T24: Dynamic Mesh Wireless Sensor Collection Network

1st Woon Jun Wei
Computing Science Joint Degree
Programme, Singapore Institute of Technology - University of Glasgow,
2837269W@student.gla.ac.uk

2nd Low Hong Sheng Jovian
Computing Science Joint Degree
Programme, Singapore Institute of Technology - University of Glasgow,
2837402L@student.gla.ac.uk

3rd Benjamin Loh Choon How
Computing Science Joint Degree
Programme, Singapore Institute of Technology - University of Glasgow,
2837341L@student.gla.ac.uk

4th Wang Rongqi Richie
Computing Science Joint Degree
Programme, Singapore Institute of Technology - University of Glasgow,
2837357W@student.gla.ac.uk

5th Poon Xiang Yuan
Computing Science Joint Degree
Programme, Singapore Institute of Technology - University of Glasgow,
2837264P@student.gla.ac.uk

***Abstract*—Wireless Sensor Networks (WSNs) are pivotal for real-time environmental monitoring applications. This paper delves into the detailed architecture and performance analysis of two prominent protocols, ESP-NOW and LoRa, within the context of a WSN setup. The ESP-NOW protocol, utilizing a Dynamic Source Routing (DSR) mesh algorithm, excels in throughput, latency, and bandwidth compared to LoRa, making it suitable for high-speed, low-latency communication needs. Conversely, LoRa demonstrates advantages in long-range communication and obstacle penetration.**

Additionally, the analysis delves into power consumption, where ESP-NOW outshines LoRa in both idle and active network states. This energy efficiency of ESP-NOW is particularly advantageous for battery-operated devices and remote deployments.

The findings presented herein provide valuable insights for optimizing protocol selection and power management strategies in WSN deployments, ensuring efficient and reliable data transmission in diverse environmental monitoring scenarios.

***Index Terms*—WSN, LoRa, ESP-NOW, Mesh, ELK, MQTT**

I. Introduction:

The absence of a dedicated Wireless Sensor Network (WSN) for monitoring harmful gas emissions, including (CO_2), and their relation to temperature fluctuations in diverse urban environments is a critical gap that has been recognized in various contexts, not specifically limited to Singapore. However, studies like "UrbanSense" [1] and "CitySee" [2] have demonstrated the urgent need for and the implementation of WSN infrastructures in urban areas to monitor environmental conditions, including (CO_2) and temperature fluctuations, which are crucial for formulating informed policies towards (CO_2) reduction and urban heat island mitigation.

II. Problem Statement

The lack of comprehensive environmental monitoring poses a significant challenge within Singapore. Currently, there is a dearth of real-time data on CO_2 levels and temperature, hindering informed decision-making and sustainable practices in the country. This data gap inhibits the establishment of correlations between environmental factors and their impact on campus well-being and sustainability. Consequently, there is a critical need to develop and deploy a scalable and energy-efficient mesh network utilizing efficient communication protocols to address this challenge.

A. Objectives

This project aims to develop a robust multi-protocol mesh system for IoT applications, with a focus on real-time environmental data monitoring. The objectives are as follows:

1) Design and implement a flexible IoT architecture capable of seamlessly integrating multiple protocols, including LoRa, ESP-Now, and traditional mesh algorithms.
2) Develop adaptive routing algorithms to dynamically switch between protocols based on environmental conditions, network congestion, and distance variations, ensuring continuous data transmission and network reliability.
3) Evaluate the performance, scalability, and reliability of each protocol in real-world scenarios, considering factors such as data latency, packet loss, and network reach.
4) Implement a centralized dashboard for visualizing and analyzing environmental data collected from sensor nodes, providing stakeholders with actionable insights for informed decision-making.
5) Integrate mechanisms for automatic protocol fallback and toggling in response to network failures, distance changes, or protocol-specific limitations, ensuring uninterrupted data flow and system resilience.

6) Conduct extensive testing and validation of the multi-protocol mesh system in a simulated and real-world environment, focusing on its ability to adapt to dynamic network conditions and maintain data integrity.

By achieving these objectives, the project aims to establish a scalable and reliable IoT infrastructure capable of supporting diverse applications while promoting data-driven decision-making and sustainable management practices in alignment with Singapore's environmental goals.

III. Literature Review

A. Routing Protocol for Low Power Lossy Networks (RPL)

Designing a robust and efficient mesh network for real-time environmental monitoring using Wireless Sensor Networks (WSNs) faces challenges due to resource constraints. RPL emerges as a promising solution [3]–[5], offering dynamic routing, energy efficiency [6], and scalability. D-RPL enhances multihop routing [7]. Security concerns [8]–[10] and protocol complexity [7] require attention. Optimizing RPL's features can facilitate timely data collection in dynamic environments, contributing to successful monitoring applications.

B. Ad hoc On-Demand Distance Vector routing protocol (AODV)

Various solutions, such as integrating Long-Range (LoRa) networks with mesh architecture [11], [12], have emerged to address the data gap in environmental monitoring. This approach enhances LoRa networks by enabling the creation of mesh structures for routing data packets through multiple intermediate nodes [13]. Among routing algorithms for mesh LoRa networks, the Ad-hoc On-demand Distance Vector (AODV) protocol emerges as promising [14]. AODV's on-demand mechanism optimizes bandwidth by establishing routes when needed, minimizing resource wastage, and its adaptive nature ensures robustness and reliability [15]. However, AODV's overhead and potential congestion are challenges [16]. Nonetheless, it shows potential for monitoring CO2 emissions and temperature fluctuations in urban areas [11], [12].

C. Hybrid Wireless Mesh Protocol (HWMP)

In the quest to enhance environmental monitoring within Singapore, the exploration of the Hybrid Wireless Mesh Protocol (HWMP) stands out for its critical relevance [17]. HWMP, celebrated for its flexibility [18], ability to scale [19], and adherence to international standards, promises easy integration with a wide array of sensor technologies. This exploration, while insightful, indicates that the protocol's potential has not been fully leveraged, particularly in comparison to other networking alternatives that might offer greater operational efficiency and minimized system overhead [20]. Recognizing the gaps in this review is essential for the development of a robust, scalable, and energy-efficient network architecture. Such a network is indispensable for Singapore, aiming to meticulously track environmental parameters and support the nation's ambitious sustainability objectives. By broadening the analysis to include a variety of protocols, we can identify optimal solutions that enhance environmental monitoring capabilities, facilitating more informed and effective policy decisions for the environmental stewardship of Singapore and beyond.

D. Dynamic Source Routing (DSR)

The Dynamic Source Routing (DSR) protocol is a reactive algorithm for projects not requiring real-time data, as it minimises unnecessary transmissions at the cost of initial latency, beneficial for environmental data analysis [21]. While the scalability issues in large networks due to route discovery flooding which may potentially increases the workload and energy consumption of the nodes[21] can be challenging, the adaptability to netowrk changes and energy efficiency of DSR protocol are remarkable, with the ability to quickly find new routes and conserve energy through dormant nodes as well as route caching [22], [23]. This caching also enhances network reliability, allowing quick switches to alternative paths if one route fails, ensuring consistent data collection [23].

E. Optimised Link State Routing (OLSR)

In networks using OLSR, nodes select one-hop reachable neighbors to form Multi-Point Relay (MPR) sets, optimizing packet transmission. OLSR's scalability benefits island-wide CO2 monitoring deployments by efficiently managing routing tables and reliably forwarding packets[24] [25] [26]. Proactive routing minimizes throughput, reducing energy consumption for non-forwarding nodes[27]. OLSR adapts to node failure, ensuring message forwarding reliability, ideal for large urban deployments like Singapore. However, continuous routing table updates increase power consumption, requiring efficient energy management[28]. While lacking inherent security, encryption and access control mitigate unauthorized access risks. Overall, OLSR offers scalability, reliability, and low latency.

