

Cloud-Integrated Smart Traffic Control for Ambulance Prioritization Using IoT and Modular Microcontrollers

Nishad Amonkar, Mayuresh Govekar, Omkar Naik, Gaurav Nemlekar, Prathamesh Bhat, Needhi Gaonkar

Electronics and Communication, Agnel Institute of Technology and Design.

Abstract - Traffic congestion in cities creates major delays for emergency response units which reduces essential life-saving operations. The research paper develops a strong cloud-based IoT system that optimizes ambulance transit through controlled traffic intersections. The system includes GPS tracking through a mobile application and cloud processing via AWS as well as dynamic traffic signal control through ESP32 and Arduino Uno microcontrollers. When drivers successfully log in their app sends GPS data to AWS which identifies the closest intersection and establishes lane prioritization. The ESP32 module receives commands to trigger signal adjustments through the Arduino Uno for unobstructed ambulance passage. The system underwent extensive testing in controlled environments to evaluate both latency performance and reliability and failure response capabilities. The system demonstrates effective ambulance movement with two main challenges regarding multiple intersection control and network connection needs. The current system architecture shows potential for development into a scalable smart city emergency services platform.

Key Words: Emergency Vehicle Prioritization, Ambulance Transit Optimization, Smart Traffic Control, IoT-Based System, Cloud Computing, AWS Integration, GPS Tracking, ESP32, Arduino Uno, Urban Mobility, Lane Clearance Automation, Real-Time Traffic Management, Smart City Infrastructure, Vehicle-to-Infrastructure Communication (V2I), Intelligent Transportation Systems (ITS)

1. INTRODUCTION

The increasing number of vehicles in urban areas creates obstacles which slow down the speed of emergency vehicles during their operations. Standard traffic control systems fail to provide sufficient responsiveness to enable ambulances move without hold during periods of heavy traffic. The Ambulance Prioritization and Traffic Management System (APTMS) functions to automatically clear lanes at junctions using ambulance tracking data and directional information. The system combines a GPS-based mobile application with cloud processing and intelligent traffic control hardware which works together to reduce ambulance transit delays. Through automated processes and direct machine communication the system fulfills smart city goals for adaptable and responsive urban transportation networks.

2. CORE

2.1 System Overview

The APTMS architecture consists of four fundamental components which make up its operational framework. The mobile application uses Kotlin to authenticate ambulance drivers and send real-time GPS location data to cloud servers after driver login. The Cloud Layer (AWS) receives GPS data to evaluate intersection proximity and select the appropriate lane for priority treatment. The ESP32 Microcontroller serves as the intermediary system that links AWS services to local traffic signal equipment. The Arduino Uno hardware units execute direct signal changes through commands issued by the ESP32. This modular framework allows developers to work in parallel while making it easier to identify system issues.

2.2 Communication Flow

Secure communication is vital. The system uses HTTPS for encrypted data transfer and JSON for light, readable payloads. MQTT was initially considered but later discarded due to Arduino integration challenges.

2.3 Software & Development Tools

- Android Studio with Retrofit for app development.
- Python-based AWS Lambda for logic processing.
- Arduino IDE with relevant libraries for ESP32 and Arduino Uno.

Version control and collaboration were handled through GitHub.

2.4 Testing Framework

A layered testing methodology was employed:

- **Unit Testing:** Verified individual module functionality.
- **Integration Testing:** Ensured smooth data flow between app, cloud, and hardware.

- **Latency Testing:** Measured end-to-end response time, achieving <1.5 seconds in optimal conditions.
- **Stress Testing:** Simulated multiple ambulances to validate lane conflict resolution.
- **Failure Testing:** Assessed system behavior under network loss and power interruptions.

2.5 Challenges and Mitigation

- GPS update latency was reduced by customizing refresh intervals.
- Signal flapping between ESP32 and Arduino was fixed using debounce logic.
- Power inconsistencies were mitigated using capacitors and a regulated supply.
- Angle-based intersection logic improved over earlier distance-only calculations.

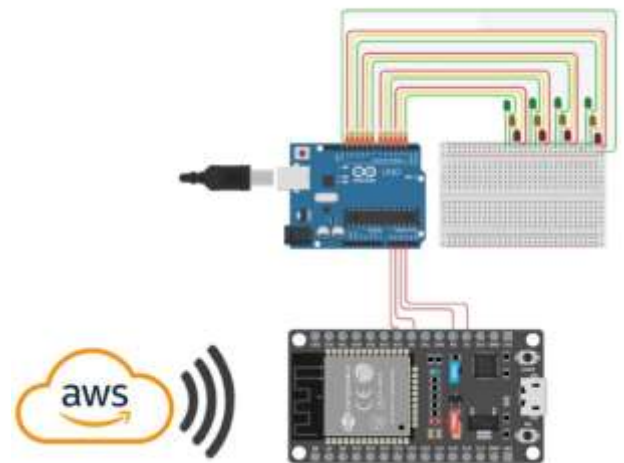


Fig -1: Figure

3. CONCLUSIONS

The APTMS prototype effectively shows how IoT together with cloud-based integration systems can address actual ambulances traffic problems. Although scaling to multiple intersections and simultaneous emergency situations presents challenges the system establishes a strong foundation. Upcoming developments will focus on adding vehicle-to-infrastructure (V2I) capabilities alongside predictive routing and multi-ambulance management features. The system functions as an effective upgrade for standard urban mobility operations which saves lives.

ACKNOWLEDGEMENT

The authors wish to thank the faculty of Agnel Institute of Technology and Design for constant support. Special appreciation goes to our project mentors for their guidance throughout development and testing.

REFERENCES

1. S. Gupta, "Cloud-Based IoT Frameworks for Emergency Services," *IEEE Smart Urban Systems*, 2021.
2. A. Mehta et al., "Real-Time Ambulance Tracking Using GPS and GSM," *International Journal of Embedded Systems*, vol. 10, no. 1, 2022.
3. R. Sharma and T. Desai, "Adaptive Traffic Management in Smart Cities," *Proc. IEEE Urban Tech Conf.*, 2020.
4. S. Banerjee, "Multi-Node IoT Systems for Public Safety," *IEEE Internet of Things Journal*, 2022.

Table -1: Lights Connection Table

Direction	Light Color	Pin Number
North	Green	2
North	Yellow	3
North	Red	4
East	Green	5
East	Yellow	6
East	Red	7
South	Green	8
South	Yellow	9
South	Red	10
West	Green	11
West	Yellow	12
West	Red	13

Direction	Detection Pin	Analog Pin Number
North	Ambulance	A0
East	Ambulance	A1
South	Ambulance	A2
West	Ambulance	A3