AUTONOMOUS DATA FERRYING FROM A SELF-ORGANIZING/HEALING WSN IN DISCONNECTED ZONES

25-26J-010

Project Proposal Report

Bandaranayake K.B.H.M.C.C

B.Sc. (Hons) Degree in Information Technology Specializing in Computer Systems & Network Engineering

Department of Computer Systems Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

August 2025

INTELLIGENT UAV-ASSISTED DATA MULING WITH PRIORITY-BASED ROUTING IN DISCONNECTED SENSOR DEPLOYMENTS

25-26J-010

Project Proposal Report

B.Sc. (Hons) Degree in Information Technology Specializing in Computer Systems & Network Engineering

Department of Computer Systems Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

August 2025

Declaration

I declare that this is my own work, and this proposal does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Name	Student ID	Signature
Bandaranayake K.B.H.M.C.C	IT22069122	

The above candidate is carrying out research for the undergraduate Dissertation under my supervision.

Signature of the supervisor: . Date: 28/08/2025

Abstract

Environmental monitoring in remote and disconnected regions remains a critical challenge due to the absence of permanent communication infrastructure, limited energy availability, and the risk of data loss during transmission. To address this, we propose a low-power, UAV-assisted data collection system for pre-deployed sensor nodes. The sensor nodes are manually deployed in the monitoring area and remain in deep sleep to conserve energy when inactive. A Cluster Head (CH) manages local data collection and stores measurements in JSON format, ensuring resilience during operations. For automatic data retrieval, the UAV carries a 433 MHz transmitter (XY-MK-5V), and each sensor node is equipped with a 433 MHz receiver connected to the ESP32 wake-up pin. When the UAV approaches, it sends a short RF signal, automatically waking the nodes. The CH then establishes a connection with the UAV's Wi-Fi access point and transmits the buffered data via MQTT. The UAV subsequently relays the data to a central server for processing and visualization. This approach provides a fully automatic, energy-efficient, and reliable mechanism for environmental data collection in hard-to-reach locations. It minimizes human intervention, reduces energy consumption through deep sleep operation, and ensures robust data transfer in disconnected environments

Keywords: UAV, ESP-NOW Communication, BLE Beaconing, MQTT Data Transfer, JSON Data Storage, Directional finding Algorithm (DFA), Energy aware routing model (EAR), Health Auditor (HA).

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Background and Literature Survey

Monitoring remote ecosystems such as biodiversity preserves, wildfire-prone forests, or disaster-hit zones remains difficult due to missing communication infrastructure, harsh terrain, and the demand for unattended, long-term sensing deployments. Conventional Wireless Sensor Networks (WSNs) often rely on static sinks or multi-hop forwarding to a base station, which rapidly depletes sensor batteries when infrastructure is absent [1].

A more sustainable approach combines WSNs with Unmanned Aerial Vehicles (UAVs), which act as mobile data ferries. UAVs periodically visit clusters of sensor nodes, retrieve their stored measurements, and deliver them to a base station. This eliminates the need for long-range or continuous transmissions, significantly extending both coverage and network lifetime [1], [2].

To further improve efficiency, we incorporate node-level data compression. Instead of transmitting full raw readings, sensor nodes apply lightweight compressive sensing (CS) techniques—using simple arithmetic operations to reduce data volume. The UAV collects these compressed packets during its scheduled flights and transports them to the base station, where powerful reconstruction algorithms decompress the information back into its original form. As demonstrated in [3], such UAV-enabled compressed data acquisition can cut down node energy consumption, improve transmission reliability, and still preserve high-quality data recovery.

By blending UAV mobility with on-site compression and centralized decompression, our system design addresses both energy and scalability challenges, providing a practical and long-lived solution for monitoring disconnected environments

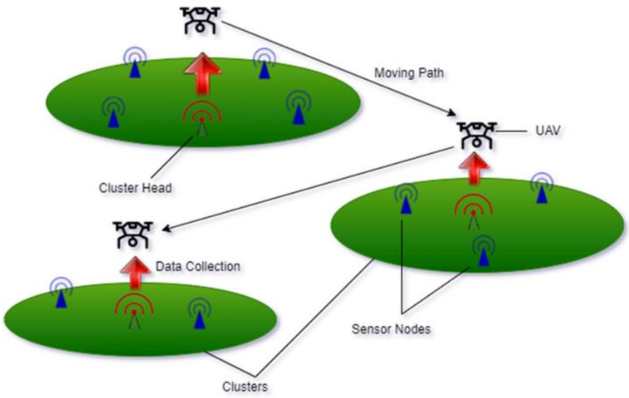

Figure 1.1.1:UAV acting as a mobile data ferry to collect buffered data from cluster heads in a WSN

Source: https://www.mdpi.com/2079-9292/10/21/2603

To further reduce energy consumption and improve scalability, cluster-based Wireless Sensor Networks (WSNs) designate a Cluster Head (CH), which aggregates data from Main Set (MS) nodes and manages upstream communication. Foundational protocols such as LEACH introduced rotating CHs and in cluster data aggregation to minimize transmissions, with subsequent studies consolidating clustering as a fundamental WSN design strategy [2][3][4]. Modern CH selection mechanisms often adopt multi-criteria trust metrics (e.g., residual energy, link quality) to maintain leadership stability, while time-division multiple access (TDMA) scheduling reduces collisions and idle listening by assigning wake-up slots to each MS node. Recent surveys reaffirm TDMA as an effective strategy for energy conservation in dense, contention-prone deployments [5].

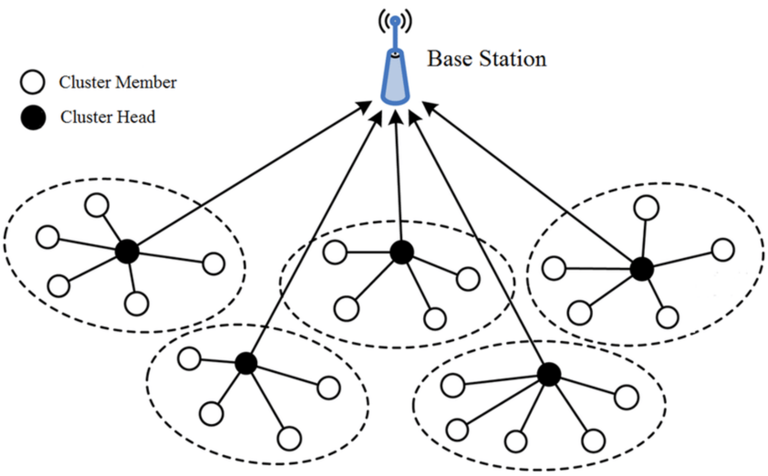

Figure 1.1.2: Cluster based WSN structure

Source: https://www.researchgate.net/figure/Cluster-based-WSN-Architecture_fig1_49600415

For short-range intra-cluster communication, ESP-NOW offers compelling benefits: it's connectionless, supports low-latency peer-to-peer exchanges, and avoids Wi-Fi overhead making it ideal for Member Set (MS) nodes that must wake briefly, transmit, and return to sleep. Empirical studies report that ESP-NOW significantly outpaces Wi-Fi in wake to transmit time about 5 ms vs 5 s yielding greater energy efficiency and responsiveness [9]. In practical deployments, such as bee colony monitoring, ESP-NOW enabled reliable long-distance communication (up to 250 m in open terrain) and achieved average active-mode currents of ~40 mA with deep-sleep currents near 0.8 mA [9]. Compared with Bluetooth Low Energy (BLE), ESP-NOW offers broader range (up to 220 m vs. BLE's <100 m) and comparable power consumption (ESP-NOW: ~60–100 mA; BLE: ~1–35 mA), though BLE excels in energy-sensitivity and device compatibility [9], [0]. One comparative study in Wireless Body Area Networks (WBANs) found BLE more energy-efficient overall, while ESP-NOW dominated in low-latency real-time performance suggesting potential for a hybrid approach leveraging both protocols' strengths [0]. Based on these findings, ESP-NOW is a promising candidate for intra-cluster data exchange, especially where fast, low-overhead transmissions are essential and MS nodes require frequent sleep cycles. BLE could serve complementary roles where ultra-low power and device interoperability are prioritized.

Automatic UAV-triggered wake-up. When a UAV arrives to collect data, the system must activate sensor nodes with minimal power overhead. To achieve this, the UAV carries a 433 MHz transmitter (XY-MK-5V), while each sensor node is equipped with a 433 MHz receiver wired to the ESP32 wake-up pin. As the UAV approaches, it sends a short RF pulse that automatically wakes the nodes from deep sleep. Once awake, the Cluster Head (CH) establishes a connection with the UAV's Wi-Fi access point and transmits buffered readings using the MQTT protocol, ensuring reliable bulk data transfer with acknowledgements. The UAV then relays this data to a central server for processing and visualization. Studies on sub-GHz RF wake-up radios confirm that such approaches drastically reduce idle listening energy while maintaining robust triggering in obstructed environments, making them well-suited for UAV-assisted WSN deployments [8], [9] Once proximity is confirmed, MQTT provides a lightweight, reliable publish–subscribe channel over Wi-Fi for bulk transfer of buffered readings from CH to UAV, with QoS levels and acknowledgements that fit intermittent contacts [10][11]. Finally, using SPIFFS with JSON at the CH enables persistent, structured storage, ensuring that data survives resets and CH re-elections. This practice, widely adopted in embedded ESP32 deployments, supports durability until data is successfully handed over [6].

