Redes e Sistemas Autónomos - Final Report

Universidade de Aveiro(*UA*)

Departamento de Electrónica, Telecomunicações e Informática(*DETI*)

Mestrado em Engenharia de Computadores e Telemática (*MECT*)

Emergency Traffic Handler

Authors

Tomás Laranjo - NºMEC 107542
tomas.laranjo@ua.pt
Gonçalo Cunha - NºMEC 108352
goncalo.cunha@ua.pt

June 16, 2025

Index

Abstract

This project presents an implementation of a cooperative intelligent transportation system (C-ITS) using the VANETZA framework for vehicle-to-everything (V2X) communication. The implemented system demonstrates advanced emergency vehicle priority management through a coordinated network of On-Board Units (OBUs), Road Side Units (RSUs), and intelligent traffic control mechanisms.

The implementation features an emergency vehicle OBU that broadcasts Decentralized Environmental Notification Messages (DENM) to request traffic light preemption, normal vehicle OBUs that respond to Signal Phase and Timing Extended Messages (SPATEM) for safe intersection navigation, and an RSU that manages adaptive traffic control based on real-time conditions. All components communicate using standardized ITS-G5 protocols with proper ASN.1 message encoding.

A web-based monitoring dashboard provides real-time visualization of vehicle positions, traffic light states, emergency mode activation, and system performance metrics. The system successfully demonstrates emergency vehicle priority scenarios where approaching ambulances trigger immediate traffic light changes, allowing safe and efficient passage through intersections.

Testing via a simulated demo validates the system's ability to handle multiple concurrent vehicles, maintain communication reliability under various network conditions, and provide seamless emergency response coordination. The modular architecture ensures scalability and supports integration with existing transportation infrastructure.

Chapter 1

Introduction

The rapid evolution of vehicular communication technologies has opened new possibilities for creating safer, more efficient transportation systems. Vehicle-to-Everything (V2X) communication, particularly through the ITS-G5 standard, enables vehicles, infrastructure, and other road users to exchange critical information in real-time, forming the foundation of Cooperative Intelligent Transportation Systems (C-ITS).

This project presents a simulated implementation of a C-ITS system that aims to be a solution for optimizing traffic at intersections, particularly in situations involving an emergency vehicle. The created system relies on Road Side Units for central infrastructure, complete with a centralized monitoring dashboard, and integrates multiple On-Board Units (OBUs) representing different vehicle types, including ambulances and other emergency vehicles.

1.1 Project Objectives

The created system aims to complete the following objectives:

- Implement a fully functional V2X communication system using the VANETZA framework;
- Develop emergency vehicle priority protocols through DENM-based signaling;
- Create an adaptive traffic management system responsive to emergency situations;
- Allow for monitoring and visualization via a created platform;
- Validate intended performance in simulated traffic scenarios.

Chapter 2

System Architecture

The implemented C-ITS system consists of 3 components, interconnected using the VANETZA framework, that work together to achieve the system's objectives. The architecture follows a distributed approach where each component operates autonomously.

2.1 System Components

Emergency Vehicle OBU

Simulates an ambulance with priority signaling capabilities, broadcasting both CAM messages for position awareness and DENM messages when approaching intersections to request traffic light preemption.

Normal Vehicle OBU

Represents standard vehicles that comply with traffic signals, received via SPATEM, and adapt their behavior accordingly to maintain safe intersection crossing.

Road Side Unit (RSU)

Manages intersection traffic lights, processes emergency requests from DENM messages, and coordinates traffic flow through SPATEM and MAPEM message broadcasting.

Using a dashboard, offers real-time visualization of the entire system, including vehicle positions, traffic light states and emergency status.

2.2 Communication Flow

The system operates through the following communication patterns:

1. **Vehicle-to-Everything (V2X)**: Vehicles broadcast CAM messages to the RSU for position awareness.

2. **Infrastructure-to-Vehicle (I2V)**: RSU transmits MAPEM for intersection information and SPATEM for traffic signal information.

3. **Vehicle-to-Infrastructure (V2I)**: Emergency vehicles send DENMs to the RSU to request priority at the intersection.

