ISSN: 2093-5374 / E-ISSN: 2093-5382

Performance Evaluation of Collision Avoidance for Multi-node LoRa Networks based on TDMA and CSMA Algorithm

I Gde Dharma Nugraha[1*], Edwiansyah Zaky Ashadi[2], and Ardiansyah Musa Efendi[3]

[1*]Electrical Engineering Department, Faculty of Engineering, Universitas Indonesia, Depok, Indonesia. i.gde@ui.ac.id, https://orcid.org/0000-0002-4960-7968

[2]Electrical Engineering Department, Faculty of Engineering, Universitas Indonesia, Depok, Indonesia. edwiansyah.zaky@ui.ac.id, https://orcid.org/0009-0001-0655-8019

[3]Connectivity Chipset Design Department, Huawei Singapore Research Centre, Singapore. ardiansyah.musa.efendi@huawei.com, https://orcid.org/0000-0002-2783-8677

Received: September 04, 2023; Revised: November 10, 2023; Accepted: January 09, 2024; Published: March 30, 2024

Abstract

LoRa can be used as the communication technology for the intelligent monitoring system. However, LoRa is usually used for outdoor communication. The usage of LoRa as indoor communication has many challenges. One of the challenges is that collision happens when using standard LoRa devices with only one channel. The algorithms based on TDMA (Time-division Multiple Access) and CSMA (Carrier-sense Multiple Access) protocols can be used to address this challenge. These two algorithms can be modified by applying the device that is the center of the network (gateway) as a central control and the data transmitter (sensor node) as a passive device. The test was conducted on the Intelligent Laboratory Monitoring System to evaluate this design on a multi-node LoRa network. RSSI testing proves that the distance and building interference affect the signal strength or RSSI of sensor nodes, so the average RSSI value is -73.75 with an RSSI threshold of value -106. The gateway successfully collected each sensor node data with an average success of about 64.953%. The experiment results show the success rate of the CSMA-based algorithm is 10% versus 100% in TDMA-based algorithm; the delay is 4125 ms for CSMA-based and 428.3 ms for TDMA-based. This result means that the CSMA-based algorithm is more complex, takes more time to process the data than the TDMA-based algorithm, has a low success rate, and is more prone to collisions.

Keywords: IoT, IoT Network, Multi-node, Smart Monitoring System, Indoor Monitoring, LoRa, Algorithm, Architecture, Protocol, Wireless Sensor Network, RSSI, TDMA, Time-division Multiple Access, CSMA, Carrier-sense Multiple Access.

1 Introduction

The monitoring system is essential to collect information about the environment that is being monitored (Su et al., 2019) Various environments require an active and real-time monitoring system, especially in the laboratory environment. Laboratories generally have a variety of components, basic materials, and equipment that are hazardous or flammable. The monitoring system is an important part of the laboratory to ascertain the situation and provide warnings if there are anomalies in the laboratory environment.

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA), volume: 15, number: 1 (March), pp. 53-74. DOI: *10.58346/JOWUA.2024.I1.005*

*Corresponding author: Electrical Engineering Department, Faculty of Engineering, Universitas Indonesia, Depok, Indonesia.

IoT technology can be integrated into monitoring systems to provide specific and effective computing and communication capabilities. IoT also allows monitoring to be carried out in multiple locations and controlled remotely. Various technologies can be applied for transmitting communication between IoT devices. The recent technology that can be used as an alternative is LoRa. LoRa or Long Range is a technology included in the Low-Profile Wireless Area Networks (LPWAN) category, which has the advantage of using relatively small power with a comprehensive transmission area coverage (Shahjalal et al., 2020). LoRa itself is a part of which functions to form long-range communication by applying the concept of chip spread spectrum (CSS) (Tsoi et al., 2020). The LoRa devices that make up the IoT network are generally composed of several sensor nodes connected to the gateway. The protocol and architecture regulating communication between LoRa devices has a standard, the Long-Range Wide Area Network or LoRaWAN, which was formed by the LoRa Alliance association (Eridani et al., 2019; LoRa Alliance; Kim, H. 2020; Sreenivasu, M., et al., 2022).

The communication protocol used in IoT networks that use LoRa is not limited to LoRaWAN, but other protocol standards are intended for communication between devices. Factors that must be considered in choosing the appropriate LoRa network protocol are the environmental density at the network location and the channel usage mechanism (Pham 2018). There are limitations, especially in simple IoT devices that prioritize simplification and low resource usage, such as there will be only one channel available for transmission because each device operates on a single frequency (Ratuloli et al., 2021). Transmission is done simultaneously, and the same frequency will cause collisions and reduce network performance (Baddula et al., 2020). Therefore, we need an algorithm suitable for simple devices so that channel management and communication between devices can be carried out as effectively as possible without the complexity and excessive use of resources (Camgözlü & Kutlu 2023).

The protocols mostly used to avoid collision in communication are TDMA (Time-division Multiple Access) and CSMA (Carrier-sense Multiple Access). TDMA is a protocol that implements communication and scheduling based on timeslots or time intervals (Ratuloli et al., 2021). CSMA is a protocol that applies the Listen Before Talk (LBT) scheme so that the transmitter will ask about the channel status before sending data (Baddula et al., 2020), (To and Duda 2018). These two protocols can be used as the basis for algorithms that can be modified in such a way as to form a mechanism that is suitable for the device used. One of the modifications that can be done is to apply the device that is the center of the network (gateway) as the central control, and the data transmitter (sensor node) will be a passive device (Majeed & Zia 2017).

Protocols implementing TDMA and CSMA-based algorithms can become communication standards and the basic architecture for developing intelligent monitoring systems to supervise laboratories. The characteristics of LoRa technology are the main attraction to forming a LoRa network system that can optimize the features of TDMA and CSMA. The development of the LoRa communication mechanism has a broad scope for exploration, especially in indoor network communications, which generally use other types of technology (Senthilkumar, M., & Uma, S.2013).

2 Basic Theory

Internet of Things (IoT)

Internet of Things or IoT is one of the technologies or platforms in the form of interconnection and data exchange between devices, which gradually becomes smarter, has more effective processing, and more efficient information communication (Liya and Aswathy 2020). Aspects of the traditional lifestyle have evolved into high-tech lifestyles. This technology continues to develop and has been implemented in various fields such as asset tracking, smart agriculture, smart cities, smart homes, pollution control, energy saving, intelligent transportation, and intelligent industries (Mekki et al., 2019; Kumar et al., 2019). Many important studies and research have been carried out to improve technology through IoT. Innovations that use the word 'Smart' are generally the result of the transformation caused by IoT (Kumar et al., 2019). In the future, IoT devices are expected not to be limited to connecting via the Internet or locally. Still, they are also likely to be able to communicate with other devices on the internet directly with each other without any intermediary nodes (Hassija et al., 2019).

The application of the IoT system can be divided into four layers: the sensing layer, network layer, middleware layer, and application layer (Hassija et al., 2019). Each layer uses different technologies and functions. The sensing layer is the layer that operates on the physical and actuator parts of the IoT. The network layer is a layer that works by transmitting information received from the sensing layer to the computing device. The middleware layer is the layer in charge of preparing abstractions between the network layer and the application layer. The application layer is a layer that interacts directly with the user and applies the final result of the IoT system that is formed.

Long Range (LoRa)

Long Range or LoRa is a telecommunications technology that has the characteristics of using relatively small power with a comprehensive transmission area coverage. LoRa itself is a part of which functions to form long-range communication. LoRa is also one of the telecommunications technologies included in the LPWAN category. LPWAN is a telecommunications protocol and technology in WAN coverage with relatively low resource usage.

Several parameters in LoRa can be configured and impact LoRa's performance. These indicators include bandwidth, spread factor, coding rate, and transmission power, as shown in Table 1. Each parameter has a range of values that will affect the performance of LoRa as the value increases or decreases.

Table 1: LoRa Parameters

Parameter	Value Range
Bandwidth	125 - 500kHz
Spreading factor	$2^2 - 2^{12} \left(\frac{chips}{symbol}\right)$
Coding rate	4/5 - 4/8
Tx Power	-4 - 20dBm

Regarding network architecture, the network concept implemented in LoRa can be based on IoT network concepts such as the Wireless Sensor Network. The Wireless Sensor Network is one of the results of developing IoT technology. Various IoT devices on the WSN are connected and eventually form a wireless network through transmission technology such as LoRa. The constituent components of the network created in the WSN generally consist of two main parts: a group of sensor nodes, also known as motes, and an IoT device that functions as a central node, whether a gateway or base station (Acharyya et al., 2019).

