universidade de aveiro

Relatório do Projeto ‘Platoon Merging’ de Redes e Sistemas Autónomos

Github link to the project: https://github.com/HugoMoinheiro/rsa_proj.git

Introduction

The idea of our project was to develop a program that simulates the rush hour traffic and make the cars exchange messages to speed up the overall traffic and avoid congestions. The main idea was to organize the cars in the available lanes corresponding to the exit they will take. After being organized, the cars on the left can increase their speed and use platooning to improve the traffic flow.

Development steps

To develop our work, we started by building a simple program to publish and subscribe messages in the broker by the 6 OBU.
Our project was meant to be focused on platooning, but the teacher suggested focusing on a traffic jam environment. That environment would consist in a highway with lots of cars traveling at a low speed due to their lane not being the most appropriate to their exit, because in traffic jam hours people are always changing to a faster lane, sometimes compromising the best position on the road for their trip. So our goal would be to rearrange the vehicles to the correct lane, and then speed up the cars on the middle and left lane to decongest the traffic.

The next step was to divide the timeline in phases, which are described in Demonstration and Timeline. After concluding the previous tasks, we began to develop the front end interface to see the results of the developed simulation. We used the Google Maps API because of the simplicity and also because our simulation uses real coordinates.
The last steps of development were the detection and correction of bugs and improving the overall simulation with a few details, such as updating the speed of the cars depending on the lane which they are in.

Architecture

The overall architecture is present in Figure 1. We have 6 OBU simulated in Vanetza and the simulation program and web app in the host PC. Each OBU has its own

broker where it publishes its messages. The messages are exchanged in the virtual network in Docker.

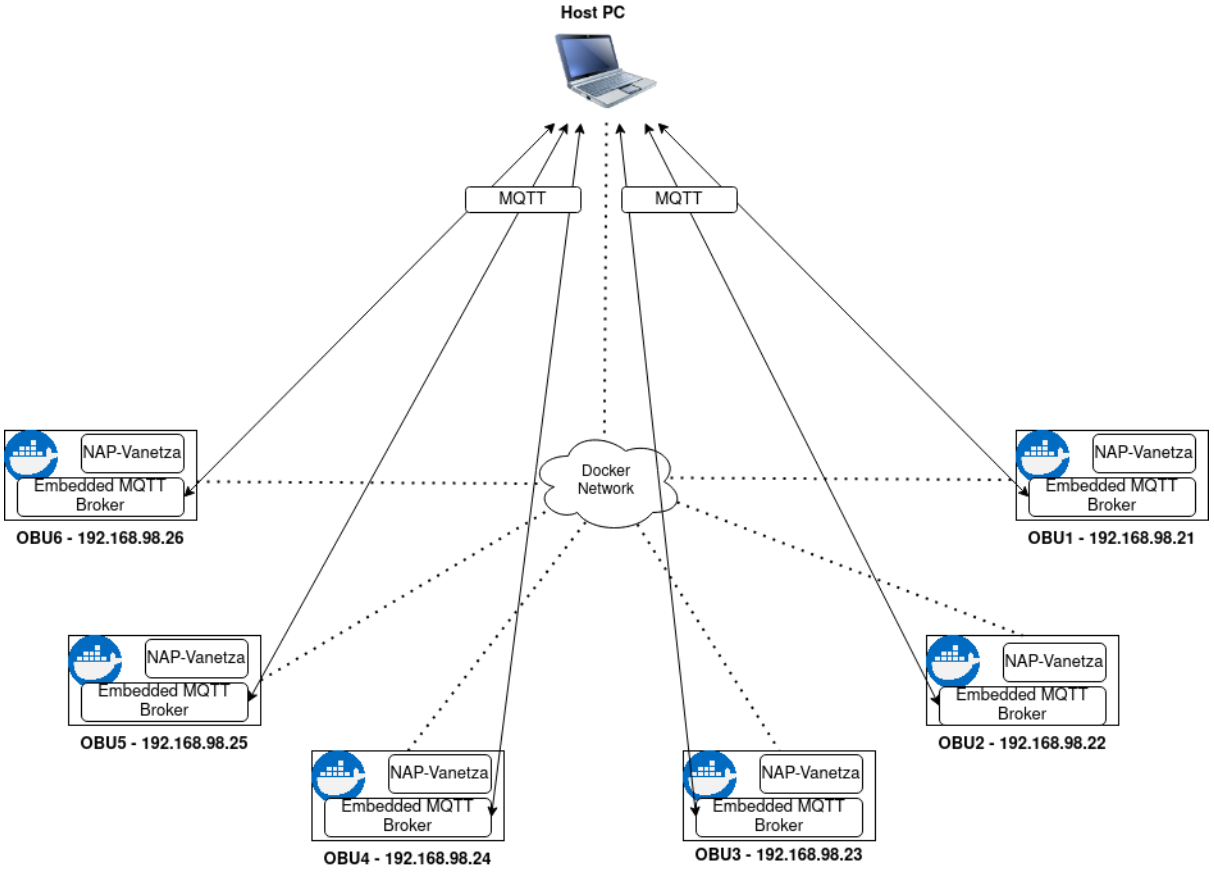

Figure 1 - Architecture of the developed solution

Each OBU communicates with each other though the vanetzalan network and subscribes to the CAM and DENM topics on the broker.

