LoRa Wireless Mesh Network for Lake Biology Study

Trey Lindquist[1], Kolton Chambers[1], Noah Benish[1], Dr. Devin R. Berg[1, *] and Dr. Ahmet Turkmen[1]

[1] Engineering and Technology Department, University of Wisconsin-Stout, Menomonie, WI 54751 USA
*Corresponding Author: bergdev@uwstout.edu

***Abstract*—The LoRa Wireless Mesh Network was developed to collect lake health data and to make the data readily accessible to researchers or other individuals who wish to study or monitor lake health. This project will include buoys which are deployed as nodes at key locations on a lake, such as in bays or other low traffic monitoring points. Each node includes a LoRa Arduino board capable of wireless communication on the 915 MHz LoRa band and four different sensors (temperature, turbidity, DO2, and pH). Each node collects sensor data and passes it from node to node until it reaches the hub node. The hub node will store that data and upload it to an online dashboard, segregated by node location, where it may be presented graphically. The focus of his project is to utilize a mesh network to pass data from node to node without data loss over long distances and without line-of-site. Initially, data communications were established between two nodes. The network was then subsequently expanded through the inclusion of additional data source nodes and finally the hub node. The result is that sensor data can be collected and transmitted from each of the nodes at defined intervals and the network protocol will drop duplicate data transmissions to reduce network congestion. This provides an expandable and dependable node-based mesh network which enables real-time lake health monitoring capability, replacing current labor and time intensive manual measurements. The resulting data set will be valuable to biologists and ecologists conducting limnological research**.

I. INTRODUCTION

The LoRa Wireless Mesh Network for Lake Biology Study project was created to assist researchers with data collecting on lakes or other waterways. Lake data was focus of this project because of a project done focusing on an agricultural application [1]. With the sensors and data that was collected, a project focused on lake biology seemed like an important biome to cover [1][2]. The LoRaWAN board allows for a lower power consumption need and while also allowing the multi-node data hopping to boost the transmission range [3][4][5]. Four different measurements, temperature, turbidity, dissolved oxygen (DO2), and acidity (pH), are collected by sensors connected to a LoRa-capable Arduino boards and transmitted by each other so that the data can be collected, stored, and published in an accessible format. This central data storage will allow the researchers to collect the data continuously without manual retrieval of water samples. This will help provide accurate and more readily available data. This takes into consideration the studies conducted into transmission time delay and data loss [6]. Allowing researchers to collect data on a broader scale is one of the primary motivations for the project.

II. PROCEDURE

A. Selecting the Data Sensors

The first step of the project was selecting the sensors so that the desired data could be collected. Not only would the sensors be needed to collect the correct data, but they also needed to be compatible with the selected LoRa Arduino Board. The sensors selected to provide essential lake health monitoring included:

- Temperature Sensor: Measures the temperature of the water in the lake.

- Turbidity Sensor: Measures the opacity by shining light through the water. As the turbidity of the water increases, a lower percentage of the transmitted lite passes through the water sample. The higher the opacity of the water, the higher the turbidity reading.

- DO2 Sensor: Measures the concentration of oxygen that exists in the water.

- pH Sensor: Measures the pH level of the water.

Each node will consist of a set of these sensors that are then wired to the corresponding LoRa Arduino board. The Arduino board concatenates the sensor readings into a data array and transmits the array as a data packet.

B. Transmission and Receiving Code Construction

The transmission and receiving code were written in separate files in the Arduino IDE[a]. The backbone for both files were sourced from the transmitting and receiving open-source code from the supplier and from a previous project that was started at the University of Wisconsin-Stout [7].

The data from is read from the sensors using supplier provided code or libraries suitable to each sensor. The data is concatenated into an array and the temperature, turbidity, DO2 and pH values form a segment of each data array. Additionally, the node identifying number and the timestamp of the data and appended to the beginning and end of the data array, respectively, as shown in Fig. 1. The array is then transmitted on the 915 MHz LoRa frequency to be received by either the next node or the hub node. To ensure that the data will be passed along even if the node is outside of the transmitter range from the hub node, each node is also capable of receiving data packets. This means that the nodes will not only transmit their own data, but also the data of the other nodes. The number of nodes and the frequency in which the data is collected can be easily manipulated to the desired values in these files.