F. Summary

In Wireless Sensor Networks (WSNs) tailored for environmental monitoring, the selection of the Dynamic Source Routing (DSR) protocol as the primary routing method for both ESP-NOW and LoRa technologies will significantly boost the network's adaptability and operational efficiency [29]. The choice of DSR, out of the five algorithms considered, stands as the optimal choice for our project due to its flexible and on-demand routing capabilities, which are particularly beneficial in addressing the dynamic conditions typical of environmental monitoring scenarios [21]. Leveraging DSR for our network's routing strategy enhances resource efficiency and ensures reliable data transfer across both short-range (ESP-NOW) and long-range (LoRa) communications. This approach not only streamlines the network setup by consolidating the routing protocol across different communication technologies but also achieves superior scalability, dependability, and minimal latency. These attributes are vital for the effective functioning of large-scale environmental monitoring systems, making DSR the most fitting routing protocol for our project's specific needs.

IV. Design

A. Architecture

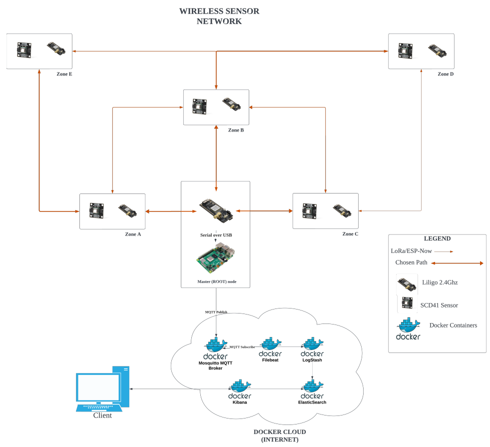

Fig. 1. Proposed System Architecture

This system architecture is meticulously crafted to efficiently collect and analyze data within a Wireless Sensor Network (WSN) mesh. The architecture encompasses two principal components: the WSN mesh and the Data Collection Sink.

The WSN mesh constitutes sensor nodes equipped with Liligo T3S3 LoRa 2.4Ghz modules, facilitating inter-node communication via ESP-Now and LoRa protocols. This setup ensures optimal node discovery and routing, essential for seamless data transmission within the mesh.

The Data Collection Sink, a pivotal aspect of the architecture, comprises several interconnected components. At its core lies the Root Node, functioning as the master node within the WSN mesh. This Root Node establishes a vital link with the MQTT broker using a Raspberry Pi 4b, serving as the conduit for data transmission from the mesh to the subsequent processing stages.

Facilitating communication between the Root Node and data processing components is the MQTT Broker, realized through the Mosquitto software encapsulated within a Docker container. This broker plays a pivotal role in ensuring seamless data flow between the WSN mesh and downstream processing units.

The backbone of data processing and analysis within this architecture is the ELK Stack, comprising Filebeat, Logstash, and Elasticsearch. Filebeat, contained within a Docker environment, serves to gather data from the MQTT broker. Subsequently, Logstash, also deployed as a Docker container, undertakes the parsing and transformation of collected data before forwarding it to Elasticsearch. The latter, functioning as a robust search and analytics engine within its Docker container, serves as the repository for processed data.

The data flow within this architecture is orchestrated as follows: sensor nodes within the WSN mesh capture environmental data, which is then routed through ESP-Now and LoRa protocols towards the Root Node. From here, the data is relayed to the MQTT broker via a Raspberry Pi 4b. Filebeat, operating within a Docker container, subscribes to the MQTT broker to receive incoming data, which is subsequently processed and transformed by Logstash before being stored in Elasticsearch for visualization and analysis through Kibana.

Key points of this architecture include the utilization of ESP-Now and LoRa protocols for efficient WSN mesh communication, the pivotal role of the MQTT broker in facilitating communication between the WSN mesh and data processing components, the comprehensive data processing and analysis capabilities offered by the ELK Stack, and the efficient isolation and management of software components through Docker containers.

This architecture presents a streamlined and efficient approach for collecting and analyzing data within a WSN mesh, rendering it suitable for a myriad of applications requiring real-time data insights.

B. Prototyping

For prototyping, our strategy involves deploying sensor nodes equipped with the Liligo T3S3 Lora 2.4Ghz module. These nodes will utilize ESP-Now and LoRa Protocols for seamless communication. One node will be connected to the SCD41 Sensor, while others will generate synthetic data. The root node will establish a connection to the sink via MQTT, ensuring smooth data collection, analysis, and visualization. As an enhancement, we aim to implement protocol switching to adapt to fluctuating network conditions.

We intend to develop a modified algorithm (**DSR**) for each protocol. Each protocol mesh will include a common master node, configured with Message Queueing Technology (MQTT). This master node will interface with the backbone ELK Stack through Mosquitto MQTT Broker and filebeat with Elasticsearch for data aggregation. Kibana will be used for visualization.

1) ESP-Now Prototype: In crafting a sophisticated network for environmental monitoring, our collective effort harnessed the combined strengths of the ESP-NOW protocol and the Dynamic Source Routing (DSR) algorithm. This integrated approach was strategically chosen to navigate the intricate challenges associated with collecting and analyzing environmental data in real time. Engineered to deliver exceptional efficiency, flexibility, and scalability, the network stands poised to fulfill the stringent demands of a wide array of environmental monitoring projects.

The initial phase involved setting up ESP32 modules to operate with ESP-NOW, a protocol distinguished by its minimal power usage and effective direct communication features. This choice was motivated by the need for a system capable of operating in energy-constrained environments while maintaining reliable data transmission over short distances. The integration of environmental sensors, such as those for measuring temperature, humidity, and air quality, was a critical step in enabling the network to collect a wide range of data, vital for comprehensive environmental analysis.

To enhance the network's adaptability and efficiency in data routing, we will develop a custom implementation of the Dynamic Source Routing (DSR) algorithm across the devices. DSR's flexible, on-demand routing mechanism will allow the network to dynamically adjust to changes and maintain data flow when required, even in the face of node failures or alterations in network topology. This will ensure uninterrupted collection and transmission of environmental data, a key requirement for real-time monitoring applications.

In our planned setup using ESP-NOW, each peer node will regularly collect data from its SCD41 sensor. It will then consult its peer list, which will serve as a routing table, containing only the MAC addresses of the root node and those peer nodes that are either directly linked to the root node or have knowledge of the route to it. The node will then send the data from the SCD41 sensor to an appropriate node available in its peer list.

Should the peer list be empty, indicating there are no known routes, the node will initiate a route discovery process by sending out a broadcast request to surrounding nodes. When a node receives such a request, it will respond if it knows the way to the root node. The node that made the request will update its peer list with this new route information from any responder. This route discovery will continue until all neighbouring nodes have been discovered. We will also implement a hop count list to refine routing efficiency, allowing the peer node to choose the route with the shortest path to the root node to ensure the most efficient data transmission within our routing framework.

Another pivotal component of our system will be the central ESP32 module, designated as the master node. This module will serve a dual purpose: it will act as a receiver for data collected via ESP-NOW from the sensor-equipped nodes and as a transmitter, forwarding this data to a cloud service or server via WiFi using the MQTT protocol. This dual functionality will enable seamless integration of the sensor network with internet-based data processing and analysis tools, extending the capabilities of our monitoring system beyond local data collection.

2) LoRa Prototype: In our planned deployment of a LoRa mesh network, distinct from our ESP-NOW prototype, we aim to construct a network architecture that features a central node equipped with internet capabilities and several leaf nodes to establish the mesh. Every node within this architecture will be assigned a distinct identifier, possibly through methods like MAC Address, to assure uniqueness throughout the network. The central node will facilitate network coordination and data transmission using the MQTT protocol, while the leaf nodes will focus on tasks such as sensing and data gathering.

