A logo with red and black stripes

Description automatically generated

**Technical University of Cluj-Napoca**

Automation and Computer Science

Year IV, Semester I

Distributed Control Systems Project

Traffic Intersection Controller

**Team**: Nagy Timea, Nemes Raluca, Oprea Florin-Octavian

**Group**: 30343

# Specifications

According to the map given to each team, develop a controller for each intersection (plant), that controller is a closed-loop one (with the in(1..n) input channels that is connected to its intersection’s output channels op(1..n) and an Intersections (with the OPs output channels). The controller must have dynamic delays feature to extend the time of the green light in case of a traffic jam.

## The intersections

A map of a city

Description automatically generated

## Simplified Model

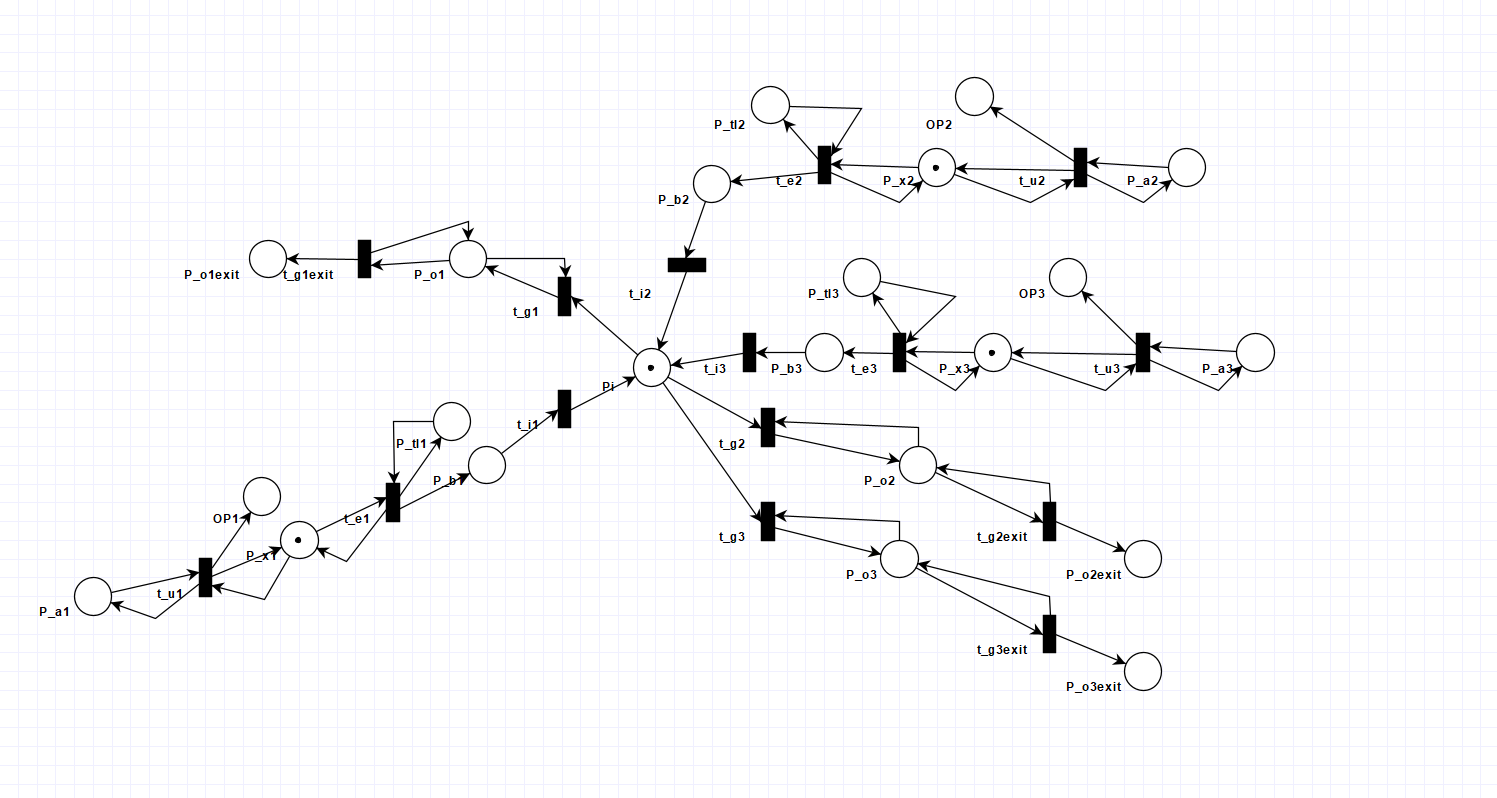
A diagram of a road with arrows and red dots

