Smart Waste Bin Network: Virtual IoT Design Challenge

Conceptual Design Document:

Submitted By:

Ane Praveen Prasad

Guru Nanak Institutions Technical Campus

praveenprasadane75@gmail.com

+9193938605543

Date: December 25,2025

For Internship Opportunity at IIITH Smart City Living Lab

Table of Contents

Contents

1. System Architecture 3

2.Data Flow Design 5

3. Route Optimization Strategy 6

4. Power Management Plan 8

5. Reliability & Fault Handling 9

6. Scalability & Network Considerations 10

7. Cost & Feasibility Discussion 11

Reference 11

1. System Architecture

I decided to go with ultrasonic sensors (like HC-SR04) paired with ESP32 microcontrollers for each bin node. Ultrasonic is simple and good enough for measuring fill levels from the top, and ESP32 has built-in Wi-Fi/Bluetooth plus support for LoRa if needed.

For communication, I'm using LoRa (with modules like SX1276) because it gives long range with very low power – perfect for spreading bins across city zones without needing Wi-Fi everywhere or paying for cellular. Data goes from bins to a few LoRa gateways, then to the cloud.

Cloud side: I'll use something like AWS IoT or Thingsboard for storing data and processing. No heavy edge computing on the bins themselves to keep them cheap and battery-friendly.

Dashboard: A simple web app (maybe Node-RED or Grafana) showing real-time fill levels on a map, alerts for full bins, and route suggestions for the collection team.

Overall, it's end-to-end: sensor → ESP32 → LoRa → gateway → cloud → dashboard/app.

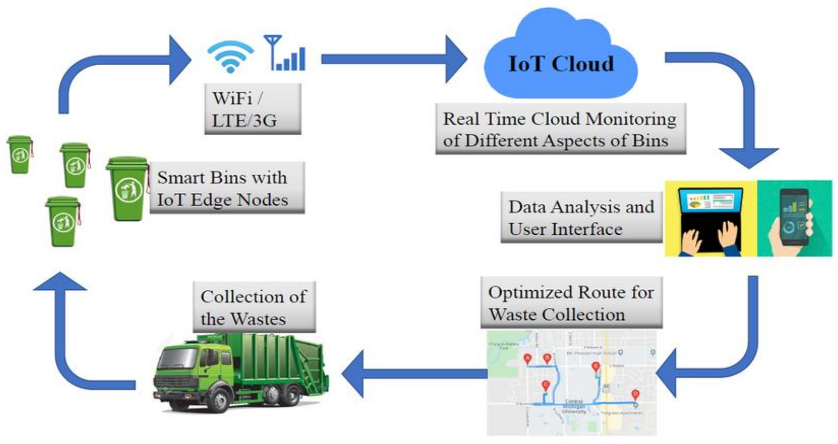

WiFi / LTE/3G
IoT Cloud
Real Time Cloud Monitoring of Different Aspects of Bins
Smart Bins with IoT Edge Nodes
Data Analysis and User Interface
Collection of the Wastes
Optimized Route for Waste Collection