Robustness under NLOS and obstructed terrain. Because 2.4 GHz links suffer foliage and obstacle attenuation, clusters cannot assume clear LoS between MS nodes and the CH [7]. To harden the intra-cluster plane, we adopt a dual strategy: (i) keep management/proximity on BLE 5 LE Coded PHY (S=8, 125 kbps) for better penetration and sensitivity (≈ 6–10 dB gain vs. 1 M PHY), improving discovery/handshake reliability in NLOS [8], [9]; and (ii) use ESP-NOW with relay fallbacks for data, enabling short multi-hop paths around blockers when direct MS→CH links degrade. Recent evaluations show BLE 5 Coded PHY sustains links at much lower RSSI in both indoor and outdoor NLOS settings, and BLE-Mesh can maintain coverage across obstacle-rich industrial layouts when nodes are sparsely duty-cycled [9], [10]. For ESP-NOW, indoor/outdoor studies report higher penetration and lower loss than TCP/Wi-Fi under obstacles, and demonstrate feasibility of multi-hop ESP-NOW to bypass NLOS, with acceptable latency/throughput for bursty sensor uploads [11]. Where clusters remain highly obstructed, we schedule a brief concurrent-flood fallback (e.g., Glossy-style flooding) to blast small control/data chunks with high reliability during the CH's TDMA window, minimizing duty-cycle overhead yet coping with deep fades [12]. Together, BLE-Coded PHY (management) + ESP-NOW (data with relay) + rare, timed flooding fallback yields resilient operation in forests and built-up terrain without abandoning your low-power sleep-heavy design

In addition to these intra-cluster optimizations, the UAV itself plays a critical role as the human-facing interface of the WSN. Recent literature emphasizes that UAVs are no longer passive ferries, but active participants that provide intelligence to the network. One emerging approach is map-before-mule scanning, where the UAV first performs a lightweight sweep to discover cluster locations, estimate node density, and build a coverage heatmap. Such UAV-assisted localization and mapping has been shown to significantly improve mission planning efficiency and situational awareness for operators [15], [16].

Building on this, UAVs can perform energy-aware mission routing, dynamically adjusting their paths based on predicted cluster head (CH) urgency. Instead of visiting all CHs on every sortie, the UAV maintains a priority list—with low-battery or critical nodes ranked higher—enabling urgent sweep, full sweep, and adaptive mission modes. Prior studies demonstrate that prioritizing data collection according to node energy or deadlines improves network lifetime and reduces data loss [17], [18].

Finally, UAVs can act as aerial node health auditors, probing for silent or faulty nodes during their scanning passes. This allows the UAV to flag failing clusters (e.g., offline, low-trust, or depleted batteries) and display them as alerts on the base station dashboard. Surveys on WSN fault diagnosis and UAV-assisted recovery highlight that integrating such lightweight diagnostics improves resilience in obstructed and disconnected environments [19], [20].

Together, these strategies position the UAV as more than just a data collector: it becomes a mapper, a smart planner, and a diagnostic agent—greatly enhancing the adaptability and sustainability of WSN deployments in harsh, infrastructure-less environments.

1.2 Research Gap

While UAV-assisted data mulling, RSSI-based localization, and energy-aware routing have each been explored individually, there is a notable lack of UAV–WSN systems that integrate robust node localization, adaptive mission planning, and health diagnostics into a cohesive architecture for disconnected environmental monitoring.

Most current research efforts still exhibit one or more of the following shortcomings:

- Dependence on GPS or RSSI-only localization: Many studies assume GPS-equipped nodes for precise localization, which is impractical in low-cost, energy-constrained WSNs [15]. RSSI-based methods have also been attempted [16], but they are highly unreliable due to multipath fading, interference, and antenna orientation—leading to unstable and inaccurate mapping.
- Static or exhaustive UAV routing: Existing UAV–WSN systems often require the UAV to visit all cluster heads (CHs) in each mission, regardless of data urgency or energy levels. This results in inefficient energy use, delayed collection from priority CHs, and reduced UAV endurance [17], [18].
- Lack of adaptive mission modes: Most path planning strategies are purely distance-based. They rarely incorporate mission objectives, such as urgent sweep (high-priority CHs only), full sweep (all CHs), or adaptive sweep (balanced energy/data trade-offs) [19].
- Absence of UAV-based health auditing: Current research focuses on data collection but neglects network diagnostics. Silent nodes, failing CHs, or rapidly draining clusters often go undetected until data loss occurs. There is little work on UAVs acting as aerial auditors to proactively identify faults and relay them to operators [20].
- No map-based visualization of network health: Base station dashboards in most designs only show collected data, not network condition alerts. Human operators lack tools to quickly identify failing nodes or degraded regions in the WSN map [21].

- The research gap therefore lies in designing a unified UAV-assisted system that:
    - Provides accurate, GPS-free node localization using BLE 5.1 Direction Finding to overcome RSSI unreliability [22].
    - Dynamically prioritizes CHs for UAV routing based on predicted battery and mission urgency [18].
    - Enables UAVs to act as aerial health auditors, probing for silent/failing nodes and visualizing alerts on a live WSN map [20].

1.3 Research Problem

The core research problem can be stated as: How can a UAV-assisted WSN operating in disconnected environments be designed to achieve accurate GPS-free localization, enable energy-aware mission planning, and provide real-time network health auditing without overburdening resource-constrained sensor nodes?

This problem encompasses several interlinked technical challenges:

- Reliable, low-overhead node localization: The UAV must replace GPS or RSSI-only localization with BLE 5.1 Direction Finding (AoA/AoD) using an onboard antenna array, allowing accurate CH/MS mapping during the scanning phase with minimal node energy cost [22], [23].
- Dynamic energy-aware routing: The UAV must plan its routes not only based on distance, but also CH energy levels, data urgency, and mission objectives. A priority list of CHs should be dynamically updated, with onboard ML predicting CH battery health between muling rounds [17], [18].
- Mission mode flexibility: The UAV should support urgent sweep, full sweep, and adaptive sweep modes, giving operators and the UAV flexibility to balance endurance, coverage, and data criticality [19].
- Aerial node health auditing: The UAV must perform diagnostic sweeps to identify silent, failing, or energy-starved nodes, reducing the risk of unnoticed data loss [20].

- Map-based visualization of WSN health: The base station must display a real-time, map-based dashboard showing CH locations, prioritized routes, and flagged health alerts, enabling proactive operator intervention [21].

By addressing these aspects, the proposed research aims to create an intelligent UAV-assisted WSN framework that ensures accurate node mapping, adaptive energy-aware routing, and resilient network health monitoring in challenging environments such as wildlife reserves, disaster zones, or remote agriculture.

.

2 Objectives

2.1 Main Objective

To design and implement an autonomous UAV-assisted data muling framework that enables reliable cluster head–to–UAV data collection, BLE/ESP-NOW–based communication, and real-time dashboard visualization for resilient Wireless Sensor Network monitoring in disconnected environments.

2.2 Specific Objectives

Develop a GPS-Free Localization Mechanism Using BLE 5.1 Direction Finding (AoA/AoD).

- Integrate a BLE 5.1 antenna array module (SKYA21039 EVB with U.FL to MMCX connectors) connected to the UAV's ESP32 companion board for angle-of-arrival (AoA) (Directional finding Algorithm).
- Implement a real-time signal processing pipeline on the Raspberry Pi 4 to calculate bearing angles of BLE packets from cluster nodes, fusing multiple measurements for accurate 2D/3D positioning.
- Design a calibration procedure for the antenna array mounted on the UAV frame (S500 Quadcopter with carbon landing gear) to minimize multipath errors in outdoor/forest environments.
- Develop algorithms to map sensor cluster positions without GPS, ensuring robust operation in GPS-denied environments such as disaster zones or dense urban areas.

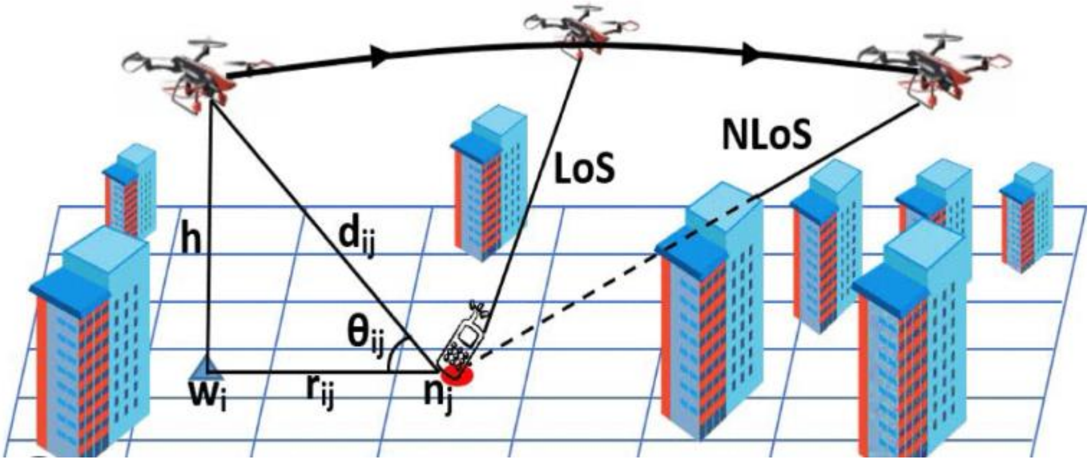

Figure 2.2.1: Autonomous UAV Trajectory for Localizing Ground Objects: A Reinforcement Learning Approach

Design a Dynamic Cluster Head (CH) Prioritization and UAV Routing Algorithm

- Implement an energy prediction model on the Raspberry Pi 4 to estimate the remaining energy of each CH, using historical consumption and battery capacity data from sensor nodes.
- Develop a multi-mode routing strategy in UAV control software (Pixhawk PX4 + RPi interface):
  - Urgent Mode – prioritize UAV visits to CHs reporting critical health events.
  - Full Mode – sequentially visit all CHs for complete data collection.
  - Adaptive Mode – dynamically adjust UAV paths based on CH energy status, distance, and node density.
- Synchronize the UAV routing engine with Pixhawk flight controller telemetry using MAVLink protocols to ensure safe path adjustments mid-flight.
- Simulate and optimize the routing algorithms before field deployment using ROS/Gazebo UAV simulation environment.