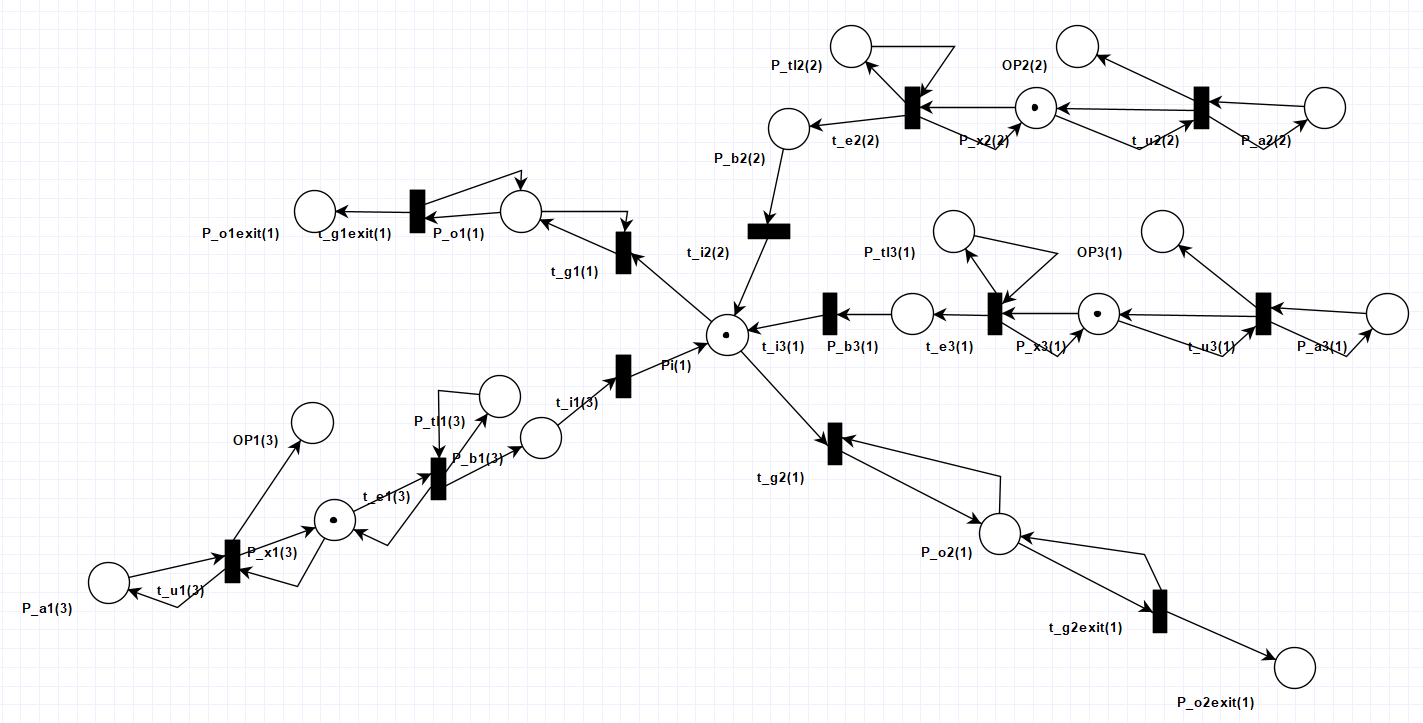
Description automatically generated

# Design

## OETPN Model for Plant

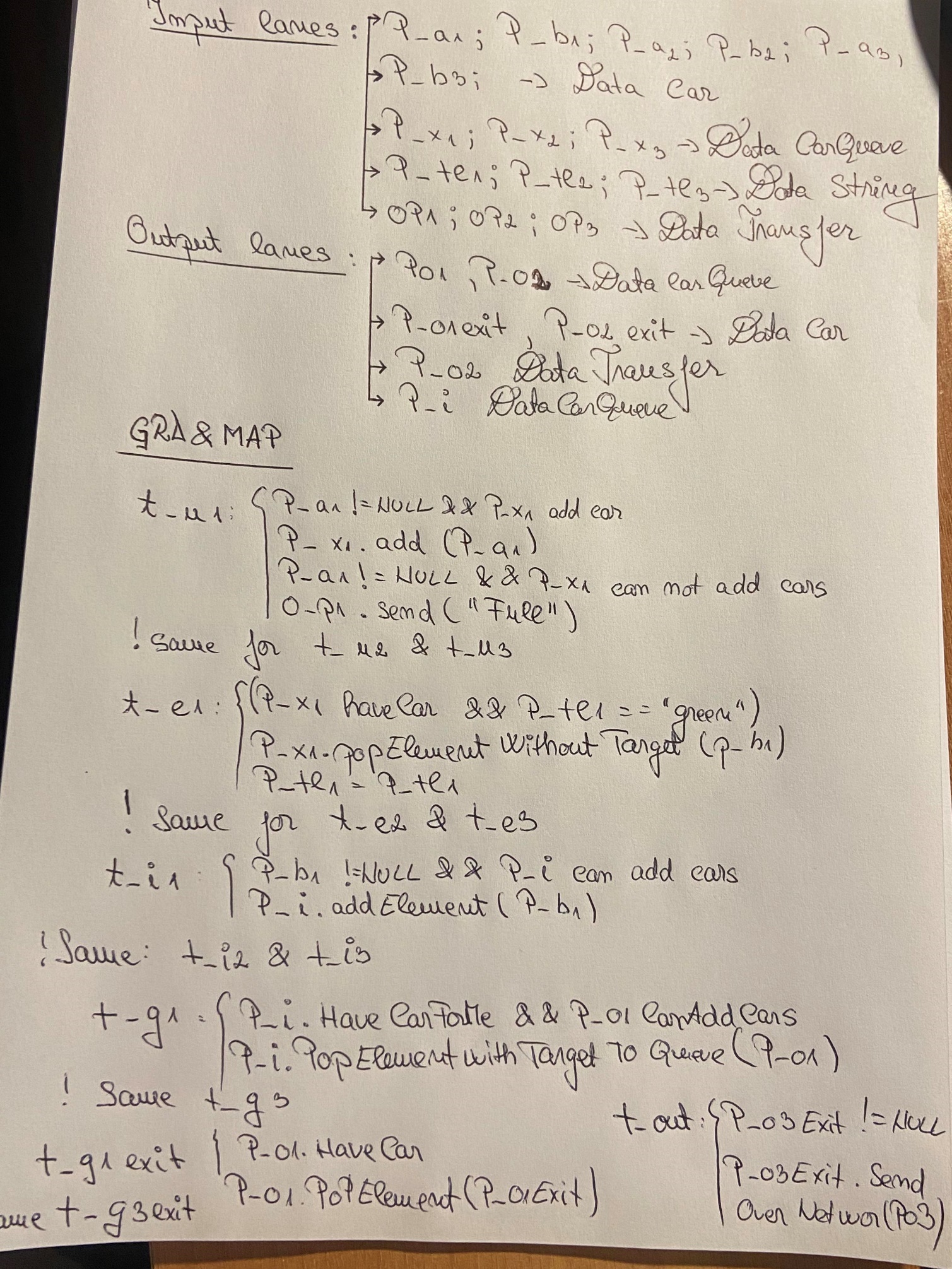
First Intersection

**

Second Intersection**

## Guards & Mappings for Plant

First Intersection – for the second intersection is the same but with one more input lane.