Upon starting up the master nodes and leaf nodes for setup, the master node will start listening for a "Discovery message" (DM). This DM will be structured to contain a message type. The leaf nodes broacast its DM to its neighbouring nodes. When a node receives a DM, if its route table is not empty, it sends a reply with its MAC address and level. When a leaf node that sends out a DM receives a "Discovery Reply Message" (DRM), it save only the route with the lowest level.

After saving the the address and level of the peer node RDM a leaf node has received, the node raises its node level by one, indicating that it is one hop further from the master node or from a known route to the master node. The node then broadcast its DM to its surrounding nodes. Nodes will ignore messages with a level higher than itself. This process will continue until the timer ends and all niehgbouring peer routes have been stored in each node's "routing table" (RT), lining up multiple potential routes.

For data transmission, the leaf node will pick the first route saved in its routing table and now send the message packet to that neighbouring peer. The message sent will be successful if it receives a "data reply message" (DRM) and the message passing continue down each level until the master node. The node will try resending the message packet via a different route in its RT, making use of various DM to find another viable path.

In our LoRa32 DSR mesh network, the master node acts as the central hub, collecting data from leaf nodes distributed throughout the network. It then efficiently transmits this data to our server for visualization and analysis. This pivotal role enables real-time monitoring and informed decision-making based on the insights gained from the data collected by our LoRa mesh network.

To optimise our LoRa mesh to better suit our project needs, we propose several strategies aimed at enhancing energy efficiency, reliability, and fault tolerance. By developing tailored routing algorithms and leveraging multi-path routing techniques, we aim to achieve these objectives while maintaining the scalability and adaptability inherent in LoRa networks.

Energy-Effecient routing can be achieved by saving and prioritising the use of shorter hops to the master node only swapping to other routes in the case of neighbouring node failure. By selecting paths with smaller hop counts, we can reduce transmission power and decrease energy consumption during data forwarding.

Multi-path routing ensures reliability and fault tolerance in our LoRa mesh network. to achieve this, multiple routes can be stored in the routing table aside from the shortest route hop discovered during route discovery phase.

3) Elasticsearch and Kibana Integration: In the proposed network architecture, each protocol node will incorporate a central master node equipped with a Liligo MCU, seamlessly connected to a Raspberry Pi 4 via USB. A dedicated script running on the Raspberry Pi will efficiently manage the serialization and transmission of serial data to the server using MQTT (Message Queueing Telemetry Transport), ensuring seamless integration with the ELK Stack infrastructure. The combination of Mosquitto MQTT Broker and Filebeat will facilitate streamlined data aggregation and storage management, while Kibana will serve as the intuitive visualization interface.

Data received from the LoRa/ESP-Now protocols will undergo serialization into JSON format before being dispatched to a designated topic on the MQTT Broker through the Raspberry Pi.

The deployment of the ELK Stack and MQTT will be orchestrated using Docker Compose, selected for its straightforward setup and deployment processes. Filebeat will operate as an MQTT Client, subscribing to a predefined topic to capture data packets transmitted from the master node. Subsequently, it will convert this data into JSON format and transmit it to Elasticsearch for storage and indexing. Kibana will then retrieve the data from the database, presenting it on the dashboard for clear visualization. Networking components will be seamlessly integrated within the Docker Compose environment.

In an optimal scenario, the master node will supervise a limited number of interconnected devices, typically ranging from 10 to 15 nodes, organized in either a tree or mesh structure. It will efficiently aggregate, organize, and transmit received data to the MQTT Broker, where Filebeat will undertake further processing. As a proof of concept, the mesh Wireless Sensor Network (WSN) will initially consist of a small number of nodes and one master, with data streams flowing from the master to the Mosquitto Broker for centralized collection.

C. Testing

In our upcoming system evaluation, we will start with unit testing on sensor nodes to confirm their communication and data generation capabilities by testing protocols between nodes. This will be followed by integration testing to assess the Wireless Sensor Network (WSN) Mesh's interoperability and its ELK stack integration, confirming efficient data transmission to the sink node. To further assess the system's adaptability and endurance, we will undertake stress testing under conditions of intense load. The outcomes of these tests are expected to demonstrate successful intra-mesh communication, reliable data conveyance to the sink, and overall system integrity, confirming our system's readiness for deployment.

Building on this foundation, our forthcoming evaluations will concentrate on meticulously analyzing the system's throughput and power consumption across both peak and normal operational phases. For measuring throughput, we will conduct a series of structured tests, using the number of packets received or transmitted over a set duration, to evaluate the data transmission capabilities of the WSN Mesh protocols across different scenarios, from optimal to high-traffic conditions. This entails systematically increasing the data load to gauge the system's capacity and pinpoint any potential bottlenecks.

Simultaneously, our approach to assessing power consumption involves real-time monitoring of the sensor nodes' energy usage across different operational states. Utilizing the S1 Energy Socket, we aim to capture and compare the power demands during idle, normal, and peak activity periods. This detailed power usage profiling over time is designed to shed light on the system's energy efficiency and spotlight areas for potential optimization.

This integrated testing framework, leveraging both simulation tools and real-world environments, is devised to offer a holistic evaluation of the system's performance and operational efficiency. By rigorously verifying that the system meets our defined benchmarks for throughput capacity and energy consumption, we ensure its suitability for widespread deployment across varied contexts, underscoring our commitment to delivering a robust and efficient solution.

V. Implementation:

A. ESP-Now Mesh

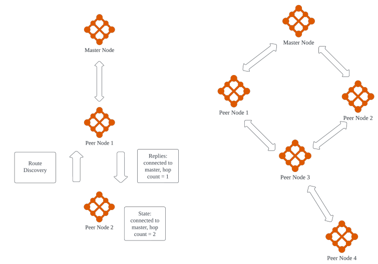

Fig. 2. ESP-NOW Nodes joining network & data transfer

The ESP-Now Mesh implementation adopts a tailored version of the Dynamic Source Routing (DSR) mesh algorithm, designed specifically for environmental monitoring applications. This algorithm optimizes route building and peer node deployment for real-world scenarios. The mesh algorithm assumes the best case approach when it comes to building routes along with the careful considerations of deployment of peer nodes in real life. Within this network, it assumes that there is an initial precense of a Master node and each peer node keeps a record of nearby nodes capable of reaching the master node, leveraging the peer list feature of the ESP-NOW library. When a node needs to send data collected from the SCD41 sensor but lacks the route information to the master, it initiates the route discovery process by sending out a handshake packet to its immediate neighbors. Upon receiving this packet, if a neighbor knows the path to the master, it responds by indicating its route knowledge and hop count. The originating node then updates its routing information, registers the responding peer's MAC address into its list, and proceeds to transmit the data either directly to the master node or through peers that have a known route to the master.

The deployment strategy for this mesh network emphasizes optimal node positioning to prevent data redundancy and congestion. Nodes needs to be strategically placed to avoid collecting redundant data and to minimize the inclusion of redundant peers in their peer lists. This approach aims to ensure that each area of deployment ideally contains only 1-2 nodes. Nodes are configured to include only peers that are closer to the master in their peer lists, as the route building in this algorithm progresses upwards towards the master node. This strategy helps reduce the number of redundant and repeated packets transmitted in the mesh network, thereby optimizing network performance and efficiency.

