

2022-02

Agro-Meteorological data Collection using a LoRaWAN-Based IoT Sensor Network

Mathew, Gaudence

International Journal of Advances in Scientific Research and Engineering

<https://doi.org/10.31695/IJASRE.2022.8.2.15>

Provided with love from The Nelson Mandela African Institution of Science and Technology

Agro-Meteorological data Collection using a LoRaWAN-Based IoT Sensor Network

Gaudence L. Mathew¹, and Dr. Mussa Ally Dida²

Research Scholar ¹, Lecture²

Department of Communications Science and Engineering

Nelson Mandela African Institution of Science and Technology

Arusha, Tanzania

ABSTRACT

Generally, the variation of meteorological conditions has a significant influence on the agriculture sector in general. In Tanzania, the study conducted in 2015 found most of the existing automatic meteorological stations are not working, and the surveyed manual meteorological stations record meteorological data in the cards daily, but cards are sent in a month to month time interval to the respective organs, this makes the concept of early warning difficult. Therefore having near-real-time meteorological data will enable the crop monitoring and forecasting systems to monitor crop development and provide early warning to the farmers and the government. The advancement of the Internet of Things (IoT) offers possibilities for developing an integrated sensors network to collect meteorological data on a near-real-time basis. In this study, the developed integrated sensor network includes sensors, which record remote meteorological data such as rainfall, humidity and temperature, a communication network, and a web application that enhance data visualization in both graphical and tabular format. In the communication system, the LoRa technology was used, which is more preferable compared to other Low Power Wide Area Network (LPWAN) technologies such as NB-IoT and Sigfox for this application. The developed system uses the LoRa Gateway Operating system which provides capabilities to build a private network, and make it more cost-efficient by reducing the operation cost for account subscription, in online platforms such as The Things Network (TTN), ThingPark, ThingSpeak, and Loriot to avoid free accounts limitations. Moreover, the developed system can work in remote areas with limited Internet access as the meteorological stations can communicate with the gateway at a distance of up to 25 km.

Key Words: Internet of Things, LoRaWAN, LPWAN, Meteorological station, Web Application.

1. INTRODUCTION

Access to reliable meteorological data is very crucial in the agriculture sector since it provides vital inputs in monitoring crop growth. Therefore the availability of meteorological data measured in near-real-time such as rainfall, relative humidity, wind, and solar radiation can be used in crop monitoring and forecasting systems (Toreti et al., 2019) and provide early warning to both farmers and the government. Tanzania Meteorological Agency (TMA) and the Ministry of Agriculture, Livestock and Fisheries (MOA) have installed several manual and automatic meteorological stations across the country. Even though the existing meteorological stations face some challenges, they provide crucial input to the agriculture sector. The manual meteorological stations are not only facing the problem of delaying meteorological data to reach the respective organs but also digitalizing all the collected meteorological data is a challenge. The survey done in 9 manual meteorological stations owned by the Tanzania Ministry of agriculture around the Arumeru district in the Arusha region found that weather data is recorded in cards daily; however, cards are sent in a month to month interim to the respective organs this delay make crop growth monitoring difficult. The study done by E.Kondela et al (Kondela et al., 2015) found most of the automated meteorological stations are not functioning due to several reasons, including poor communication systems, and the push to densification and replacing non-working automated meteorological stations is hindered by significant expenses in maintaining and purchasing the new ones. Hence having local owned automated meteorological stations won't just diminish the purchasing and maintenance expenses yet, also, will provide full access and control on collected meteorological data.

The ongoing progression of the IoT and wireless sensor networks (WSNs) provides the possibility of developing an integrated sensor network for agro-meteorological data collection by deploying Low Power Wide Area Network (LPWAN). Having an adaptive data rate and the ability to deploy a private network gives LoRa technology advantages over Sigfox and NB-IoT technologies. Furthermore, LoRa has a higher data rate and unlimited maximum number of messages per day, which makes it better than Sigfox which has only 140 uplink and 4 downlink messages per day. Again LoRa has higher noise immunity and coverage distance compared to NB-IoT (Mekki et al., 2019), and in addition to that, LoRa can send recorded meteorological from rural areas where there is limited network infrastructure (Byanyuma et al., 2013). For the reasons mentioned above, this study aimed to use the LoRaWAN network to develop a wireless sensor network for agro-meteorological data collection in Tanzania.

This study is organized as follows: Section 2 contains an overview of LoRaWAN, while section 3 contains material used and methods. Section 4 presents results and discussion, and lastly, section 5 is the conclusion.