Fig 1.1: The proposed system architecture

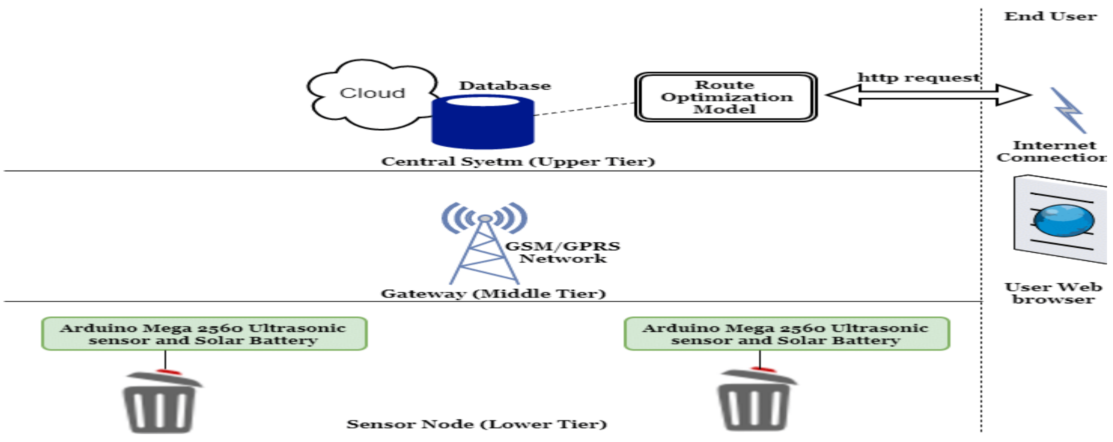


Fig 1.2: Smart bin and Central system architecture

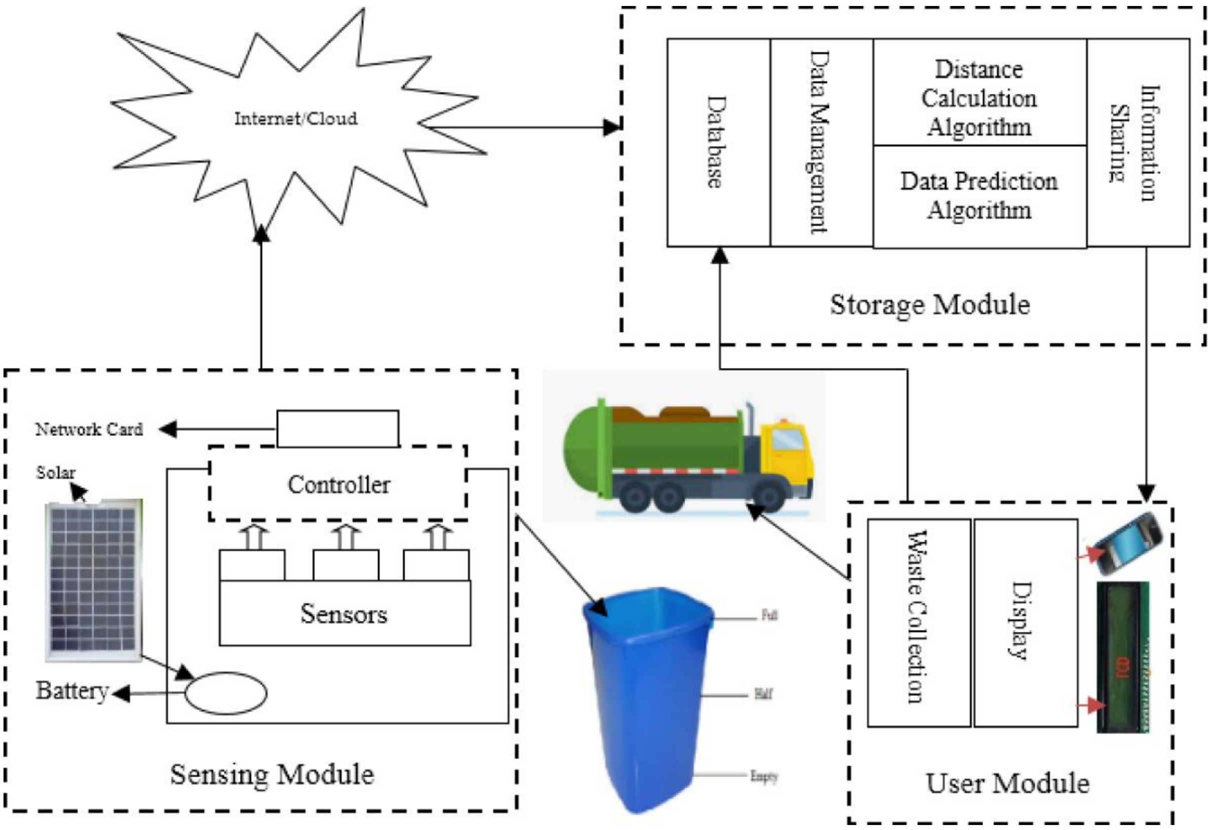


Fig 1.3

Fig1: Simple System IOT Based Smart Waste Bin Monitoring and Municipal Solid Waste Management Architecture

Fig 2: Data Flow

2.Data Flow Design

Data starts at the bin: ultrasonic sensor pings every few hours, calculates fill percentage, adds bin ID and timestamp.

ESP32 packs it into a small payload and sends via LoRa to the nearest gateway.

Gateway forwards to cloud over internet (Wi-Fi or Ethernet).

In cloud, data lands via MQTT protocol – I picked MQTT because it's lightweight, reliable for IoT, supports publish/subscribe so multiple services can listen (like alerts and dashboard), and works great over LoRaWAN setups.

