensure that SNs can communicate with FDRN more effectively. At the same time, we will study the expansion system with multiple LoRaWAN gateways that support multiple FDRNs and SNs.

References

1. Raj, A., Prakash, S.: Internet of everything: a survey based on architecture, issues and challenges. In: Proceedings of the 5th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON), Gorakhpur, India, pp. 1–6. IEEE (2018)
2. Shahraki, A., Taherkordi, A., Haugen, O., Eliassen, F.: A survey and future directions on clustering: from WSNs to IoT and modern networking paradigms. IEEE Trans. Netw. Serv. Manag. **18**(2), 2242–2274 (2020)

3. Olatinwo, D.D., Abu-Mahfouz, A., Hancke, G.: A survey on LPWAN technologies in WBAN for remote health-care monitoring. Sensors **19**(23), 5268 (2019)
4. Liya, M., Arjun, D.: A survey of LPWAN technology in agricultural field. In: Proceedings of the Fourth International Conference on IoT in Social, Mobile, Analytics and Cloud, pp. 313–317 (IEEE) (2020)
5. Zourmand, A., Hing, A.L.K., Hung, C.W., AbdulRehman, M.: Internet of Things (IoT) using LoRa technology. In: Proceedings of the IEEE International Conference on Automatic Control and Intelligent Systems, Selangor, Malaysia, pp. 324–330. IEEE (2019)
6. Erturk, M.A., Aydin, M.A., Buyukakkaclar, M.T., Evirgen, H.: A survey on LoRaWAN architecture, protocol and technologies. Future Internet **11**(10), 216 (2019)
7. Tokmakov, D., Asenov, S., Dimitrov, S.: Research and development of ultra-low power LoraWan sensor node. In: Proceedings of the IEEE XXVIII International Scientific Conference Electronics, Sozopol, Bulgaria, pp. 1–4. IEEE (2019)
8. Codeluppi, G., Cilfone, A., Davoli, L., Ferrari, G.: LoRaFarM: a LoRaWAN-based smart farming modular IoT architecture. Sensors **20**(7), 2028 (2020)
9. Lundell, D., Hedberg, A., Nyberg, C., Fitzgerald, E.: A routing protocol for LoRa mesh networks. In: Proceedings of the IEEE 19th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM), Chania, Greece, pp. 14–19 (2018). https://doi.org/10.1109/WoWMoM.2018.8449743
10. Osorio, A., Calle, M., Soto, J.D., Candelo-Becerra, J.E.: Routing in LoRaWAN: overview and challenges. IEEE Commun. Mag. **58**(6), 72–76 (2020)
11. Truong, V.-T., Nayyar, A., Showkat, A.L.: System performance of wireless sensor network using LoRa-zigbee hybrid communication. Comput. Mater. Continua **68**(2), 1615–1635 (2021). https://doi.org/10.32604/cmc.2021.016922. http://www.techscience.com/cmc/v68n2/42201
12. Cotrim, J.R., Kleinschmidt, J.H.: LoRaWAN mesh networks: a review and classification of multihop communication. Sensors **20**(15), 4273 (2020)
13. Mistry, S.: Arduino LoRa library (2021). https://github.com/sandeepmistry/arduino-LoRa. Accessed 27 Dec 2021
14. IBM, Kooijman, M., Moore, T., Won, C., Rose, F.: MCCI LoRaWAN LMIC library (2021). https://www.arduino.cc/reference/en/libraries/mcci-lorawan-lmic-library/. Accessed 27 Dec 2021
15. Wang, J., Yi, S., Zhan, D., Zhang, W.: Design and implementation of small monitoring wireless network system based on LoRa. In: Proceedings of the IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chengdu, China, vol. 1, pp. 296–299. IEEE (2019)
16. Van Truong, T., Nayyar, A., Masud, M.: A novel air quality monitoring and improvement system based on wireless sensor and actuator networks using LoRa communication. PeerJ Comput. Sci. **7**, e711 (2021)
17. Coutaud, U., Heusse, M., Tourancheau, B.: LoRa channel characterization for flexible and high reliability adaptive data rate in multiple gateways networks. Computers **10**(4), 44 (2021)
18. Lee, S.L., Darsono, A.M.: Enhancement of smart street light monitoring system based on LoRa technology. In: INOTEK 2021, vol. 1, pp. 67–68 (2021)