2. OVERVIEW OF LoRaWAN

LoRa radio modulation technology was developed by Cycleo and was acquired and patented by Semtech in 2014 (Sornin et al., 2016). Unlike NB-IoT technology, which operates in licensed LTE (long-term evolution) frequency bands, the LoRa physical layer operates in the unlicensed industrial, scientific, and medical (ISM) radio bands that is 433 MHz and 868 MHz in Europe and 915 MHz in North America. The ability to spread signal over wide frequency bands makes LoRa stronger against interference and jamming. LoRa uses the Chirp Spread Spectrum (CSS) modulation technique, whereby the carrier signal consists of chirps signals, whose frequency increases and decreases with time (Cattani et al., 2017). LoRa technology has two different types of chirps that is the base chirp and modulated chirp. The base chirp starts with the minimum frequency of $f_{min} = -\frac{Bw}{2}$ and the maximum frequency of $f_{max} = +\frac{Bw}{2}$: Bw donates signal's bandwidth, while the modulated chirp is cyclically time shifted base chirp (Haxhibeqiri, 2018). The LoRa's chirps facilitate the LoRa signal to travel over long distances (Haxhibeqiri, 2018). The LoRa modulation has a link budget of above 140 dB with the transmission power of 14 dBm and the receiver sensitivity of -137 dBm, these allow it to cover a long communication range (Lavric, 2019).

LoRaWAN network supports star topology, where the multiple end devices send data to the LoRaWAN gateway. Several gateways can be connected to one server. The role of LoRaWAN gateways is to forward raw data packets from the end devices to the network server. LoRaWAN standard supports three classes of end devices that is class A, B, and C depending on the requirement of the Internet of Things (IoT) application. Class A devices have the basic sets and options that are implemented by every LoRa end device for them to join the LoRaWAN network. This class of devices supports bidirectional communication, whereby a device's uplink transmission is followed by short downlink receive windows. When an end device sends an uplink message, it waits for acknowledgements (Fehri et al., 2018), and if no acknowledgement is received, the LoRaWAN uses the backoff algorithm that has been implemented in the LoRaWAN application layer to calculate waiting time before the retransmission of the message (Polonelli et al., 2019). The message retransmission process proceeds until the acknowledgement is received or the maximum number of retransmissions is attained (Fehri et al., 2018). Most of the time, class A devices are in sleeping mode and hence consume the least power. Class B devices consume more power compared to class A devices, and this is because Class B devices open more receiving window for downlink communication. The class B devices resemble class A devices in network joining procedures, an uplink message transmission since they both use ALOHA-like channel access. Class C devices are the most power-consuming devices since they continuously open the downlink receive window except the time for uplink transmission.

The Adaptive Data Rate (ADR) mechanism is implemented in LoRaWAN to dynamically manage end devices link parameters, and to increase communication reliability in the LoRaWAN network. ADR mechanism allows the end devices to adapt the data rate and the transmission power depending on the changes in the transmission link (Li et al., 2018). When an end device sends a large number of uplink messages back to back without a downlink response from the network server, the end devices assume the network connection loss and resolve the problem by decreasing data rate and increasing transmission power until the connection is regained.

The variation of LoRa physical layer settings such as bandwidth (BW), spreading factor (SF), and coding rate affects LoRa communication performance. BW is the range of frequencies used to transmit the message, and LoRa technology permits the utilization of scalable BW of 125 kHz, 250 kHz, or 500 kHz (Noreen et al., 2017). With the higher bandwidth, you can achieve more data rate, but as a trade-off for receiver sensitivity and communication coverage distance. Another physical layer setting is SF, where LoRa offers the flexibility of utilizing spreading factors at the range of 6 to 12 (Boano et al., 2017) which results in spreading rate ranging from 2^6 to 2^{12} Chips/symbol. The higher the spreading factor, the higher the signal to noise ratio and the radiosensitivity, but this results in more power consumption. To increase receiver sensitivity, LoRa uses Forward Error Correction

(FEC) technique. This technique is used to control errors in a communication channel, whereby end devices send redundant bits, which enables the receiver to detect and correct errors in a transmitted message. The variable number of redundant bits ranges from 1 to 4, whereby when transmitting with the maximum code rate of 4/8 makes the channel more resilient to interference and reduces the message decoding errors. However, sending more redundant bits increases the size of the packet, and therefore more power will be used during transmission. By considering the mentioned LoRa physical layer settings above, more communication range can be obtained by using narrow bandwidth, high SF and high code rate though it comes at the expense of transmission power.

