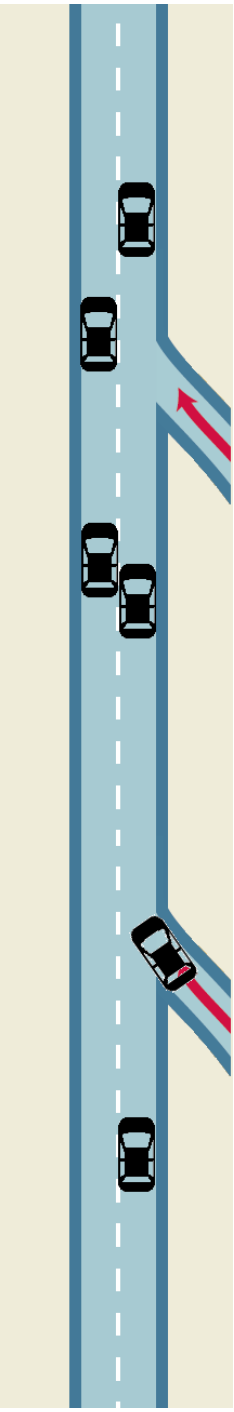
Kern Krisztián ZD3VV3G
Muhari Benjámin IG8S88
Kotán Tamás B0I5EM

Team 2

Homework Project Idea – Specification

Highway Traffic Control System

Our project is based on the idea of a system of independent moving entities that have no vision, or any other information about each other. Let's say these entities are cars driving on a two lane highway in the same direction. New cars can enter to the right lane at entrances. Our system will be capable of controlling these cars so that they avoid collisions.



In our proposed system every car is moving in the same direction in either the left or the right lane. Lane changing takes time, during which the car is in both lanes. Each car has a default movement speed, an acceleration and a deceleration parameter.

The cars are controlled by a centralized "puppet master". This central control system will keep every car in the right lane with its default speed whenever possible. When the distance between two cars reaches a critical threshold, the system orders the car approaching from behind to switch lanes. If the cars are already in the left lane, the faster car will be ordered to brake. The system will try to prioritize entering vehicles in such a way that they can accelerate without being disturbed, so the vehicle approaching from behind should move into the left lane. The control system will need to take into account each car's parameters. Naturally, the final system will need to use a complicated algorithm to handle every possible scenario, but these are the main principles.

Our planned real-time system will implement the required architecture in the following way:

Each car will correspond to an instance of the same client program. This program will be implemented in C language, and is planned to run on the ELTE Pandora / Opsys linux server. When reaching the end of the road interval, the program terminates. (They will use the MQTT-C lib.)

Instantiating this program in separate processes will be handled by a shell script. This script creates a given number of cars with random positions and parameters at start. From time to time, it also creates a new one, entering through an entrance, or at the start of the road interval.

The control system will be implemented in a python script and will also display a real-time visualization.

Communication between the clients and the control system will be made possible using an mqtt broker. This broker will be hosted on the Internet (Google Cloud IoT services).

Kern Krisztián ZD3VV3G
Muhari Benjámin IG8S88
Kotán Tamás B0I5EM

We designed the communication's architecture as follows:

Each client will have a number of flags, like isSwitchingLane, isAccelerating, isDecelerating. They will also store their own position (this a coordinate along a single axis), their current speed, their parameters (default speed, accelerations) and the lane they are in. As for entering vehicles, they will be in a third lane, until they reach the entrance point.

The control system will continuously receive data from the broker, determine the desired commands for each vehicle (with the discussed control algorithm), display the current state, and send the commands back.

Each car will correspond to a single topic, and have a few subtopics, for example for commands, lane, speed, position. E.g. the lane topic could have messages like "entered lane 1", "left lane 2". The commands topic could have messages like "start braking", "stop braking", "switch lane to left". The clients are only subscribed to their own subtopics, but naturally, the master is subscribed to them all. When a client terminates, it indicates it with a message.

Each client will be in an infinite loop, which ends when the critical distance is reached. In the loop they will first set their own flags if they received a new command to do so. Then they update their position accordingly, and also applying a small normally distributed error (this way the system will not be deterministic). Finally they send back their current position, speed, and lane information.