B. LoRa Mesh

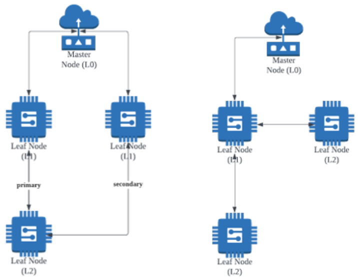

Fig. 3. LoRa Nodes joining network & data transfer

The LoRa Mesh is implemented with a modified DSR with hierarchy architecture design. Nodes are ranked by their hop count towards the master node, whereby nodes further away from the master node holds a higher rank value. This architecture design allows the nodes to store only the addresses of the nodes of lower rank value, instead of the traditional way whereby nodes store the routes to destination. The flooding effect during route discovery can be mitigated as the discovery messages will only be propagated once per discovery. In order to build a mesh network, the route table, discovery and reply messages are needed. The discovery message is the simplest message format, which contains only the message type. Nodes will regularly check its route table, if it is empty, the discovery message will be sent. Nodes that receive the discovery message will check whether its own route table is empty, if it is not, it will reply the discovery message with its own MAC address and rank. Other nodes that receive the dicovery reply message, regardless if it has started a discovery, will update its route table if applicable. As such, the route table will be regularly updated with the nearest nodes with the lowest rank. The primary objective of the nodes are for environmental data collection and transmission. Therefore, when a data packet is generated by the sensor or is received from the other nodes, the leaf nodes will forward the data packet to a lower rank node in the route table. The data packet, other than the sensor data, contains the MAC address of the original sender, the MAC address of the next receiver and a random generated number for verification. When forwarding, nodes will change the MAC address that indicates the next receiver to the first address of its route table. Upon receiving a data packet successfully, the node will reply back to the previous sender immediately. The reply message, typically contains the message type and the random number from the corresponding data packet. When the node that sends out the data packet receives the reply, it will check the random number against the record to determine if the reply correct. This was designed for nodes to be able to listen out for multiple reply packets. However, due to the constraint of the LoRa module, the nodes will only listen for one reply message at a time. As LoRa does broadcast communicaiton only, nodes will receive data packets that are not intended for them, and are unable to tell if the data packets sent are successfully received by the receiver node. Therefore, the data reply mechanism is implemented to ensure important data packets are received at least once. The LoRa module only can receive or transmit one data packet at a time. As a result, messages cannot be reliably transmitted. To resolve this issue, data queues are implemented for data receiving, data to send and data sending. Because the main loop of the system is running in single thread, data receiving will be prioritised. As such, multiple data packets can be received and directly stored in the receiving queue before any filtering or processing. In the ideal situation, it will process the queued data packets from the queue. When data packet that requires a reply is sent, it will the packet be removed from the queue; if the reply is not received after a certain duration, the data packet in the sending queue will be moved back to the data to send queue again. Moreover, packets that does not require a reply, such as the reply packets, will be prioritised by placing them at the start of the queue to prevent deadlocks. Network interrupt handling is one of the greatest strength of the DSR protocol. Whenever a data reply is not received after a fixed duration, it will retry up to three times, before switching to the next address in the route table. As the data packets are all stored in a queue, there will be lesser chance of having missed a data packet, although it poses a potential risk of buffer overflow. If the error occurs too many times, it will attempt to switch to use ESP-NOW mesh to maintain the connectivity. This LoRa implementation approach showcases the robustness, swiftness, and efficiency in environmental data transmission. Its design significantly reduces the complexity of route tables and mitigates the flooding effect, enhancing network reliability. With swift data packet handling through prioritized queues and a retry mechanism, it ensures data integrity and timely responses, even with the single-thread limitations of the LoRa module.

C. Master Node Infrastructure

The Master Node Infrastructure encompasses a suite of pivotal components meticulously designed to harmoniously facilitate the monitoring and management of IoT devices. Charged with duties spanning data aggregation, processing, and user interface provisioning, the Master Node assumes a central role in orchestrating system functionalities. Core elements of this infrastructure comprise the Liligo Gateway, a Raspberry Pi MQTT Client and Broker, and the ELK Stack. Furthermore, our implementation integrates a router to establish an artificial cloud-based server for the ELK Stack.

1) Protocol Master: Serving as the protocol master within the Master Node setup, the Liligo Module shoulders the vital responsibilities of traffic monitoring, serialization, and MQTT-based transmission to the ELK stack. In collaboration with a Raspberry Pi configured as both MQTT Client and Broker, the Liligo Module establishes a USB connection for seamless data transfer to the Raspberry Pi. Subsequently, the Raspberry Pi orchestrates the dissemination of serialized data to the ELK stack via MQTT.

Configured to monitor Lora and ESP-Now Protocol packets, the Liligo Module acts as the primary data acquisition unit. Upon reception, data undergoes serialization and is then forwarded to the MQTT Broker by the Raspberry Pi 4b, facilitating streamlined integration with the ELK stack for subsequent processing.

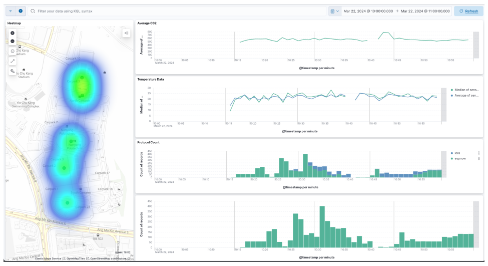

Fig. 4. ELK Dashboard

2) ELK Stack: The deployment of the ELK Stack is streamlined through the utilization of Docker and Docker Compose, enabling efficient deployment with minimal configuration overhead. Comprising Elasticsearch, Logstash, and Kibana, the ELK Stack offers a robust framework for data storage, processing, and visualization. Elasticsearch, a distributed, RESTful search and analytics engine, serves as the cornerstone for storing extensive datasets. Logstash functions as a versatile data processing pipeline, ingesting data from diverse sources, transforming it, and then forwarding it to a designated "stash" such as Elasticsearch. Kibana, a powerful data visualization dashboard, empowers users to interact with and explore data stored in Elasticsearch.

Raw data emanating from the Liligo Module is routed to the Raspberry Pi before being relayed to the ELK stack for processing. Following processing, data is stored within Elasticsearch. Kibana is subsequently employed to create visualization dashboards, offering insights into various metrics such as packet transmission, environmental parameters, device connectivity, and protocol distribution.

These visualizations furnish valuable insights facilitating the monitoring of the Wireless Sensor Network (WSN), enabling performance evaluation of protocol switching. This analysis, outlined in Section VI, encompasses metrics including packet transmission, environmental data, device connectivity, protocol distribution, and power consumption analysis.

3) Switching: A pivotal feature ingrained within all slave nodes is the capability to seamlessly transition between the LORA and ESP-Now protocols in response to evolving network dynamics. This feature assumes paramount significance in the comprehensive evaluation of each protocol's efficacy across diverse network conditions. The switching mechanism, autonomously activated by individual slave nodes, predicates its decision on the observed packet failure rate. Upon discerning a notable uptick in packet failures, the slave node instigates a judicious transition to the alternative protocol. Such a mechanism stands as a linchpin in preserving network robustness and fostering optimal performance amidst fluctuating network exigencies.

An inherent advantage of this mechanism lies in its facilitation of comparative performance assessments between the two protocols amidst varying network topologies and conditions. By juxtaposing their respective performance metrics, stakeholders can glean invaluable insights into the aptness of each protocol for specific application scenarios contingent upon the prevailing network milieu. Moreover, the dynamic switching functionality serves as a potent tool for pinpointing the protocol best suited to a given network configuration, thereby engendering heightened network efficiency and reliability.