Implement a UAV-Based Health Auditing Module for Sensor Networks

- Configure the UAV's Raspberry Pi 4 to periodically scan for missing or silent CH beacons via BLE, detecting failing or disconnected sensor clusters.
- Enable the UAV to request real-time energy status reports from CHs via ESP-NOW or MQTT over Wi-Fi once communication is established.
- Design a fault detection engine on the Raspberry Pi to classify nodes into categories: Active, Silent, Energy-Starved, or Failed.
- Visualize node health status on a real-time WSN monitoring dashboard at the base station, with alerts triggered for failing or unreachable nodes.
- Implement a data relay function, where UAV automatically syncs auditing results back to the base station via Telemetry link.

Evaluate System Performance through Simulation and Field Experiments

- Set up a multi-node WSN testbed using ESP32-based sensor nodes with role-based CH election to simulate real-world energy dynamics.
- Deploy the UAV prototype in controlled outdoor environments for real localization and routing tests.
- Measure and benchmark the following key performance indicators (KPIs):
    - Localization Accuracy – error in mapping sensor cluster positions.
    - Energy Efficiency – average node lifetime improvements under CH prioritization.
    - Routing Performance – UAV travel time, coverage rate, and mission success rate.
    - Fault Detection Rate – accuracy in detecting silent/failing nodes.
- Compare results between simulation (ROS/Gazebo) and field deployments to validate scalability and reliability.

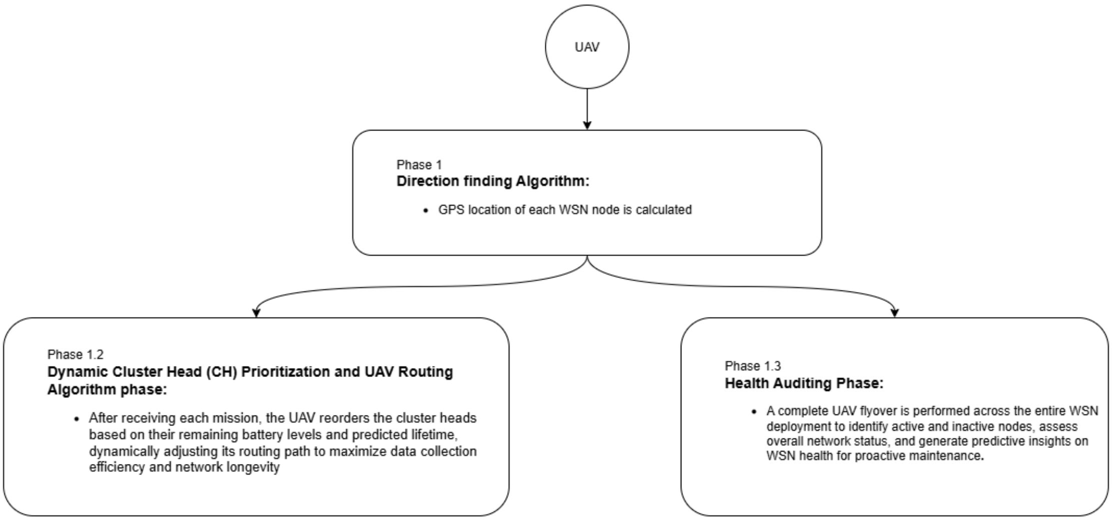

Figure 2.2.2: UAV-Assisted WSN Mission Workflow: Direction Finding, Dynamic CH Prioritization, and Health Auditing Phases

3 Methodology

The methodology for this research component focuses on the design and implementation of the UAV-assisted data muling framework, covering the scanning and localization of Cluster Heads (CHs), the generation of a real-time WSN topology map, and the energy-aware routing of the UAV for prioritized data collection. This section outlines the system design, implementation tasks, required hardware/software platforms, data handling mechanisms, mission scheduling strategies, and the anticipated outcomes of integrating UAV intelligence into the WSN data ferrying process.

3.1 System Design

Develop a simulation environment (e.g., NS-3, MATLAB, or Python-based testbeds) to train models, algorithms and practice UAV–WSN interactions.

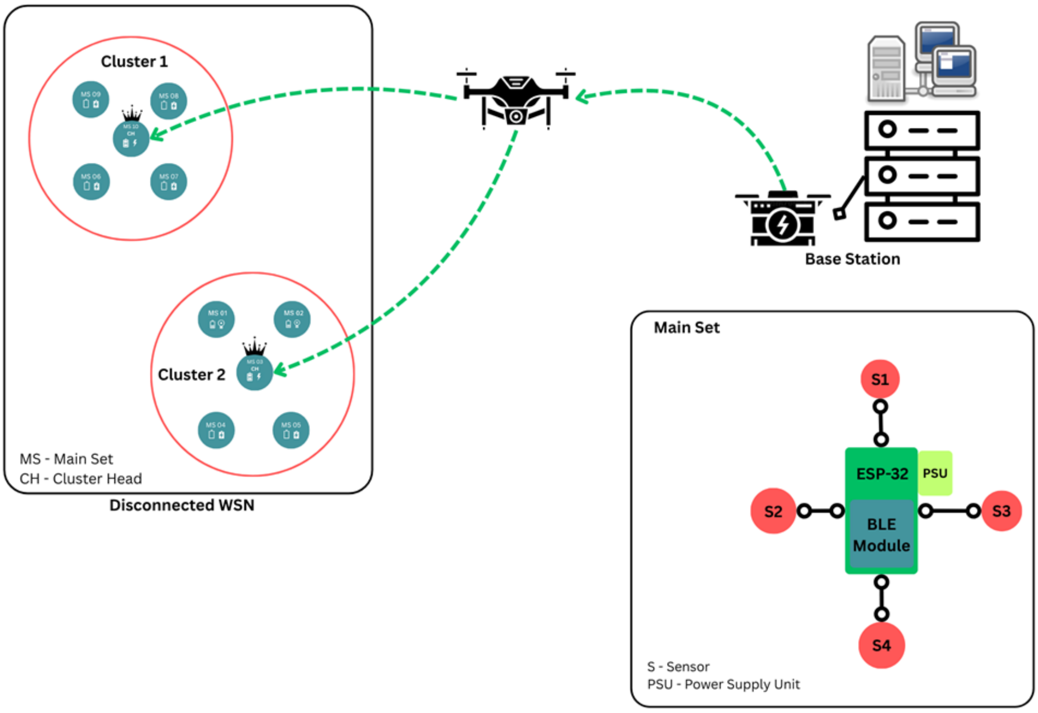

Figure 3.1.1: Proposed UAV-assisted cluster-based WSN architecture

Simulate BLE 5.1 Direction Finding localization methods for accuracy benchmarking.

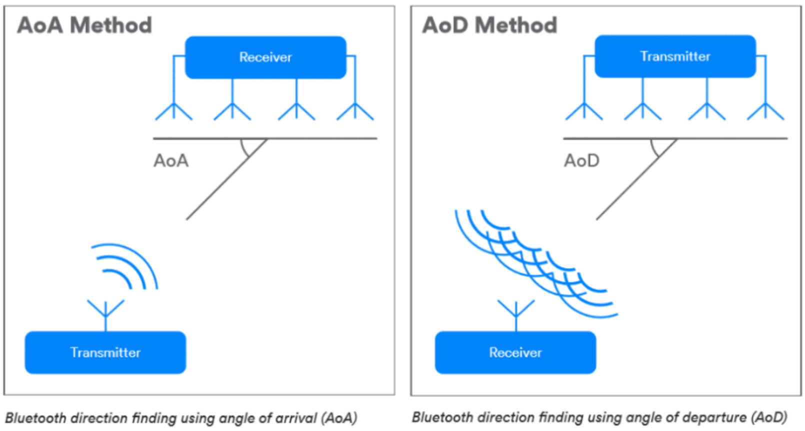

Figure 3.1.2: Angle of Arrival Method

Source: https://www.microcontrollertips.com/new-bluetooth-spec-gives-the-iot-a-sense-of-direction/

Localization & Mapping Phase

- Equip UAV with BLE 5.1 antenna array to perform scanning.

