Server based system for small cities traffic lights management

Francesco Ercolani
A.A. 2022/2023

GOALS AND REQUIREMENTS

Abstract

traffic lights are managed as independent units in each intersection. Each one has its own state cycle and it is synchronized with its corrisponding twin on the other side of the street.

But what happens if certain streets have unpredictable peaks of traffic in certain time frames? All these problems could be solved with a dynamic management of the sempahores cycles

Goal

> The goal of the project is to develop a system which optimizes the traffic lights management exploiting the tools provided by a distribute organization.

Requirements

- Simple GUI
- Distribute system
- Variable scenario
- Resilient
- Server administrator interaction
- Close to real scenario

Simple GUI

The GUI is both user side and server administrator side. The user sees the map with the vehicles crossing the streets. The admin manages the servers

Variable scenario

The user is able to set the number of vehicles that he wants to see in the map

Server administrator interaction

The admin is able to crash/restart the servers. This was made to test the resiliency of the system

Distribute system

Each semaphore is controlled by the server which sets them according to sensors placed nearby the intersections which provide data that is submitted to a function

Resilient

There can be 2 ore more servers, one is the main and the other and the backups. If the main crashes, one of the backup takes its place

Close to real scenario

The whole project aims at being as close as possible to a real life scenario

Non-functional requirements

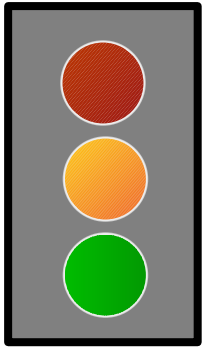
- User friendly
- Simple GUI

DESIGN AND TECHNICAL ASPECTS

Entities

Server

Computes the data and assigns the new timings to each semaphore



Semaphore

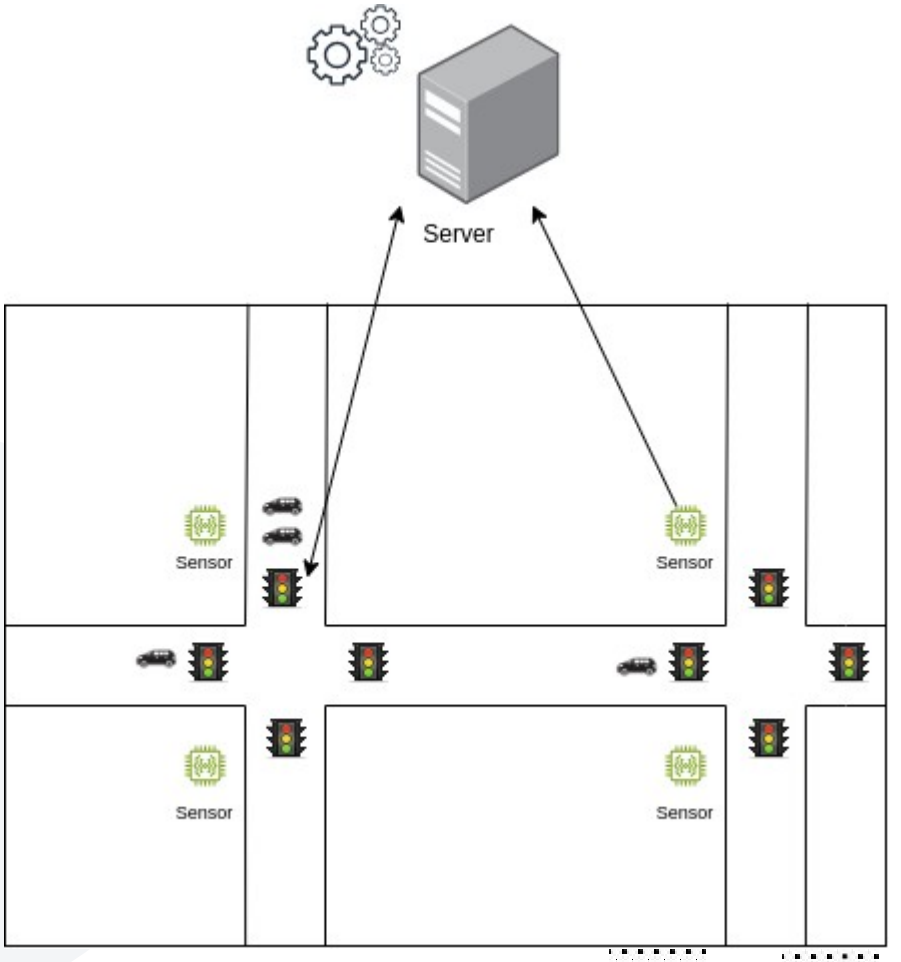
Receives the new green light and red light times