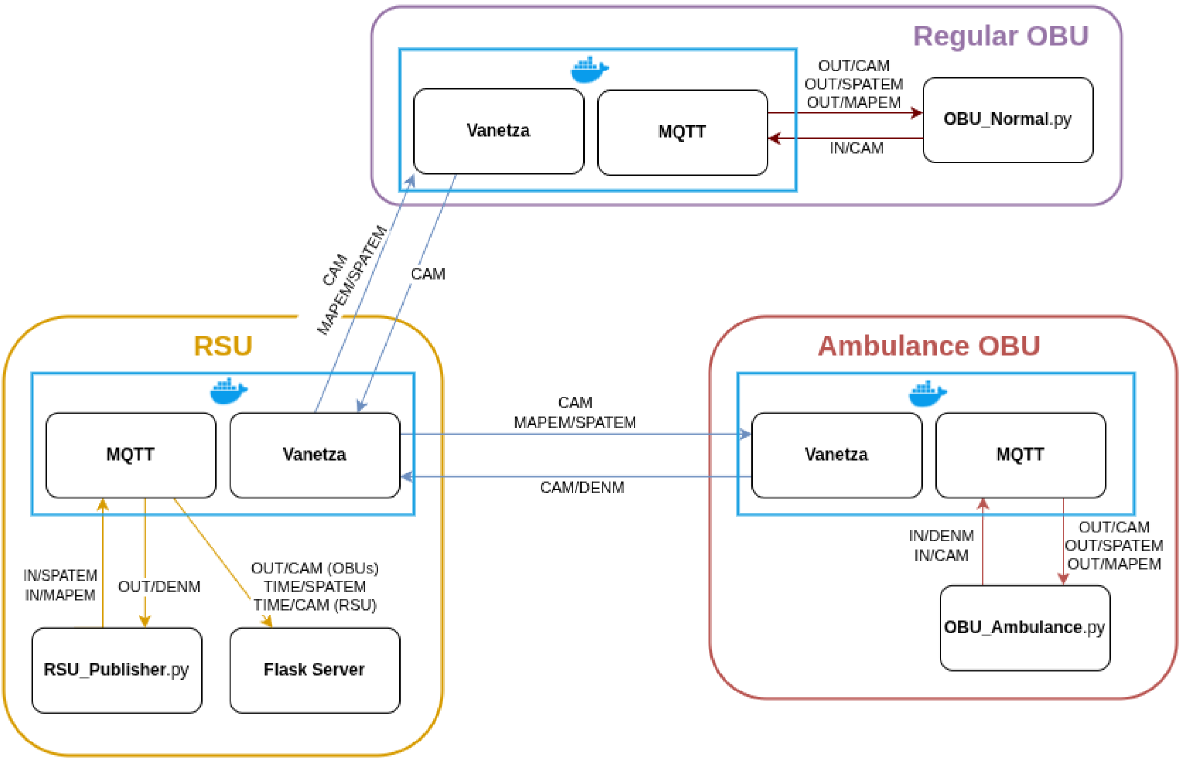

Figure 2.1: Architecture with Exchanged Messages

Chapter 3

Implementation

3.1 Communication Protocols and Message Standards

Regarding the standardized ITS communication protocols used, four message types are exchanged via the Vanetza framework between the components of the system: CAM, DENM, MAPEM and SPATEM.

Transmission Patterns

- **Normal vehicle CAM:** 0.4 second intervals;
- **Emergency vehicles CAM:** 0.7 second intervals;
- **Emergency vehicles DENM:** aperiodic, sent on contact with RSU (event driven);
- **RSU CAM:** 1.2 second intervals;
- **RSU MAPEM:** 1.2 second intervals;
- **RSU SPATEM:** 0.6 second intervals.

3.1.1 CAM

Cooperative Awareness Messages (CAM) are used by a station to share its position:

- **Station ID**: Unique station identifier;
- **Station Type**: Vehicle classification (5=car, 10=emergency, 15=RSU);
- **Position**: GPS coordinates (latitude/longitude);
- **Dynamics**: Speed, heading, and acceleration.