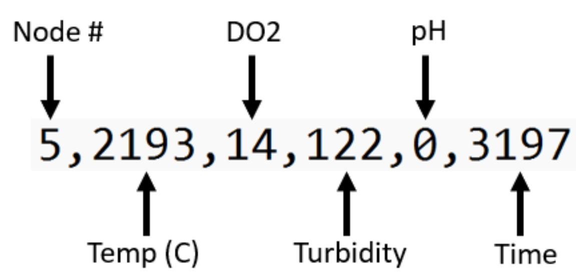

Figure 1. Transmission data array showing the arrangement of data provided by each node transmission.

The hub node is the one node in the network that does not collect data, but only serves to receive data packets from the other nodes, whether received directly or via retransmission from a nearby node. The hub node must also have internet connectivity to allow it to sync the received data to a central, online storage location for further processing.

C. Implementing the Mesh Network

In implementing the mesh network, the sensor nodes will be able to collect data, transmit their collected data, and retransmit data collected at other nodes. To guarantee that all of the data packets make are received at the hub node, deduplication of the data packets is performed at each node only after a specified number of retransmits. This way the network is more robust and packet loss due to temporary interference will be less likely. Deduplication is also performed at the hub node before the data packet is recorded to long-term storage.

D. Data Processing

Each individual sensor requires some on-board processing to render the data in a human-readable format. The resulting sensor readings are written to the data packet described in Fig. 1. Currently, data packets received at the hub node is appended to a long-term storage file using Tera Term. This file is processed using kernel files that are located on GitHub. The data from the log file is then transformed into four different graphs, with each graph displaying the data from each sensor. The graphs each have the information from the specific data type on it. For example, the temperature graph displays the temperature gathered from each node, so each graph will have lines that correspond to the number of nodes, in this case three. The resulting data is pushed to an online repository hosted on GitHub to provide a central point for researchers to readily access the information that they need. GitHub Actions is used to automatically render up-to-date graphs on a regular schedule. Historical data is also accessible via the repository's commit history.

III. Testing and Results

With a project mainly revolving around software construction and implementation, rigorous testing was needed to get the desired results. Testing included: transmission range, node longevity, and the data packet handling capability of the network. Each portion of the testing provided valuable information about how the project functions, the current restraints, and also what can be improved upon in the future.

A. Transmission Range

From the datasheet of the LoRaWAN board, the given transmission range was 2 km. This takes into consideration no interference from outside sources including buildings, cars,

[a] All code associated with this project is available at https://doi.org/10.17605/OSF.IO/89CD3

and trees. In Fig. 2, when the range was tested, it was found that the longest transmission range with some interference under normal operating conditions was only 0.8 km. This range needs to be considered with planning for node placement. This means that nodes are going to need to be closer together for the data to be passed along from node to node and to the hub. This transmission range was found to be consistent regardless of the pairing of individual nodes tested. Depending on the location of the hub node, data packets may traverse multiple hops on the network before reaching the hub and being recorded to long-term storage. Note that this testing was performed in an on-land environment with local sources of interference such as buildings and trees. If the nodes are deployed on the water, interference sources are likely to have a lower impact on the transmission range between adjacent nodes.

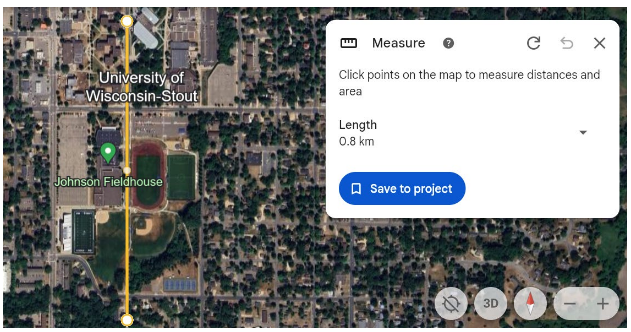

Figure 2. Image from Google Earth showing the placement of the hub node and the nearest node placement to demonstrate the working transmission range.