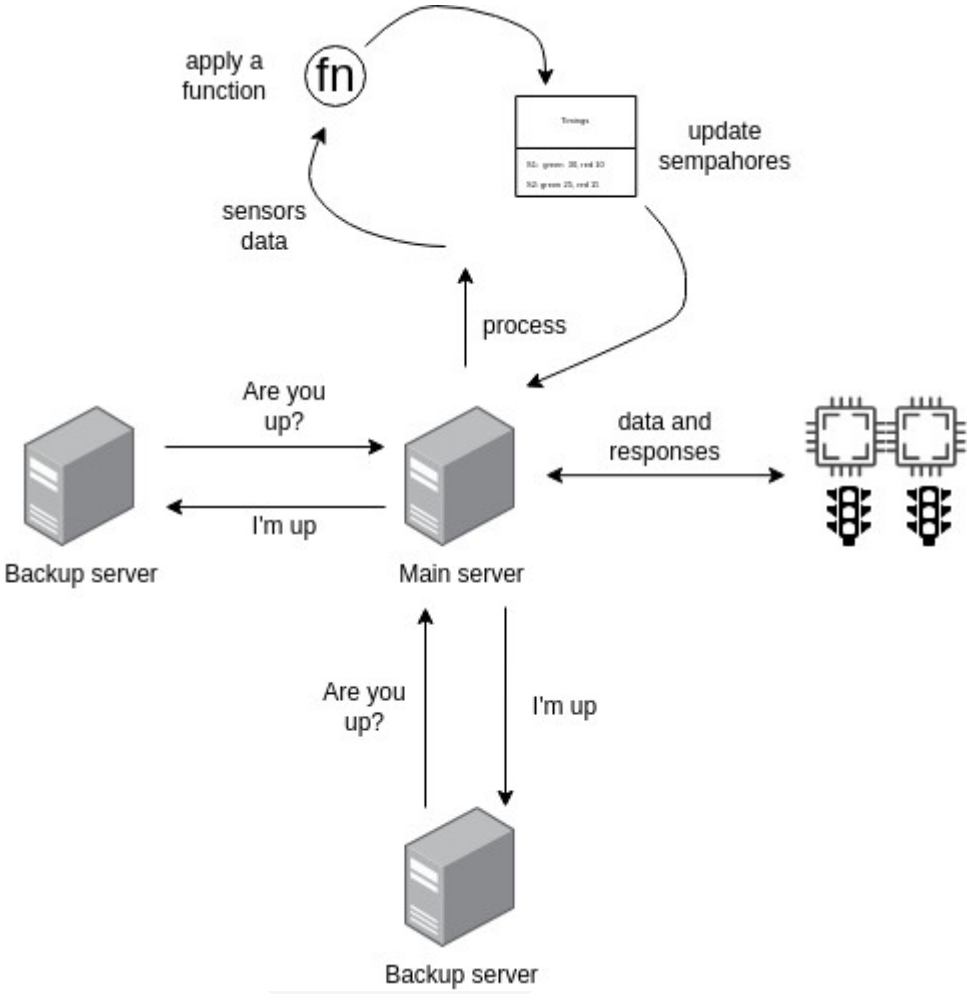
Sensor

Collects the number of vehicles nearby the intersections and send it to the server

Structure

The general architecture is client-server. The server is responsible for receiving the data from the sensors placed in the streets nearby the intersections, process it, and assigning the new timings to each semaphores when it receives the requests from them

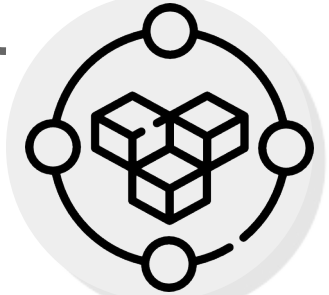



apply a function
fn
update sempahores
sensors data
process
Are you up?
I'm up
data and responses
Backup server
Main server
Are you up?
I'm up
Backup server

Pattern

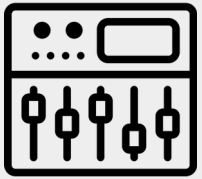
MODEL

Elements for the map logic such as the intersections, the vehicles, the streets and the map, and the element for the traffic lights management (client-server logic)