Table 1 is a summary of LoRa physical layer settings and their effect on communication performance

| Physical layer settings | values | Impact on communication performance |
|-------------------------|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Bandwidth | 125 kHz, 250 kHz, or 500 kHz | Transmitting at higher bandwidth will result in shorter transmission time but at the expense of sensitivity and hence, shorter communication coverage range. |
| Spreading factor | 6 to 12 | Signal transmission with the spreading factor of 2^{12} Chips/symbol has the highest signal to noise ratio and hence higher sensitivity and communication range. |
| Code Rate | 4/5, 4/6, 4/7, and 4/8 | Using a maximum code rate of 4/8 increases channel error detection and correction capabilities and hence make the channel more resilience to noise and interference, and therefore more communication coverage range can be achieved |

The theory suggests LoRa can cover the distance of 5 km in urban areas and 15 km in suburban areas (Dambal et al., n.d.) though the study (Jovalekic et al., 2017) indicates that LoRa can achieve a range of up to 316 km. By considering the Friis Transmission Equation:

$$P_r = \frac{P_t G_r G_t \lambda^2}{(4\pi d)^2}$$

Where P_r is the power received, P_t is the transmitted power G_r is the receiving antenna gain, G_t is the transmitting antenna gain, λ denotes the signal wavelength, and d is a distance between the two antennas, the increase in transmitting antenna gain G_t and receiving antenna gain G_r , increases the coverage distance between the two antennas. Although when choosing a transmitting and receiving antennas, it is also important to observe the maximum allowed EIRP (Equivalent Isotropic Radiated Power) in that specific region.

3. MATERIALS AND METHODS

The system conceptual framework shown in figure 1 contains meteorological stations as end devices, the gateway, and the server where users can access meteorological data through a web application.

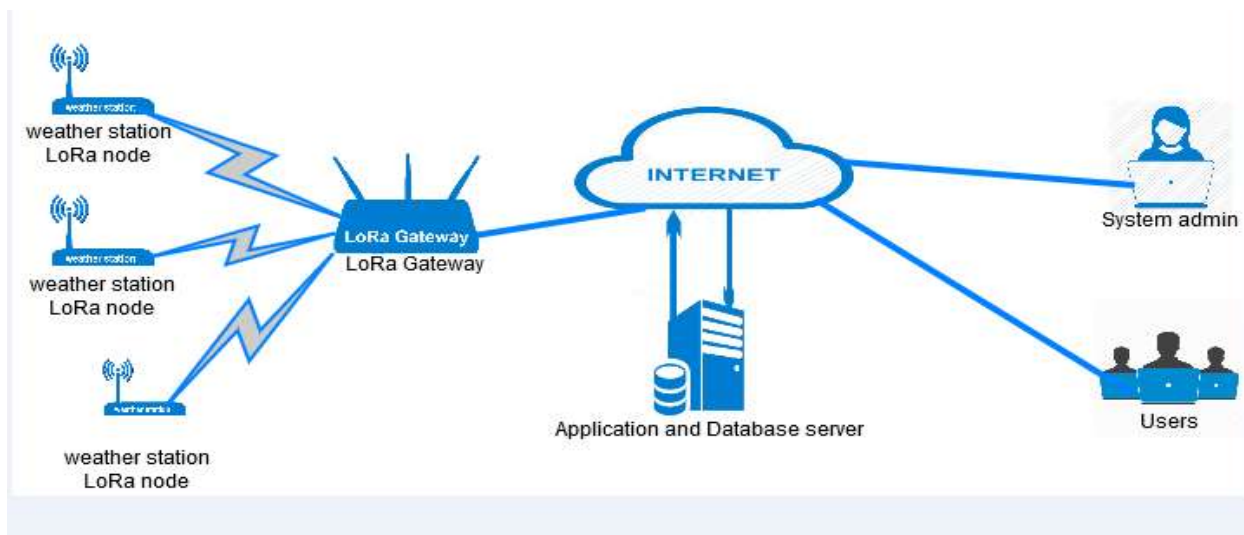


Figure 1: Conceptual framework

3.1. End device components and configuration

Figure 2 shows the designed prototype of the LoRaWAN end device, which consists of sensors such as temperature and humidity sensor, rain gauge, GPS, and LoRa module. The DHT 11 is an affordable digital temperature and humidity sensor, and it is suitable for measuring temperature from 0 °C to 50 °C, and humidity from 20% to 80%, with an operating voltage of 3-5V. Another sensor is a rain gauge tipping bucket with a resolution of 0.2mm and 0.1%, which uses a reed switch to generate digital pulses after every 0.2mm of rain. GPS module is not only used to track the location of the end device (meteorological station) but also to generate timestamps for the measured meteorological data. Lastly, the MKR1300 LoRa module operating at 868 MHz is used to send sensor readings to the gateway. Figure 2 shows the LoRaWAN node.



