POWERRANGERS

Group 12

EDGE-BASED V2X EFFICIENT TRAFFIC EMERGENCY RESPONDING PROTOCOL

CONFERENCE: DIVANET '21
NOVEMBER 22–26, 2021, ALICANTE, SPAIN

Review 2

AARON AYUB
DEPARTMENT OF CS
BROCK UNIVERSITY

THIAGO S. GOMIDES
DEPARTMENT OF CS
BROCK UNIVERSITY

ROBSON E. DE GRANDE
DEPARTMENT OF CS
BROCK UNIVERSITY

WHAT WE PRESENTED IN REVIEW-1

- Introduced the problem: emergency vehicles delayed due to congestion.
- Proposed Edge-based V2X Efficient Traffic Emergency Responding Protocol.
- Discussed three algorithms:
    a. Accident Dissemination (multi-hop broadcast).
    b. ERV Discovery & Selection (Central Edge Node based).
    c. ERV Route Broadcast & Accident Resolution.
- Simulation plan: SUMO + OMNeT++ (Veins integration).

CHANGES WE MADE

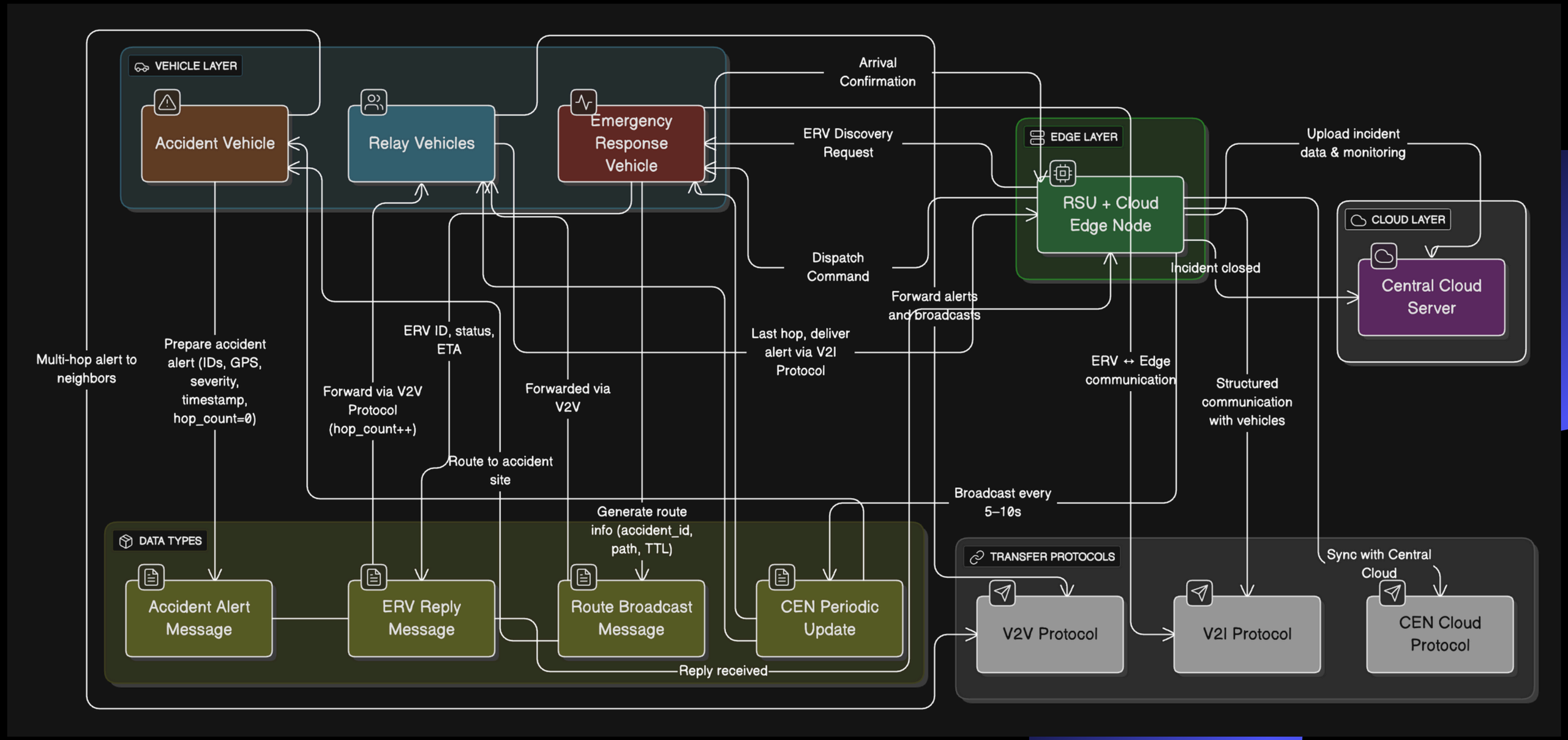
- Advanced ERV Selection (Fuzzy Logic and GA)
  - Before: Only “pick the nearest ERV” (Euclidean-distance-based).
  - Now: Fuzzy logic–based selection (even based on traffic density)

- Extended Vehicle Rerouting
  - Before: Rerouting logic was applied only to the Emergency Vehicle (ERV).
  - Now: Added rerouting for normal vehicles after the accident as well

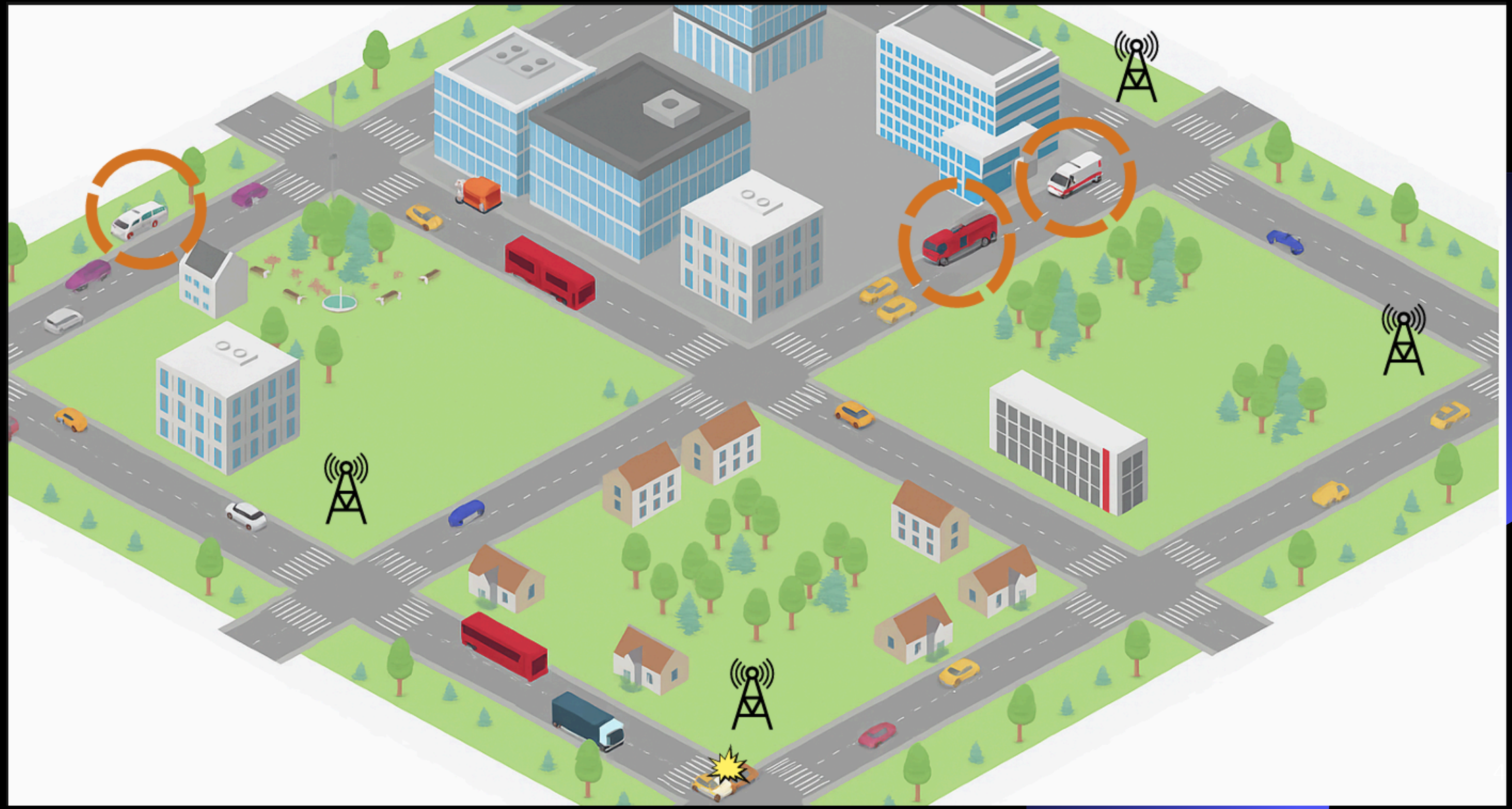
- Improvement: Accident Randomization
  - Before : Accident location in SUMO was fixed at one field location.
  - After: Accident location is now randomized across the road network.

- Change of Simulation Software
  - Before: Proposed simulation was with SUMO + Veins (OMNeT++).
  - Now: We shifted to SUMO and TRACI for network simulation.

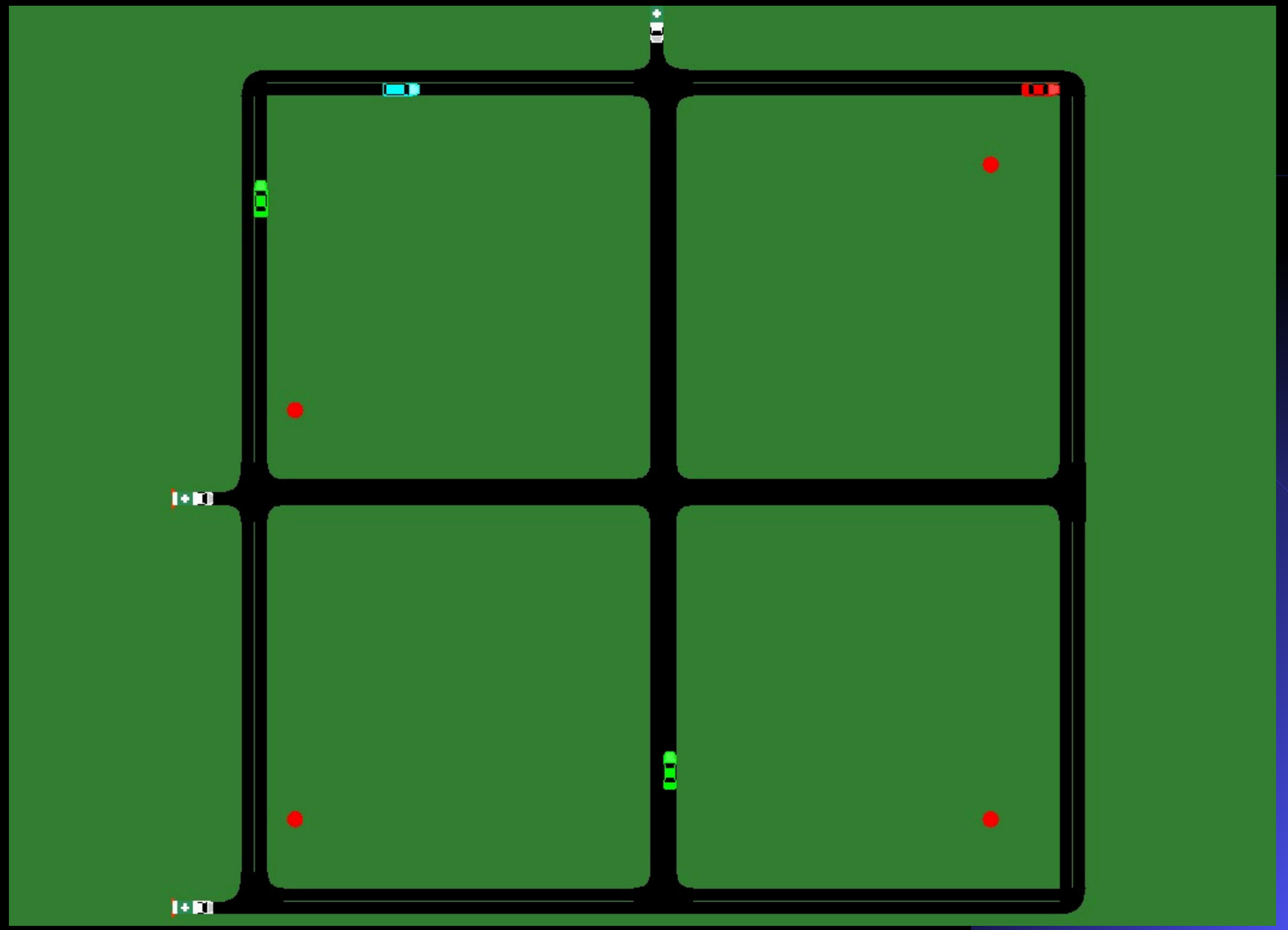
ARCHITECTURE



IMPLENTATION



SIMULATION



V2V PROPAGATE


```
> def find_vehicles_in_range(source_position, exclude_vehicle=None):···

> def find_edge_nodes_in_range(position):···

> def propagate_v2v_message(message, current_vehicle_id, current_position):···

> def broadcast_emergency_alert(source_vehicle_id, accident_location, collision_pair, accident_id):···
```

Getting Congestion in Each Path using Fuzzy

1. Collect Data

The roadside computer (edge) checks each road:

i) How many vehicles are there? ii)What is their average speed?

2. Turn into Simple Categories

i) Vehicle Count → Low / Medium / High ii) Speed → Low / Medium / High

3. Apply Smart Rules like

i) Many vehicles + slow speed → High Congestion

ii) Medium vehicles + medium speed → Medium Congestion

iii) Few vehicles + fast speed → Low Congestion

4. Calculate Congestion Score : Rules combine into one score (0–10).