3.1.2 DENM

Decentralized Environmental Notification Messages (DENM) are used by an emergency vehicle to request priority.

- **Station ID**: Unique station identifier;
- **Station Type** : Vehicle classification (10=emergency);
- **Event position:** Real-time vehicle location;
- **Event Type:** Emergency vehicle approaching (code 95-1).

3.1.3 MAPEM

Map Extended Messages (MAPEM) are used by the RSU to announce road structure information, including:

- Intersection topology and geometry;
- Lane coordinates and connections;
- Signal group assignments to intersection entrances.

3.1.4 SPATEM

Signal Phase and Timing Extended Messages (SPATEM) provide traffic signal information, including:

- Signal group identifiers (defined in the MAPEM);
- Event states (3=Red, 5=Green);
- Timing information (minEndTime).

3.2 Normal Vehicle OBU Implementation

The Normal Vehicle OBU represents standard civilian vehicles that comply with traffic regulations and respond appropriately to infrastructure signals. This component demonstrates intelligent traffic-aware navigation.

Position Management

The emergency vehicle tracks its dynamic positioning through a GPS-based coordinate system:

- Utilizes GPS-based latitude/longitude coordinates;
- Implements speed calculations;
- Supports four-lane intersection navigation (North, South, East, West);
- Maintains accurate heading information for directional awareness;
- Broadcasts all position information via CAM.

SPATEM Message Processing

The vehicle continuously monitors SPATEM messages to determine traffic light states, which it uses to control its trajectory:

- Processes signal group mappings (1=North, 3=East, 5=South, 7=West);
- Interprets event states (3=Red, 5=Green);
- Associates signal status to corresponding lane;
- Responds according to signal changes.