Figure 2: LoRaWAN end device

The MKRWAN 1300 board is programmed by C++ programming language using Arduino software. It also has analogue and digital pins which provide an interface for sensors connection. After sensors configurations and the MKRWAN 1300 board is also configured to enable transmission of sensor data to the gateway via the LoRaWAN network. For end devices to join the LoRaWAN network, they can either be activated by Activation By Personalization (ABP) or by Over-The-Air Activation (OTAA). These two activation mechanisms ensure the security of the network, though OTAA is more flexible, secured, (Butun & Pereira, 2019), and recommended on higher security demanding applications. In this study, end devices use the OTAA activation method, whereby a network join request contains DevEUI, JoinEUI, and DevNonce. The NwkKey is used to calculate message

integrity check (MIC), and only if the end device is permitted to join the network, the network server will respond with the join accept message, which contains DevAddr, JoinNonce and other parameters (Haxhibeqiri, 2018).

The Data sent from end devices is the station identification number (ID), measured temperature and humidity values, rainfall data, and data timestamp. The station location with coordinates is directly registered to the server to reduce packet size.

3.2. Gateway components and configuration

The gateway consists of RisingHF RHF0M301 gateway, 6 dBi fibreglass antenna, raspberry pi 3 model B, and SD card for operating system (OS) installation. The RisingHF RHF0M301 gateway has a connection interface to connect with raspberry pi. The raspberry pi has an ARM quad-core processor and power specifications of 3.3V, 5V, and 2A, which can either be supplied by grid or battery. The leading role of the gateway is to forward packets received from end devices via the LoRaWAN network to the network server database. To do so, the gateway requires internet access. With raspberry pi 3, the gateway can access the Internet wirelessly or through an Ethernet cable connection.

The gateway can be configured to forward end devices data to online platforms such as The Things Network (TTN), ThingPark, and Lorient. These platforms provide HyperText Transfer Protocol (HTTP) integration, which enables users to forward their sensors data from these platforms to their own servers' database. Even though these platforms are used by many, they have limitations to users with a free account. For instance, TTN allows an average of 30 seconds uplink time per day for every device and a maximum of 10 downlink messages per day per device (*Limitations: Data Rate, Packet Size, 30 Seconds Uplink and 10 Messages Downlink per Day Fair Access Policy - End Devices (Nodes) - The Things Network*, n.d.), while the Lorient platform limits number of user accounts, user applications, and only one gateway and a maximum of ten devices (*LORIENT*, n.d.) are allowed in a free account. The ThingPark platform allows only one gateway and a maximum of five devices (*Home / ThingPark LoRaWAN Developer Portal*, n.d.) in a free account.

To avoid limitations from the mentioned platforms, the LoRa Server Project developed a free and open-source LoRa Gateway Operating System, which provides full capabilities to build up a private network without limitations. The RisingHF RHF0M301 gateway is among LoRa gateways, which support the LoRa Gateway Operating system. The other ones are LORIX One, IMST - iC880A, IMST - iC980A, Pi Supply - LoRa Gateway Hat, RAK - RAK2245, RAK - RAK831 Gateway Developer Kit and Sandbox Electronics - LoRaGo PORT. This operating system comes in two versions, which are the base version and the full version. In this study, the full version was used, and it contains three main components, which are the LoRa Gateway Bridge, LoRa server and LoRa App server.

LoRa Gateway Bridge: service is responsible for communication between the gateway and network server. It also converts LoRa packet-forwarder protocols into a LoRa Server standard protocol. It uses Semtech Packet forwarder to forward the received packets by the concentrator to a server through an IP/UDP link. It also translates UDP protocol from the gateway to JavaScript Object Notation (JSON) message and publishes it to Message Queuing Telemetry Transport (MQTT) implementation at the server (Albert & Haslhofer, 2017).

LoRa Server is an open-source and core application server of the LoRaWAN network. It keeps track of active end devices and the ones that want to join the network (Torres et al., 2018). It also supports both OTAA and ABP activation mechanisms. For the end devices that want to join the network, it will check with the Application server to know if it is allowed to join the network, and if so, it verifies the end device by authenticating it and forwarding it to the application server. It also supports the ADR mechanism, and it also decrypts the received data payload before reaching the MQTT broker (Zinas et al., n.d.). Furthermore, it handles LoRaWAN mac-layer, the uplink packets received by the gateway(s), and scheduling of downlink data transmissions (*LoRa Server - LoRa Server Server Documentation*, n.d.).