A visual representation of the switch is depicted in Figure 5

VI. Analysis

A. Network Analysis

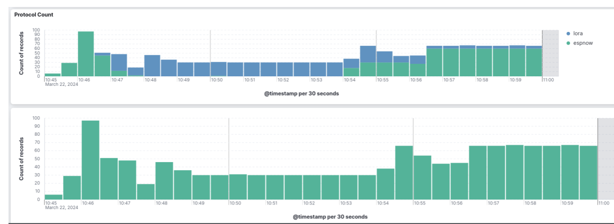

Fig. 5. Protocol Distribution

Figure 5 delineates the distribution of transmitted packets across the LoRa and ESP-Now protocols. The graph portrays LoRa Packets denoted in blue and ESP-Now Packets in green in the Protocol Count-Time Graph (Top). Evidently, the ESP-Now protocol emerges as the predominant choice, exhibiting a higher packet count vis-à-vis the LoRa protocol in the Count of records-Time Graph (bottom).

The discernible instances of protocol switching, notably observed around the timestamps 10:46 and 10:54, underscore the dynamic nature of the implemented ESP-Now Mesh and LoRa Mesh. During these transitions, a shift from LoRa to ESP-Now and vice versa is perceptible, as delineated by the transitions between the blue and green segments in the graph.

All calculations are based on the figures depicted in Figure 6 for ESP-NOW throughput and Figure 7 for LoRa throughput. These figures represent the packets received by the Master node in a network configuration consisting of 4 nodes arranged in a linear topology with 4 levels of hopping. The timing intervals for data collection were approximately 60 minutes for ESP-NOW and 50 minutes for LoRa Mesh.

Fig. 6. ESP NOW Throughput

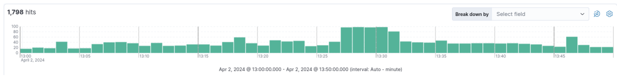

Fig. 7. LoRa Throughput

Figure 6 illustrates the received ESP-NOW packets, while Figure 7 displays the received LoRa packets. These visualizations provide insights into the performance of each protocol within the network configuration.

The data collected from these figures serves as the basis for calculating various network metrics, including throughput, latency, and bandwidth, which are essential for evaluating the efficiency and effectiveness of the communication protocols in the network.

Note that the extra multiple of 3 is used to scale the LoRa data to fit the same time delay implemented in the ESP-NOW Mesh (1 Second for ESP-NOW vs 3 second for LoRa).

1) Throughput Calculation:: To calculate the throughput of our mesh network, we utilized the formula:

$$\text{throughput} = \frac{\text{total packets received}}{\text{Total time (seconds)}}$$

For ESP NOW:

$$\text{throughput} = \frac{21694}{60 \cdot 60} \approx 6.0261 \text{ packets/second}$$

For LoRa:

$$\text{throughput} = \frac{1789}{60 \cdot 50} \cdot 3 \approx 1.789 \text{ packets/second}$$

ESP NOW achieves a throughput of approximately 6.0261 packets per second, indicating its capability to handle a higher volume of data compared to LoRa. Despite LoRa's lower throughput of approximately 1.789 packets per second, it is known for its extended range and superior penetration through obstacles. Therefore, while ESP NOW excels in terms of raw data transmission rate, LoRa's strengths lie in its ability to maintain connectivity over longer distances and in challenging environments.

2) Latency Calculation:: To calculate latency, we use the formula:

$$\text{Latency} = \frac{\text{Total time (seconds)}}{\text{total packets received}}$$

For ESP NOW:

$$\text{Latency} = \frac{60 \cdot 60}{21694} \approx 0.165 \text{ seconds/packet}$$

For LoRa:

$$\text{Latency} = \frac{60 \cdot 50}{1789} \cdot 3 \approx 1.676 \text{ seconds/packet}$$

Latency represents the time it takes for a packet to travel from the sender to the receiver in the network.

For ESP NOW, the latency is approximately 0.165 seconds per packet.

For LoRa, the latency is approximately 1.676 seconds per packet.

Comparing both results, ESP NOW exhibits significantly lower latency compared to LoRa, making it more suitable for applications requiring real-time or low-latency communication.

3) Bandwidth Calculation:: Bandwidth is calculated using the formula:

$$\text{Bandwidth} = \frac{\text{Packet Count} \cdot \text{Packet Size}}{\text{Total time (seconds)}}$$

For ESP NOW:

$$\text{Bandwidth} = \frac{21694 \cdot 30 \text{ bytes}}{60 \cdot 60} \approx 180.78 \text{ (in bytes/second)}$$

For LoRa:

$$\text{Bandwidth} = \frac{1789 \cdot 50 \text{ bytes}}{60 \cdot 50} \cdot 3 \approx 89.45 \text{ (in bytes/second)}$$

Bandwidth quantifies the rate at which data is transmitted over the network.

For ESP NOW, the calculated bandwidth is approximately 180.78 bits per second.

For LoRa, the calculated bandwidth is approximately 89.45 bits per second.

Comparing both results, ESP NOW demonstrates higher bandwidth compared to LoRa. This indicates that ESP NOW is capable of transmitting data at a faster rate, making it advantageous for applications requiring high data throughput.

4) Network analysis summary: The comparison between LoRa and ESP NOW in the context of a Wireless Sensor Network (WSN) with 4 nodes in a line and 1 master reveals distinct performance characteristics. ESP NOW demonstrates significantly higher throughput, achieving approximately 6.0261 packets per second, compared to LoRa's throughput of approximately 1.789 packets per second. However, this higher throughput comes at the expense of increased latency, with ESP NOW exhibiting a latency of approximately 0.165 seconds per packet, while LoRa has a latency of approximately 1.676 seconds per packet. Furthermore, when considering bandwidth, ESP NOW provides higher bandwidth at approximately 180.78 bytes per second, whereas LoRa's bandwidth is approximately 89.45 bytes per second. Therefore, while ESP NOW offers superior throughput and bandwidth, LoRa presents advantages in terms of latency and potentially longer range and better obstacle penetration, making it a more suitable choice for scenarios prioritizing these factors.

B. *ESP-NOW vs LORA Power Consumption Analysis*

The following segment presents an analysis of the power utilization contrasts between ESP-NOW and LoRa nodes, alongside a comparative study of the master nodes for both ESP-NOW and LoRa protocols.

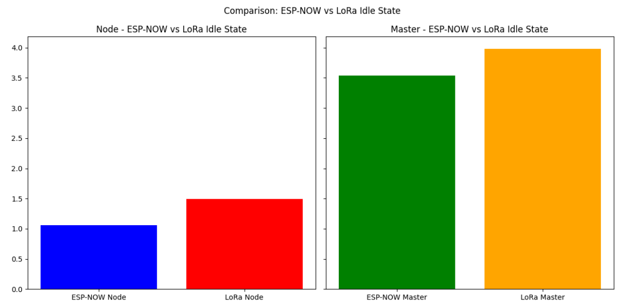

Fig. 8. Idle State of ESP-NOW VS LORA

Figure 8 shows that ESP-NOW technology demonstrates lower power consumption in the idle state for both node and master configurations. This suggests that ESP-NOW is a more energy-efficient option when devices are not actively communicating. This aspect of power efficiency is especially advantageous in applications where devices are expected to operate on battery power for extended durations and spend a significant amount of time in the idle state. The reduced energy usage can lead to longer battery life and decreased operational costs, making ESP-NOW an attractive choice for energy-conscious wireless communication implementations.