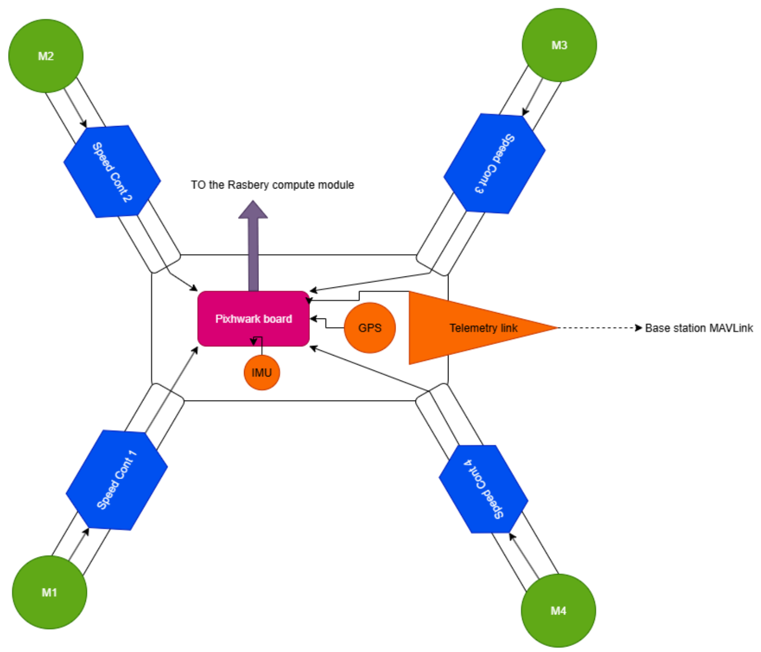

Figure 3.1.3: UAV system architecture Diagram PART1

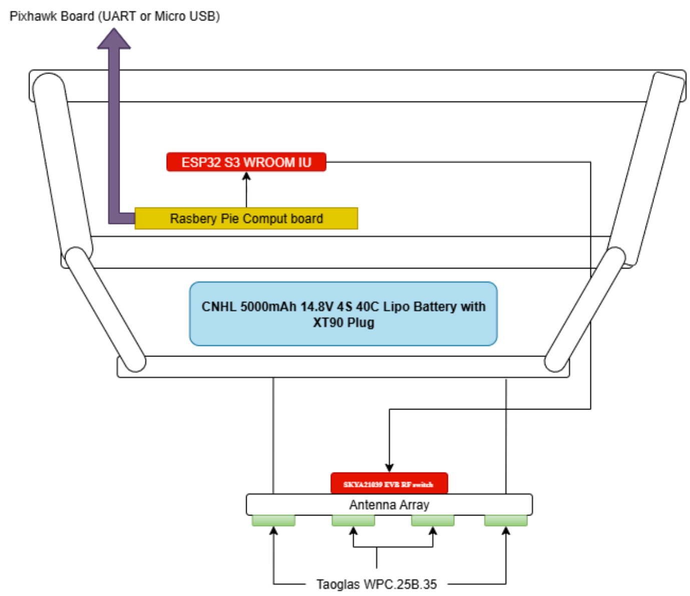

Figure 3.1.4: UAV system architecture Diagram PART2

- Use AoA/AoD data to compute cluster/node positions and repeat this process from 2 or more positions (triangulation) and compute the absolute position with minimum error.
- Construct and update a real-time WSN map at the base station.

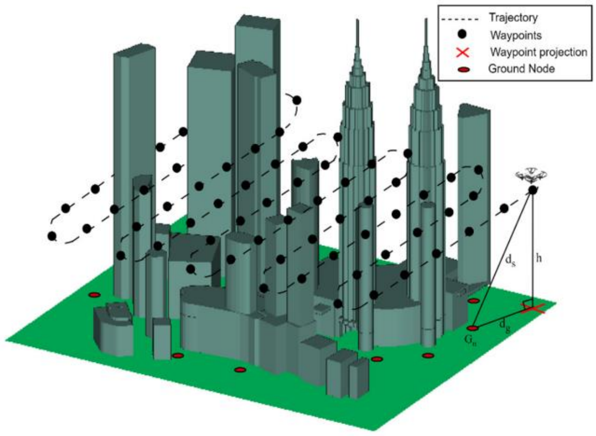

Figure 3.1.5: waypoint projection

Source: UAV-Assisted Localization of Ground Nodes in Urban Environments Using Path Loss Measurements

Energy-Aware Routing Phase

- Implement an onboard ML-based predictor for CH energy estimation.
- Generate dynamic CH priority lists and mission modes (urgent sweep, full sweep, adaptive sweep).
- Optimize UAV flight paths using shortest-path / TSP-based heuristics combined with energy-aware weighting.

Data Mulling & Health Auditing Phase

- Data mulling Pipeline.

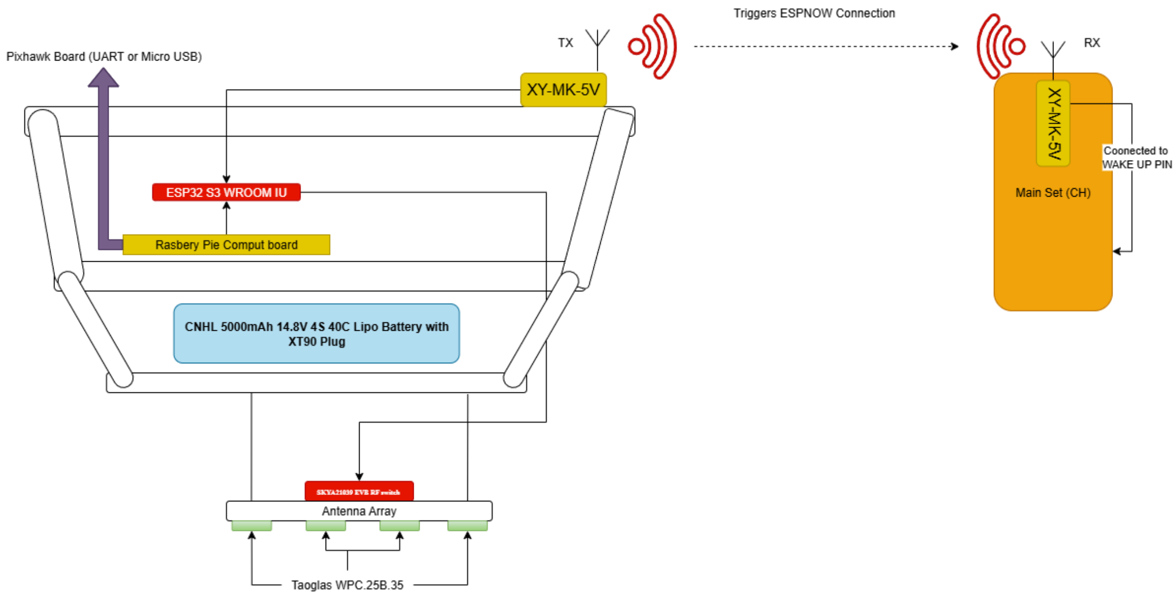

Figure 3.1.6: UAV system architecture diagram PART3

- During UAV visits, CHs transmit buffered data using MQTT.
- CH dump a battery report of each node in its cluster.
- Later this Data is being feeding to a Model to predict the healthiness of each node (Between 1st scanning and 1st data mulling mission the healthiness prediction done based on RSSI which can contain lot of error, multiple mission with each dumping a battery report can enhance its accuracy).
- Health anomalies are logged and visualized on a map-based dashboard.

Evaluation & Validation

- Metrics: localization error, UAV path length, CH fairness, data delivery ratio, energy consumption, node fault detection rate.

Comparative analysis: RSSI-only localization, static routing vs. proposed adaptive routing.

3.2 Implementation Tasks

Autonomous UAV configuration

- Configure the Pixhawk PX4 flight controller as the primary autopilot for the S500 quadcopter, integrated with the Raspberry Pi 4 companion computer via MAVLink protocol.
- Implement autonomous mission execution using ArduPilot/PX4 flight modes, allowing the UAV to follow pre-defined or dynamically generated waypoints based on cluster head (CH) locations.
- Connect the UAV-mounted ESP32 board to handle communication with sensor nodes, while the RPi4 executes higher-level tasks such as data processing, routing logic, and health monitoring.
- Design power management strategies using the 14.8V 4S LiPo battery with UBEC voltage regulation, ensuring stable power delivery to Pixhawk, Raspberry Pi, and BLE antenna modules.

Implementing BLE direction finding setup

- Integrate the BLE 5.1 antenna array (SKYA21039 EVB with U.FL to MMCX adapters) to the UAV's ESP32 board for AoA (Angle of Arrival) measurements.
- Program the ESP32 firmware to extract in-phase and quadrature (IQ) samples from received BLE packets.
- Stream IQ samples to the Raspberry Pi 4, where signal processing algorithms (MUSIC or ESPRIT) compute the direction and relative location of sensor nodes.
- Calibrate the antenna array orientation on the UAV frame to compensate for mechanical vibrations and multipath reflections during flight.
- Validate localization accuracy in controlled experiments by comparing estimated positions with known GPS ground truth (only for evaluation purposes).

Implementing Energy-Aware Routing system

- Develop an energy prediction module on the Raspberry Pi 4 that estimates each CH’s remaining energy by analyzing historical transmission intervals and battery data.
- Implement a dynamic prioritization algorithm that classifies CHs into priority groups (critical, moderate, safe) based on predicted energy levels.
- Synchronize routing decisions with the Pixhawk autopilot using MAVLink commands, enabling the UAV to:
  - Urgent mode: immediately visit CHs with critically low energy.
  - Full mode: visit all CHs sequentially for complete data retrieval.
  - Adaptive mode: adjust flight paths in real-time to balance coverage, energy efficiency, and urgency.
- Simulate routing strategies in ROS/Gazebo UAV environment before real deployment to optimize flight endurance and minimize unnecessary detours.