LoRa App server: is an open-source of the LoRaWAN application server that acts as a bridge between the IoT cloud and the LoRa Server. It has a web interface that enables end devices inventory by offering nodes and gateway administration by application and organization. It has RESTful and gRPC API for integration with external services; moreover, it provides different ways of sending and receiving device payloads that can either be through MQTT, HTTP or writing directly into an InfluxDB database. The communication with applications is through JSON and Application Programming Interfaces (APIs).

3.3 Web application development

The programming languages such as Hypertext Preprocessor (PHP), Hypertext Markup Language (HTML), Cascade Styling Sheet (CSS), JavaScript, and jQuery were used during the development of web application. In the server part, a free and open-source MySQL database was used; however, the MySQL database has a weakness in handling concurrent database reading and writing

operations but provides real-time and efficient data storage services (Zhou et al., 2019). The CSS programming language provides an excellent web interface and makes it more user-friendly and attractive. Also, CSS was used to build web applications responsive to various devices such as mobile phones, tablets, and computers with different screen resolutions.

The data received from the gateway through Application Programming Interface are in JSON format and base64 encoded. After decoding the data, JavaScript was used to split the sensor values. Before inserting decoded sensor values into the database.

4. RESULTS AND DISCUSSION

4.1. Web result interfaces

The web application provides a user authentication feature, whereby all users have to log in to be able to access meteorological information. When a user accesses the web address, they will be required to enter a username and password. The web application also provides a way to retrieve the user's forgotten password. When a user click forgot password option will be directed to the password retrieval page where will be required to enter his/her username (email address) for his/ her password to be sent to his/her email address.

After a successful login to the system, users can change his/her profile details such as profile picture, phone number, and account password. Users will also be able to access meteorological data in both numerical and graphical formats. Moreover, any user can download recorded meteorological data in CSV file format, by specifying the region and time interval he/she wants. This will give users the ability to use recorded meteorological data for further data analysis. The system administrator is the one registering new users' accounts and has the privileged to delete any unwanted account. The system admin is also responsible for registering a new station and closing the one which is no longer working.

The web application is also integrated with Google API to show the exact location of the station by using Google maps, and lastly, the web application notifies the system admin and users whenever a station has not sent weather data for more than two hours. Figure 3 shows the system admin's interface after login and tabular representation of the recent received meteorological data, while figure 4 shows the graphical representation of temperature, humidity, and rainfall data of the Node1.