In terms of network communication, one of the LoRa communication methodologies currently available is LoRaWAN. LoRaWAN is an open communication standard for LoRa realized by the non-profit association LoRa Alliance (LoRa Alliance). LoRa communication applied to the LoRaWAN standard uses a mechanism based on the ALOHA protocol where the sensor node will transmit at any time when the data is ready to be sent so that there will be a possibility of collision (Baddula et al., 2020). The difference between the ALOHA protocol and the protocol used in LoRaWAN is only in the length of the packet sent (To and Duda 2018).

Communication techniques used in LoRa are not limited to LoRaWAN. Various other standards for communication between devices can be implemented in LoRa communication. Considerations that need to be considered in implementing the desired communication mechanism are the design of the use and specifications of the LoRa device. Ra-02 is a further development of SX1278, which uses 433Mhz frequency in communication. This will be a limitation of LoRa communication, where there will only be one channel that can be used to transmit data to the same destination. Some communication standards can be applied in this condition, such as the distribution of channel usage based on time differences. Time division or scheduling can be allocated statically as is standard in Time Division Multiple Access (TDMA) or based on channel availability applied to Carrier-sense Multiple Access (CSMA).

TDMA, or Time Division Multiple Access, is an access-sharing protocol based on time differences. This standard can be used as a method of scheduling between data transmissions based on time slots or allocated time intervals (Ratuloli et al., 2021). There will be no competition between data senders because each transmission will be given its own time allocation (Piyare et al., 2018). Each sender will not transmit if not part of the time interval.

CSMA or Carrier-sense Multiple Access is a technique that operates at the medium access control layer (MAC layer), which regulates the intermediary in the transmission process. CSMA applies a scheme where there will always be a check on the channel to ensure its status. The channel will be in two possible states: idle if no transmission is in progress and busy if the channel is in use. The transmission will only be carried out after a while if it is found that the channel is in use. The transmission will be carried out if the identified channel is not used. This scheme is also known as Listen Before Talk (LBT) (Baddula et al., 2020; To and Duda 2018).

Received Signal Strength Indication (RSSI)

Received Signal Strength Indication or RSSI is an indicator that can be used to estimate or measure how strong a signal can be obtained between communicating devices. Estimates are made by detecting the received signal strength in milli-watts, and the final measurement results in dBm (Fitriawan et al., 2020).

This indicator can represent signal loss and play a role in determining the feasibility of wireless communication between devices. The range of RSSI measurements is generally in the field of negative values. The RSSI measurement in dBm units, which is getting closer to 0, will indicate that the formed communication signal is getting stronger. The further the RSSI measurement is from the value 0, the measured signal suggests that the communication signal that is established is relatively weak (Fitriawan et al., 2020).

3 System Planning

The implemented system is a system that integrates the Internet of Things network through LoRa transmission technology to be able to monitor a laboratory intelligently. Details of the proposed Intelligent Laboratory Monitoring System design will be explained further in sub-chapter 3.

Intelligent Laboratory Monitoring System Design

The components of the designed LoRa network consist of four sensor nodes placed in different laboratories, one gateway, and a web server. Environmental phenomena that are monitored in this system are temperature and gas concentration. Each sensor will observe the phenomenon in the testing laboratory and convert it into information data. The data obtained will then be controlled on the microcontroller and transmitted to the gateway device, which functions as the central control in the network. Data that is successfully received by the gateway wirelessly will be sent back to the Thing Speak web server to be processed and displayed to users.

This system is intended to determine the state of the environment around IoT devices that are observed remotely. Users can find the temperature and gas concentration values in the observed environment. The data obtained can be rearranged on the IoT platform, especially on the data display. With data storage on the platform and remote access features, the system can report as soon as possible if there is anomalous data after the data is processed. Further action when an abnormal change in the observation environment is found can be carried out as soon as possible by applicable regulations so that undesirable things do not happen.

The type of topology used in the design of this system is a star topology with a gateway as a single central node, as shown in Figure 1. Each sensor node in this topology has no relation to each other and is connected singly to the gateway.

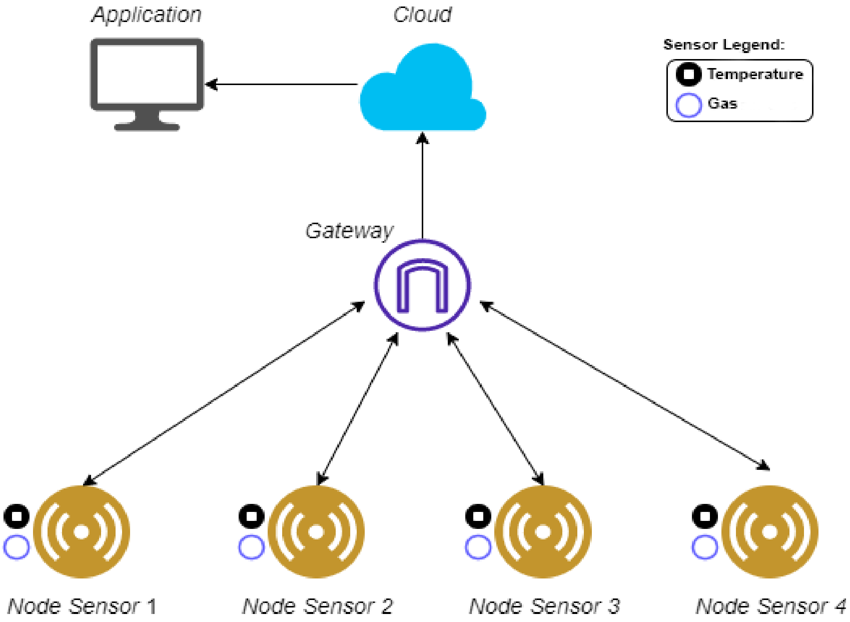


Figure 1: Intelligent laboratory monitoring system design illustration

Architectural and Topological Design

The network architecture applies the concept of a Wireless Sensor Network composed of several nodes acting as sensor nodes and gateways. Both sensor nodes and gateways have their respective functionalities according to their tasks. The sensor node, equipped with a DHT11 temperature sensor and MQ2 gas concentration, will send the sensor data to the gateway. The gateway will then receive data from all sensor nodes and send it to the IoT platform ThingSpeak (Rajesh, D., et al., 2023).

The sensor node device consists of Arduino Uno as the central controller, MQ2 as a gas sensor, and DHT11 as a temperature and humidity sensor. The device is also equipped with a LoRa Ra-02 module that operates at a frequency of 433 MHz. In addition, an external power source in the form of a 9V 1A adapter is prepared to supply Arduino Uno. Components other than Arduino Uno will obtain the resources from the Arduino Uno pinout. This is intended to provide ground and power sources that can be controlled and pre-regulated on the Arduino Uno.

The gateway device comprises Arduino Uno as a controller, LoRa Ra-02 as a LoRa module, and ESP8266 as a Wi-Fi module. The primary function of this device is as the central node in the LoRa network that controls and receives sensor data that will be transmitted to sensor nodes. Any sensor data successfully received by the gateway will be sent back to the IoT cloud, namely Thingspeak.

The gateway in the proposed system design has a vital role in the sustainability of the LoRa network. The gateway will control each transmission node sensor. Some rules and conditions are implemented at the gateway in receiving and sending data. The integrated rules use the modified standard TDMA and CSMA protocols. The gateway will determine the distribution of time slots on TDMA and channel status on CSMA. After going through a predetermined path, the data obtained by the sensor node will be retransmitted using the Wi-Fi module if the time-slot rotation condition has been reached or the gateway has fully received the data from each sensor node.

Hardware Specifications

In the proposed design, several hardware devices are used to build the Intelligent Laboratory Supervision Monitoring System. The details can be seen in Table 2.

Table 2: Hardware specifications on Intelligent Laboratory Monitoring System

Hardware	**Total**	**Description**
Arduino Uno	Five units	*Microcontroller*
DHT11	Four units	Temperature and humidity sensors
MQ2	Four units	Gas sensors
SX1278 Ra-02	Five units	LoRa Modules
ESP8266	One unit	Wi-Fi Module
Breadboards and Jumper cables	Five sets	additional peripherals

LoRa Communication Algorithm Design

The communication algorithm used at each sensor node and gateway is based on the modified TDMA and CSMA protocols. The main modification is in the network's communication settings, which the gateway will fully control. The sensor node will function as a passive device and will not transmit when it does not receive a command from the gateway. Apart from these two things, the details of the modifications applied to the TDMA-based and CSMA-based algorithms differ.

The design of the TDMA-based LoRa network algorithm modifies the gateway as a device that regulates scheduling, synchronization, and timeslot allocation for each sensor node. The timeslot or the length of time given by each sensor node to be able to transmit has the same duration. The sensor node acts as a passive device and will not transmit anything on the channel when it is outside the given timeslot. The order of channel usage is defined on the gateway statically. The gateway will synchronize by sending a key as a marker that the timeslot belonging to the sensor node with the same key is in progress.