VIEW

User and server administrator GUI



CONTROLLER

Controller for the communication between the entities

Technologies

- Java
- AWT e Swing
- Java.net
- Gson

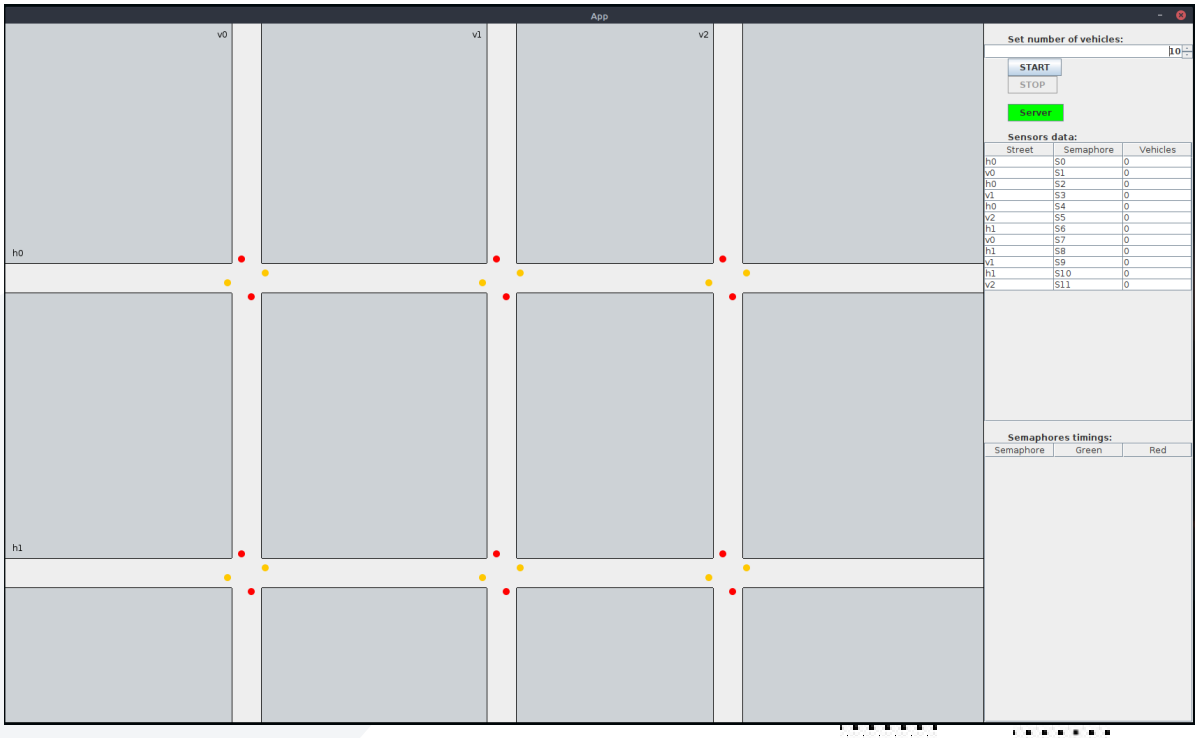
USAGE EXAMPLES

Servers running