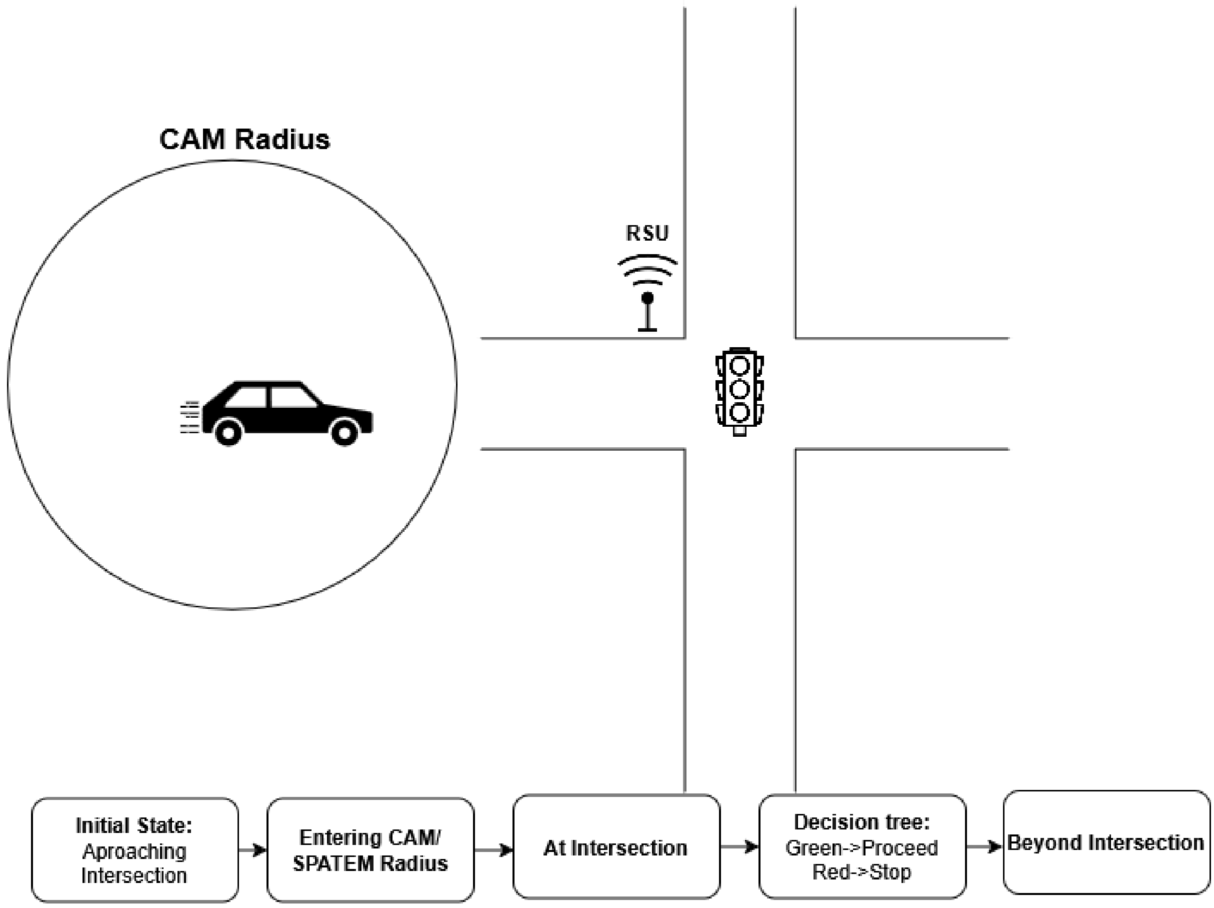

Figure 3.1: Normal OBU Flow

3.3 Emergency Vehicle OBU Implementation

The emergency vehicle OBU is an extension of the normal OBU, with priority signaling capabilities. This component demonstrates how emergency vehicles can request traffic signal preemption using standardized V2X protocols.

DENM Emergency Signaling

When approaching intersections (and getting within a certain distance), the system automatically:

1. Generates DENM messages with emergency cause codes
2. Includes precise vehicle position and heading
3. Transmits priority request to infrastructure
4. Tracks acknowledgment and response times

Distance Calculation

Uses Haversine formula for accurate message radius distance measurement:

```
# === Utility Functions ===

def load_json(filepath):
    with open(filepath, "r") as file:
        return json.load(file)

def haversine_distance(lat1, lon1, lat2, lon2):
    # Calculate distance in meters between two lat/lon pairs
    R = 6371000
    phi1, phi2 = math.radians(lat1), math.radians(lat2)
    dphi = math.radians(lat2 - lat1)
    dlambda = math.radians(lon2 - lon1)
    a = math.sin(dphi/2)**2 + math.cos(phi1) * math.cos(phi2) * math.sin(
    dlambda/2)**2
    c = 2 * math.atan2(math.sqrt(a), math.sqrt(1-a))
    return R * c
```