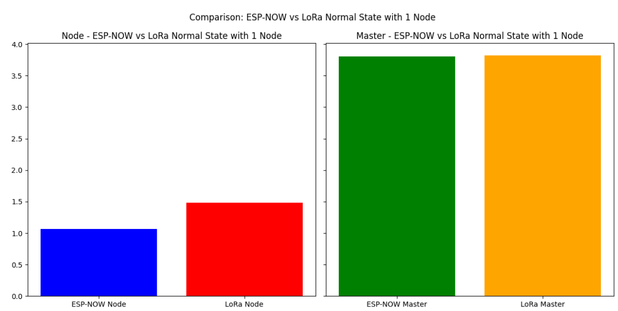

Fig. 9. Normal State with 1 node of ESP-NOW VS LORA

Figure 9 highlights the enhanced power efficiency of ESP-NOW in a single-node setup, where it outshines LoRa by registering a lower rate of power consumption, as illustrated by the blue bar for the ESP-NOW Node versus the red bar for the LoRa Node. This efficiency also extends to master nodes, where the ESP-NOW Master shows energy consumption that is nearly identical to that of the LoRa Master. The advantage of the ESP-NOW's lower power demand becomes particularly significant in configurations with a sole node, potentially extending the device's usable life and minimizing the need for frequent maintenance. The energy-conserving trait of ESP-NOW is also of paramount importance in applications where the power supply is constrained, such as with battery-operated sensors in isolated locations, making it a strategic choice for energy management.

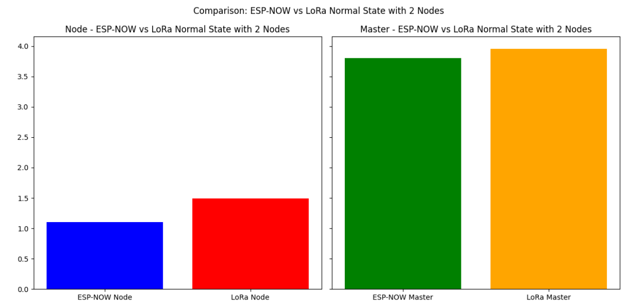

Fig. 10. Normal State with 2 nodes of ESP-NOW VS LORA

In the two-node scenario depicted in figure 10, the ESP-NOW Node (blue bar) consumes less power than the LoRa Node (red bar), demonstrating ESP-NOW's greater power efficiency. The Master nodes for both ESP-NOW (green bar) and LoRa (yellow bar) show similar power consumption, suggesting that for master nodes, the choice between ESP-NOW and LoRa may not significantly affect power usage. Overall, in a two-node system, ESP-NOW offers an advantage for individual nodes in terms of power efficiency.

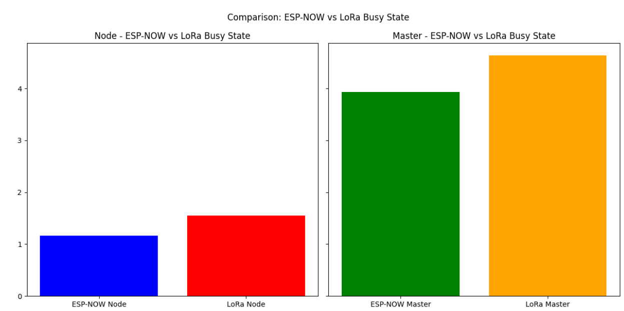

Fig. 11. Busy State of ESP-NOW VS LORA

Figure 11 depicts a busy state scenario, ESP-NOW technology (blue bar) still maintains lower power consumption compared to LoRa (red bar) for individual nodes. This indicates that even when the network is busy with higher levels of communication and data transmission, ESP-NOW nodes operate more efficiently than LoRa nodes.

For master nodes, the ESP-NOW Master (green bar) consumes significantly less power than the LoRa Master (yellow bar). In a busy network state, this difference suggests that ESP-NOW is notably more power-efficient for master nodes as well. The graph reflects that in high-activity situations, where nodes and masters are frequently communicating, ESP-NOW's power efficiency is a substantial advantage.

C. *Summary of Analysis*

In terms of performance within a wireless sensor network with four nodes, the ESP-NOW protocol surpasses LoRa with a higher throughput of 6.0261 packets per second against 1.789 packets per second for LoRa. It also benefits from considerably lower latency at approximately 0.165 seconds per packet compared to LoRa's 1.676 seconds per packet and greater bandwidth at 180.78 bytes per second over LoRa's 89.45 bytes per second. ESP-NOW's capabilities make it preferable for high-speed, low-latency communication needs, while LoRa remains a viable choice for its long-range and obstacle penetration abilities.

Regarding power consumption, ESP-NOW showcases lower power usage than LoRa in both idle and active (busy) network states. For single-node configurations, ESP-NOW's power efficiency could lead to longer battery life and less frequent maintenance, beneficial for remote and battery-operated devices. In networks with two nodes, ESP-NOW nodes use less energy, though master node power consumption is similar between the protocols. Even under busy conditions, ESP-NOW sustains its power-saving benefits for individual and master nodes alike, indicating a significant advantage for energy-sensitive deployments.

VII. Conclusion

A. Use Case

Our solution embodies a versatile application spectrum, catering adeptly to both research and practical scenarios. It facilitates comprehensive comparisons of latency, throughput, and bandwidth between the LoRa and ESP-NOW protocols, offering a framework easily adaptable to other protocol pairs such as Bluetooth Low Energy (BLE) and Zigbee. The Kibana dashboard serves as an intuitive visualization platform for protocol usage and sensor data, showcasing scalability to manage substantial data loads. Leveraging this data, predictive models, such as weather forecasting, can be trained, significantly broadening the solution's utility.

In practical contexts, our solution presents potential enhancements to communication infrastructures in government enforcement agencies like the military and police. Its reactive nature ensures seamless network adaptation, crucial for scenarios demanding swift failover mechanisms. For instance, in military logistics, real-time metadata collection—such as vehicle speed and location—can streamline operational planning. Our solution enables this by leveraging both ESP-NOW for short-range communication and LoRa for broader deployments, facilitating data transmission to a central server for analysis and decision-making.

While the implementation employs an MCU supporting both LoRa and ESP-Now, other implementations replicating this solution would similarly need to consider MCU capabilities for their respective use cases. Utilizing BLE and Zigbee could mirror ESP-NOW and LoRa, where either protocol can be toggled depending on the distance between endpoints. However, the comprehensive implementation must also account for potential clashes in protocol technologies.

B. Future Works

The current implementation of the ESP-NOW mesh serves as an effective short to medium-distance communication solution, seamlessly transitioning to LoRa for extended-range mesh communications within a Wireless Sensor Network (WSN). However, there are areas for future enhancement and refinement to maximize the network's capabilities.

One key area for improvement is the implementation of routing based on hop count and the introduction of health checks for network stability. These enhancements were not feasible within the initial implementation due to time constraints but could significantly improve the efficiency and reliability of the network.

In addition, addressing the observed slight packet reception discrepancies between LoRa and ESP-NOW meshes requires further investigation. This may involve exploring factors such as protocol and mesh algorithm implementations, as well as inherent differences in transmission rates, to ensure optimal performance across both protocols.

Furthermore, streamlining protocol switching through the master node presents an opportunity to enhance overall system efficiency and control. By optimizing the switching process, the network can adapt more seamlessly to changing conditions and communication requirements.

While the current implementation focuses on a specific use case - a sensor collection mesh network - future iterations could explore broader applications and configurations. For instance, considering a fully implemented Dynamic Source Routing (DSR) algorithm for both protocols and implementing an election algorithm to select a master node with internet connectivity could automate mesh construction and data collection processes.

Moreover, the potential integration of a LoRaWAN gateway as the internet-connecting node opens up possibilities for transmitting data to a centralized database for analysis. Additionally, exploring the feasibility of a hybrid mesh network, where switching occurs between specific nodes rather than the entire network, could further optimize communication efficiency.

In summary, the multi-protocol mesh network with reactive switching algorithm presents significant potential for enhancement and customization to meet specific use cases. By addressing the identified areas for improvement and exploring new configurations, the network can achieve greater reliability, efficiency, and adaptability in diverse environments.

