<u>Smart Waste Bin Network (Virtual IoT Design Challenge)</u>

Submitted by: Pratik Tiwari

1. System Architecture:

This IoT system employs an end-to-end framework designed to monitor fill levels and optimize logistics across city zones.

- **Edge Hardware:** Each bin is equipped with an **HC-SR04 Ultrasonic Sensor** for level detection and an **ESP32 microcontroller** with an integrated **LoRa module**.
- **Data Communication:** We will use **LoRaWAN** (Long Range Wide Area Network). This is chosen over Wi-Fi for its superior range in urban environments and significantly lower power consumption.
- **Cloud Strategy:** Data is processed at the edge (the bin) to filter noise before being sent via a LoRa Gateway to a **Central Cloud Server** (e.g., AWS IoT Core) for long-term storage and analytics.
- **Dashboard:** A web-based **GIS Dashboard** will visualize bin statuses (Green/Yellow/Red) and provide real-time location mapping for city authorities.

2. Data Flow Design:

The data movement is structured to ensure minimal latency and high reliability.

- **Protocol:** MQTT (Message Queuing Telemetry Transport).
- **Justification:** MQTT is ideal because it is a lightweight, publish-subscribe protocol that works efficiently even with low bandwidth and unstable network connections.
- **Flow Steps:**
    1. **Sensor** measures distance.
    2. **Microcontroller** calculates percentage full.
    3. **ESP32** publishes a JSON payload to the MQTT broker.
    4. **Cloud Application** subscribes to the topic and stores data in a database.
    5. **Dashboard** fetches data via API for visualization.

3. Route Optimization Strategy:

A. ***<u>Decision Logic</u>***

Priority Level	Fill Level	Action Required
High	**>80%**	Must Collect: Add to today's route immediately.
Medium	**50%−80%**	Optional: Collect only if the bin is on the way to a "High" priority bin.
Low	**<50%**	Skip: Do not visit; plenty of capacity remaining.

B. ***Routing Logic (The "How")***

Once the bins are prioritized, use the **Greedy Nearest Neighbor** logic for the truck's path:

1. **Start** at the Garbage Depot.
2. **Identify** all bins in the "High" priority category.
3. **Find** the "High" priority bin closest to the truck's current location.
4. **Check** for any "Medium" priority bins that are physically located between the current position and that "High" priority bin. If it adds less than 5 minutes to the trip, collect it too.
5. **Move** to that bin, empty it, and repeat the process until all "High" priority bins are cleared.
6. **Return** to the Depot.

C. ***Logic Flowchart & Pseudocode***

pseudocode.c

```

bins_to_collect = []

for bin in city_zones:

    level = bin.get_fill_level() //Detect and report fill level

    if level >= 80:
        bin.status = "CRITICAL"
        bins_to_collect.append(bin)
    elif level >= 50:
        bin.status = "STALE" # Collect if convenient
    else:
        bin.status = "OK"

// Sort by distance from current truck position
optimized_route = sort_by_nearest_neighbor(current_location, bins_to_collect)

//Notify authorities of the suggested route [cite: 14, 15]
send_to_dashboard(optimized_route)
```

D. ***Why this works?***

- **Operational Efficiency:** It prevents "dry runs" where trucks visit half-empty bins, reducing fuel costs.
- **Hygiene:** It prioritizes bins that are nearly overflowing to prevent poor urban hygiene.
- **Feasibility:** This logic is easy to implement in standard coding languages (Python/C++).

4. Power Management Plan

To maximize battery life for bins in remote locations, we propose the following:

- **Periodic Sensing:** The sensor only activates once every 30 minutes rather than continuous monitoring.

- **Deep Sleep Mode:** The ESP32 stays in a ultra-low-power "Deep Sleep" state between transmissions.
- **Event-Driven Transmission:** If the fill level remains unchanged, the system sends a "heartbeat" only once every few hours to save energy.

5. Reliability & Fault Handling

Accuracy is critical for preventing unnecessary truck deployments.

- **Median Filtering:** To handle false readings (e.g., a plastic bag momentarily blocking the sensor), the system takes five readings and uses the **median value** to eliminate outliers.
- **Redundancy:** A secondary PIR (Passive Infrared) sensor or a Tilt sensor can be used to verify if the bin has actually been used or tipped over.
- **Calibration:** The system performs an auto-calibration "zero-check" when the bin is empty to adjust for debris at the bottom of the bin.

6. Scalability & Network Considerations

The system is designed to scale to 100+ bins across multiple zones.

- **Topology:** Star-of-Stars Topology
  Why: Each bin connects to the nearest LoRa Gateway (Star). Multiple Gateways then connect to the central cloud. This ensures that adding more bins doesn't overwhelm a single point of entry and allows for easy expansion into new city zones.

7. Cost & Feasibility Discussion

Component	Approximate Price (INR)
Microcontroller: ESP32 with Integrated LoRa Module	₹1,800 – ₹2,200
Sensor: Ultrasonic Distance Sensor (HC-SR04)	₹60 – ₹100
Power: Li-ion 18650 Battery (2500mAh) + BMS	₹300 – ₹500
Enclosure: IP65 Waterproof Industrial Housing	₹400 – ₹800

Miscellaneous: Antennas, Jumper Wires, Mounting Brackets	₹200 – ₹400
Total Estimated Cost per Unit	**₹2,760 – ₹4,000**

Prices are based on retail rates; bulk procurement for 100+ bins would likely reduce costs by 20-30%.

Trade-offs & Feasibility:

- **Cost vs. Accuracy:** We chose Ultrasonic sensors over high-end LiDAR. While LiDAR offers millimeter precision, it costs upwards of ₹10,000 per unit. Ultrasonic sensors provide sufficient accuracy for "fill-level" detection at a fraction of the price, ensuring the project remains financially viable for a city-wide rollout.
- **Connectivity (LoRaWAN vs. NB-IoT):**
  - **LoRaWAN:** High initial setup cost (Private Gateways needed), but **zero monthly data charges** and superior battery life (5+ years).
  - **NB-IoT:** Zero initial infrastructure cost (uses existing 4G/5G towers), but requires **recurring monthly SIM rentals** and consumes more power.
  - **Decision***:* We propose LoRaWAN for maximum long-term cost-efficiency.
- **Scalability:** By using a **Star Topology**, we can add hundreds of bins to a single LoRa Gateway. This "pay-as-you-grow" model allows city authorities to start with one ward and scale to the entire city without changing the core software architecture.