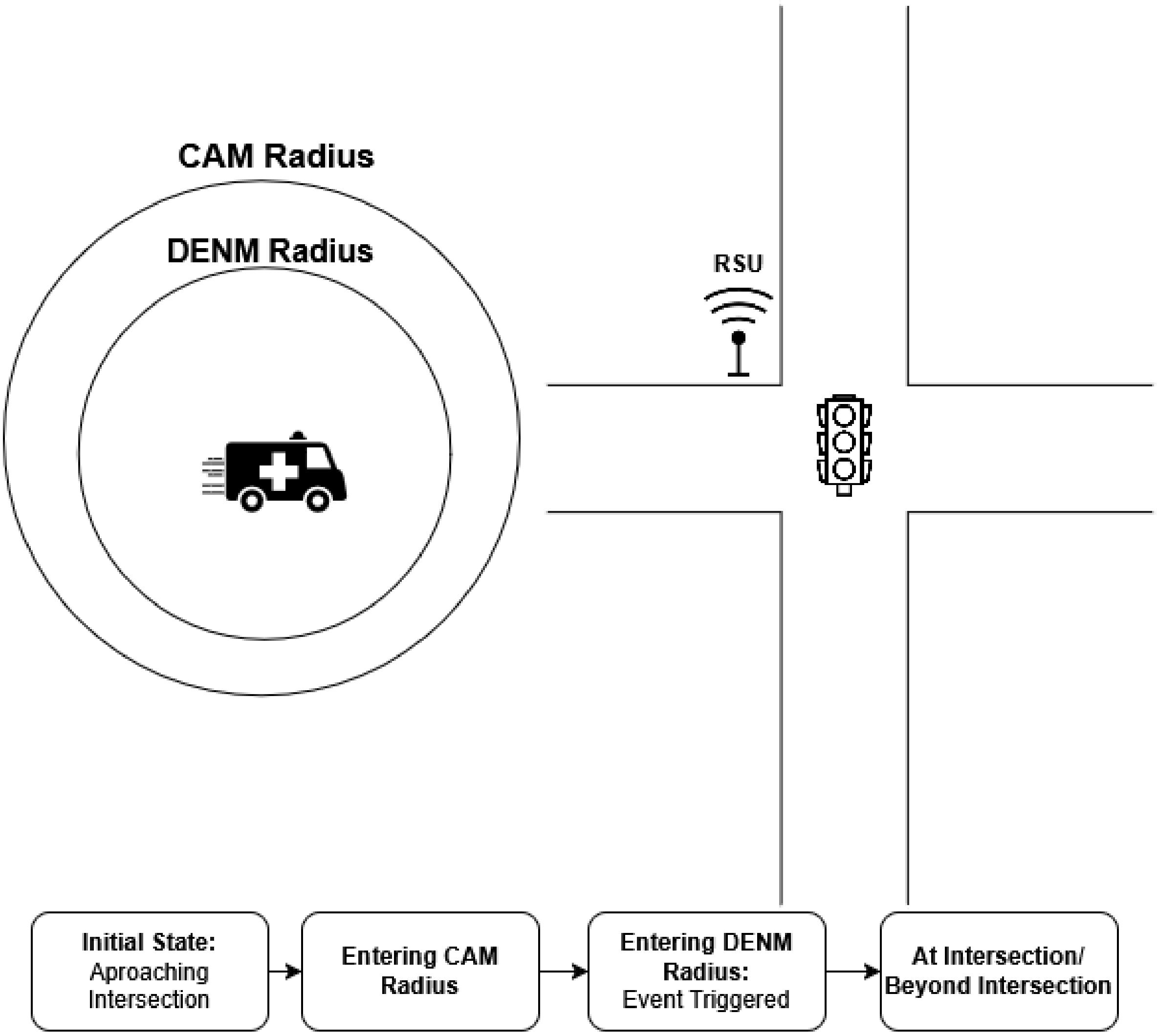

Figure 3.2: Ambulance OBU Flow

3.4 Road Side Unit (RSU) Implementation

The Road Side Unit serves as the main infrastructure component, managing traffic signals, processing emergency requests, and coordinating intersection traffic flow through advanced message broadcasting.

3.4.1 SPATEM Broadcasting

Normal Operation Mode

Implements standard traffic light cycling:

- 10-second cycle duration for responsive traffic flow;
- Alternating North-South and East-West phases;
- Pattern: [Red, Green, Red, Green] → [Green, Red, Green, Red];
- Automated timing calculations based on system clock.

Emergency Mode Activation

Upon receiving DENM messages:

1. **Immediate Response**: All signals set to red within 1 second;
2. **Direction Analysis**: Determines emergency vehicle heading;
3. **Priority Granting**: Sets appropriate signal to green;
4. **Duration Management**: Maintains emergency mode while emergency vehicle is inside the intersection;
5. **Normal Restoration**: Returns to normal operation.

3.4.2 MAPEM Broadcasting

MAPEM provide the following information:

- Intersection geometry and lane configuration;
- Signal group assignments and phases;
- Timing parameters and constraints.