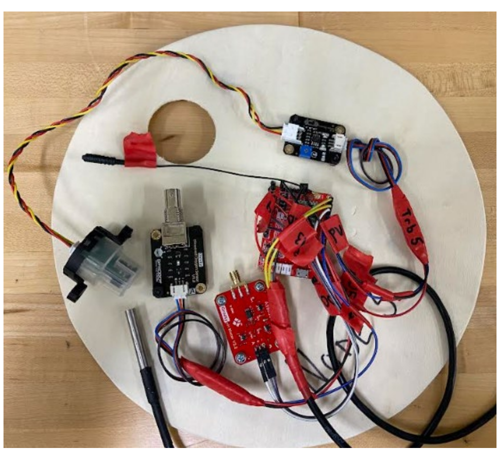

Figure 3. Photo of the various sensors used at each node wired to the LoRa Arduino module.

B. Longevity Testing

As one of the goals of this project was for the sensors to run for an extended period, a crucial test for this project was operating it under a long period of time to see how the system handled it. When these tests were conducted, the prototype system held up well, but it did run into problems after 9-hours due to how the time measurement was configured. This issue could likely be resolved through the addition of a GPS module to provide accurate time stamping for each data packet. Fig. 4 and Fig. 5 provide an example of our graphs after some long-term tests. Note that the pH for Node 5 is initially negative due to a temporary sensor malfunction.

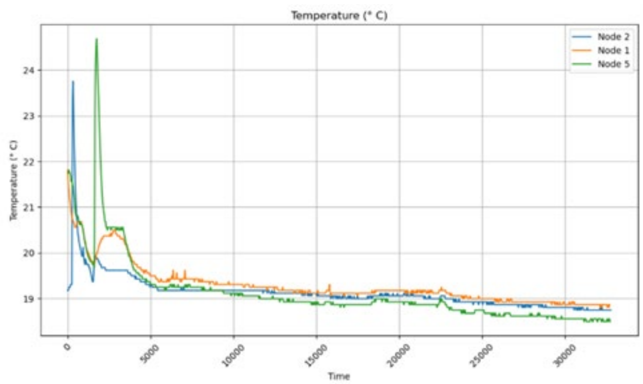

Figure 4. Plot of ambient temperature data collected during testing of Nodes 1, 2, and 5.

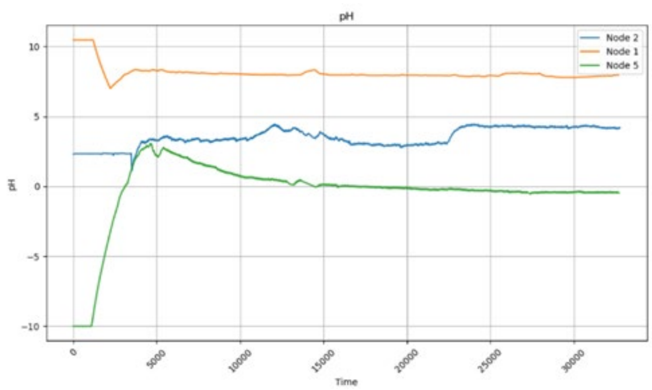

Figure 5. Plot of pH data collected during testing of Nodes 1, 2, and 5.

C. Data packet handling

There was a significant amount of software testing that was conducted during this project. The first testing portion was checking to see if two nodes were able to talk to each other. The testing was successful as the hub node, which was running the receiving code, and the sensor node, which was running the transmitting code, were able to transmit the data. The next testing that was completed was implementing the mesh network. To do this, three sensor nodes each ran the transmitting code, and the hub node ran the receiving code. The results showed a transmission from each of the three nodes but not any duplicated data. Based on how it was set up, data packet duplicates were expected due to each of the nodes being within transmission range of each other. This issue was quickly fixed and then each of the nodes were not only transmitting their own data, but also the data of the other nodes. The last main software testing that was conducted was checking the graphs. This was done to ensure that the data that was being graphed was from the correct sensor and that it was producing the correct results. In Fig. 4 and Fig. 5, the data from the temperature and pH sensors are shown with each node present of the graphs.

