Smart Parking System using LoRaWAN and Edge Computing: A Computing Continuum Approach

Hoza Violeta Maria

1 Introduction

Urbanization has significantly increased parking demand, contributing to traffic congestion and pollution as drivers search for available parking spaces [1]. Intelligent parking systems aim to address these issues by leveraging Internet of Things (IoT) technologies to monitor and manage parking space availability in real time [2].

Modern smart parking systems can significantly benefit from the computing continuum paradigm, which integrates computational resources across the device, edge, and cloud layers. This approach optimizes latency, energy consumption, and scalability by allocating tasks to the most suitable layer based on resource availability and application requirements [3].

This project focuses on designing a smart parking system that combines long-range, low-power communication protocols such as LoRaWAN (Long Range Wide Area Network), edge computing for local data processing, and cloud services for data storage and user interaction, all within a computing continuum framework that makes it possible to build a cost-effective, real-time parking availability system. This approach allows quick decision making at the device and edge layers, reducing cloud dependency, latency, and network load while enhancing responsiveness and scalability for urban smart city applications.

This report outlines the proposed system architecture, communication protocols, software tools for development, and a detailed implementation plan.

2 Proposed Solution

To address the limitations of cloud-centric parking solutions, this project proposes a smart parking solution that strategically distributes computation across multiple layers — namely, the device, edge, and cloud layers — leveraging the computing continuum paradigm. This layered approach not only reduces latency and bandwidth consumption but also enhances scalability and energy efficiency.

- **Device layer:** Parking sensors equipped with microcontrollers perform preliminary preprocessing, such as noise filtering and event debouncing, directly at the source. This reduces unnecessary transmissions and conserves battery life.
- **Edge layer:** LoRaWAN gateways serve as intermediaries that aggregate data, perform real-time analytics like occupancy detection and demand prediction, and execute local decision-making, providing immediate parking guidance. The proximity to the data source enables rapid responses and lowers reliance on cloud services.

- **Cloud layer:** The cloud provides comprehensive data storage, advanced analytics, and machine learning services for city-wide coordination, long-term trend analysis, and integration with user applications such as reservation and payment platforms.

Communication between layers is realized using LoRaWAN for efficient, long-range sensor-to-edge transmission and MQTT for low-overhead messaging between edge gateways and cloud services. Together, these technologies facilitate an energy-efficient, reliable, and scalable network infrastructure.

A continuum orchestration mechanism is incorporated to intelligently manage task allocation across layers based on system parameters such as latency sensitivity, network bandwidth, and computational capacity. This policy-driven orchestration dynamically adapts the workload distribution, improving system responsiveness and resilience under varying urban traffic and network conditions.

2.1 System Architecture

Components are organized as follows:

- **Parking Sensors (Device layer):** Deployed in parking spots to detect vehicle presence using ultrasonic or magnetic sensor [4] . Connected via LoRaWAN for long-range, low-power communication. Perform local preprocessing to filter redundant data.

- **LoRaWAN Gateway (Edge layer):** Collects sensor data and performs event filtering, data aggregation, and real-time analytics locally, forwarding only relevant information to the cloud using MQTT.

- **Cloud Platform (Cloud layer):** Hosts the central dashboard, stores historical data, and provides APIs (RESTful) for applications.

- **User Application:** Displays available parking spots in real time and supports optional reservation or payment functionality.

2.2 Workflow

1. Sensors detect vehicle presence and preprocess data locally to eliminate noise and redundant signals before transmitting occupancy data to the gateway via LoRaWAN, reducing unnecessary network traffic.

2. The edge nodes process data locally (removing redundant messages and applying basic logic), implementing edge computing principles to reduce cloud dependency [5].

3. Only relevant updates are sent to the cloud using MQTT (which is recognized as an efficient protocol for IoT applications [6]), minimizing bandwidth usage and reducing latency for local operations.

4. The cloud handles city-wide analytics, long-term storage, and integration with external services and user applications.

5. End users access real-time parking data via a mobile or web application connected through secure REST APIs.

The workflow ensures efficient resource utilization, fast reaction times for drivers, and scalability to city-wide deployments while minimizing energy consumption and network load.