3.4.3 DENM Processing Implementation

```python
def handle_emergency_denm(denm_payload):
    """
    Immediately set all semaphores to red and then set the semaphore for the
    lane
    corresponding to the ambulance (determined from the DENM's heading) to
    green.
    """
    global emergency_mode, emergency_mode_expiry, emergency_target_signal
    emergency_duration = 10

    spatem_msg = load_json(SPATEM_FILE_PATH)

    for intr in spatem_msg.get("intersections", []):
        for state in intr.get("states", []):
            if state.get("state-time-speed"):
                state["state-time-speed"][0]["eventState"] = 3
                state["state-time-speed"][0]["timing"] = {"minEndTime": 30}

    # Determine the lane from the DENM's heading
    heading = denm_payload.get("location", {}).get("eventPositionHeading", 0)
```

The RSU implements an emergency response algorithm:

- **Message Validation**: Verifies DENM authenticity and format;
- **Heading Analysis**: Extracts vehicle heading from accompanying CAM;
- **Signal Mapping**: Determines corresponding traffic signal;
- **Preemption Execution**: Immediately changes signal states in SPATEM;
- **Timer Management**: Tracks the duration of emergency mode.

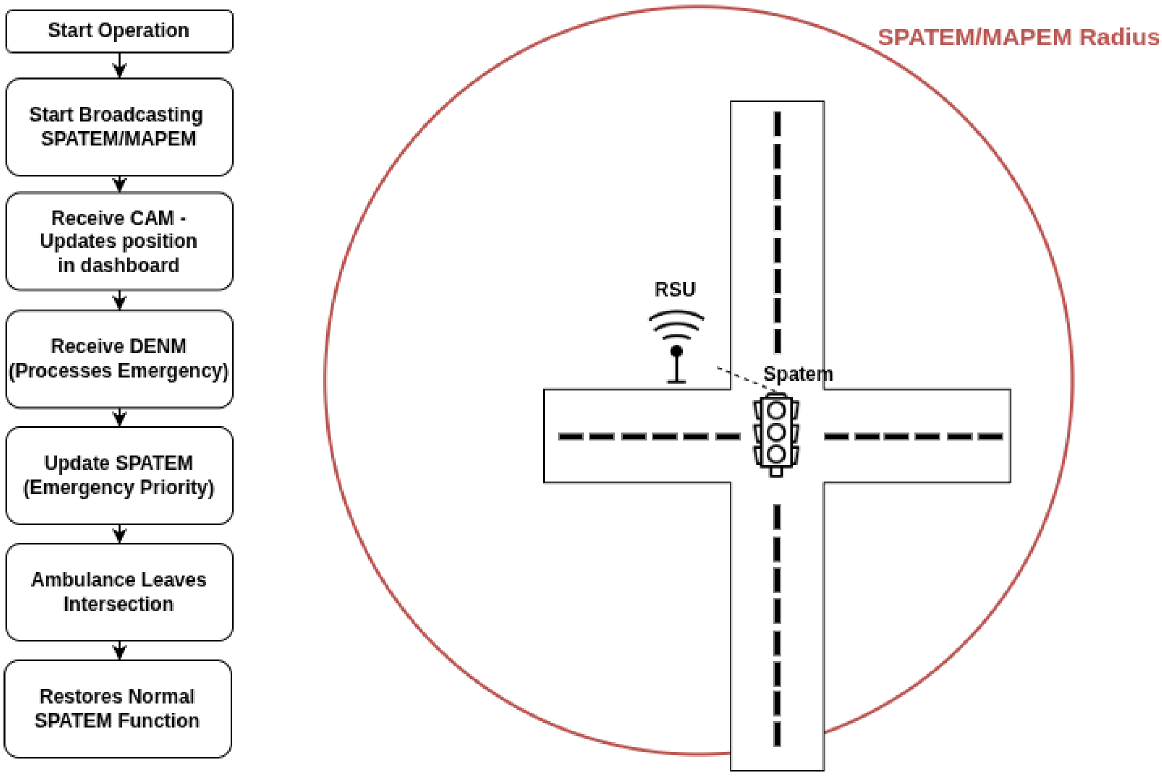

Figure 3.3: RSU Flow

3.4.4 Monitoring Dashboard

The dashboard, integrated in the RSU, facilitates real-time monitoring of the entire C-ITS system. Built using Flask and common web tools such as React, it features a map that displays the following:

- Intersection and road layout;
- Road Side Unit location;
- Vehicle positions with real-time updates;
- Traffic light states with color-coded indicators.