> Main server and backup are running normally as their GUI shows

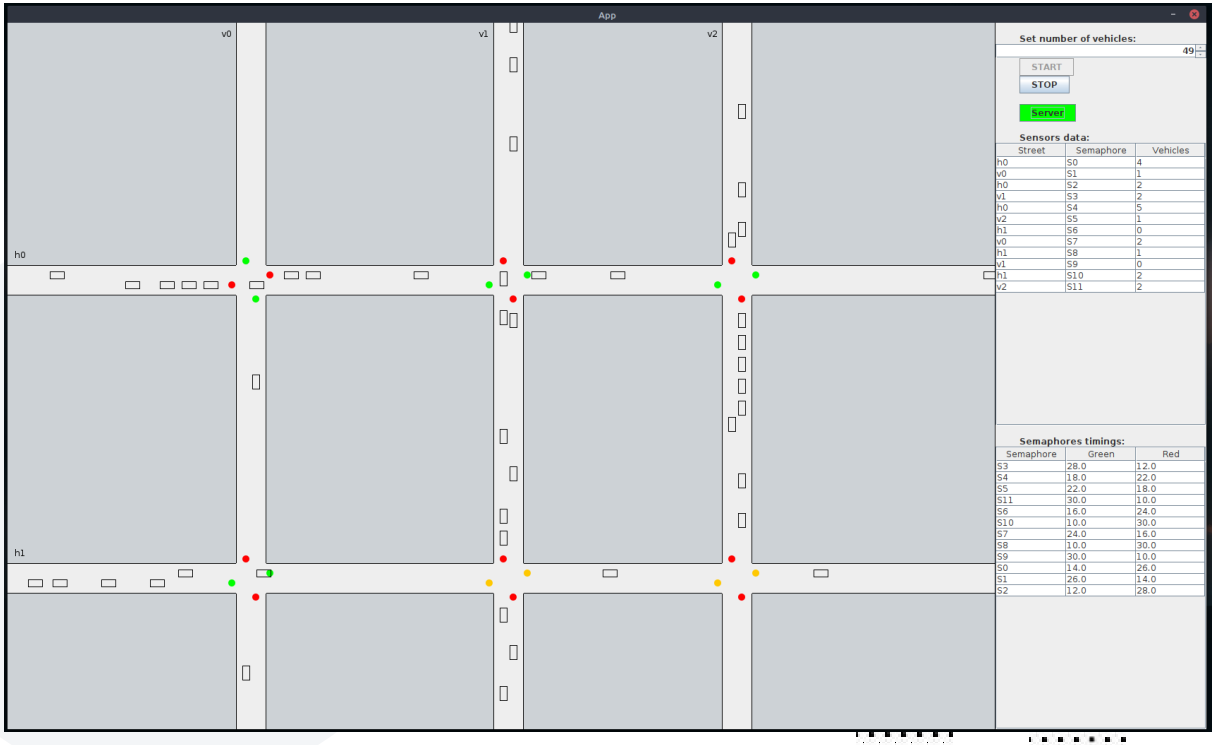
Map with no traffic

> The traffic has not started yet



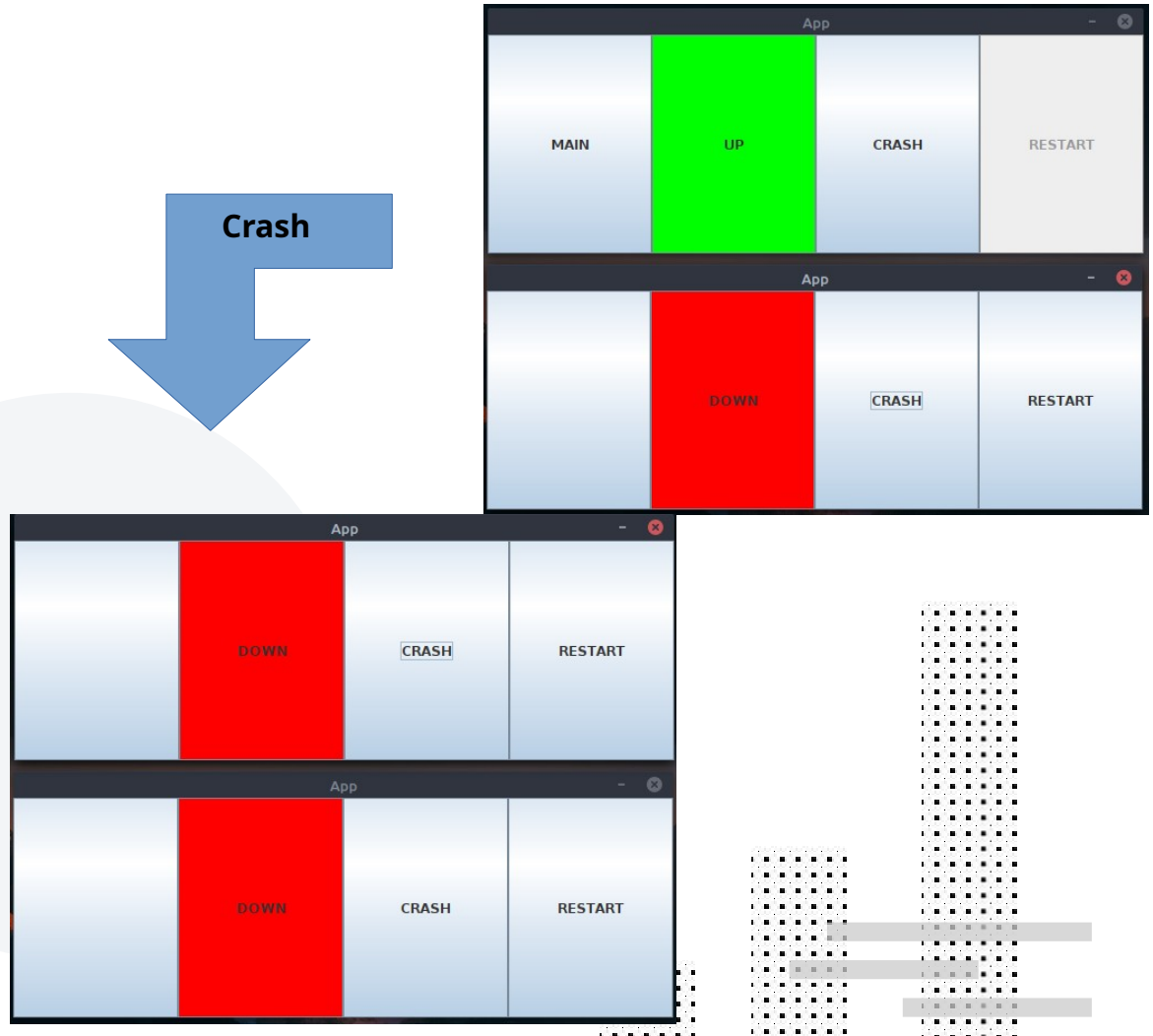
Traffic started

When the button start is pressed, the vehicles start appearing in the map