References

[1] D. Rainham, "A wireless sensor network for urban environmental health monitoring: Urbansense," *IOP Conference Series: Earth and Environmental Science*, vol. 34, 2016. DOI: 10.1088/1755-1315/34/1/012028.

[2] Y. Liu, X. Mao, Y. He, K. Liu, W. Gong, and J. Wang, "Citysee: Not only a wireless sensor network," *IEEE Network*, vol. 27, pp. 42–47, 2013. DOI: 10.1109/MNET.2013.6616114.

[3] M. Rechache, "Study of performance evaluation of RPL objective functions (MRHOF and OF0) for IOTs," Accepted: 2022-05-23T07:21:53Z, Thesis, UNIVERSITY OF KASDI MERBAH OUARGLA, 2021. [Online]. Available: http://dspace.univ-ouargla.dz/jspui/handle/123456789/29110 (visited on 02/11/2024).

[4] R. Alexander, A. Brandt, J. P. Vasseur, J. Hui, K. Pister, P. Thubert, P. Levis, R. Struik, R. Kelsey, and T. Winter, "RPL: IPv6 routing protocol for low-power and lossy networks," Internet Engineering Task Force, Request for Comments RFC 6550, Mar. 2012, Num Pages: 157. DOI: 10.17487/RFC6550. [Online]. Available: https://datatracker.ietf.org/doc/rfc6550 (visited on 02/12/2024).

[5] H. Kharrufa, H. Al-Kashoash, and A. Kemp, "Rpl-based routing protocols in iot applications: A review," *IEEE Sensors Journal*, vol. 19, pp. 5952–5967, 2019. DOI: 10.1109/JSEN.2019.2910881.

[6] L. Zhu, R. Wang, and H. Yang, "Multi-path data distribution mechanism based on rpl for energy consumption and time delay," *Information*, vol. 8, no. 4, 2017, ISSN: 2078-2489. DOI: 10.3390/info8040124. [Online]. Available: https://www.mdpi.com/2078-2489/8/4/124.

[7] H. Kharrufa, H. Al-Kashoash, Y. Al-Nidawi, M. Q. Mosquera, and A. Kemp, "Dynamic RPL for multi-hop routing in IoT applications," in *2017 13th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*, Feb. 2017, pp. 100–103. DOI: 10.1109/WONS.2017.7888753. [Online]. Available: https://ieeexplore.ieee.org/document/7888753 (visited on 02/12/2024).

[8] A. Arena, P. Perazzo, C. Vallati, G. Dini, and G. Anastasi, "Evaluating and improving the scalability of RPL security in the internet of things," *Computer Communications*, vol. 151, pp. 119–132, Feb. 1, 2020, ISSN: 0140-3664. DOI: 10.1016/j.comcom.2019.12.062. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366419307479 (visited on 02/12/2024).

[9] A. Mayzaud, R. Badonnel, and I. Chrisment, "A distributed monitoring strategy for detecting version number attacks in RPL-based networks," *IEEE Transactions on Network and Service Management*, vol. 14, no. 2, pp. 472–486, Jun. 2017, Conference Name: IEEE Transactions on Network and Service Management, ISSN: 1932-4537. DOI: 10.1109/TNSM.2017.2705290. [Online]. Available: https://ieeexplore.ieee.org/document/7930501 (visited on 02/12/2024).

[10] A. Verma and V. Ranga, "Security of rpl based 6lowpan networks in the internet of things: A review," *IEEE Sensors Journal*, vol. 20, no. 11, pp. 5666–5690, 2020. DOI: 10.1109/JSEN.2020.2973677.

[11] H.-C. Lee and K.-H. Ke, "Monitoring of large-area iot sensors using a lora wireless mesh network system: Design and evaluation," *IEEE Transactions on Instrumentation and Measurement*, vol. 67, no. 9, pp. 2177–2187, 2018. DOI: 10.1109/TIM.2018.2814082.

[12] R. Berto, P. Napoletano, and M. Savi, "A lora-based mesh network for peer-to-peer long-range communication," *Sensors*, vol. 21, no. 13, 2021, ISSN: 1424-8220. DOI: 10.3390/s21134314. [Online]. Available: https://www.mdpi.com/1424-8220/21/13/4314.

[13] M.-D. Ong and T.-M. Phan, "Mesh lora node design with reactive ad-hoc routing protocol," vol. 2, pp. 10–15, Jul. 2023.

[14] E. M. Belding-Royer and C. E. Perkins, "Evolution and future directions of the ad hoc on-demand distance-vector routing protocol," *Ad Hoc Networks*, vol. 1, no. 1, pp. 125–150, 2003, ISSN: 1570-8705. DOI: https://doi.org/10.1016/S1570-8705(03)00016-7. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1570870503000167.

[15] E. M. Royer and C. E. Perkins, "Multicast operation of the ad-hoc on-demand distance vector routing protocol," in *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, ser. MobiCom '99, Seattle, Washington, USA: Association for Computing Machinery, 1999, pp. 207–218, ISBN: 1581131429. DOI: 10.1145/313451.313538. [Online]. Available: https://doi.org/10.1145/313451.313538.

[16] R. Bhardwaj, *Aodv routing protocol " network interview*, Nov. 2020. [Online]. Available: https://networkinterview.com/aodv-routing-protocol/.

[17] Y. Kai, M. J. Feng, and M. Z. Hui, *Hybrid routing protocol for wireless mesh network*, Nov. 2009. [Online]. Available: https://ieeexplore.ieee.org/document/5375910/.

[18] M. Bahr, "Update on the hybrid wireless mesh protocol of ieee 802.11s," in *2007 IEEE International Conference on Mobile Adhoc and Sensor Systems*, 2007, pp. 1–6. DOI: 10.1109/MOBHOC.2007.4428721.

[19] B. Nassereddine, A. Maach, and S. Bennani, "The scalability of the hybrid protocol in wireless mesh network 802.11s," in *2009 Mediterrannean Microwave Symposium (MMS)*, 2009, pp. 1–7. DOI: 10.1109/MMS.2009.5409759.

[20] M. Ng and K.-L. Yau, "An energy-efficient hybrid wireless mesh protocol (hwmp) for ieee 802.11s mesh networks," Nov. 2013, pp. 17–21, ISBN: 978-1-4799-1506-4. DOI: 10.1109/ICCSCE.2013.6719925.