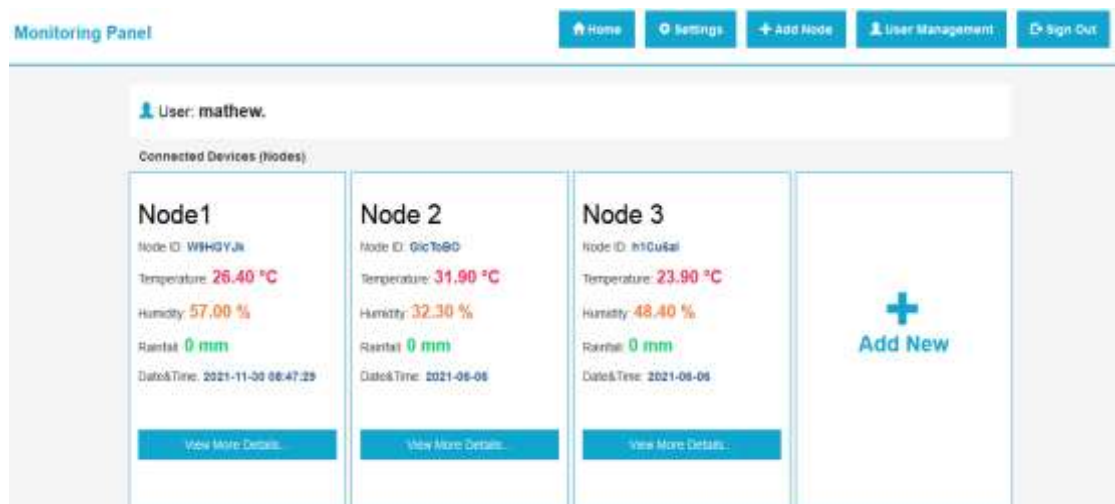


Figure 3: System admin's interface with tabular data visualization

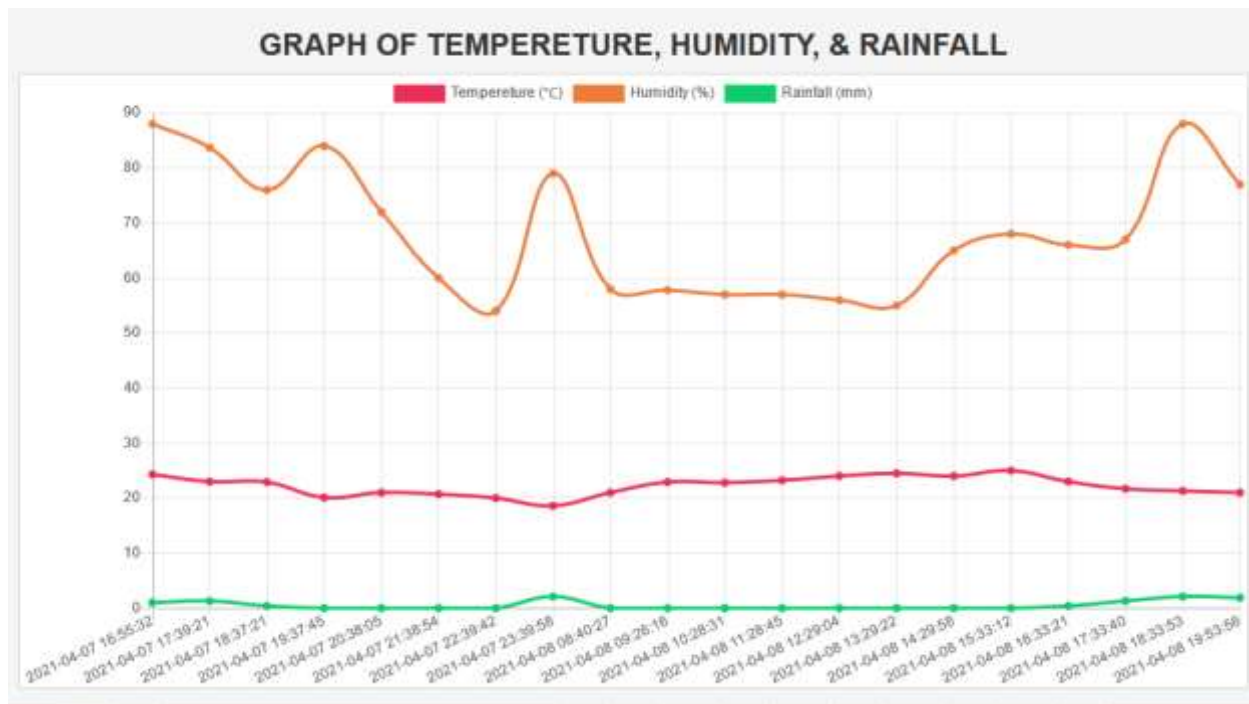


Figure 4: Graphical data visualization

4.2. Communication range test

Among the benefits of the LoRaWAN network is the ability to send data over a long-distance, this serves a purpose for the meteorological stations being able to send data to the gateway even if they are located in a remote area with a limited Internet connection. As the theory suggests LoRaWAN

can cover the distance of 5 km in urban areas and 15 km in suburban areas (Dambal et al., n.d.), the system was tested to ensure if the meteorological stations could communicate with a gateway in a distance of at least 15 km in suburban areas. To ensure there is a line of sight between the meteorological station (Node) and LoRa gateway BotRf simulation software was used to simulate the communication path before the devices were taken to the field

BotRf simulation software was used to ensure there is a line of sight between the gateway and the end device. Even though there are many commercially available software intends to solve the same problem, most of them use digital elevation maps, while some of them are quite expensive, and others can only be used by the radio and antenna of a particular manufacturer (Zennaro et al., n.d.). BotRF is an open-source wireless link planning tool running on Telegram bot technology, and it is supported by any smartphone or computer with Telegram software installed. This software is capable of simulating a terrain profile of the specific area and calculating the path loss in a specified wireless link. For useful wireless link must have at least 60% of the Fresnel zone cleared. Figure 5 shows the simulation results of the transmission link between the two selected points. The brown line represents the apparent earth's bulge, the green line represents the apparent terrain profile, and the magenta line represents 60 % of the first Fresnel zone. The simulation results show there is no obstruction between the two points, and therefore the link can be used for communication between the end device and the gateway.