Why MQTT? It's better than HTTP for frequent small updates (less overhead), and way more efficient than raw TCP for battery-powered nodes.

From cloud, processed data goes to dashboard for visualization and to a mobile app for drivers/authorities.

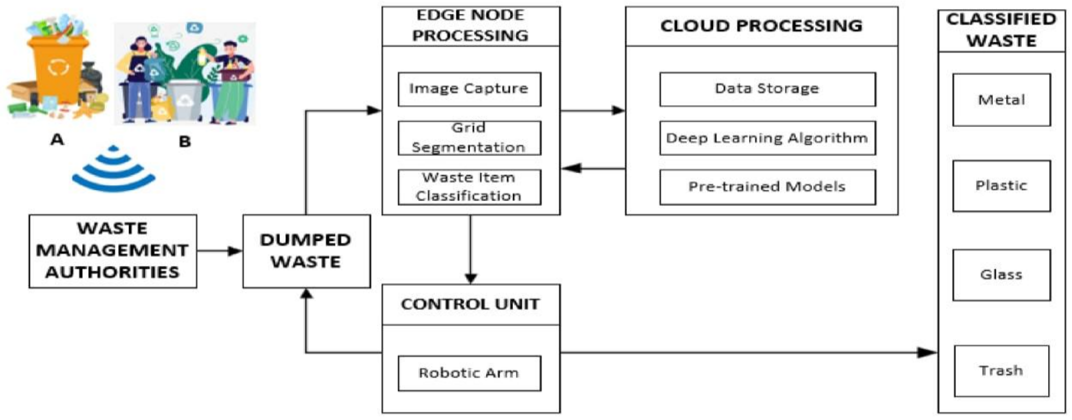


Fig 2.1

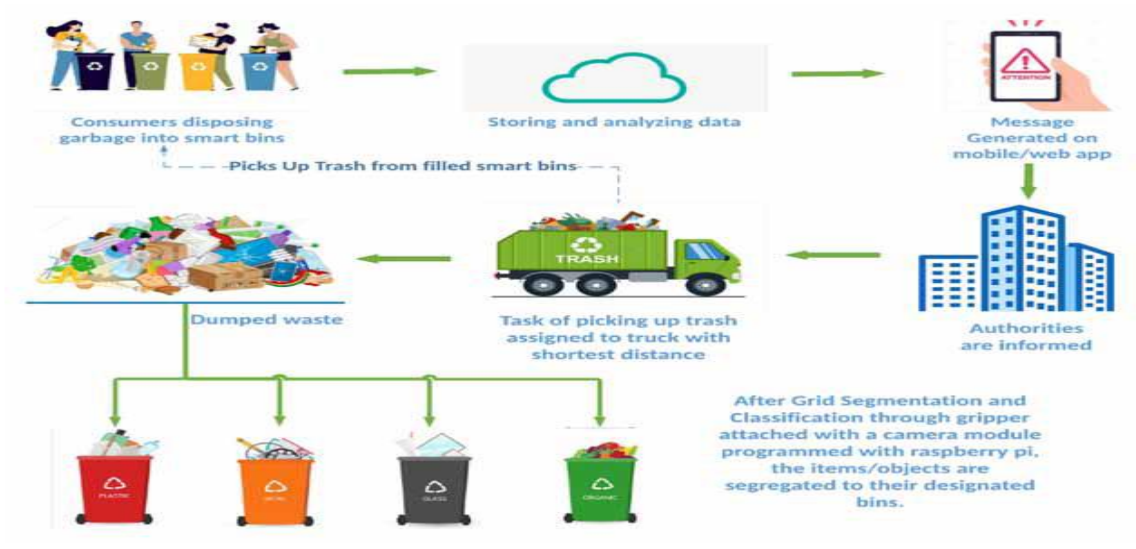

Consumers disposing garbage into smart bins
Storing and analyzing data
Message Generated on mobile/web app
Picks Up Trash from filled smart bins
Dumped waste
TRASH
Task of picking up trash assigned to truck with shortest distance
Authorities are informed
After Grid Segmentation and Classification through gripper attached with a camera module programmed with raspberry pi, the items/objects are segregated to their designated bins.

Fig 2.2

3. Route Optimization Strategy

Basic rule: Collect any bin over 80% full, or if it's been >3 days since last empty.

For smarter routing: Group full bins by zone (clustering based on location), then prioritize by fill level (highest first) and distance.

Algorithm: Simple nearest-neighbor approach – start from depot, go to closest full bin, then next closest, until truck capacity is reached. Repeat for multiple trucks.

If possible, add basic shortest path (like Google Maps API) between points.

Pseudocode idea:

```
full_bins = list of bins >80%
sort full_bins by fill_level descending
clusters = group by city zone
for each cluster:
    route = [depot]
    current = depot
    while bins left and truck not full:
        next_bin = closest to current in cluster
        add next_bin to route
        current = next_bin
    route += [depot]
send route to driver app
```

This saves fuel by avoiding empty bins and long detours.

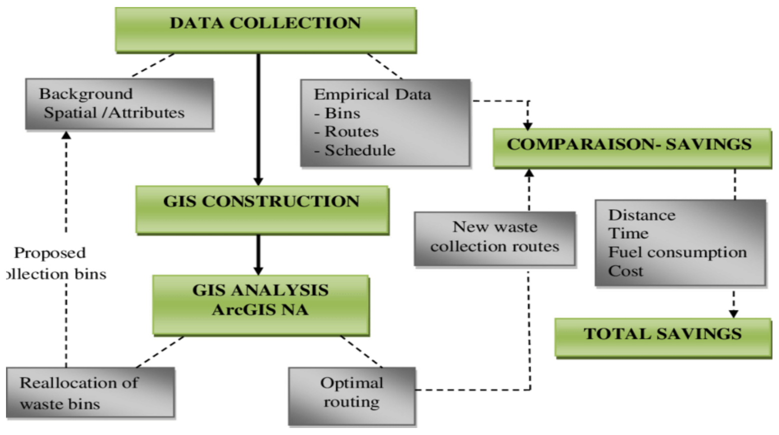

DATA COLLECTION
Background Spatial /Attributes
Empirical Data - Bins - Routes - Schedule
COMPARAISON- SAVINGS
GIS CONSTRUCTION
New waste collection routes
Distance Time Fuel consumption Cost
Proposed ollection bins
GIS ANALYSIS ArcGIS NA
TOTAL SAVINGS
Reallocation of waste bins
Optimal routing

Fig 3.1: Flow Chart of data analysis steps

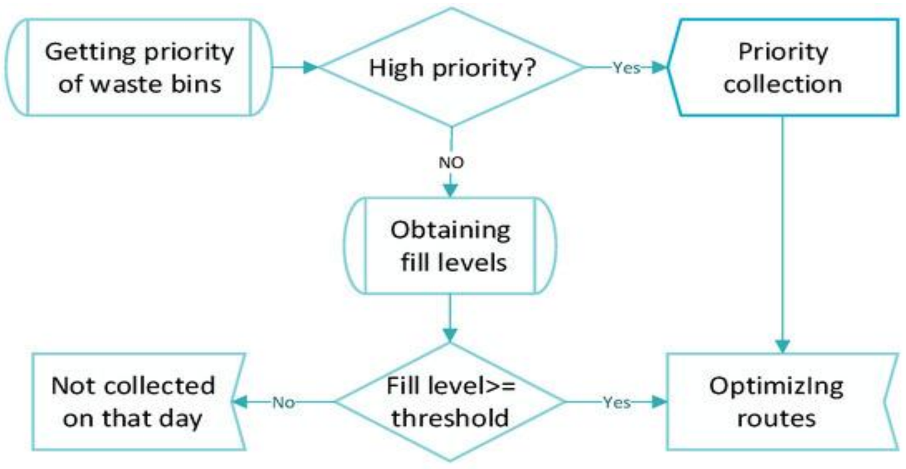


Fig 3.2: Optimization of vehicles routing for waste collection management

Fig 3: Route FlowChart

4. Power Management Plan

- Deep sleep on ESP32 (wake every 30 min).
- Solar panel (5W) + Li-ion battery.
- Transmit only on significant change.
- Bins battery-powered: Use deep sleep, periodic sensing, solar recharge.

The bins run on batteries, so we need to save power.

I use deep sleep mode on the ESP32. It sleeps most of the time and wakes up only every 30 minutes to check the fill level.

I also send data only when the bin fill changes a lot (like 10% more full). This saves battery.

I add a small solar panel on the bin lid to charge the battery from sunlight.

I think solar is good because it makes the bin work longer in places without electricity.

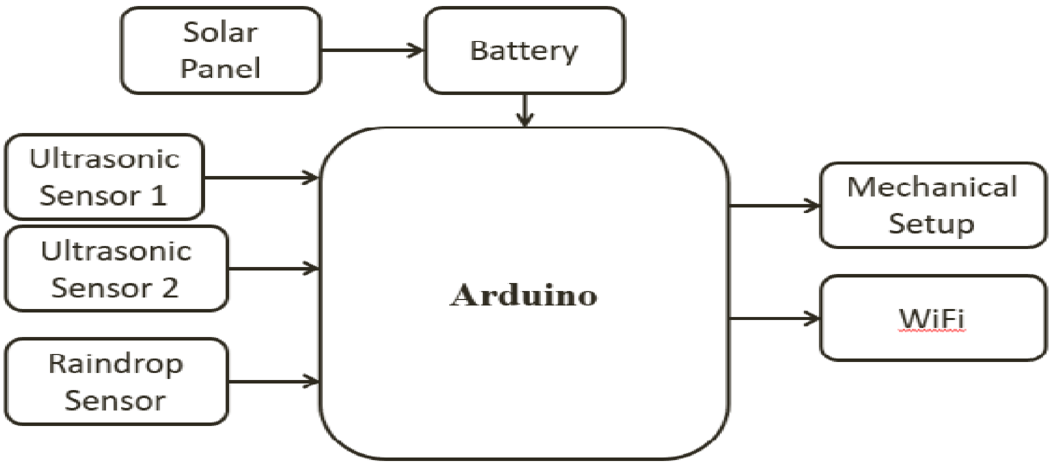


Fig 4.1: Solar Power for Waste management collection

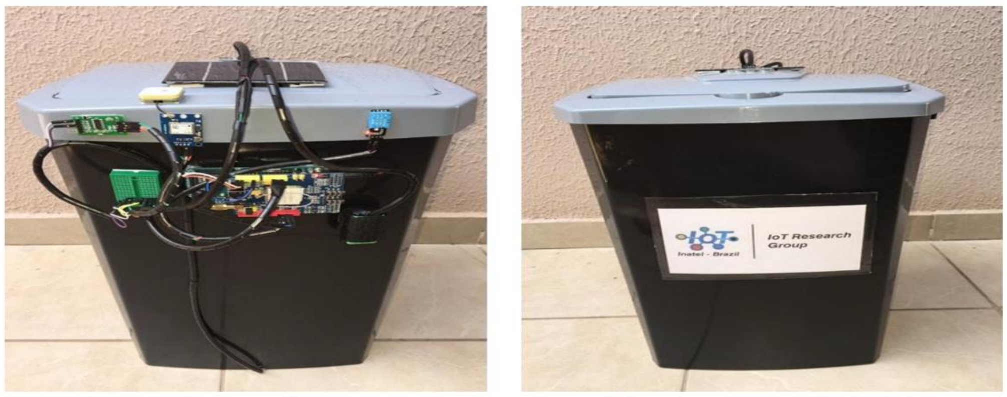


Fig 4.2: A smart waste management solution geared towards citizens

Fig 4: Different power management plans

5. Reliability & Fault Handling

Ultrasonic can give wrong readings if something hangs inside or rain/snow blocks it.

Handling:

Take multiple readings (say 5) and average/median to filter noise.

If reading jumps wildly, flag as possible error and alert for manual check.

Add a simple tilt sensor to detect if bin is tipped over.

Redundancy: Maybe pair with a basic weight sensor later, but for now keep it simple.

Periodic calibration: Send a "heartbeat" daily so we know the node is alive.

If no data for 48 hours, mark as faulty on dashboard.

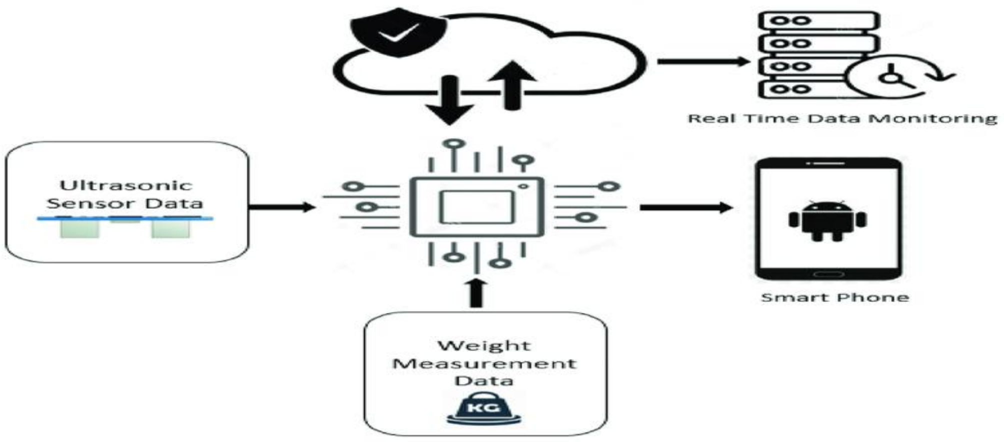
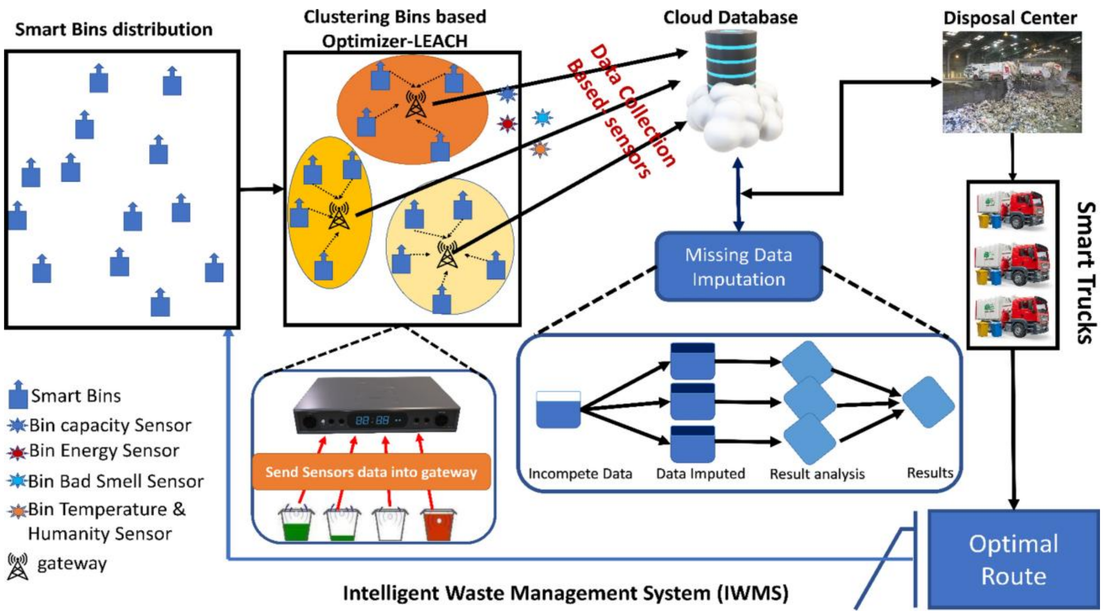


Fig 5.1: A block diagram and working principle of smart trash box



6. Scalability & Network Considerations

- LoRaWAN supports 1000+ nodes/gateway.
- Mesh topology for better coverage in dense areas.

The system should work for 100 or more bins in different city areas.

I use LoRaWAN network. It can connect many bins to one gateway.

I choose star topology because it is simple and uses less power and each bin talks directly to gateway. Simpler, no relay battery drain, and LoRa range is 5-10km in urban areas.

With LoRaWAN, one gateway can handle hundreds of nodes easily since transmissions are rare.

Scales well – add more gateways for bigger areas, cloud handles the data no problem.

But for better coverage in big cities, I can add some mesh (bins help send data for others).

Mesh is fault-tolerant because if one path fails, data goes another way.

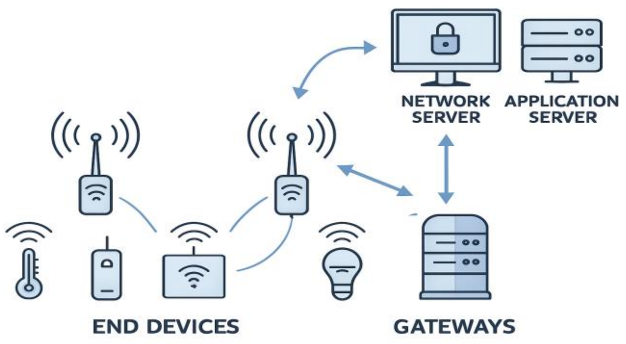


Fig 6.1: Complete guide to LoRaWAn Architecture for IOT

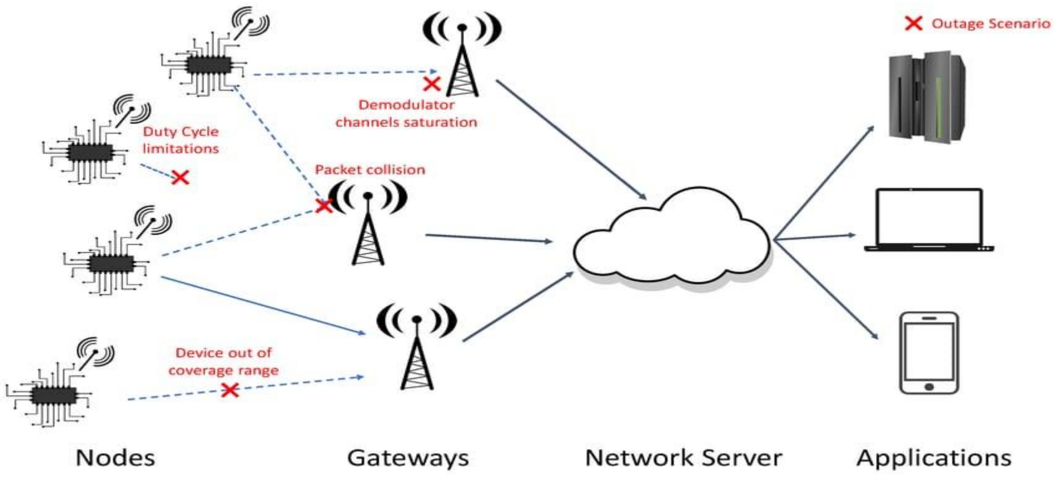


Fig 6.2: LoRaWAN star-of-stars network topology

Fig 6: Network Topology

7. Cost & Feasibility Discussion

- Per bin: ~₹4,000-5,000 (2025 prices).
- Trade-offs: Higher accuracy increases cost but saves 30% on collections (like Barcelona 2025 pilots).
- Highly feasible with proven LoRa deployments

Estimating costs conceptually for 2025 market prices in India, a single smart bin unit (ESP32 microcontroller ~₹600-800, HC-SR04 ultrasonic sensor ~₹200, LoRa module ~₹1,500, HX711 weight sensor ~₹300, battery/solar panel ~₹800, enclosure and misc ~₹1,000) comes to approximately ₹4,500-5,500 per bin. For a pilot with 100 bins, total hardware cost would be around ₹4.5-5.5 lakhs, excluding gateways and cloud setup.

Trade-offs: Higher accuracy from dual sensors and mesh networking increases initial cost by 20-30% but improves reliability and reduces operational expenses (e.g., fewer unnecessary truck trips, 25-40% savings as seen in 2025 pilots). Scalability is excellent with LoRaWAN, but requires planning for gateway placement. Overall, the system is highly feasible, aligning with India's Swachh Bharat initiatives and global trends, offering quick ROI through reduced fuel/labor costs and better hygiene.

LoRa is cheap and long-range but slower data rate (fine for us). Wi-Fi cheaper per node but needs power and coverage everywhere. NB-IoT great coverage but ongoing SIM fees.

This setup is feasible – low cost, easy to deploy, accurate enough for fill levels, and scales without huge bills. Biggest win is saving on truck fuel by smarter routes.

Reference

1. ResearchGate & MDPI papers on LoRaWAN smart waste systems (architecture, data flow).
2. SCS Engineers & NextBillion.ai for route optimization examples.
3. WasteHero & Binimise dashboard inspirations.
4. Arduino/Hackster.io prototypes for ESP32 + LoRa + ultrasonic hardware.

GitHub Repository:

https://github.com/Praveenprasadane/WM-AnePraveenPrasad-GuruNanakInstitutionsTechnicalCampus