3 Software Tools for Implementation

For developing, simulating, and testing the proposed system, the following tools can be used:

- **Eclipse Mosquitto:** An open-source MQTT broker for lightweight and reliable message delivery between edge nodes and the cloud, supporting the edge-cloud communication layer.
- **Node-RED:** A visual programming tool for integrating sensors, gateways, and cloud APIs. Supports MQTT and rapid prototype development for continuum orchestration logic.
- **ChirpStack:** An open-source LoRaWAN Network Server used to manage LoRa communication and simulate device-edge layer interactions.
- **Cooja/Contiki-NG:** IoT network simulators to validate the communication performance of LoRaWAN and MQTT under constrained conditions and evaluate continuum task distribution.
- **The Things Network (TTN):** A community-driven global LoRaWAN server for prototyping and testing IoT applications across the computing continuum.
- **Grafana:** Dashboard visualization for real-time parking data and continuum performance monitoring across all layers.
- **Flutter or React Native:** Frameworks for developing responsive, cross-platform mobile applications to display parking data and handle user interactions.
- **Docker/Kubernetes:** For containerized deployment and orchestration of edge and cloud services, supporting dynamic scaling based on continuum requirements.

4 Implementation Plan

1. **System Architecture Design** — Define the integration of LoRaWAN, MQTT, and edge logic with clear task distribution across device, edge, and cloud layers. Establish orchestration policies for dynamic task allocation.
2. **Device Layer Implementation** — Develop sensor data preprocessing algorithms for noise filtering and data reduction. Generate synthetic parking sensor data simulating vehicle presence and absence.
3. **Edge Logic Development** — Implement filtering, aggregation, real-time analytics, and local decision-making algorithms at the edge to reduce cloud traffic and improve responsiveness.

4. **Continuum Orchestration Setup:** Develop policy-based orchestration mechanisms that dynamically allocate tasks based on network conditions, latency requirements, and resource availability.

5. **MQTT Communication Setup** — Configure the Eclipse Mosquitto broker for secure edge-cloud message transmission with QoS levels appropriate for different data types.

6. **User Interface Development** — Build dashboard and mobile app prototypes for parking visualization and interaction.

7. **Testing and Evaluation** — Conduct network simulations with Cooja/Contiki-NG, and measure performance metrics (latency, bandwidth usage, energy consumption) across all continuum layers, and evaluate orchestration effectiveness.

8. **Security Implementation** — Integrate TLS for MQTT and verify LoRaWAN encryption for secure communication.

5 Conclusion

This project proposes a smart parking system leveraging the computing continuum, with careful distribution of computation across devices, edge nodes, and cloud infrastructure. The system integrates LoRaWAN for energy-efficient, long-range communication, MQTT for lightweight messaging, and a continuum orchestration mechanism to dynamically manage workload allocation based on real-time conditions.

This approach addresses key challenges in smart parking including latency, bandwidth usage, energy consumption, and system scalability. Use of open-source tools enables extensive simulation and prototyping without immediate hardware deployment.

Future directions include implementing real-world prototypes, integration of AI-driven analytics for dynamic demand prediction, enhanced security, and exploring scalability in larger smart city deployments.

References

[1] V. K. Sarker, T. N. Gia, I. Ben Dhaou, and T. Westerlund, "Smart parking system with dynamic pricing, edge-cloud computing and lora," *Sensors*, vol. 20, no. 17, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/17/4669

[2] W. A. Jabbar, L. Y. Tiew, and N. Y. Ali Shah, "Internet of things enabled parking management system using long range wide area network for smart city," *Internet of Things and Cyber-Physical Systems*, vol. 4, pp. 82–98, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2667345223000494

[3] A. Al-Dulaimy, M. Jansen, B. Johansson, A. Trivedi, A. Iosup, M. Ashjaei, A. Galletta, D. Kimovski, R. Prodan, K. Tserpes, G. Kousiouris, C. Giannakos, I. Brandic, N. Ali, A. B. Bondi, and A. V. Papadopoulos, "The computing continuum: From iot to the cloud," *Internet of Things*, vol. 27, p. 101272, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2542660524002130

[4] J. J. Barriga, J. Sulca, J. León, A. Ulloa, D. Portero, J. García, and S. G. Yoo, “A smart parking solution architecture based on lorawan and kubernetes,” *Applied Sciences*, vol. 10, no. 13, 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/13/4674

[5] W. Yu, F. Liang, X. He, W. G. Hatcher, C. Lu, J. Lin, and X. Yang, “A survey on the edge computing for the internet of things,” *IEEE Access*, vol. 6, pp. 6900–6919, 2018. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8123913

[6] D. Soni and A. Makwana, “A survey on mqtt: a protocol of internet of things (iot),” in *International conference on telecommunication, power analysis and computing techniques (ICTPACT-2017)*, vol. 20, no. April, 2017, 2017. [Online]. Available: https://www.researchgate.net/profile/Dipa-Soni/publication/316018571_A_SURVEY_ON_MQTT_A_PROTOCOL_OF_INTERNET_OF_THINGSIOT/links/58edafd4aca2724f0a26e0bf/A-SURVEY-ON-MQTT-A-PROTOCOL-OF-INTERNET-OF-THINGSIOT.pdf