[21] K. Shobha and K. Rajanikanth, "Efficient flooding using relay routing in on-demand routing protocol for mobile adhoc networks," in *2009 IEEE 9th Malaysia International Conference on Communications (MICC)*, 2009, pp. 316–321. DOI: 10.1109/MICC.2009.5431521.
[22] L. Meng and W. Song, "Routing protocol based on grover's searching algorithm for mobile ad-hoc networks," *China Communications*, vol. 10, no. 3, pp. 145–156, 2013. DOI: 10.1109/CC.2013.6488843.
[23] K. Murugan, S. Balaji, P. Sivasankar, and S. Shanmugavel, "Cache based energy efficient strategies in mobile ad hoc networks," in *2005 IEEE International Conference on Personal Wireless Communications, 2005. ICPWC 2005.*, 2005, pp. 90–94. DOI: 10.1109/ICPWC.2005.1431308.
[24] S. Kakade and P. Khanagoudar, "Performance analysis of olsr to consider link quality of olsr-etx/md/ml in wireless mesh networks," *2017 2nd International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS)*, pp. 1–10, 2017. DOI: 10.1109/CSITSS.2017.8447791.
[25] T. C. Ed. and P. J. Ed., "Optimized link state routing protocol (olsr). ietf," *IEEE Sensors Journal*, p. 4, 2003.
[26] J. H. Ahn and T.-J. Lee, "Multipoint relay selection for robust broadcast in ad hoc networks," *Ad Hoc Networks*, vol. 17, pp. 82–97, Jun. 2014, ISSN: 1570-8705. DOI: 10.1016/j.adhoc.2014.01.007. [Online]. Available: http://dx.doi.org/10.1016/j.adhoc.2014.01.007.
[27] Z. Guo, S. Malakooti, S. Sheikh, C. Al-Najjar, M. Lehman, and B. Malakooti, "Energy aware proactive optimized link state routing in mobile ad-hoc networks," *Applied Mathematical Modelling*, vol. 35, no. 10, pp. 4715–4729, Oct. 2011, ISSN: 0307-904X. DOI: 10.1016/j.apm.2011.03.056. [Online]. Available: http://dx.doi.org/10.1016/J.APM.2011.03.056.
[28] A. H. Wheeb, R. Nordin, A. A. Samah, and D. Kanellopoulos, *Performance evaluation of standard and modified olsr protocols for uncoordinated uav ad-hoc networks in search and rescue environments*, Mar. 2023. [Online]. Available: https://www.mdpi.com/2079-9292/12/6/1334#:~:text=Compared%5C%20to%5C%20other%5C%20protocols%5C%2C%5C%20the,%5C%E2%5C%80%5C%9Chello%5C%E2%5C%80%5C%9D%5C%20messages%5C%20conveyed%5C%20increases..
[29] H. Li and L. Fen, "An improved routing protocol for power-line sensor network based on dsr," in *2010 2nd International Conference on Future Computer and Communication*, vol. 1, 2010, pp. V1-246-V1–249. DOI: 10.1109/ICFCC.2010.5497794.

APPENDIX

A. Raw Power Consumption: ESP-NOW vs LORA

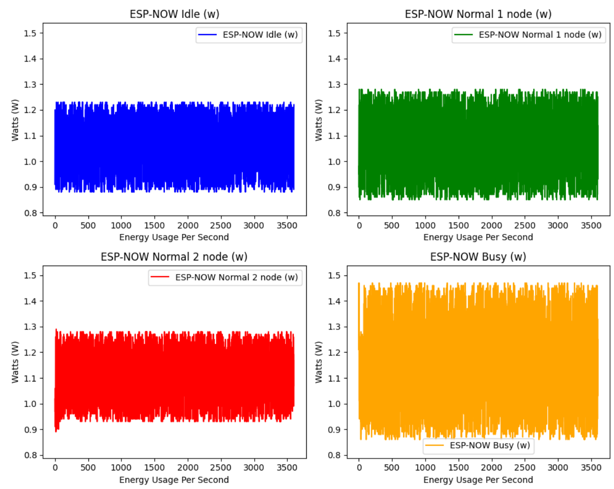

Fig. 12. ESP-NOW Node Power Consumption

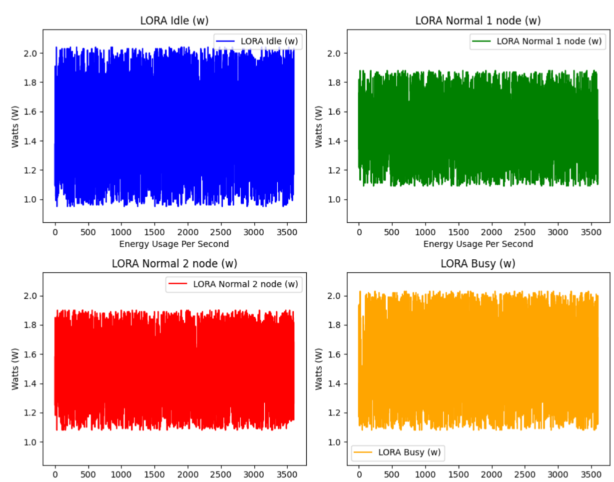

Fig. 13. LORA Node Power Consumption

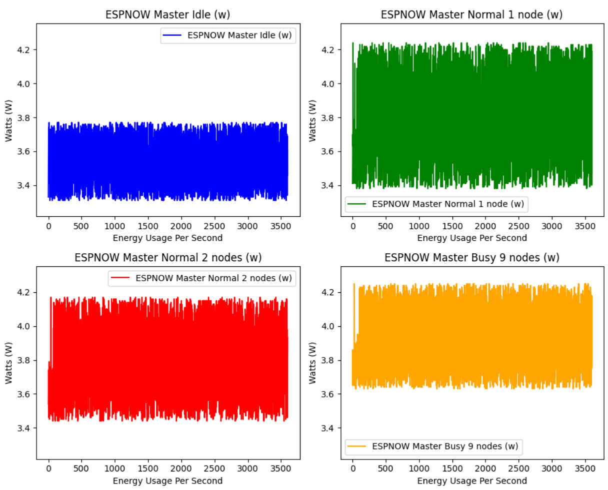

Fig. 14. ESP-NOW Master Power Consumption

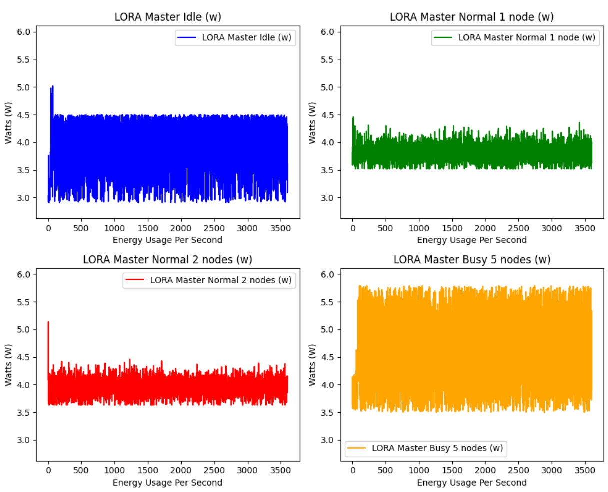

Fig. 15. LORA Master Power Consumption

B. ELK Stack Snapshots

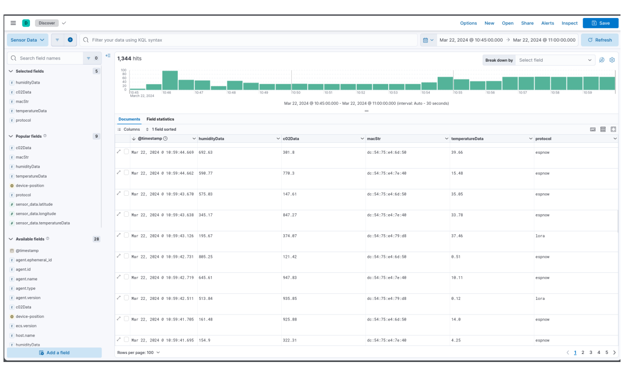

Fig. 16. Raw Data Dashboard

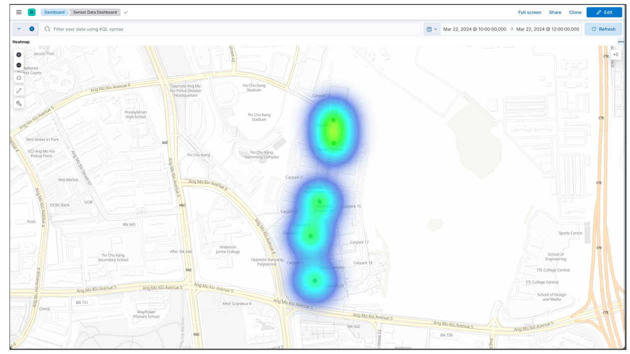

Fig. 17. Heatmap view