## OETPN Model for Controllers

First Intersection

* The 3 traffic lights are green one at a time.

A diagram of a network

Description automatically generated

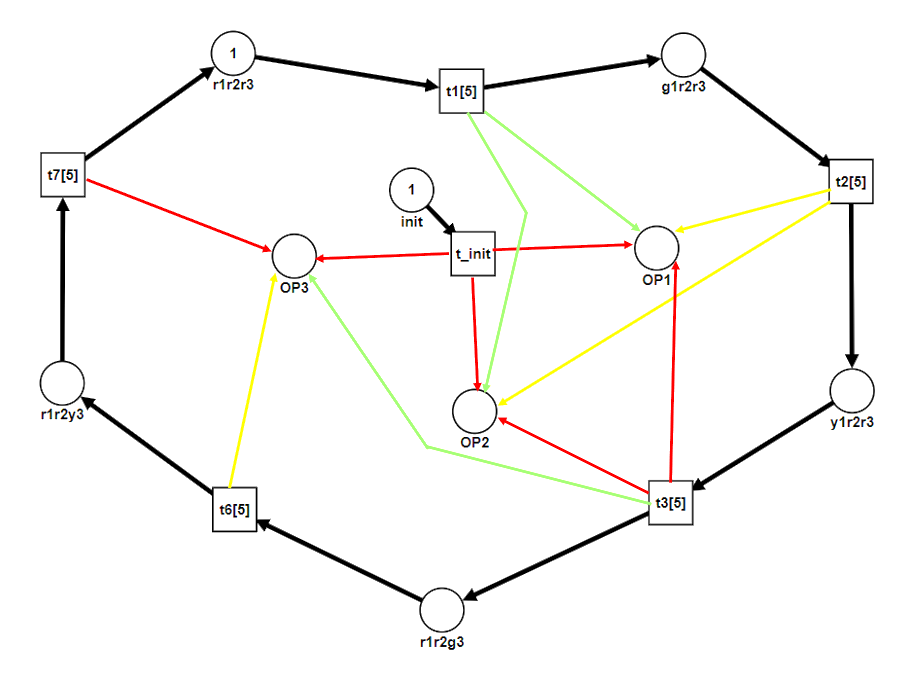
**In3**

**In2**

**In1**

Second Intersection

* OP3 traffic light is green: the other 2 must be red.
* OP3 TL is red: the other 2 can be green simultaneously.



**In2**

**In3**

**In1**

## Guards & Mappings for Controllers

A paper with writing on it

Description automatically generatedA paper with writing on it

Description automatically generatedA paper with writing on it

Description automatically generatedA paper with writing on it

Description automatically generated

## Component Diagram

A diagram of a traffic light controller

Description automatically generated

# Implementation

The entire system (plants + controllers) is implemented in Java using OERTPN Framework. The repository can be found on GitHub:

github.com/NTimea302/Traffic\_Intersection\_Controller

# Testing

We tested the application for the following use cases:

1. Send a car from the 1st intersection, that should go through the middle street and exit from one of the exit lanes from the 2nd intersection.

2. Traffic jam: for each intersection, create a traffic jam case by sending the maximum number of cars to the input lane of the intersection, start the controller, then send the last car. The controller should receive a signal from the plant (intersection) and the transition that is responsible for sending a yellow light to that lane where you input the cars to, should have changed the delay to 10 sec. Let the controller OETPN run until it reaches the same transition (2 loops) to show that the delay is changed back to 5 sec.

Screen shots from the execution

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated