RapidEMS Progress Report 3

Tejas R. Annapareddy,* Jacob E. Lee,† and Eyosyas B. Samuel‡
Governor's School @ Innovation Park, Manassas, Virginia 20110, USA
(Dated: February 29, 2024)

In this report, we outline progress on RapidEMS. We use RapidGLD, a version of Greenlight District Simulator, for simulations. Steps include updating GLD, creating an EMS vehicle model, and simulating road systems. We have been working on preparing skeleton software for use in the raspberry pi's. We use PuTTy as a terminal emulator to SSH into our pi's and remotely install software and configure our GPS and Python software to communicate with each other and simulate a civilian and an ambulance.

I. INTRODUCTION

As stated previously, our project revolves around decreasing Emergency Medical Services (EMS) response times to increase overall patient discharge success rates.

II. TESTING

We have established a new version of Green Light District (GLD) Simulator which we call RapidGLD. The original version of GLD Simulator can be found at https://sourceforge.net/projects/stoplicht/. Our version, RapidEMS, can be found at https://github.com/RapidEMS/RapidGLD.

A new Ambulance Roaduser was created to effectively simulate an EMS vehicle. Vehicles will be programmed according to process layed out by RapidEMS, which is to alert vehicles via our vehicle-to-infrastructure communications system before the sound of the Ambulance's sirens can be detected by the human ear. Each vehicle will switch lanes and to create a free lane for the ambulance.

A. Location

We have modeled our map based on the fastest route from an arbitrary position in an urban area [1] to the nearest hospital [2]. A map of the route can be found here. The map(s) that we have created for this project can be found here.

III. ANALYSIS

A table that represents that data received from our simulation is as follows:

TABLE I. A table of the values computed from our experiment.

	Response Time (cycles)	
Spawnrate %	Disabled	Enabled
2	3.28	2.52
3	8.30	6.82
4	18.58	14.28
5	44.86	17.78

The response times were computed in cycles, a unit used within the RapidGLD simulator. The data shows that the implementation of RapidEMS (and as a result other communications systems) decrease the response times of EMS vehicles. Something to note is that, the spawn chances that were changed only affect other vehicles in our simulation—the spawn chance for EMS vehicles was constant, at 0.01f = 1.00 %. The data in the table has been compiled into a graph, as follows:

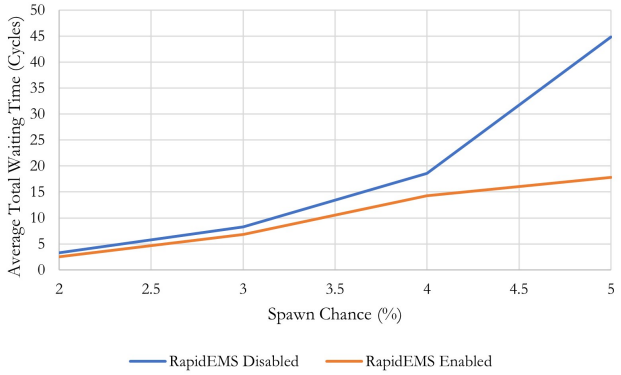

FIG. 1. Average total waiting time of EMS vehicles simulated in RapidGLD with and without the implementation of RapidEMS

The spawn chance (x-axis) in Figure 1 is best conceptualized as the level of activity in our simulation. In the context of RapidEMS, it is the amount of traffic on the roads serving as obstacles for an EMS vehicle transporting a patient to the nearest hospital.

* tannapar@gmu.edu
† jlee418@gmu.edu
‡ esamuel3@gmu.edu

IV. HARDWARE

Now that we have received our Raspberry Pis and GPS modules, we will start with creating a real life demonstration of our software. We have begun imaging the Pis and developing the prototyping software for each. We will designate two of the Pis to represent an ambulance and a civilian.

GLDCivilian will be either a stationary vehicle in traffic or a moving vehicle on a highway. GLDAmbulance will be a moving ambulance entering traffic or navigating to an emergency scene. The goal is to have a proof of concept that would integrate into vehicle information displays.

V. FUTURE PLANS

Once our items arrive, we will do software testing and make sure all the radios and GPS modules are properly calibrated and ready for use. We then will preform mock tests to ensure that the system communicates correctly and with as little delay as possible.

---

[1] 917 607 5894, 1025 Atlantic Ave, Brooklyn, NY 11235.
[2] NewYork-Presbyterian Brooklyn Methodist Hospital, 506 6th St, Brooklyn, NY 11215.