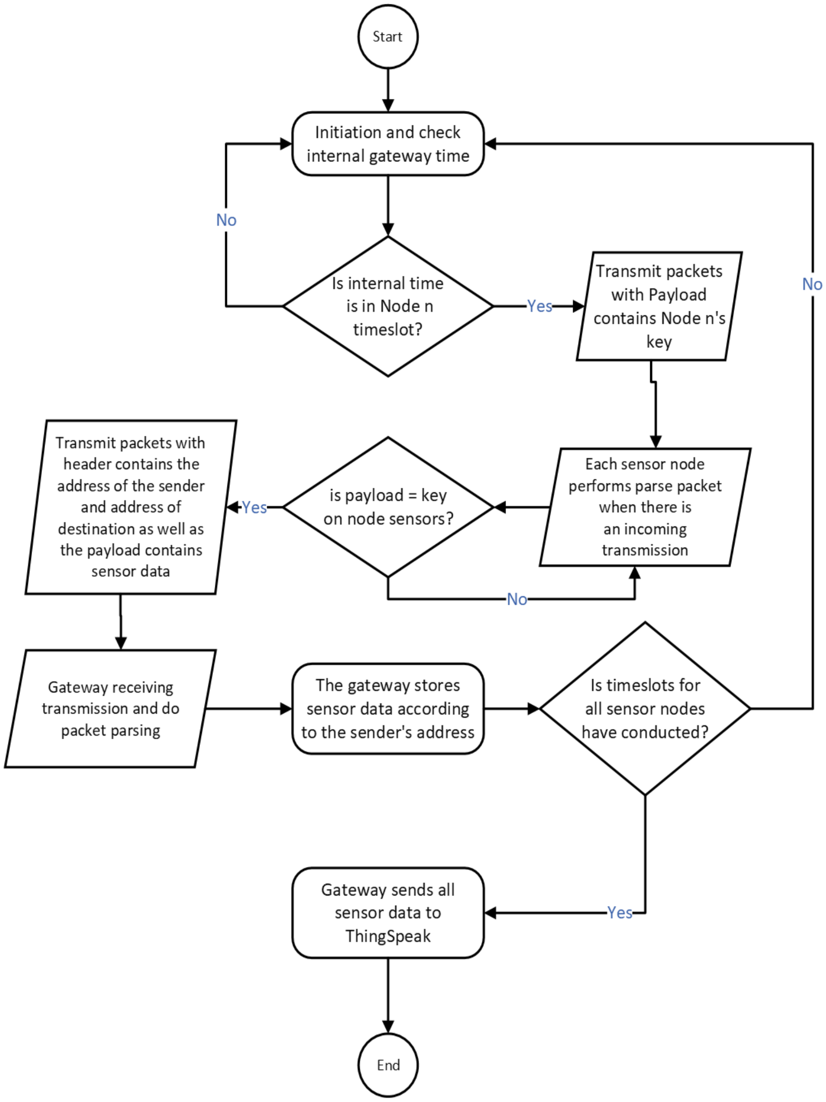


Figure 2: TDMA-based LoRa network system flowchart

Figure 2 depicts a flowchart of a LoRa network system that implements a TDMA-based algorithm. The system starts with the gateway initiation along with its internal time. The internal time at the gateway will be used as a synchronization parameter for all sensor nodes. At the specified second, the gateway will send a packet with a payload containing the key from the sensor node that has a timeslot in that time interval. Each sensor node will parse the packet to find the payload of the received packet. If the payload matches the key owned by the sensor node, the sensor node will send a reply in the form of temperature sensor data and gas concentration sensor to the gateway. The gateway will parse the packet and store the sensor data obtained internally. This flow is looped back with the gateway's internal time as the parameter determining timeslot synchronization and transition. If the rotation of all timeslots is successful, the sensor data stored internally in the gateway will be sent simultaneously to Thingspeak.

The design of the CSMA-based LoRa network algorithm uses the gateway status as a parameter that determines the gateway's availability to transmit. Gateway availability is a modification of the CSMA standard, which uses channel availability as a transmission parameter. Sensor nodes on the system will function as passive devices and operate based on information actively transmitted by the gateway.

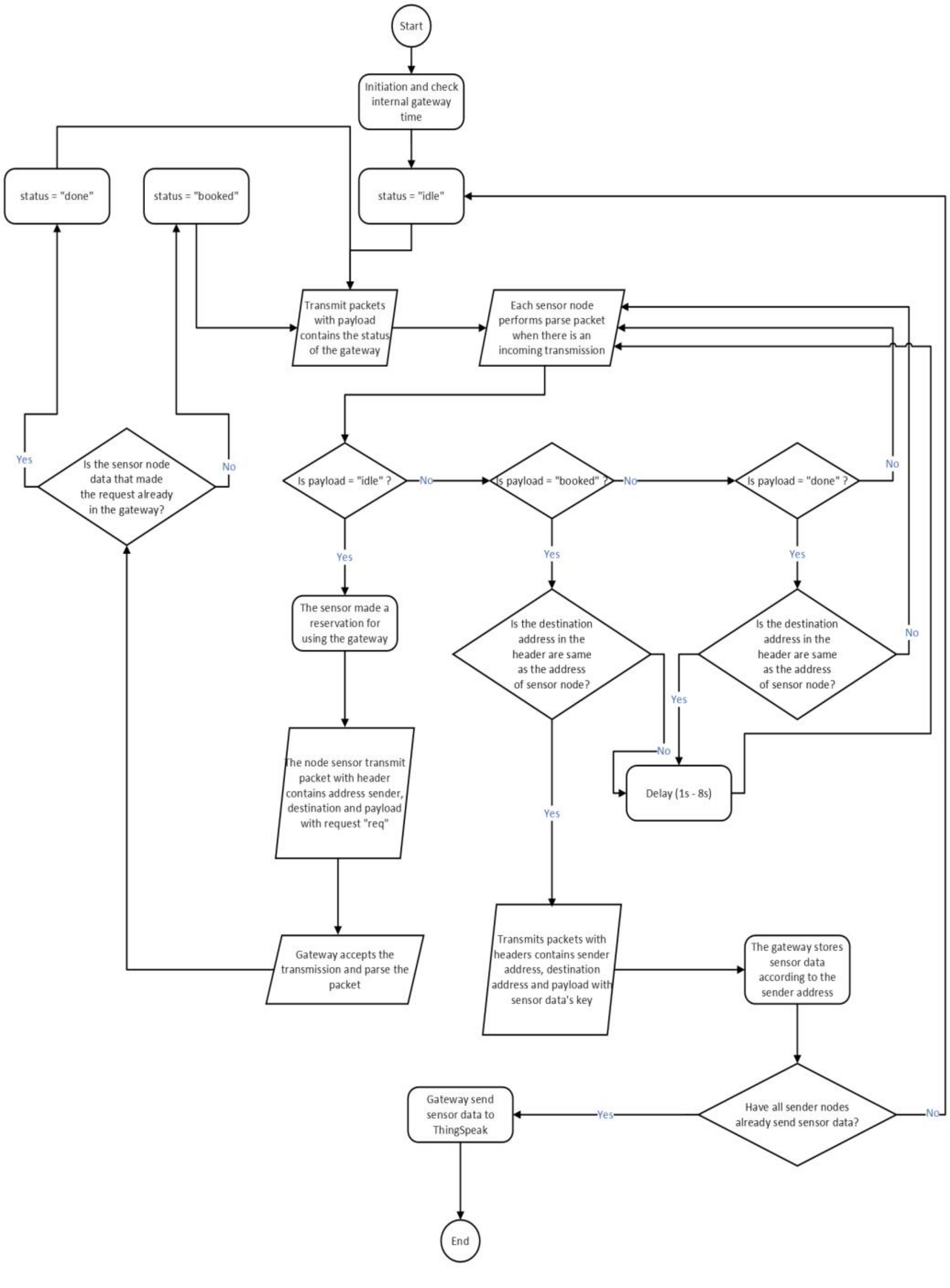


Figure 3: CSMA-based LoRa network system flowchart

Figure 3 depicts a flowchart of a LoRa network system that implements a CSMA-based algorithm. The gateway at the beginning of the initiation will be in an “idle” state and inform all sensor nodes of that status. The sensor node will parse the packet to identify the gateway's status. Suppose the gateway status is "idle". In that case, each sensor node will make a reservation to the gateway by sending a payload containing "req" and a header containing the address of the sending sensor node. The gateway can only accept one sensor node reservation at a time. The gateway's status will change to “booked” when one sensor node has successfully made a reservation. The gateway will inform the status that has changed to "booked" to the sensor node. The gateway includes the address of the sensor node in the header so that only sensor nodes that successfully make a reservation will reply in the form of temperature sensor data and gas concentration to the gateway. Other sensor nodes that are not intended will delay with a random time starting from 1 second to 8 seconds. Sensor nodes that successfully transmit data also delay with random duration from 5 seconds to 8 seconds. Once this flow is complete, The gateway will return to an “idle” state. If a sensor node is found that made a reservation, the sensor node has previously sent sensor data in the same cycle. The gateway status will change to "done." Next, the gateway will instruct the sensor node to delay with a random duration from 5 seconds to 8 seconds. The status of the gateway will then return to “idle.” The flow in this system will continue until the gateway successfully stores the sensor data of all sensor nodes. Then, the gateway can send the data to Thingspeak.