i) 0 = free road ii) 10 = heavy congestion

5. Path Congestion

Each road segment gets a score.

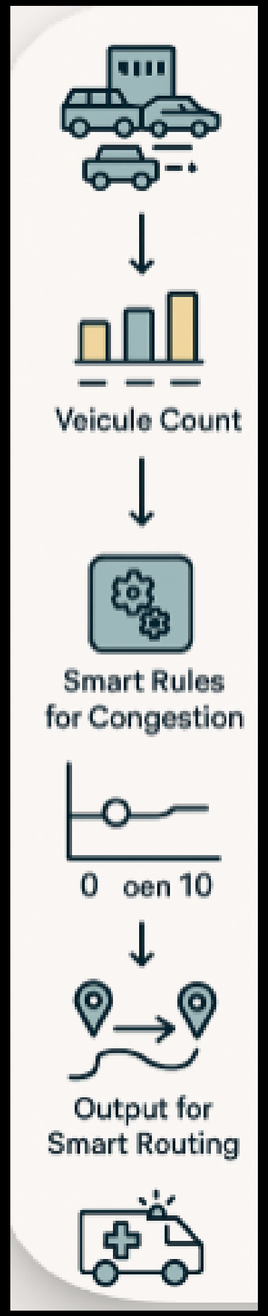
Scores are averaged → overall path congestion

6. Share Results for Routing

The edge node shares the congestion result as:

Low (0–3), Medium (4–6), High (7–10)

Ambulances → to the quickest, least congested path



BEST ERV SELECTION

The GA simulates evolution to rapidly identify the most suitable ERV for an incident by evaluating multiple real-world factors at once.

1. Defining a "Good" Choice: The Fitness Function

Each potential ERV is assigned a "fitness score". A higher score means a better choice. The score combines three key factors:

FACTOR	Goal	Impact of Fitness
Distance	Shorter is Better	1/textDistance
Traffic	Less Congest-ion is Better	1/textTraffic
Readiness	Higher is better	textReadiness (e.g., availability, equipment)

2. Evolution to the Optimal Solution

The Genetic Algorithm process iteratively compares the candidates. Through this "survival of the fittest" process, the ERV with the highest fitness score is chosen as the optimal response unit.