D. GPS Module

The main issue that was encountered was incorporating a GPS module. There were two focuses of the GPS module. The first would pass along the latitude and longitude coordinates of the nodes. With that information, biologists would be able to track the location of the nodes and to keep track of drifting and mobility of the nodes. The other piece of data was a timestamp. Sending a timestamp with each string would improve two aspects of the project. The first would ensure the data reliability and transmission delay that occurs. The second is the readability and long-term function of the graphs. The timestamp from the data would replace the x-axis data on the graphs. This would allow the graphs to improve their accuracy and readability. Without the timestamp, a current constraint is that the time on the x-axis of the graphs is the time from the start of the program running (in seconds). This only correctly works for just over nine hours before the time reaches the max bit size.

The GPS module that was tried in this project was the NEO-6M GPS Module (GT-U7, 161296561) for Arduino GPS. When using this GPS, a timestamp was not appearing. When testing that GPS module on an Arduino Uno board, the data from the GPS was working. When comparing this information, the prevailing theory is that the processor chip on the LoRaWAN board used in this project is not compatible with the GPS module.

IV. CONCLUSION

The LoRa Wireless Mesh Network for Lake Biology Study was successful, but some future work remains. The data is successfully stored, and researchers are able to quickly monitor and assess the real-time health of the lake. The goal of the project was completed but, there are some imperfections that can be improved upon. One of the factors that played a role was the time constraints. Further, one enhancement that would improve the project is the addition of a GPS module, which would provide more precise time stamps for each data packet. Additionally, precise coordinates for each node would enable researchers to receive notification of drift in the node's location or if the node would break free from its mooring. In conclusion, the goal of collecting data and storing the data in a centralized location that is accessible to researchers was accomplished, but there can be improvement that will enhance the project and make it more efficient and valuable.

ACKNOWLEDGEMENT

The authors acknowledge the financial support provided by the Stout University Foundation through the G.A. Taft Professorship.

REFERENCES

[1] W. Chanwattanapong, S. Hongdumnuen, B. Kumkhet, S. Junon and P. Sangmahamad, "LoRa Network Based Multi-Wireless Sensor Nodes and LoRa Gateway for Agriculture Application," 2021 Research, Invention, and Innovation Congress: Innovation Electricals and Electronics (RI2C), Bangkok, Thailand, 2021, pp. 133-136, doi: 10.1109/RI2C51727.2021.9559804.

[2] Loraute: Routing Messages in backhaul lora networks for underserved regions | IEEE Journals & Magazine | IEEE Xplore. (n.d.). https://ieeexplore.ieee.org/document/10142009/

[3] R. M. Liaqat, P. Branch and J. But, "A Novel Approach to Collision Avoidance in LoRa Networks," 2023 Fourteenth International Conference on Ubiquitous and Future Networks (ICUFN), Paris, France, 2023, pp. 412-417, doi: 10.1109/ICUFN57995.2023.10200139.

[4] F. Basili, S. Parrino, G. Peruzzi and A. Pozzebon, "IoT Multi-Hop Facilities via LoRa Modulation and LoRa WanProtocol within Thin Linear Networks," 2021 IEEE Sensors Applications Symposium (SAS), Sundsvall, Sweden, 2021, pp. 1-6, doi: 10.1109/SAS51076.2021.9530117.

[5] Y. Yi, H. Zhao and Y. Wang, "LoRa Signal Monitoring System of Multi-Node Software Define Radio," 2020 IEEE Wireless Communications and Networking Conference Workshops (WCNCW), Seoul, Korea (South), 2020, pp. 1-5, doi: 10.1109/WCNCW48565.2020.9124898.

[6] C. -W. Liang, Y. -L. Wu, C. -Y. Shi, S. -M. Lu and H. -C. Lee, "Poster Abstract: Evaluation of a LoRa Mesh Wireless Networking System Supporting Time-Critical Transmission and Data Lost Recovery," 2019 18th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), Montreal, QC, Canada, 2019, pp. 317-318, doi: 10.1145/3302506.3312607.

[7] Chundu, S., Lundquist, C., and Berg, D.R., 2023. MenomiNet: A prototype network for real-time public lake data. In National Conference on Undergraduate Research, Eau Claire, WI. doi: 10.5281/zenodo.6977169