4 Implementation and Analysis

This chapter describes the implementing, testing, and analysis of the Intelligent Laboratory Monitoring System implemented with TDMA and CSMA-based algorithms for communication between IoT devices via LoRa. Variables discussed and evaluated include delivery success ratio, RSSI, transmission time for one sensor node, and execution time to perform one rotation on each algorithm.

System Implementation

The system is implemented on IoT devices with Arduino Uno as controller, DHT11 and MQ2 as gas sensors, and LoRa-02 as LoRa module. The gateway has an additional module, namely ESP866, as a Wi-Fi module. Modules with controllers are connected via cables and integrated into a prototype on a breadboard. The IoT device is supplied with a 9V 1A adapter, with the overall implementation illustrated in Figure 4 and Figure 5.

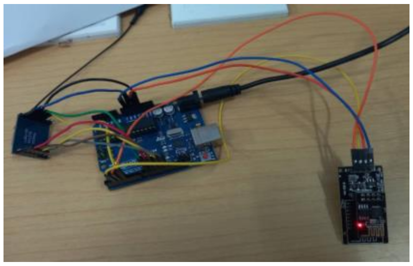


Figure 4: Gateway device

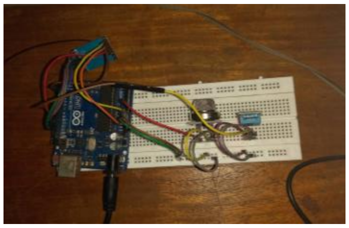


Figure 5: Sensor node device

System programming is done via Arduino IDE with some additional libraries. Some of the libraries used are SPI.h, SoftwareSerial.h, and LoRa.h. In terms of programming, the system is implemented with the gateway as the primary orientation and sensor nodes as passive devices. Systems with TDMA or CSMA-based algorithms have the same device address and essential functions in sending packets, parsing packets, separating sensor data, and using the Thingspeak API. The difference in implementing these two algorithms lies in the flow and conditions in scheduling data transmission.

Table 3: Address of Sensor Node and Gateway

Devices	***Address* (*byte*)**
gateway	0xFF
Node1	0xBB
Node2	0xCC
Node3	0xAA
Node4	0xDD

Table 3 shows the address of each device implemented on the LoRa network. Each address is allocated to each device in bytes. The gateway as the primary network controller has an address of 0xFF, while the sensor node will have an address of 0xBB for Node1, 0xCC for Node2, 0xAA for Node3, and 0xDD for Node4.

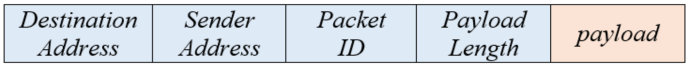


Figure 6: Packet headers and payload

This system's arrangement of the packets consists of a header as packet information and a payload as the data to be sent. In Figure 6, the header is marked with a blue block, while a red block indicates the payload.

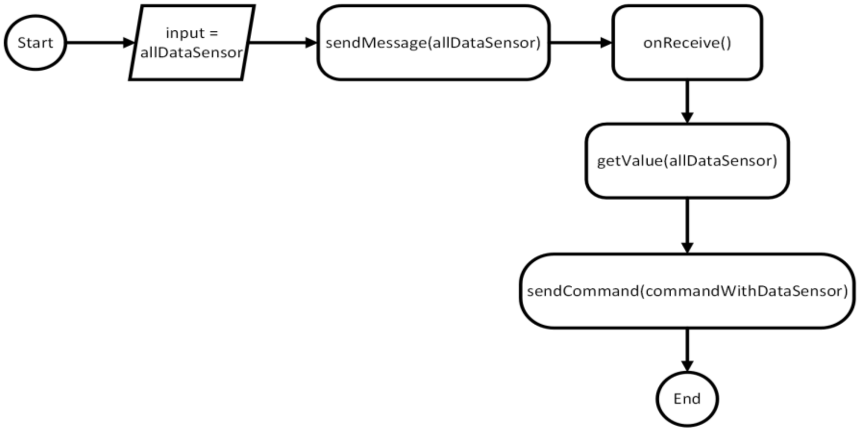


Figure 7: Sending and receiving sensor data flowchart

Figure 7 represents the flow of all sensor data successfully guessed by the sensory node, included in the packet's payload to be transmitted via LoRa until it was sent to ThingSpeak. The all Data Sensor variable is the input variable obtained by the temperature and gas concentration sensors. The sendMessage() function is a function used to send packets. This function is implemented with three parameters to construct a package. The parameters needed are the contents of the message that want to be sent (allDataSensor), the sender's address, and the recipient's address. The length of the message and the message ID will be automatically determined by this function to be included in the packet structure.

The onReceive() function shows a function to parse packets. The function will not return any value when it finds no incoming packets. The identified packets will be processed in this function by reading the header and payload individually. The payload received is sensor data such as all Data Sensor in Figure. In addition, this function will check whether the payload's length matches the payload's length before sending it. If they do not match, the following flow in the function will not continue. The following flow of this function will adjust to the LoRa communication algorithm used.

The getValue() function is a function to separate strings containing more than one concatenated sensor data. This function will accept the whole string, the delimiter as a data separator marker, and order the data based on the delimiter. The function will return a string value according to the parameters entered. The allDataSensor value will be split into multiple sensor data according to the included separator or delimiter.

The sendCommand() function is used to send AT command messages through ESP8266 in a structured manner. The parameters needed to call this function are the AT command string that wants to be used, the number of attempts at sending the AT command message, and the expected reply from ESP8266. An AT command string will be included with the Thingspeak API containing sensor data to be shipped to the Thingspeak platform. AT Command "AT+CIPMUX=1" is used to open a connection via ESP8266. "AT+CIPSTART=0 is used to initiate a connection as a client. The parameters for this AT command are accompanied by a TCP connection, port 80¸, with the Thingspeak domain as the host. "AT+CIPSEND=0," is used to transmit sensor data via ESP8266. The last AT command, "AT+CIPCLOSE=0, " closes the TCP connection.

Testing Scenario

Four scenarios are carried out to carry out the test: testing data retrieval on the Intelligent Laboratory Monitoring System, testing using a TDMA-based algorithm, trying using a CSMA-based algorithm, and testing the RSSI threshold value on the Intelligent Laboratory Monitoring System. The three initial tests used the same topology and architecture, namely the Star topology and a single gateway architecture with four sensor nodes. The device is placed in a room and several laboratories in the Electrical Engineering Department Building, Universitas Indonesia.