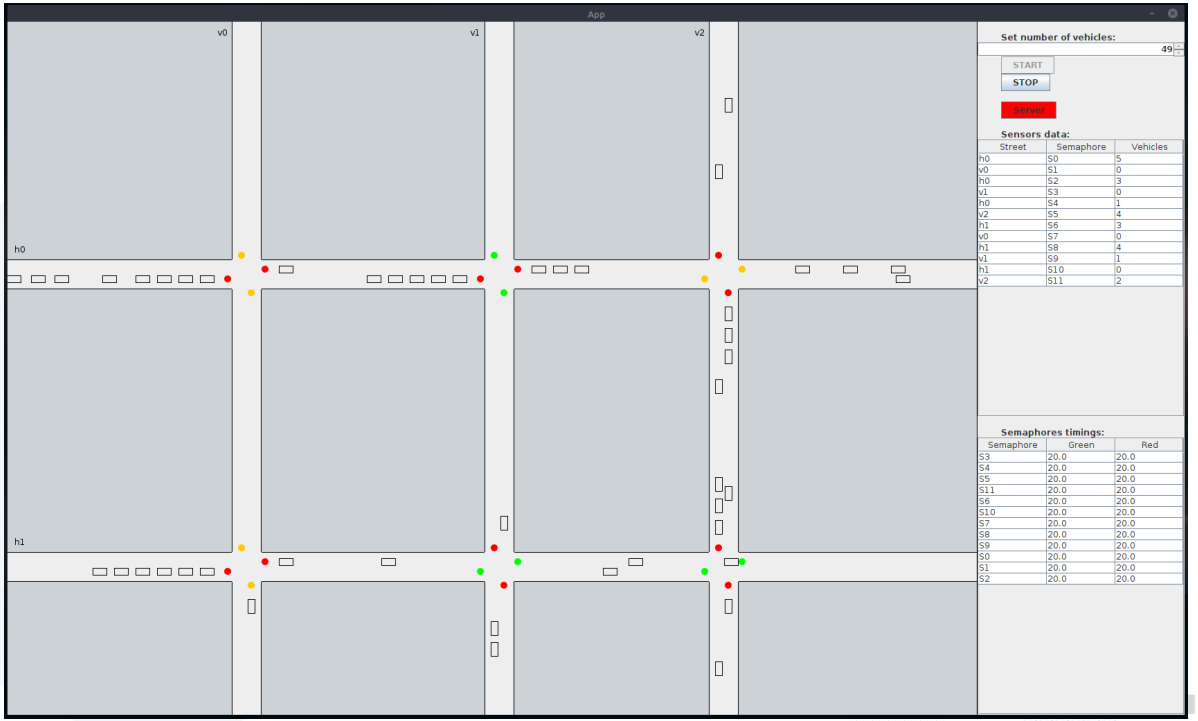
Server crash

At first, the main server crashes, and so the backup becomes the main. Then also the other one crashes



Sempahores go autonomous

> If both servers crash, the semaphores go in autonomous mode



Demo