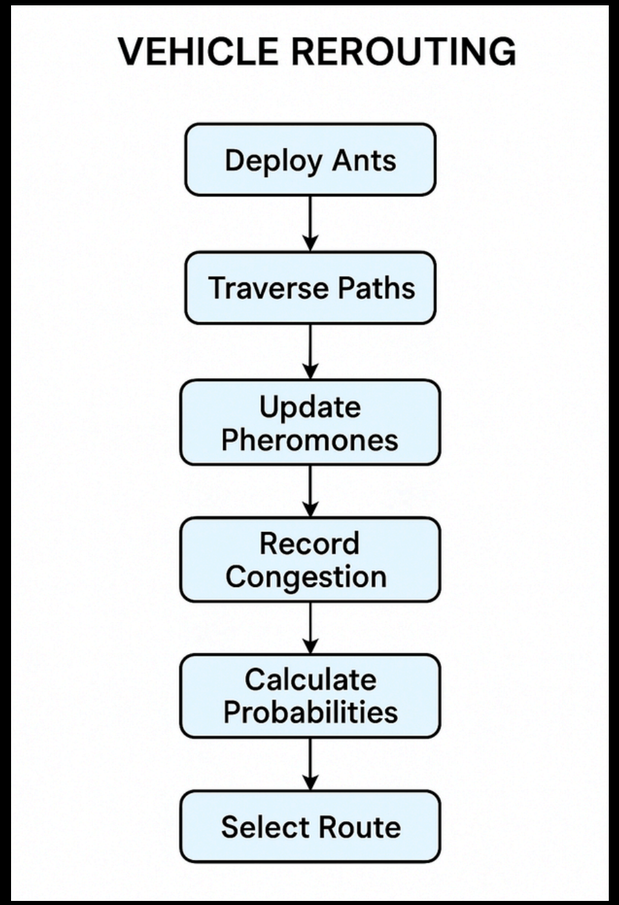
VEHICLE REROUTING

Goal : Vehicles need to avoid ERV route and accident route, avoid Congested routes, minimize travel time to destination.