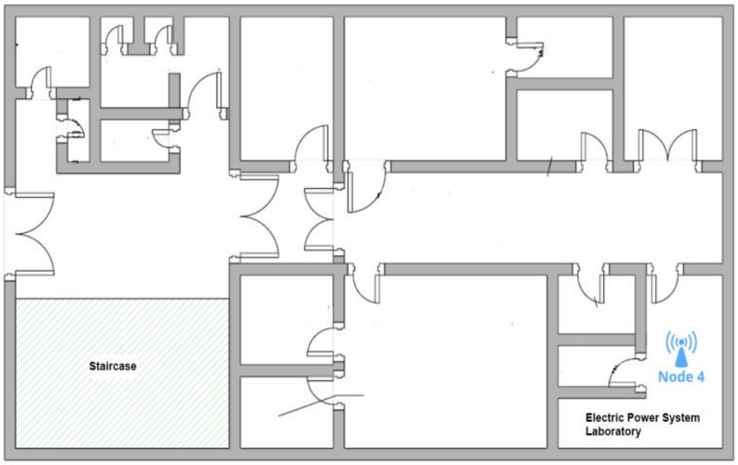


Figure 8: Device placement plan on 1st floor

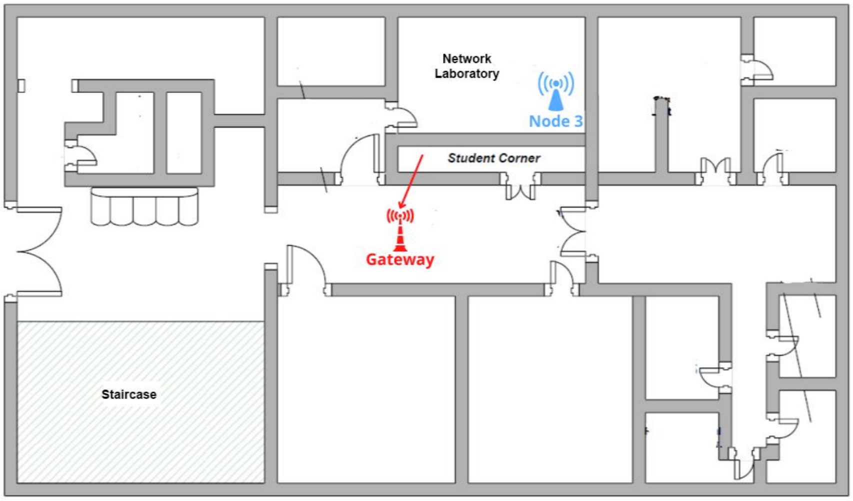


Figure 9: Device placement plan on 2nd floor

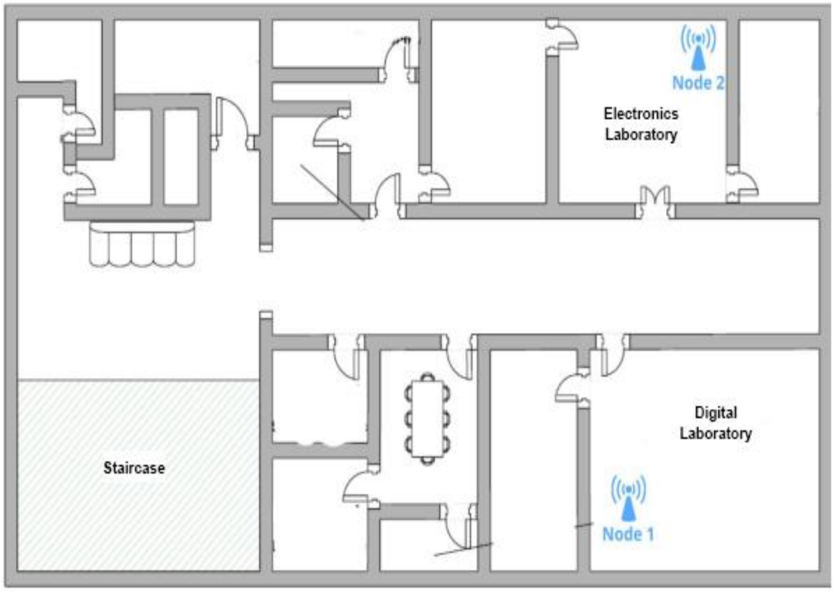


Figure 10: Device placement plan on 3rd floor

Figure 8, Figure 9, and Figure 10 illustrate the placement of sensor nodes for sensor data retrieval scenarios and scenarios for implementing TDMA and CSMA algorithms. Sensor nodes are placed on the 1st, 2nd, and 3rd floors of the Electrical Engineering Department Building, University of Indonesia, respectively. Each sensor node is given a name using numbering. Node1 is placed in the Digital Lab, Node2 in the Electronics Lab, Node3 in the Network Lab, and Node4 in the Electrical Power Lab.

The first scenario tests data retrieval in a closed loop to determine the success rate and results of sensor data collection carried out by the gateway. This scenario was carried out for eight days, on 30-31 May 2022 and 2-8 June 2022.

The second and third scenarios test the implementation of TDMA and CSMA-based algorithms on LoRa networks aimed at knowing RSSI, transmission duration for data collection of one sensor node, and overall time for data collection from all sensor nodes obtained by the gateway.

The fourth scenario is testing the RSSI value threshold on the Intelligent Laboratory Monitoring System, which is intended to determine the correlation of the RSSI value with the threshold based on variations in device placement in the Electrical Engineering Department Building. The test was carried out with two LoRa devices connected point-to-point. One of the devices will function as a sensor node and send three sensor data from DHT11. Sensor data will be received by the LoRa receiver device to parse and evaluate RSSI. The test was carried out under two conditions: conditions between devices in the same room and between devices in different rooms.

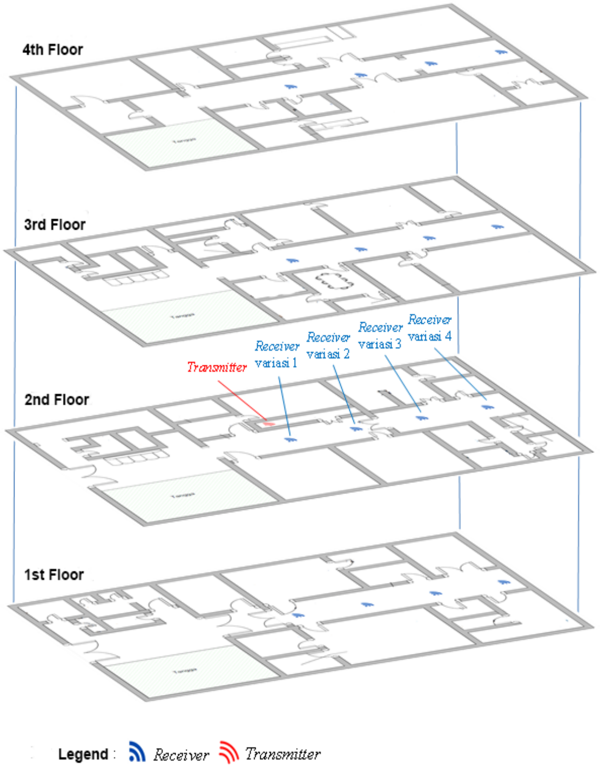


Figure 11: Transmitter placement plan and receiver placement variations on each floor

Figure 11 is an illustration that represents the placement between devices in different room conditions. Four placement variations have a distance difference of 4 meters between variations. The four variations of receiver placement are then carried out on floors 1 to 4, with the transmitter devices placed in the exact location, namely on the 2nd floor. The final distance between the devices is obtained from the results of a straight line using the Pythagorean theorem with a difference between floors of 3 meters.

Testing Device Parameters

Parameter determination defines the LoRa module configuration in the test scenario. Parameters determined in LoRa are frequency, Tx power, spreading factor, bandwidth, coding rate, and preamble length. The parameter values used can be observed in Table 4.

Table 4: LoRa Device Parameters

Frequency	443 Mhz
Tx power	17 dB
Spreading factor	7
Bandwidth	125.000 Hz
Coding rate	4/5
Preamble length	8

Results and Analysis of Data Collection Tests on Intelligent Laboratory Monitoring Systems

The results of data retrieval that have been collected can be observed on the Thingspeak display. The test was carried out for eight days; the gateway collected data from the sensor node network as much as 10966 times and stored the data obtained on Thingspeak. One data collection cycle consists of collecting temperature sensor data and gas concentration sensors from each sensor node. The data will be collected as much as 2 (sensor variations) multiplied by 4 (many sensor nodes), namely 8. The total data collected by the gateway and stored in Thingspeak is 10966 multiplied by 8 or 87728 data.

Based on testing data collection for eight days in a closed environment of the Department of Electrical Engineering building on the Intelligent Laboratory Supervision System, it was found that the success rate of data collection by the gateway was varied. Figure 12 is the percentage of successful collection of each sensor node.

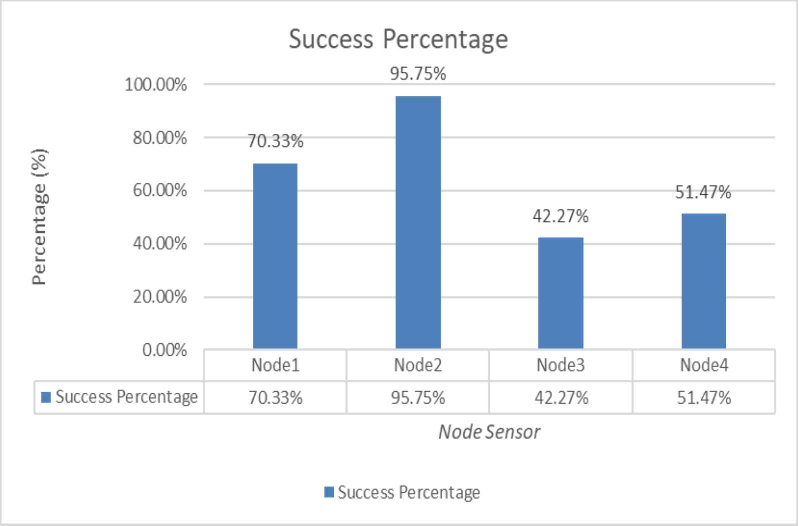


Figure 12: Graph of the percentage of successful data collection from each sensor node

It can be observed that the most significant percentage is in the collection of sensor data from Node2, which is placed in the Electronics Lab with a rate of 95.75%. At the same time, the smallest percentage is in data collection from Node3, which is placed in the Network Lab with a percentage of 42.27%. The percentage results that are not maximized are caused by several things, and one of them is that there is a sensor node device that is not active when the data collection stage is in progress. The test period is past the holiday on June 4-5, so some sensor nodes are inactive because they are not supplied with power through the adapter connected to the plug. Even if a sensor node is passive, the gateway will continue to collect data and send it to Thingspeak on each rotation. Another factor that affects this success rate is not having a watchdog timer on the sensor node as a follow-up when the device overheats, memory or RAM is full, infinite looping, or other obstacles that cause the device not to operate. With the watchdog timer, the device will restart when the preset timer has exceeded the limit. This feature has been implemented on the gateway but not the sensor node. This is because the maximum limit of the watchdog timer is 8 seconds, and the sensor node in this system is implemented as a passive device. The sensor node has a high probability of triggering an unnecessary restart of the watchdog timer because the sensor node will do nothing when the gateway does not issue any commands. This scheme occurs when the gateway transmits data to Thingspeak, which can take more than 8 seconds.