Implementing Health Auditing System

- Configure CH nodes to periodically broadcast BLE beacons indicating their operational status.
- Enable the UAV to detect missing or weak beacons, identifying silent or failing nodes during flyovers.
- Develop a lightweight UAV-side monitoring module on the Raspberry Pi 4 to classify nodes into Active, Silent, Energy-Starved, or Failed categories.
- Implement visualization at the base station with a real-time WSN map dashboard showing UAV-collected health reports and triggering alerts when anomalies are detected.
- Enable UAV-based emergency reporting, where critical node failures are immediately relayed to the ground station mid-flight

Testing the Data mulling function

- Implement the CH-to-UAV MQTT-based data exchange where the UAV acts as a mobile broker.
- Ensure QoS-based acknowledgements to guarantee that no data is erased from CH storage until UAV confirms successful reception.
- Perform controlled ground tests with single-node and multi-node clusters to verify reliable data transfer under weak Wi-Fi conditions.
- Storage Mechanism in the onboard computer of the Rasbery Pie.
- Extend tests to in-flight scenarios, measuring packet loss, latency, and UAV hovering stability during data collection.

Testing Each system pair wise or parallel

- Conduct pair-wise testing (e.g., UAV–CH link only, intra-cluster communication only, BLE direction finding only) to isolate and debug subsystems.
- Progress to parallel system integration testing, running all modules together under simulated field conditions.
- Evaluate system performance using defined KPIs: localization accuracy, routing efficiency, network lifetime improvement, and fault detection reliability.
- Finalize with field experiments where the UAV performs end-to-end missions, combining localization, energy-aware routing, health

3.3 Required Materials

To implement the proposed stages, the following hardware and software resources will be used:

- Hardware:

  UAV Platform & Flight Control

  - S500 Quadcopter Frame Kit with PCB version and carbon fiber landing gear.
  - 4 × 2812 900KV Brushless Motors (3–6S compatible, optimized for long-range UAV missions).
  - Carbon Fiber 1045 Propellers for efficient lift and stability.
  - ESC 40A Electronic Speed Controllers (SimonK firmware) for precise motor control.
  - Pixhawk PX4 PRO 32-bit Flight Controller supporting ArduPilot/ArduPlane/ArduRover.
  - Flysky FS-i6X 6–10CH 2.4GHz RC Transmitter with FS-iA6B Receiver for manual override and testing.
  - Raspberry Pi 4 (4GB) as UAV companion computer for localization, routing, and health auditing.
  - 6 mm CS-Mount Lens + Raspberry Pi HQ Camera (12.3 MP) for visual monitoring and optional mapping support (optional).

Communication & Localization Modules

- BLE 5.1 Direction Finding (AoA/AoD)
    - SKYA21039 EVB: BLE 5.1 evaluation board for IQ sample extraction and direction finding.
    - 4 × Taoglas WPC.25B.35 antennas, connected to the EVB via U.FL-to-MMCX adapters, arranged in a planar array on the UAV belly for Angle of Arrival measurement.

WSN Communication

- ESP32 Microcontrollers for Main Set (MS) and Cluster Head (CH) nodes.
- ESP-NOW wireless communication using ESP32 chip antenna for short-range intra-cluster data transfer.
- XY-MK-5V 433 MHz RF modules for UAV-to-CH wake-up signaling.

Telemetry & UAV Communication

- Telemetry antennas (2.4 GHz / 915 MHz) for Pixhawk ↔ Ground Control Station link.
- GPS antenna (for UAV navigation and ground-truth reference).

Power Systems

- CNHL 5000mAh 14.8V 4S 40C LiPo Battery with XT90 Plug for primary UAV power.
- UBEC 5V 5A BEC (shielded) for stable RPi and ESP32 power.
- 3S/4S XT60 Parallel Charging Board for simultaneous multi-battery charging.
- CNHL LiPo Battery Safe Bag for fireproof storage.
- ToolkitRC M6D Charger for balanced LiPo charging.

- Battery Voltage Tester (1–8S) with buzzer alarm for in-field voltage monitoring.
- Amass XT60-to-XT90 adapters for connector compatibility.

WSN Node Components

- ESP32-based CH and MS nodes, each with:
- Environmental sensors (temperature, humidity, soil moisture, air quality).
- Solar charging modules for energy harvesting.
- SPIFFS flash memory for persistent data buffering.
- RPi GPIO Ribbon Cable + Breakout Adapter for interfacing UAV-side Raspberry Pi with sensor boards.

Tools & Accessories

- UNI-T UT33D+ Multimeter for electrical debugging.
- 2 × RC Anti-Vibration Plate Dampers to minimize UAV vibrations on Pixhawk and antennas.
- U.FL-to-MMCX adapters/cables for antenna array integration.
- Miscellaneous mounting hardware for UAV belly array placement.

- Software & Tools
    - Arduino IDE & PlatformIO for ESP32 firmware development.
    - ESP-IDF Libraries for ESP-NOW, BLE AoA/AoD, SPIFFS, and MQTT integration.
    - PX4/ArduPilot Firmware for Pixhawk configuration.
    - MAVLink Protocols for RPi ↔ Pixhawk communication.
    - ROS & Gazebo for UAV flight simulation and routing testing.

- Python (NumPy, SciPy, MQTT libraries) for signal processing, health auditing, ML and visualization.
- Draw.io / Lucidchart for workflow and system diagrams.
- Dashboard
    - React frontend.
    - Spring boot micro service backend developed on a Kubernetes cluster.
    - PostgreSQL DB for Data storage
- GitHub for version control and collaboration.
- Serial debugging tools (Arduino Serial Monitor, minicom) for low-level communication testing

3.4 Data Handling & Testing

This study collects experimental data using a UAV platform (S500 quadcopter with Pixhawk PX4, Raspberry Pi 4, ESP32, and BLE 5.1 antenna array) interfacing with CH nodes from the WSN. No human surveys or interviews are conducted. Data handling focuses on:

- Sensor localization (BLE AoA/triangulation)
- Energy-aware CH routing with ML
- Health auditing with ML
- Reliable UAV data retrieval and system integration

Testing follows a staged methodology to validate each subsystem individually, then in fully integrated operation.

- UAV-Based Sensor Data Collection & Node Localization
  - Purpose: Accurately determine CH and MS cluster positions using BLE 5.1 Direction Finding without consuming UAV compute for other tasks.
  - System Inputs:
    - BLE IQ samples from CH nodes (via XY-MK-5V transmitter at CH)
    - UAV GPS & IMU data for position references
  - System Outputs:
    - 2D/3D positions of each CH/MS cluster
    - Localization confidence metrics
  - Procedure:
    - UAV Scanning Flight (Dedicated):
      - UAV focuses exclusively on BLE scanning.
      - Antenna array (Taoglas SKYA21039 + SKYA EVB boards) collects IQ samples at multiple UAV positions along the flight path.
    - Triangulation & Processing:

    - After scan completion, Raspberry Pi 4 executes MUSIC/ESPRIT algorithms to extract Angle of Arrival (AoA).
    - Multiple AoA readings combined via geometric triangulation to compute precise 2D/3D positions.
    - Outlier filtering and weighted averaging mitigate multipath and noise effects.
  - Validation & Metrics:
    - Compare estimated positions with known reference positions.
    - Calculate localization error (expected ~10^{-5} relative error).
- Test Cases:
  - UAV fly-over at different altitudes and orientations.
  - Evaluate localization accuracy vs number of AoA samples.
  - Test robustness under BLE interference conditions.

- Energy-Aware Cluster Head Routing (ML-Based)
  - Purpose: Optimize UAV route to visit CHs efficiently based on predicted energy levels.
  - ML Inputs:
    - Last known CH battery level (from previous UAV flights / CH JSON dumps)
    - Number of buffered packets per CH
    - Distance from UAV to CH waypoints
    - Historical energy consumption trends
  - ML Outputs:
    - Priority score for each CH
    - Recommended visit sequence for UAV

- Workflow:
  - Offline Training:
    - Use simulated CH energy depletion datasets to train ML model (e.g., regression with gradient descent optimization).
    - Evaluate using MAE/RMSE.
  - Deployment on UAV:
    - Load model weights on Raspberry Pi 4.
    - During real flights, UAV receives CH JSON dumps, predicts CH energy levels, and dynamically ranks CHs for efficient routing.
  - Flight Testing:
    - Conduct few controlled flights to compare predicted energy vs actual CH battery levels.
    - Refine ML model incrementally with new flight data.
  - Efficiency Considerations:
    - Model inference uses lightweight frameworks (TensorFlow Lite / PyTorch Mobile).
    - CPU and memory monitored to avoid affecting BLE scanning or flight control.
- Test Cases:
  - Verify ML predictions against actual CH battery measurements.
  - Measure energy savings and total flight distance.
  - Compare planned routes vs shortest path or baseline routing algorithm.

- Health Auditing with ML
  - Purpose: Detect failing or energy-starved CH/MS nodes for prioritized UAV visits.
  - ML Inputs:
    - Time series of RSSI / AoA signal strength
    - Last reported CH battery levels
    - Missing or delayed packets from CHs/MS nodes

- ML Outputs:
  - Fault probability per node
  - Flags for nodes exceeding threshold for immediate inspection
- Workflow:
  - Offline Training:
    - Use historical node failures or simulated faults to train ML classifier/regressor.
    - Optimize via gradient descent to minimize detection error.
  - Deployment on UAV:
    - Model stacked on the same Raspberry Pi 4 alongside routing ML.
    - Parallel inference during flight predicts node health in real-time.
  - Testing:
    - Simulate node failures, battery starvation, or packet loss.
    - Compare predictions with ground truth.
    - Evaluate precision, recall, F1-score for detection reliability.
  - Efficiency Considerations:
    - Parallel processing ensures BLE scanning, ESP-NOW reception, and ML inference run without interference.
    - Models are quantized and optimized for low-latency execution.