The DENM messages used the following cause codes and sub-cause codes:

- First DENM messages:
  - 41 - Vehicle leaves on the 1st exit
  - 42 - Vehicle leaves on the 2nd exit
  - 43 - Vehicle leaves on the 3rd exit
  - 5# - Position where the vehicle is
- Second DENM messages:
  - 44 - Vehicle at the correct position
  - 45 - Vehicle wants to move 1 lane to the left
  - 46 - Vehicle wants to move 2 lane to the left
  - 47 - Vehicle wants to move 1 lane to the right
  - 48 - Vehicle wants to move 2 lane to the right
  - 5# - Position of the vehicle he wants to trade position
  - 57 - There is no vehicle to trade position
- Last DENM messages:
  - 58 - Vehicle leaves at his exit

Demonstration and timeline

The timeline of the project is present in Figure 2. As referred before, the project is separated in different phases.
For each car we created a Python Object from the class Vehicle. That class has attributes to describe the vehicle, such as the vehicle exit, the vehicle id, the vehicle coordinates, the vehicle speed and some control variables. The class also has some important methods to update the vehicle coordinates, using the Geopy library. The speed and direction are used to update the coordinates. Right now, our implementation only works in the North direction.
6 cars are created in 3 different lanes. Each one of these vehicles has a random exit, randomized in the beginning of the simulation program**.** Every exit has at least one car leaving on that exit. All the cars are periodically exchanging CAM messages, at a frequency of 2Hz.
We start the simulation exchanging only CAM messages. Each OBU receives the other OBU messages and calculates their position on the road. That calculation is made with a comparison with their own coordinates with the other OBU coordinates.
In Figure 2 we can see the initial formation of the cars.

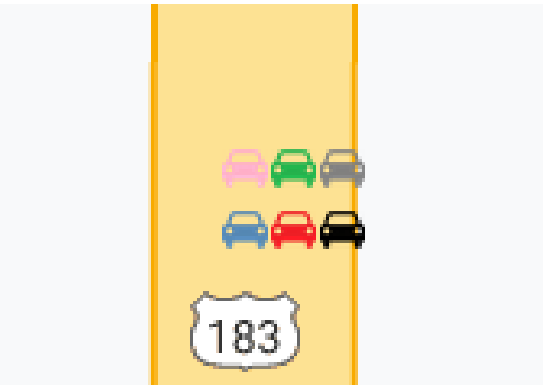

Figure 2 - Initial car formation

In this example, the cars will get this result:

- pink car is in the first position;
- green in the second position;
- gray in the third position
- blue in the fourth position;
- red in the fifth position;
- black in the sixth position.

After calculating their initial position, they send a DENM message to the other cars informing them of the exit they will take and their position on the road. After exchanging those DENM messages, they process what they need to do, if they need to change lanes or if they can stay in the same position. If they need to change lanes, they check if there is any car to trade.

The next phase includes a new DENM exchange with the amount of lanes they need to cross to go to the desired lane and with which car they need to switch (if there is a car to switch).
On this stage, the cars move to the correct position, and then the cars in the left and middle lane increase their speed in order to make the traffic faster.
The next and final phases correspond to the vehicles exiting in their exits and publishing a new DENM message to inform other cars that they have exited. When the vehicles on the road receive a message from a vehicle exiting they move one lane to the right, because they are approaching their exit.

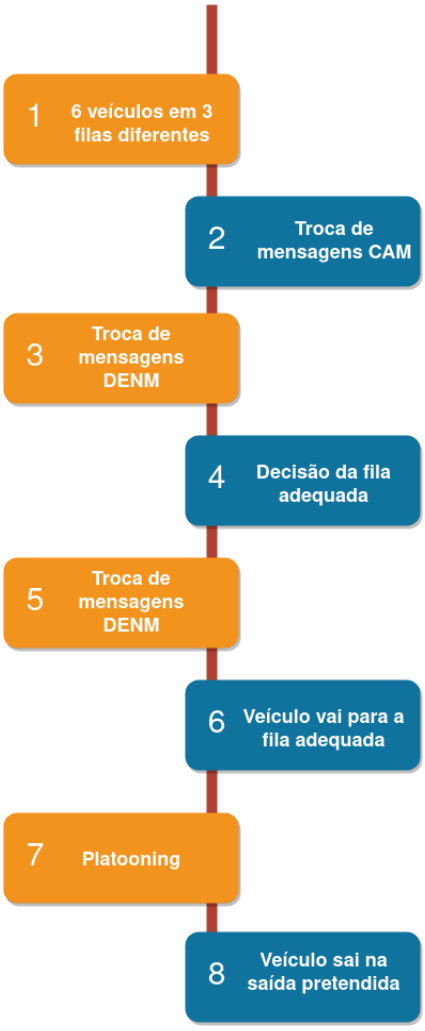

Figure 3 - Timeline of the project

universidade de aveiro

Technologies used

To simulate the OBUs we used Vanetza. The simulation program was written in Python for the ease of use and libraries available, such as geopy, that we used to work with the coordinates in the real world. The web app was developed using the Django framework and the Google Maps API.

Tests and Results

The overall results that we obtained were good. Because the exits of the cars are randomized, sometimes the cars are overlapped, due to the fact that we don't check if they collide when exchanging position, and the platooning is not correctly implemented because they only travel at the same speed, so the CAM messages are not used for that purpose. Rather than that we achieved the work that we agreed at the beginning of the development. These flaws can be corrected in future work.

Hugo Moinheiro, 84931
Pedro Silva, 89228