Another test carried out is the measurement of RSSI or received signal strength indicator to determine the signal strength between the gateway and sensor nodes placed in the building in the Intelligent Laboratory Monitoring System. The RSSI value will show the signal strength between LoRa devices connected.

Table 5: Estimated distance and placement difference between sensor node and gateway

Node Sensor	**Estimated Distance from the gateway**	**Floor difference from the gateway**
Node1	22 meter	One floor
Node2	20 meter	1 l floor
Node3	2 meter	0 floor (different room)
Node4	25 meter	One floor

Table 5 shows the difference between sensor nodes regarding distance and interference in the form of floor differences. Node3, placed in the Network Lab, has the closest space, is placed on the same floor as the gateway, and has the highest RSSI value compared to other sensor nodes. This shows that Node3 has the strongest signal in the tests carried out. Other sensor nodes on different floors and a considerable distance from the gateway have a significant difference in RSSI value when compared to the RSSI of Node3, Node1, Node2, and Node4 are in identical conditions with different placements on one floor and a distance of 20 meters to 25 meters from the gateway. Node2 is the second closest sensor node to the gateway; it has the smallest RSSI value of -86. This proves that the smaller the difference in distance between devices does not guarantee the RSSI value will increase. Other factors, such as interference and floor structures, such as walls between rooms and other physical objects, can be a barrier to using signal transmission media. For Node1 and Node4, the RSSI value obtained by the two sensor nodes has the same pattern as Node3, where the amount of interference is not too dominant.

This analysis proves that the distance and interference from the building arrangement that limits the floors affect the RSSI value of devices communicating via LoRa. As the distance between devices increases and the amount of interference increases, the RSSI value will decrease or further away from the 0 value. This applies to the opposite; the RSSI value will increase or approach the 0 value when the distance between devices decreases, and the amount of interference decreases.

Results and Analysis of LoRa Networks with TDMA and CSMA-based Schemes

The implemented system was tested with two different algorithms: TDMA (Time Division Multiple Access) and CSMA (Carrier Sense Multiple Access). The tests carried out show that there are differences in results between these two algorithms. Differences in outcomes can be observed in the success rate, data collection of one sensor node, and data collection of all sensor nodes.

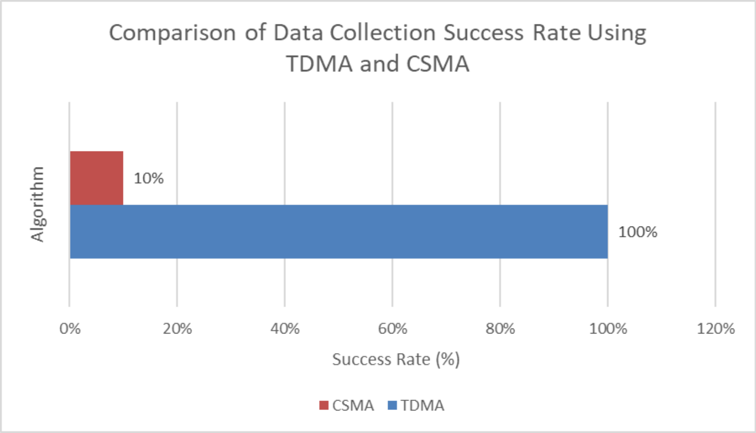
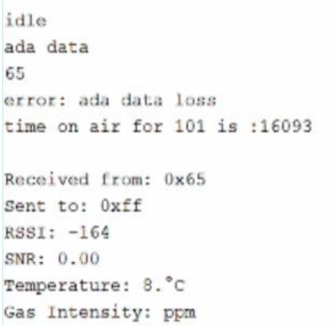


Figure 13: Comparison graph of the percentage of success in data collection using TDMA and CSMA

The graph in Figure 13 shows the difference in the percentage of success when the system is implemented with the TDMA and CSMA algorithms. It can be observed that the use of the CSMA algorithm has a much smaller success percentage compared to TDMA. The use of the CSMA algorithm only succeeded in collecting data from the entire sensor node once out of 10 trials. At the same time, using the algorithm is much better by successfully retrieving sensor node data in each experiment.

The low success rate of using CSMA-based algorithms is caused by the gateway's scheme or flow of data collection. When the gateway informs that it is in an "idle" state, each sensor node receiving this information will transmit it to make a reservation. Sensor nodes that do not consist of only one device will transmit at identical times to each other. Especially on Node1, Node2, and Node4, which have RSSI with a difference of 6 dB from each other. The LoRa manufacturer's documentation page, Semtec, explains that collisions are very likely to occur if there is a transmission-reception with an RSSI difference of less than 6 dB (LoRa and LoRaWAN). Implementing the CSMA algorithm in this system has a scheme with four different signal waves on one channel and the exact destination so that collisions and distortions will occur. The signal received by the gateway will change from its proper form because the signal from the sensor node will interfere with each other.

Interference results between sensor nodes can be observed in the serial display on the gateway. The gateway has a problem when the device's status is "idle," so four different signal waves will use the same channel. The high level of signal interference between sensor nodes will increase the possibility of a collision so that the data is distorted. It can be observed in Figure 14 that the packet parsing carried out by the gateway has distortions where the address sender does not exist in the LoRa network and the sensor data is incomplete or has no value at all.


```
idle
ada data
65
error: ada data loss
time on air for 101 is :16093

Received from: 0x65
Sent to: 0xff
RSSI: -164
SNR: 0.00
Temperature: 8.°C
Gas Intensity: ppm
```

Figure 14: Sensor data distortion on CSMA-based system

The Intelligent Laboratory Monitoring System, both a system with a TDMA and CSMA-based system, applies a mechanism in the form of a function to check the length of the payload before it is sent and after it is received. The checksum function will compare the value of the variable incoming Length and the value of incoming.length(). The variable incoming Length is a variable obtained from the packet header that is included before the packet is sent, representing the length of the payload before it is sent. At the same time, incoming.length() is an incoming variable that is included with the syntax length() to find out the length of the variable. The incoming variable is the payload value that has passed the parsing stage at the gateway.

The length of the payload that does not match when the packet is sent and received proves that the data was successfully transmitted, but there is an obstacle that causes the data to be distorted. One of the obstacles that can occur is the collision between signals at the same time and channel, which is a phenomenon that occurs in CSMA-based systems. The checksum function will compare the value of the variable incomingLength with the value of incoming.length() and drop the packet if the comparison result does not match. Incorrect payload length due to collision or interference will be automatically decreased on this function, affecting the communication success rate, especially on CSMA-based systems.

Another reason that affects the low percentage of successful use of the CSMA algorithm is the parameters used in LoRa devices. Parameters configured for each device include a spread factor of 7 and a coding rate of 4/5. The selected parameter value is the smallest option compared to other value options. This will affect the device's characteristics in receiving and transmitting LoRa signals. The spread factor value will affect the number of chips that comprise the symbol with the 2^SF formulation. The composition that defines the character will be more specific if the spread factor value is increasing so that the impact of noise can be minimized. Another parameter, namely the coding rate, is intended so that the system has redundancy to make corrections when there is a failure. A coding rate with a value of 4/5, applied to the test, uses 1-bit redundancy from the existing 4 bits. The higher the coding rate used, the prepared redundancy will increase so that corrections can be made better. Increasing the value of these two parameters will improve the system's reliability, although the transmission will take longer due to the lower bit rate. Based on the tests carried out, the parameters used must be according to the conditions and needs of the system to get maximum results.

Another comparison between TDMA and CSMA that can be observed is the difference in the collection duration of each sensor node individually. The results of the collection duration are represented in the graph of Figure 15.