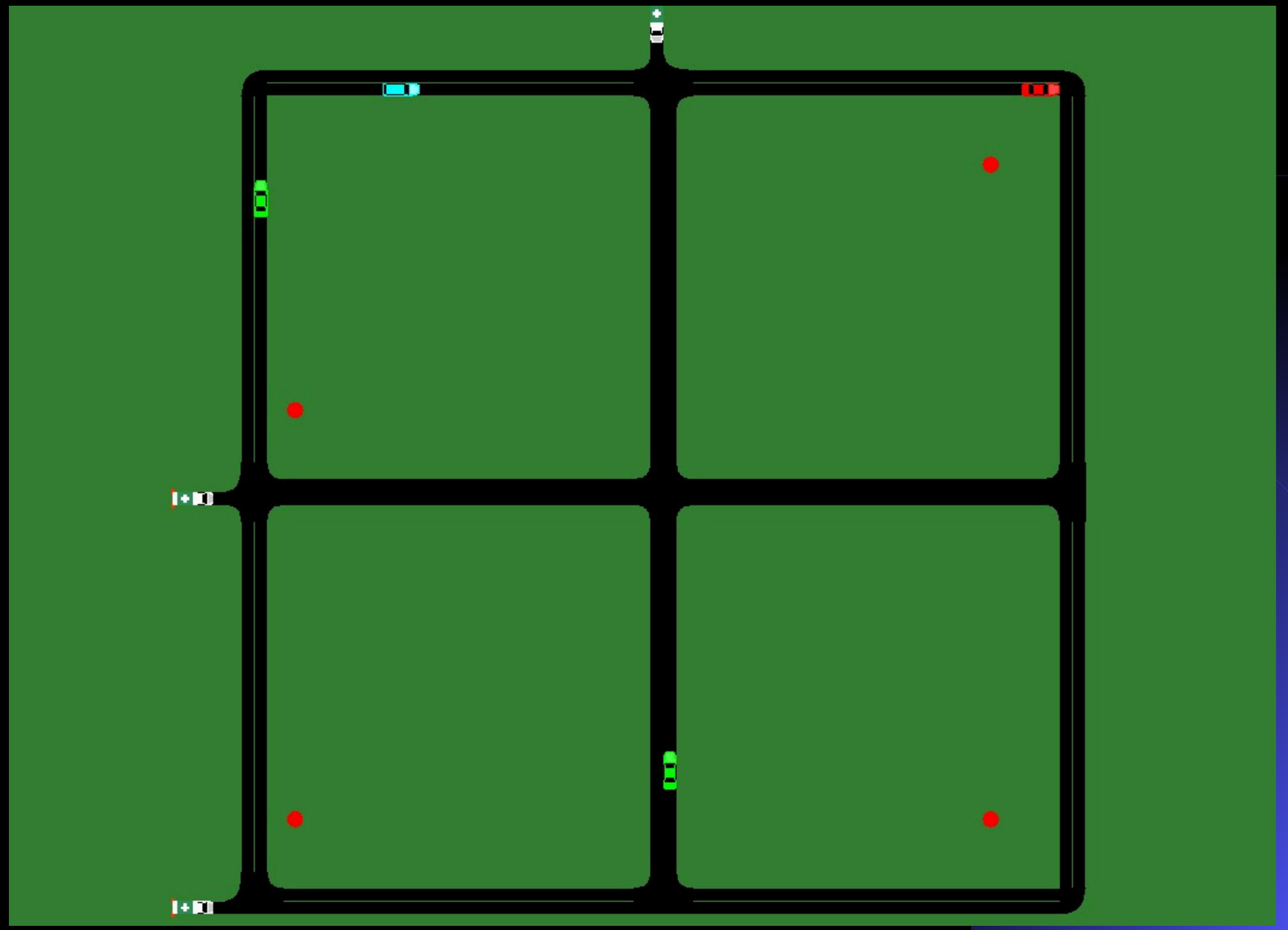
Input : Listens to the Broadcasts

Algorithm Used : Ant Colony Optimisation

First, the ERV route and accident route is blocked to prevent entry. Then, all vehicles already on the path are rerouted.



SIMULATION



SIMULATION

```
[ACCIDENT] Vehicles involved: ('veh6', 'veh8') at position (216.799, 198.4) (Total accidents: 1)
[CEN EdgeCEN_C REGISTER] Accident ACC_001 at (216.799, 198.4) involving vehicles veh6 (sim time 85.0s)
Selected Best Ambulance: ambulance2 (Route: routeAmbulance2)
[V2I] Vehicle veh6 received accident ACC_001 info from CEN EdgeCEN_C at t=85.0s (distance 24.9)
[REROUTE] Vehicle veh6 old route: ['E_H', 'H_I', 'F_I']
[REROUTE] Vehicle veh6 new route: ['H_I', 'E_H', 'H_I', 'F_I']
[V2I] Vehicle veh8 received accident ACC_001 info from CEN EdgeCEN_C at t=85.0s (distance 23.4)
[REROUTE] Vehicle veh8 old route: ['G_H', 'H_I']
[REROUTE] Vehicle veh8 new route: ['H_I']
[V2I] Vehicle veh9 received accident ACC_001 info from CEN EdgeCEN_C at t=92.0s (distance 197.5)
[REROUTE] Vehicle veh9 old route: ['A_D', 'D_G', 'G_H']
[REROUTE] Vehicle veh9 new route: [':D_7', 'A_D', 'D_G', 'G_H']
[CEN EdgeCEN_C BROADCAST] Accident ACC_001 at (216.799, 198.4) being broadcast at sim time 95.0s
    [CEN EdgeCEN_C RECEIVED] Accident ACC_001 info received from EdgeCEN_C (distance: 24.9)
    [CEN EdgeCEN_I RECEIVED] Accident ACC_001 info received from EdgeCEN_C (distance: 179.2)
[CEN EdgeCEN_C BROADCAST] Accident ACC_001 at (216.799, 198.4) being broadcast at sim time 105.0s
    [CEN EdgeCEN_C RECEIVED] Accident ACC_001 info received from EdgeCEN_C (distance: 24.9)
    [CEN EdgeCEN_I RECEIVED] Accident ACC_001 info received from EdgeCEN_C (distance: 179.2)
[V2I] Vehicle veh10 received accident ACC_001 info from CEN EdgeCEN_C at t=108.0s (distance 196.1)
[REROUTE] Vehicle veh10 old route: ['A_D', 'D_G']
[REROUTE] Vehicle veh10 new route: [':D_7', 'A_D', 'D_G']
[V2I] Vehicle veh11 received accident ACC_001 info from CEN EdgeCEN_C at t=114.0s (distance 198.7)
[CEN EdgeCEN_C BROADCAST] Accident ACC_001 at (216.799, 198.4) being broadcast at sim time 115.0s
```

INDIVIDUAL CONNTRIBUTION

Congestion.py - Vijay CSE23754
Vehicle.py - Mohan CSE23409
Best erv .py - Vamsidhar CSE23163
Route generator.py - Akshar CSE23547
Cen broadcast.py - Keerthi CSE23031
Run simulation.py - Isha CSE23029
Sumo network and accident.py - Raghav CSE23032

Thank **You**