- Compute Module Stack & Real-Time Task Scheduling
    - Raspberry Pi 4 (UAV) Responsibilities:
        - High-Priority: BLE direction finding, IQ sample collection, ESP-NOW reception.
        - Medium-Priority: ML inference for energy-aware routing and health auditing.
        - Low-Priority: JSON logging, telemetry to base station, dashboard updates.
    - Efficiency Measures:
        - Multi-threaded or asynchronous Python tasks.
        - Monitoring memory usage to prevent swapping or CPU throttling.
        - Quantized ML models for fast inference without affecting UAV flight stability.
- Staged Testing & Full System Integration
    - Scanning Phase:
        - UAV flies dedicated scanning route to collect BLE IQ samples.
        - Processes triangulation offline or in-flight after completing scan.
        - Predictive CH positions sent to base station via telemetry link.
    - Energy Prediction & Routing Phase:
        - UAV receives CH list, predicts energy levels using ML.
        - Generates optimal UAV path considering distance and predicted battery.
        - Conducts limited flight to validate predictions vs actual CH energy.
    - Data Collection Phase:
        - UAV visits selected CHs.
        - Each CH transmits JSON dump of MS energy/data via ESP-NOW or BLE-triggered Wi-Fi.

- Health Auditing Phase:
    - ML predicts failing or low-energy nodes using newly collected JSON data.
    - Updates UAV route for prioritized inspections in future flights.
- Metrics and Error Analysis:
    - Localization Accuracy: 2D/3D error of CH positions vs ground truth.
    - Energy Prediction Accuracy: MAE/RMSE of predicted vs actual CH energy.
    - Routing Efficiency: Total flight distance, energy consumption, % of critical CHs visited.
    - Health Audit Accuracy: Precision/recall/F1 of failing/silent node detection.
    - Data Delivery Reliability: % of JSON packets successfully collected.
    - Latency & Resource Usage: Time from data reception to ML decision; CPU/memory utilization

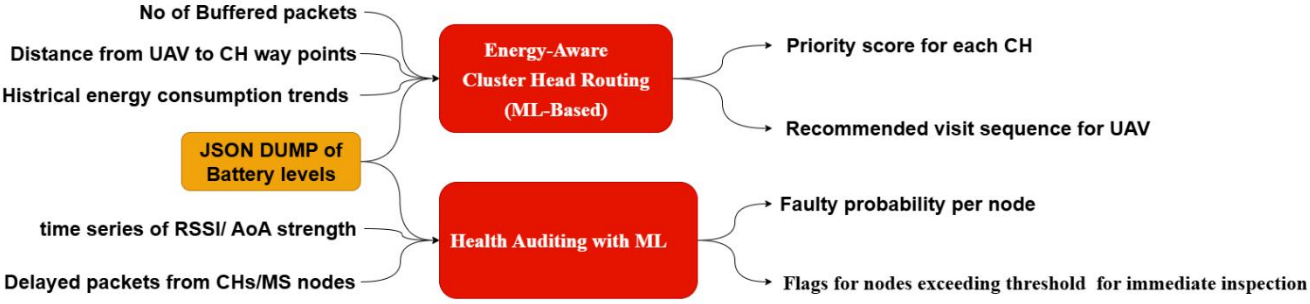

Figure 3.4.1: Model Architecture V1

3.5 Anticipated Outcomes

- Reliable Cluster-to-UAV Data Offloading

All CHs are able to securely transmit buffered JSON packets to the UAV through MQTT with acknowledgement-based clearing. This process ensures that data is not only transferred reliably but also cleared from the CH once successful delivery is confirmed. As a result, the system guarantees zero data loss during the UAV's aerial collection windows, thereby maintaining the integrity of sensing operations.

- Seamless UAV-Assisted Network Awareness

Beyond data collection, the UAV actively contributes to situational awareness by detecting CHs through BLE beaconing during its scanning phase. This mechanism allows the UAV to construct and update a live cluster availability map, which reflects the reachability and operational status of nodes in real time. Such visibility ensures that both the UAV and the operator remain informed about network health throughout mission execution.

- Energy-Aware CH Prediction and Scheduling

Following each data muling phase, the UAV gathers JSON reports containing the battery levels of CH and MS nodes. These reports feed into a lightweight machine learning model, based on gradient descent regression, which refines predictions of CH battery depletion across multiple flights. The resulting predictions are integrated into mission planning so that future UAV flights can proactively adapt to the dynamic energy states of the network, improving sustainability and reducing risk of node failures.

- Interactive Operator Dashboard

An interactive web-based dashboard is used to display UAV-collected data, CH health status, and predicted node lifetimes. Through this interface, operators are able to designate which CHs should be prioritized in upcoming missions while receiving clear visual alerts when CHs become unresponsive or show signs of imminent failure. These early warnings provide an opportunity for timely intervention, thereby preventing potential cluster collapse and extending network lifespan.

- Mission-Aware UAV Decision Making

The UAV adapts its scanning, routing, and offloading behavior by considering predicted CH lifetimes, its own remaining energy, and the urgency of sensed data distinguishing, for instance, between critical alarms and routine readings. This adaptability allows the UAV to achieve an optimal balance between mission coverage, responsiveness to urgent events, and efficient use of its energy resources, ultimately increasing overall mission effectiveness.

- Continuous Refinement Through Auditing

Each mission contributes to the UAV's long-term learning process by enriching its dataset of CH health and energy reports. Over time, this iterative auditing enables the UAV to continuously refine the accuracy of its predictions. As a result, the UAV becomes increasingly "intelligent," reducing unnecessary flights, avoiding redundant operations, and ensuring that high-priority CHs are never overlooked.

4 Project Requirements

4.1 Functional requirements.

The system must provide the following core functions:

- CH-to-UAV Communication

  A reliable communication framework is essential between CHs and the UAV. The UAV detects CHs using BLE 5.1 direction-finding and ranging, enabling accurate localization. Buffered JSON data is subsequently offloaded to the UAV via Wi-Fi using MQTT with acknowledgement mechanisms, ensuring secure and lossless transfer of information.

- UAV-Assisted node GPS location finding

  The UAV is tasked with assisting in the localization of network nodes. Through beacon scanning, the UAV facilitates the generation of a cluster map that captures node positions. This information is further integrated into a geo-tagged dashboard, allowing operators to visualize the spatial distribution of CHs and MS nodes in real time.

- UAV-Assisted Path Planning

  Path planning is a critical function for optimizing UAV operations. The UAV computes optimal flight routes by considering multiple parameters such as CH priority, predicted node lifetimes, UAV battery availability, and the mission mode. This adaptive routing ensures balanced coverage, energy efficiency, and responsiveness to high-priority tasks.

- Aerial Health Auditing

  The UAV also functions as a health auditor for the WSN. During its mission, it probes CHs and MS nodes to determine their operational status, classifying them as alive, silent, low-energy, or failed. This information is then used to generate a health status map for operators, offering early warning of potential network issues.

- Dashboard Interface

  A web-based dashboard provides live visualization of the entire system. It displays CH and MS health status, UAV-collected data logs, and the current flight status of the UAV. Additionally, the dashboard allows operators to select specific CHs to prioritize and configure mission profiles, thereby offering direct human-in-the-loop control.

- Machine Learning Integration

  Machine learning models are integrated into the UAV's computational workflow to refine predictions of node lifetimes. Gradient-based models, executed on the UAV's onboard compute module, enable real-time estimation of energy depletion patterns. These predictive insights guide mission planning and improve long-term system resilience.

4.2 Non-Functional Requirements

- Scalability

  The system must scale to support hundreds of MS nodes and tens of CHs without significant degradation in performance. This ensures applicability in large deployments and future expansions.

- Reliability

  High reliability is essential for effective operation. The system must achieve a data delivery success rate exceeding 99%, enabled by MQTT acknowledgement mechanisms that safeguard against data loss.

- Energy Efficiency

  The system must operate with strict energy efficiency constraints. MS nodes minimize idle listening to conserve energy, while UAV flights are scheduled adaptively to optimize mission success against available resources.

- Latency

  Data retrieval from CH to UAV must occur within 3–5 seconds of CH detection. This ensures timely offloading of data and responsiveness to critical events.

- Robustness

  The system must demonstrate robustness in the face of failures. It should recover gracefully from CH node failures as well as UAV mission aborts, thereby maintaining operational continuity.

- Maintainability

  The system is designed with modularity in mind, with MS, CH, UAV, and dashboard components developed as independent modules. This modular design supports independent upgrades, testing, and long-term maintainability.

- Security

  All data transfers must be authenticated and encrypted to prevent spoofing, tampering, or unauthorized interception. This ensures the confidentiality, integrity, and trustworthiness of collected data.