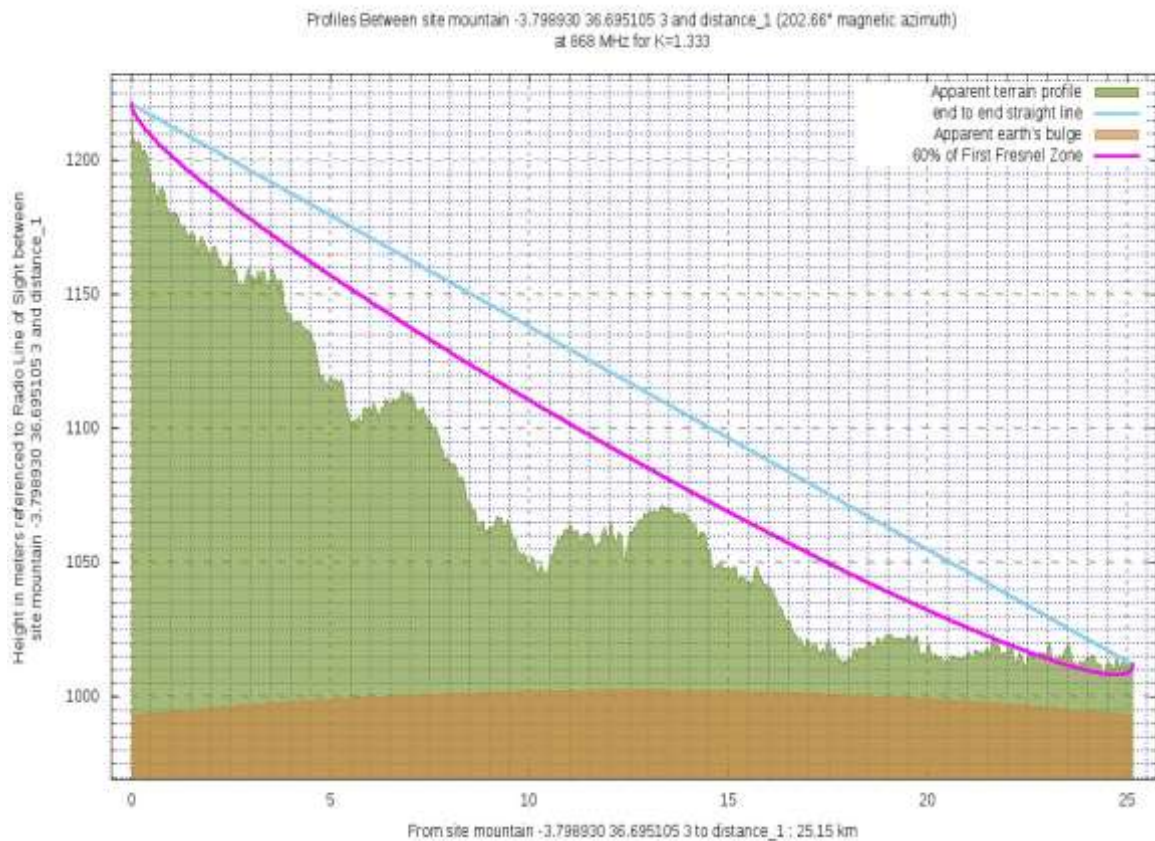


Figure 5: BotRF simulation results

In this test, the LoRaWAN gateway was able to receive meteorological data from a station at a distance of 25.15 km which exceeds the expected distance of 15 km.

5. CONCLUSION

In this study, the development of integrated sensors for agro-meteorological data collection in Tanzania prototype was described. The system records environmental values such as temperature, humidity, and rainfall and uses the LoRaWAN network to transmit recorded data from the end device to the gateway, and the gateway uses the Internet to forward received data to the server database. The developed meteorological station can communicate with the LoRaWAN gateway at a distance of 25.15 km in a suburban area via the LoRaWAN network. Also, users will not only be able to visualize the recorded data in tabular and graphical format, but also they will be able to download the recorded data in CSV format for other uses. Furthermore, the LoRa Gateway Operating system was used to come up with a cost-efficient system that doesn't require any full-access account subscription in online platforms such as TTN, ThingSpeak, Lorient, and ThingPark.