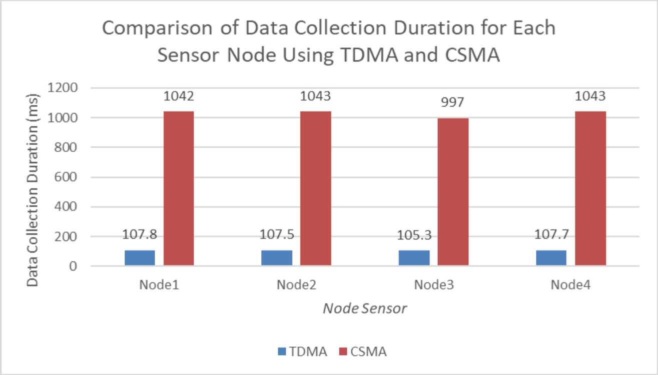


Figure 15: Comparison graph of data collection duration for each sensor node using TDMA and CSMA

It can be observed that there is a significant difference between the duration of data collection through the TDMA and CSMA algorithms. The system that implements the CSMA algorithm takes about ten times longer than the system with the TDMA algorithm. This is due to differences in scheduling schemes that affect the amount of information exchanged before the gateway receives sensor data. The system that implements the TDMA algorithm only requires one-way, two-way communication to collect sensor data. In this scheme, the gateway only needs to send the key from the timeslot and then wait for a reply in the form of sensor data from the relevant sensor node. The system with the CSMA algorithm requires two-way communication twice to get data from one sensor node. The gateway must transmit to an "idle" state and wait for transmission from the sensor node to make a reservation. The gateway then retransmits to inform the relevant sensor node so that it can send sensor data. After that, the gateway is successful. Programming for determining the status and reservation process is a function only found on the gateway based on the CSMA algorithm, so the system will need more time to collect sensor data.

Another comparison based on the duration that can be observed is the difference in the overall duration of data collection in a purely systematic manner, specifically for TDMA or CSMA. The TDMA algorithm has a special systematic scheduling, namely timeslot, while the CSMA algorithm has a systematic delay on the sensor node device according to the gateway's status. The total duration of data collection is the entire duration of data collection of the individual sensor nodes that are summed. This entire duration on TDMA is the sum of 107.8 ms (Node1) + 107.5 ms (Node2) + 105.3 ms (Node3) + 107.7 ms (Node4) or 428.3 ms. As for CSMA, the total duration is purely the sum of 1042 ms (Node1) + 1043 ms (Node2) + 997 ms (Node3) + 1043 ms (Node4) or 4125 ms. This total duration will increase when the timeslot on TDMA and delay on CSMA are included in the calculation. For one cycle of collecting all sensor data, the system with the TDMA algorithm takes 2760.5 ms while the system with the CSMA algorithm takes 12966 ms. The results of these two types of duration are represented in the graph of Figure 16.

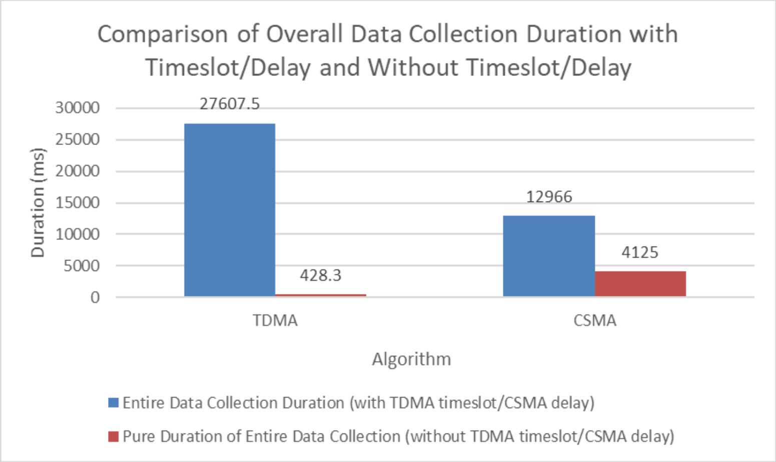


Figure 16: Comparison graph of overall data collection duration with timeslot/delay and without timeslot/delay

When viewed from the aspect of pure duration without any timeslot or delay, the system with the TDMA algorithm requires a shorter time than the system using the CSMA algorithm. This is due to the difference in the number of communications the gateway makes with the sensor node in each algorithm. Meanwhile, in collecting all data in one complete cycle accompanied by timeslot on TDMA and delay on CSMA, the system with the CSMA algorithm requires a shorter time than the system based on the TDMA algorithm. The use of the CSMA algorithm only takes 12966 ms, while the TDMA algorithm takes about twice as much time as CSMA with a duration of 27607.5 ms. This is due to the timeslot duration, set in the test with a period of 6 seconds. The TDMA-based test system has a relatively static overall duration with a minimum duration of 24000 ms, obtained from 4 (number of sensor nodes) multiplied by 6 seconds (timeslot duration).

The duration of the tested CSMA-based system will always be faster than the TDMA-based system if the obtained duration is below 24000 ms. The CSMA algorithm allows for more flexible scheduling and faster collection because sensor nodes can send sensor data directly if a gateway is available. On the other hand, CSMA-based algorithms can also cause slower data collection due to competition between sensor nodes to use the gateway. Gateways in an "idle" state will not receive transmissions as long as there is a continuous delay on two or more sensor nodes that receive the same random delay result. The duration of the CSMA-based system also increases when the maximum delay is found, which is 8 seconds at any time. Reservations made by sensor nodes that have previously made reservations also increase the duration of the data collection cycle because the gateway requires additional action to reject the reservation and instructs the sensor node to perform a random delay.

Results and Analysis of RSSI Threshold Values in Intelligent Laboratory Monitoring Systems

The results of the RSSI values obtained based on testing on the placement of devices in the same room with placements in different rooms can be compared to determine the correlation of RSSI values with the placement of devices that communicate via LoRa.

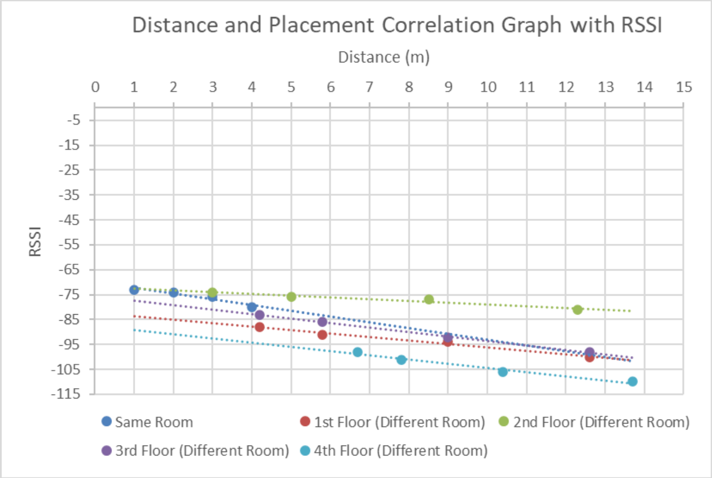


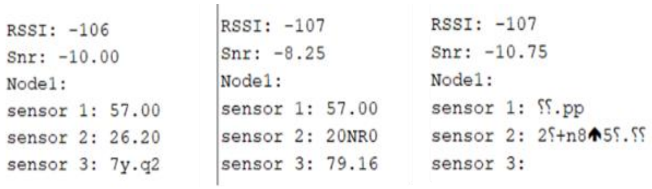
Figure 17: Graph of distance and placement correlation with RSSI

Figure 17 shows the RSSI value based on distance, placement location, and placement conditions. Five lines represent placement conditions, with each state having four variations. These four variations in the form of data are then used to construct a linear line that describes the characteristics of the RSSI value in the placement conditions. The five placement conditions tested in this test are one for placing the device in the same room and four for placing the device in a different room. The difference in floors from the 1st to the 4th floor is the placement conditions for testing instruments in other rooms.

Based on the characteristics of the linear line for each receiver placement condition and the variation in its placement, the more significant the difference in floor placement, the more the RSSI value will decrease or weaken. This is evidenced by the placement of the receiver on the 4th floor, where the RSSI value obtained is the weakest compared to other conditions. For device placement on the same floor, namely on the 2nd floor, the RSSI value for devices placed in different rooms is more substantial than for devices in the same room. This is due to interference for different room conditions in the form of a room dividing wall, while conditions in the same room have more interference than walls due to the crowd or density of visitors.

Another result that can be observed is comparing the RSSI value when the receiver is placed on the 1st and 3rd floors. These two conditions have the same distance for each variation but have different RSSI values. This proves that the RSSI value is not limited to distance and floor difference interference but also other interference between LoRa devices.