4.3 System Requirements

- Hardware Requirements
  - MS Node: ESP32, low-power sensors, 433 MHz receiver.
  - CH Node: ESP32-S3, SPIFFS flash, BLE 5.1, Taoglas antenna array.
  - UAV: DJI-class quadcopter with onboard computer (e.g., Raspberry Pi CM4 or Jetson Nano), dual radios (BLE 5.1 + Wi-Fi), and extended battery.
  - Operator Side: Laptop/server running dashboard + MQTT broker
- Software Requirements
  - Node firmware: ESP-IDF/Arduino framework.
  - UAV compute: Linux-based with Python/C++ mission control scripts.
  - ML models: Scikit-learn/TensorFlow Lite for regression and auditing models.
  - Dashboard: Web-based UI (React + spring boot backend).
  - Protocols: ESP-NOW, BLE 5.1, Wi-Fi, MQTT.
- Networking Requirements
  - MS ↔ CH: ESP-NOW (short-range, low-power).
  - CH ↔ UAV: BLE 5.1 (scanning/detection) + Wi-Fi MQTT (data offload).
  - UAV ↔ Dashboard: Telemetry Link.
  - Fault Tolerance: Store-and-forward model with retries and ACKs
- Energy Requirements
  - MS: <100 mW average consumption.
  - CH: Sustained operation >1 month on battery + solar backup.
  - UAV: Endurance ~30–40 minutes per flight; adaptive planning required.

4.4 Test Cases

- MS-to-CH Data Integrity Test
  - Deploy 10–20 MS nodes with sensors.
  - Validate slot synchronization, JSON structure correctness, and persistence after power cycles.
  - Threshold: >98% packet integrity, <2% synchronization error.
- CH Failure Recovery Test
  - Simulate CH power loss during active data aggregation.
  - Validate SPIFFS persistence and automatic recovery.
  - Threshold: No more than 5% data loss after reboot.
- UAV Scanning & Localization Test
  - Fly UAV over test field with multiple CHs.
  - Validate BLE 5.1 direction-finding accuracy (target: 10^{-5} error rate in predicted GPS positions).
  - Compare predicted CH positions vs. ground truth.
- CH-to-UAV Data Offload Test
  - UAV hovers within comms range of CHs.
  - Validate MQTT data offload with ACK.
  - Threshold: 100% acknowledgement delivery, <5 s transfer time per CH.
- UAV Mission Planning Test
  - Load UAV with route planner considering CH battery predictions.
  - Fly adaptive sweep and full sweep missions.
  - Metrics: flight time saved vs. naive sweep, % coverage of critical CHs.
- Aerial Health Auditing Test
  - UAV probes live CHs and record their responses.
  - Validate health dashboard flags “silent/failing” CHs within one mission cycle.
  - Threshold: >95% detection accuracy of dead/failing CHs.

- End-to-End System Integration Test
    - Deploy MS–CH network with UAV flights.
    - UAV collects, audits, and relays data to dashboard.
    - Operators visualize in dashboard with alerts.
    - Metrics: Overall delivery success >98%, mean latency <10 s from sensing to dashboard update.
- Stress & Scalability Test
    - Gradually increase MS nodes (up to 100) and CHs (up to 10).
    - Measure UAV scanning time, data integrity, dashboard responsiveness.
    - Threshold: System stable with <10% increase in latency under scale.

5 Description of Personal and Facilities

5.1 Work breakdown structure

Table 5-1: Work Breakdown Structure

Level 1	Level 2	Level 3 (Prototype Tasks)
1. System Design & Simulation	1.1 Literature review	Finalize design approach
	1.2 System diagrams	UAV–WSN–Dashboard architecture
	1.3 Simulation models	ROS/Gazebo test runs
	1.4 Test environment	Define lab/field test site
2. UAV Hardware Prototype	2.1 Frame assembly	S500 frame, motors, ESCs, propellers
	2.2 Flight controller	Install Pixhawk PX4
	2.3 Companion computer	Setup Raspberry Pi 4
	2.4 Power distribution	LiPo battery + UBEC regulator
	2.5 Flight testing	Manual + autonomous basic flight
3. Sensor Node Prototype (CH & MS)	3.1 ESP32 setup	Configure MS and CH roles
	3.2 Sensor integration	Attach temperature, humidity sensors
	3.3 MS → CH comms	Implement ESP-NOW
	3.4 CH → UAV comms	Implement MQTT over Wi-Fi
	3.5 Local storage	SPIFFS with JSON buffering
4. Localization & Mapping	4.1 BLE setup	Integrate BLE 5.1 antenna array
	4.2 Data collection	Collect IQ samples via ESP32
	4.3 Processing	Run MUSIC/ESPRIT on RPi
	4.4 Triangulation	UAV scanning from 2–3 positions
	4.5 Mapping	Basic real-time cluster map
5. Routing (Simplified)	5.1 Static routing	Pre-set waypoint path

	5.2 Priority list	Manual CH priority selection
	5.3 UAV sync	MAVLink integration with Pixhawk
	5.4 Field testing	Small-area adaptive routing test
6. Health Auditing (Minimal)	6.1 CH beaconing	Periodic BLE beacon broadcast
	6.2 UAV detection	Detect silent CH nodes
	6.3 Node status	Active / Silent classification
	6.4 Alerts	Dashboard alert for silent CHs
	6.5 Reporting	UAV → base station telemetry
7. Data Muling	7.1 MQTT broker	Run on UAV (Raspberry Pi)
	7.2 Data upload	CH sends JSON packets
	7.3 ACK mechanism	Confirm receipt before delete
	7.4 Temporary storage	Save on Raspberry Pi
	7.5 Forwarding	Send to base station
8. Dashboard (Prototype UI)	8.1 Backend	Local MQTT + PostgreSQL
	8.2 Frontend	Basic React UI
	8.3 Visualization	Show cluster positions (map markers)
	8.4 Alerts	Silent CH warnings
	8.5 Operator panel	Start/stop mission, select mode
9. Testing & Validation	9.1 Subsystem tests	UAV–CH, CH–MS, UAV–Dashboard
	9.2 Ground test	UAV hovering + data upload
	9.3 Flight test	UAV visits 2–3 CHs
	9.4 Metrics	Data delivery, localization error
	9.5 Refinement	Adjust prototype performance
10. Documentation & Finalization	10.1 Prototype design	Document build steps
	10.2 Evaluation	Results tables/graphs
	10.3 Final report	Dissertation write-up

5.2 Timeline

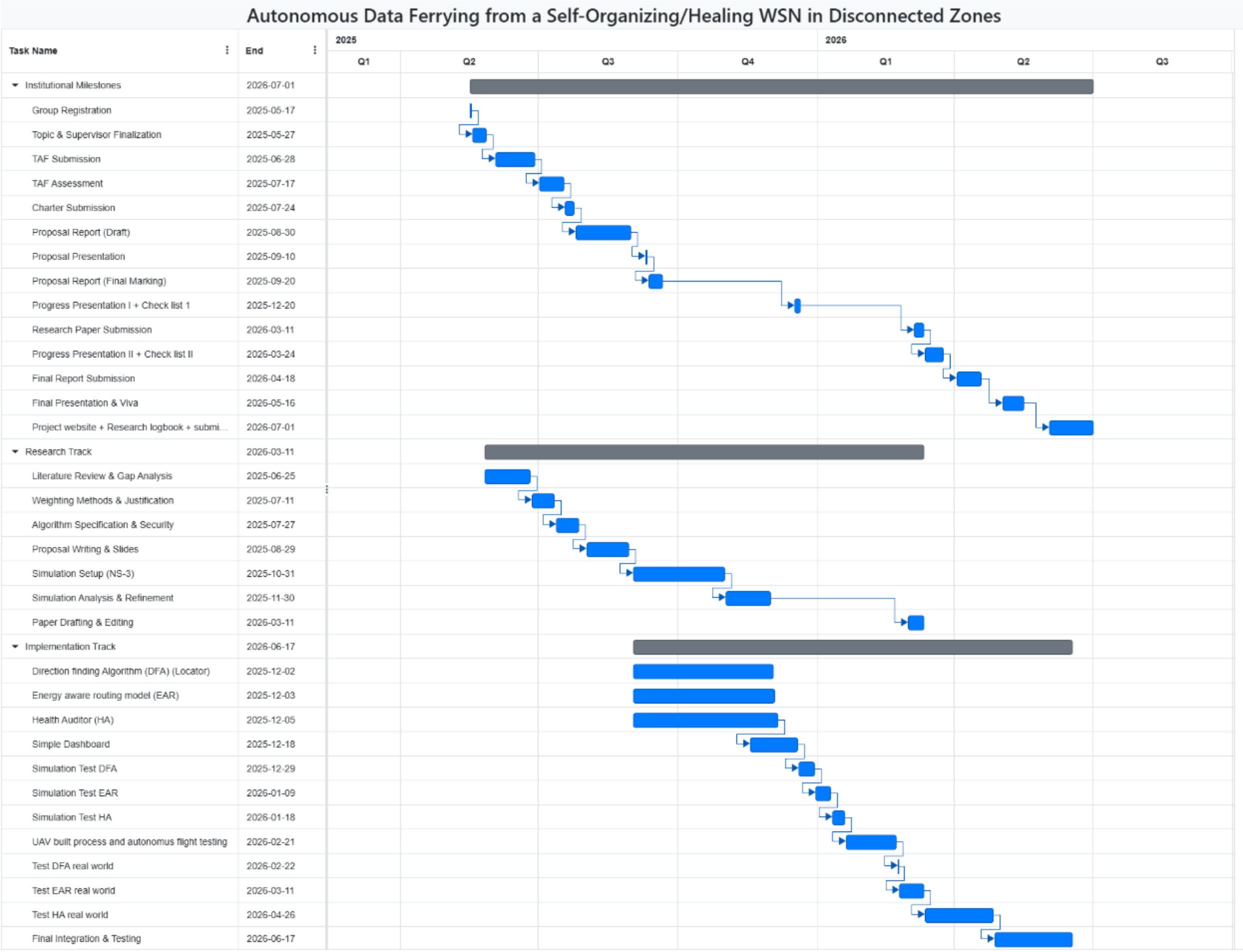

Figure 5.2.1: Autonomous Data Ferrying from a Self-Organizing Healing WSN in Disconnected Zones Gantt chart