Chapter 4

Testing and Validation

With the purpose of validating our system, we created a demo for a common use case: an ambulance approaches an intersection at the same time that another vehicle approaches in a different direction. This demo video can be found in the appendix section of this document.

4.1 Flow of the demo

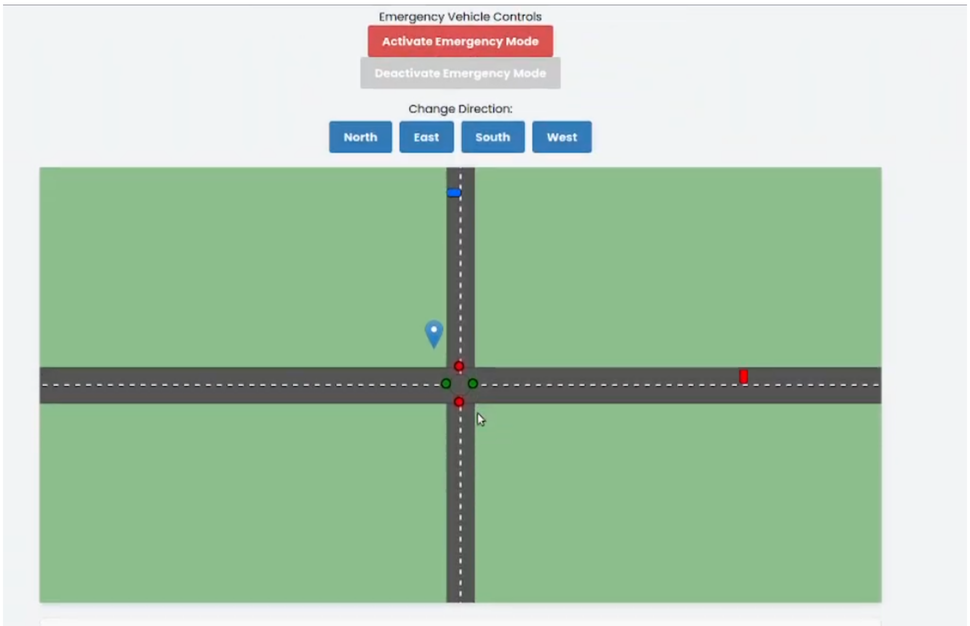

Figure 4.1: Demo Setup

In this demo, we can see an ambulance, represented by a red rectangle, approaching the intersection from the east side, while another vehicle, represented by a blue rectangle, approaches from the north, both broadcasting CAM messages with their position. The ambulance signals via a DENM message that it is in an emergency, this message is then received by the RSU, and based on its heading, sets the corresponding signal light in the SPATEM message to green, setting all other signals to red, with the other vehicle, based on the received SPATEM, stopping before the intersection. Once the RSU detects the ambulance left the intersection and restores the normal functioning of the SPATEM, the vehicle can cross the intersection when the signal it receives eventually changes to green.

Chapter 5

Results and Discussion

The previous demo showcases a functional scenario for the system created in this project that demonstrates its intended use, although in a simulated environment, validating the utilization of the system, at least in a small scale. The created system also establishes the fundamental communication tools that can be used in a larger scale implementation.

5.1 Future Work

For possible future work, the obvious next step is to test the implemented system using real hardware instead of docker containers in the same device (one possibility would be using one Raspberry Pi for each OBU, and another for the RSU, with emulated coordinates). Another possible evolution would be to add another RSU/intersection, allowing for coordination between both RSUs, and enabling the implementation of smart traffic coordination using the SPATEM in each device, and based on the perceived vehicle traffic on each RSU via the CAMs they receive.

Appendix

This appendix provides links to the project's demo, source code, and related resources.

Project Demo Video

`https://youtu.be/CdfzDY3ZdXI`

GitHub Project Repository

`https://shorturl.at/SQxbY`

Vanetza-NAP GitHub Repository

`https://shorturl.at/Ik18d`