REFERENCES

1. Albert, P., & Haslhofer, F. (2017). *Practical Limitations for Deployment of LoRa Gateways*.
2. Boano, C. A., Cattani, M., & Kay, R. (2017). *Impact of Temperature Variations on the Reliability of LoRa An Experimental Evaluation*.
3. Butun, I., & Pereira, N. (2019). *Security Risk Analysis of LoRaWAN and Future Directions*. 1–22. <https://doi.org/10.3390/fi11010003>
4. Byanyuma, M., Kalolo, S., Mrutu, S. I., Nyakyi, C., & Sam, A. (2013). *Affordable Broadband Connectivity for Rural Areas*. 62–65.
5. Cattani, M., Boano, C. A., & Römer, K. (2017). *An Experimental Evaluation of the Reliability of LoRa Long-Range Low-Power Wireless Communication*. <https://doi.org/10.3390/jsan6020007>
6. Dambal, V. A., Mohadikar, S., Kumbhar, A., & Guvenc, I. (n.d.). Improving LoRa Signal Coverage in Urban and Sub-Urban Environments with UAVs. *2019 International Workshop on Antenna Technology (IWAT)*, 210–213.
7. Fehri, C. El, Kassab, M., Abdellatif, S., Berthou, P., & Belghith, A. (2018). LoRa technology MAC layer operations and Research issues. *Procedia Computer Science*, 130, 1096–1101. <https://doi.org/10.1016/j.procs.2018.04.162>

8. Haxhibeqiri, J. (2018). *A Survey of LoRaWAN for IoT: From Technology to Application*. <https://doi.org/10.3390/s18113995>
9. *Home | ThingPark LoRaWAN developer Portal*. (n.d.). Retrieved October 21, 2019, from <https://partners.thingpark.com/en>
10. Jovalekic, N., Pietrosemoli, E., & Zennaro, M. (2017). *Smart and Very Distant Objects*. <https://doi.org/10.1145/3127502.3127514>
11. Kondela, E., Nungu, A., & Pehrson, B. (2015). *Status of Existing Weather Observation Station Network in Tanzania and the Possibility to Automate and Densify it*.
12. Lavric, A. (2019). *LoRa (Long-Range) High-Density Sensors for Internet of Things*. 2019.
13. Li, S., Raza, U., & Khan, A. (2018). *How Agile is the Adaptive Data Rate Mechanism of LoRaWAN ?*
14. *Limitations: data rate, packet size, 30 seconds uplink and 10 messages downlink per day Fair Access Policy - End Devices (Nodes) - The Things Network*. (n.d.). Retrieved October 14, 2019, from <https://www.thethingsnetwork.org/forum/t/limitations-data-rate-packet-size-30-seconds-uplink-and-10-messages-downlink-per-day-fair-access-policy/1300>
15. *LoRa Server - LoRa Server Server documentation*. (n.d.). Retrieved October 14, 2019, from <https://www.loraserver.io/loraserver/overview/>
16. *LORIoT*. (n.d.). Retrieved October 14, 2019, from <https://www.loriot.io/technology.html>
17. 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. <https://doi.org/10.1016/j.ict.2017.12.005>
18. Noreen, U., AHCENEBOUNCEURUNIV-BRESTFR, E., & Clavier, L. (2017). *A Study of LoRa Low Power and Wide Area Network Technology*.
19. Polonelli, T., Brunelli, D., Marzocchi, A., & Benini, L. (2019). Slotted ALOHA on LoRaWAN-design, analysis, and deployment. *Sensors (Switzerland)*, 19(4). <https://doi.org/10.3390/s19040838>
20. Sornin, N., Cited, R., & Sornin, N. (2016). (12) *United States Patent*. 2(12).
21. Toreti, A., Maiorano, A., De Sanctis, G., Webber, H., Ruane, A. C., Fumagalli, D., Ceglar, A., Niemeyer, S., & Zampieri, M. (2019). Using reanalysis in crop monitoring and forecasting systems. *Agricultural Systems*, 168(September 2017), 144–153. <https://doi.org/10.1016/j.agry.2018.07.001>
22. Torres, F., Soriano, J., & Riva, G. (2018). First Steps in the Development of a LoRaWAN Testbench. *2018 Ninth Argentine Symposium and Conference on Embedded Systems (CASE)*, 7–12.
23. Zennaro, M., Rainone, M., & Pietrosemoli, E. (n.d.). *Radio Link Planning made easy with a Telegram Bot*. 1–10.
24. Zhou, Q., Zheng, K. A. N., Member, S., Hou, L. U., Xing, J., & Xu, R. (2019). Design and Implementation of Open LoRa for IoT. *IEEE Access*, PP, 1. <https://doi.org/10.1109/ACCESS.2019.2930243>
25. Zinas, N., Kokkonis, G., & Valsamidis, S. (n.d.). *Proposed open source architecture for Long Range monitoring . The case study of cattle tracking at Pogoniani*. 1–6.