5.3 System diagram – V1

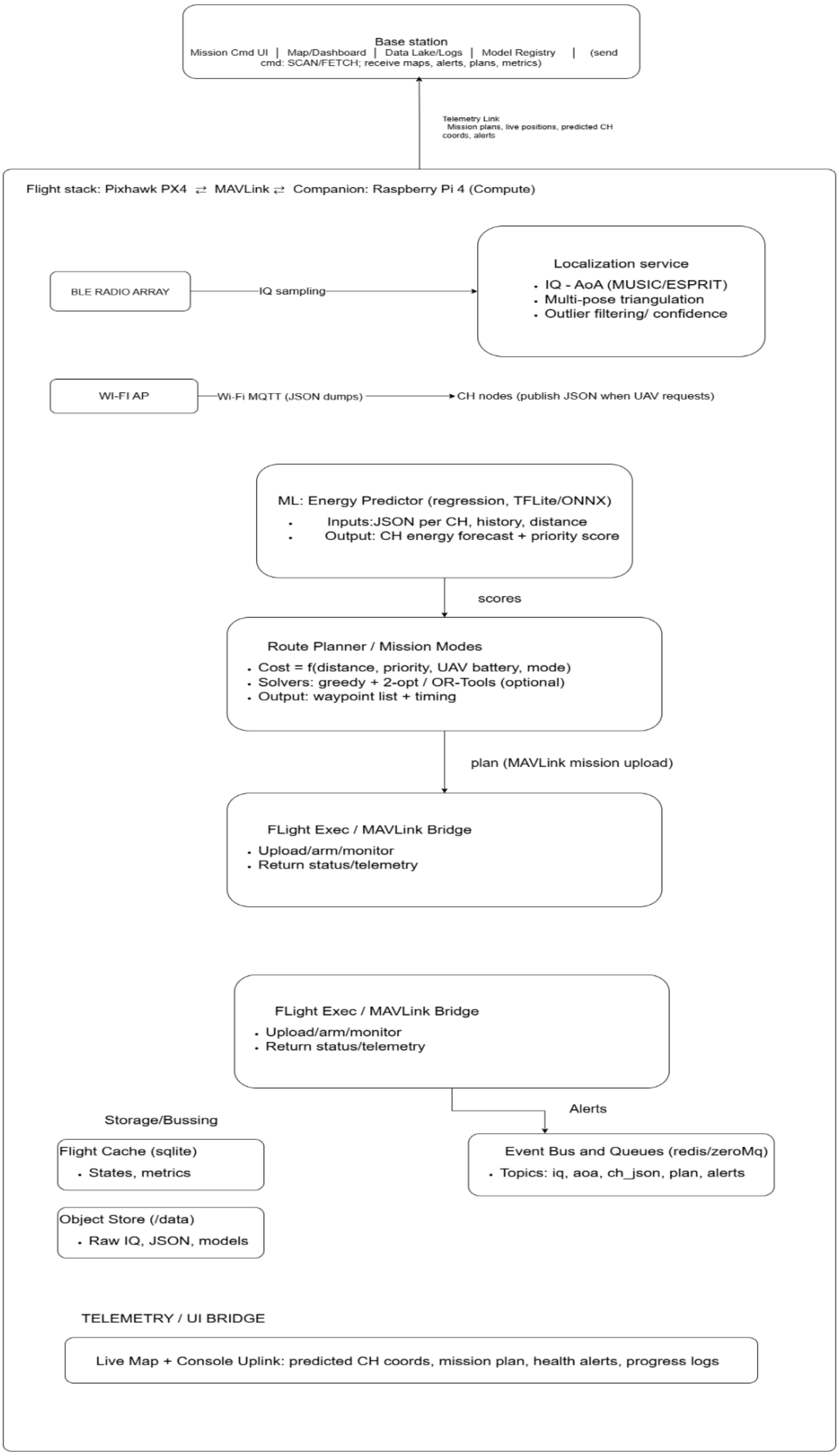

Figure 5.3.1: software system architecture diagram V1

6 Budget and Budget Justification

Table 6-1: Estimated Budget

Component	Prize
UAV + Raspberry pie 4 + Antenna setup + Datamuling components	120K ~60K
ESP32-S3-DevKitC-1U (8MB Flash, 8MB PSRAM) x 7	1,850.00
18650 Li-ion 3500 mAh battery (Samsung)	780
3x18650 Battery Holder (parallel wiring)	120
CN3065 Solar Li-ion Charger module	350
INA219 Current sensor (I^2C)	350
Mini Solar Panel 6V 1W (110x60mm)	350
SMA antenna + IPEX to SMA cable	690
ENS160 + AHT21 air quality sensor module x 6	7,100.00
INMP441 I^2S MEMS microphone module x 6	5040
HMC5883L 3-axis magnetometer module x 6	2700
BME280 pressure/temperature/humidity sensor x 6	5700
Total	145,030.00

7 References

[1] J. Luo, J. Hu, D. Wu, and R. Li, "Opportunities and challenges of UAV-enabled wireless sensor networks," *IEEE Network*, vol. 32, no. 4, pp. 193–199, Jul.–Aug. 2018.

[2] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proc. 33rd Annu. Hawaii Int. Conf. Syst. Sci.*, 2000, pp. 1–10.

[3] S. Pudlewski, T. Melodia, and A. Prasanna, "Compressed sensing for real-time energy-efficient wireless multimedia sensor networking," *IEEE Commun. Mag.*, vol. 50, no. 6, pp. 40–49, Jun. 2012.

[4] M. Younis and K. Akkaya, "Strategies and techniques for node clustering in wireless sensor networks: A survey," *Ad Hoc Netw.*, vol. 6, no. 4, pp. 621–648, Jun. 2008.

[5] I. Demirkol, C. Ersoy, and F. Alagöz, "MAC protocols for wireless sensor networks: A survey," *IEEE Commun. Mag.*, vol. 44, no. 4, pp. 115–121, Apr. 2006.

[6] Espressif Systems, "ESP32 technical reference manual," 2020. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf

[7] "ESP-NOW - ESP32 - — ESP-IDF Programming Guide v5.5 documentation," *Espressif.com*. [Online]. Available: https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/network/esp_now.html. [Accessed: 19-Aug-2025].

[8] Mdpi.com, "Low-Power Wake-Up Radio Techniques for UAV–WSN," *Sensors*, vol. 24, no. 22, p. 7170, 2024. [Online]. Available: https://www.mdpi.com/1424-8220/24/22/7170. [Accessed: 19-Aug-2025].

[9] C. Hernández-Goya, R. Aguasca-Colomo, and C. Caballero-Gil, "BLE-based secure tracking system proposal," *Wirel. Netw.*, vol. 30, no. 6, pp. 5759–5770, 2024.

[10] "MQTT - the standard for IoT messaging," *Mqtt.org*. [Online]. Available: https://mqtt.org/. [Accessed: 19-Aug-2025].

[11] "MQTT V3.1 protocol specification," *IBM.com*. [Online]. Available: https://public.dhe.ibm.com/software/dw/webservices/ws-mqtt/mqtt-v3r1.html.

[12] ResearchGate/USENIX/ACM DL, "High-reliability concurrent flooding (Glossy/Splash/COFlood) for NLOS bursts," 2018.

[15] S. A. Shah et al., “Collaborative UAV–WSN Systems for Civilian Applications: A Survey,” *Sensors*, vol. 19, no. 18, p. 4015, 2019.

[16] J. Caballero et al., “Mobile Robot SLAM-Assisted Localization of WSN Nodes,” *IEEE Trans. Mobile Comput.*, vol. 7, no. 9, pp. 1111–1125, 2008.

[17] J. Ermis et al., “Priority-Based Data Collection for UAV-Aided Mobile Wireless Sensor Networks,” *Sensors*, vol. 20, no. 7, p. 2014, 2020.

[18] S. Chowdhury et al., “Data Acquisition Control for UAV-Enabled Wireless Rechargeable Sensor Networks,” *Drones*, vol. 7, no. 1, p. 34, 2023.

[19] A. Ramadan et al., “A Comprehensive Survey on Wireless Sensor Network Fault Diagnosis,” *Information Fusion*, vol. 55, pp. 60–79, 2019.

[20] N. Varghese et al., “Fault Detection and Recovery in UAV-Assisted Sensor Networks,” *Computer Networks*, vol. 237, p. 110013, 2024.

[21] Network visualization/dashboard tools in UAV–WSN systems. *ResearchGate/Elsevier/IEEE Access*, 2024.

[22] BLE 5.1 Direction Finding (AoA/AoD) – Nordic Semiconductor Application Note. *Nordicsemi.com*, 2021.

[23] Experimental evaluation of BLE 5.1 AoA for indoor positioning. *IEEE Access*, 2022.