The threshold is obtained by finding the maximum RSSI limit where the LoRa device in the Intelligent Laboratory Monitoring System can still function properly. This test found distorted data on the RSSI with values of -106 and -107 when taking measurements on the 4th floor. The distortion displayed on the serial receiver device can be observed in Figure 18.



```
RSSI: -106          RSSI: -107          RSSI: -107
Snr: -10.00         Snr: -8.25          Snr: -10.75
Node1:              Node1:              Node1:
sensor 1: 57.00     sensor 1: 57.00     sensor 1: ʕʕ.pp
sensor 2: 26.20     sensor 2: 20NR0     sensor 2: 2ʕ+n8⬆5ʕ.ʕʕ
sensor 3: 7y.q2     sensor 3: 79.16     sensor 3:
```

Figure 18: Distortion of data on threshold RSSI

Based on the overall retrieval of the RSSI value on the RSSI threshold value test on the Intelligent Laboratory Monitoring System, there are three incidents where the sensor data sent by the transmitter is distorted. This event occurs in the state between devices when the RSSI values are -106 and -107. Data distortion occurs as a letter in the sensor data that should be numeric. The data gets distorted when the RSSI value decreases or weakens, as in the RSSI test with a value of -107. Sensor data has a data length outside the specified length and is accompanied by letters, symbols, and punctuation marks that should not be found in sensor data. It can be concluded that the limit of the device operating optimally or the RSSI threshold on the Intelligent Laboratory Monitoring System is in the placement between devices with an RSSI value of less than -106.

5 Conclusion

The LoRa communication algorithm scheme in the Intelligent Laboratory Monitoring System applies the concept based on the TDMA and CSMA protocols. The TDMA-based system is implemented with a timeslot of 6 seconds, while the CSMA-based system is implemented with a random delay of 1 to 8 seconds. The system has basic functionality by setting a gateway that acts as an active device while the sensor node is a passive device.

There are differences between TDMA-based algorithms and CSMA when implemented on LoRa networks. Compared to the TDMA algorithm, CSMA is more prone to collisions; CSMA takes longer to collect data from a single sensor node; more flexible CSMA scheduling; and CSMA has a more complex mechanism. In addition, the distance and interference from the building arrangement that limits the floors affect the RSSI value of the devices that communicate via LoRa. The limit of the device operating optimally or the RSSI threshold on the Intelligent Laboratory Monitoring System is in the placement between devices with an RSSI value less than -106.

6 Acknowledgments

This work has been supported by Hibah Riset Kompetitif (PUTI Q2 2022) Research Grant 2022 from Universitas Indonesia No. NKB-1335/UN2.RST/HKP.05.00/2022.

References

[1] Acharyya, I.S., Al-Anbuky, A., & Sivaramakrishnan, S. (2019). Software-defined sensor networks: towards flexible architecture supported by virtualization. *In Global IoT Summit (GIoTS)*, 1-4.

[2] Baddula, M., Ray, B., & Chowdhury, M. (2020). Performance evaluation of Aloha and CSMA for LoRaWAN network. *In IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*, 1-6.

[3] Camgözlü, Y., & Kutlu, Y. (2023). Leaf Image Classification Based on Pre-trained Convolutional Neural Network Models. *Natural and Engineering Sciences*, *8*(3), 214-232.

[4] Eridani, D., Widianto, E.D., & Augustinus, R.D.O. (2019). Monitoring system in LoRa network architecture using smart gateway in simple LoRa protocol. *In international seminar on research of information technology and intelligent systems (ISRITI)*, 200-204.

[5] Fitriawan, H., Rohman, R.C., Herlinawati, H., & Purwiyanti, S. (2020). RSSI Measurement of ZigBee Based Wireless Sensor Networks on Various Topologies. *Journal of Electrical Engineering*, *16*(2), 15-21.

[6] Hassija, V., Chamola, V., Saxena, V., Jain, D., Goyal, P., & Sikdar, B. (2019). A survey on IoT security: application areas, security threats, and solution architectures. *IEEE Access*, *7*, 82721-82743.

[7] Kim, H. (2020). 5G core network security issues and attack classification from network protocol perspective. *Journal of Internet Services and Information Security (JISIS)*, *10*(2), 1-15.

[8] Kumar, S., Tiwari, P., & Zymbler, M. (2019). Internet of Things is a revolutionary approach for future technology enhancement: a review. *Journal of Big data*, *6*(1), 1-21.

[9] Liya, M.L., & Aswathy, M. (2020). LoRa technology for Internet of Things (IoT): a brief survey. *In fourth international conference on I-SMAC (IoT in social, mobile, analytics and cloud) (I-SMAC)*, 8-13.

[10] LoRa Alliance, "What is LoRaWAN® Specification - LoRa Alliance®." https://lora-alliance.org/about-lorawan

[11] LoRa Semtech Website, "LoRa and LoRaWAN: Technical overview | DEVELOPER PORTAL," LoRa Semtech Developer Portal Documentation. https://lora-developers.semtech.com/documentation/tech-papers-and-guides/lora-and-lorawan

[12] Majeed, A., & Zia, T.A. (2017). Multi-Layer Network Architecture for Supporting Multiple Applications in Wireless Sensor Networks. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JOWUA)*, *8*(3), 36-56.

[13] Mekki, K., Bajic, E., Chaxel, F., & Meyer, F. (2019). A comparative study of LPWAN technologies for large-scale IoT deployment. *ICT express*, *5*(1), 1-7.

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

[15] Piyare, R., Murphy, A.L., Magno, M., & Benini, L. (2018). On-demand TDMA for energy efficient data collection with LoRa and wake-up receiver. *In 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 1-4.

[16] Rajesh, D., Giji Kiruba, D., & Ramesh, D. (2023). Energy Proficient Secure Clustered Protocol in Mobile Wireless Sensor Network Utilizing Blue Brain Technology. *Indian Journal of Information Sources and Services (IJISS)*, *13*(2), 30–38.

[17] Ratuloli, F.H.E., Budi, A.S., & Bhawiyuga, A. (2021). Implementation of an Anti-Collision Scheme Using TDMA and TPSN Methods on a LoRa Based WSN System. *Journal of Information Technology and Computer Science Development*, *5*(1), 283-290.

[18] Senthilkumar, M., & Uma, S. (2013). High Speed 3d DWT VLSI Architecture for Image Processing Using Lifting Based Wavelet Transform. *International Journal of Communication and Computer Technologies (IJCCTS)*, *1*(2), 87-91.

[19] Shahjalal, M., Hasan, M.K., Islam, M.M., Alam, M.M., Ahmed, M.F., & Jang, Y.M. (2020). An overview of AI-enabled remote smart-home monitoring system using LoRa. *In International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, 510-513.

[20] Sreenivasu, M., Kumar, U.V., & Dhulipudi, R. (2022). Design and Development of Intrusion Detection System for Wireless Sensor Network. *Journal of VLSI Circuits and Systems*, *4*(2), 1-4.

[21] Su, Z.Y., Lin, Y.N., & Shen, V.R. (2019). Intelligent environmental monitoring system based on LoRa long range technology. *In IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE)*, 354-357.

[22] To, T.H., & Duda, A. (2018). Simulation of lora in ns-3: Improving lora performance with csma. *In IEEE International Conference on Communications (ICC)*, 1-7.

[23] Tsoi, M.H., Ng, T.H., Lun, D.P., Choy, Y.S., & Mung, S.W. (2020). LoRa Data Throughput Enhancement by Slotted Channel Activity Detection. *In IEEE Asia-Pacific Microwave Conference (APMC)*, 466-468.

Authors Biography

I Gde Dharma Nugraha received his B.E. degree in Electrical Engineering from Universitas Indonesia (UI), Depok, Indonesia, in 2005; M.E. degree in Computer and Electronic Engineering from UI in 2009; and Ph.D. degree in Electrical and Computer Engineering from Chonnam National University in 2020. He is working as a lecturer in the Computer Engineering Study Program, Electrical Engineering Department, Faculty of Engineering, Universitas Indonesia. His research is about the Internet of Things, Embedded Systems, Cyber Security, Big Data Architecture, and Machine Learning.

Edwiansyah Zaky Ashadi received his B.E. degree in Computer Engineering from Universitas Indonesia (UI), Depok, Indonesia, in 2022. He is working as a professional in field of computer security. His research is about the Internet of Things, Wireless Sensor Networks.

Ardiansyah Musa Efendi is a senior staff engineer in the IoT Chipsets Design Algorithm Lab at Huawei Singapore Research Centre, Singapore. He received an M. Eng in Electronics & Computer Engineering, and a Ph.D. in Artificial Intelligence Convergence from Chonnam National University, Gwangju, South Korea, in 2014 and 2021, respectively. He got a B.Eng. Degree in Computer Engineering from Universitas Indonesia, Indonesia in 2006. Dr. Ardi has published several journal and conference papers related to IoT and sensor network systems. In addition to sensor networks, Dr. Ardi is interested in algorithm development and optimization for wireless sensing and machine learning models.