LA Team: DCS Project Report

Year IV, AIA English, Group 30342&30341

Students: Malita Alin, Tudor Ada, Mosnegutu Vlad

Table of Contents

1. Introduction

1.1 Premise and requirements

The following report represents the detailed description of the project of team LA, involving two given intersection which require control through a traffic light system. This project is composed of 4 main sections:

1. Specifications: the illustration of the subject intersections through map and drawings.

2. Design: presentation of the OETPN models for the plants, controllers and connection street, including place types and guard maps. This section also includes the component diagram of the system.

3. Implementation: Git repository link to all the components of the project, especially the code and the activity history of the team members.

4. Testing: The demonstration of the functionality of the OETPN models and control through testing 2 requested scenarios

1.2 Specifications

Figure 1.1 Google Earth view

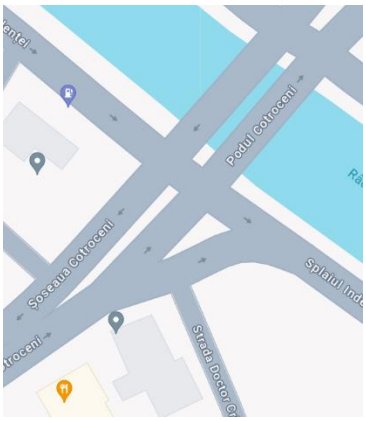

Figure 1.2 Intersection 1 (Pin1)

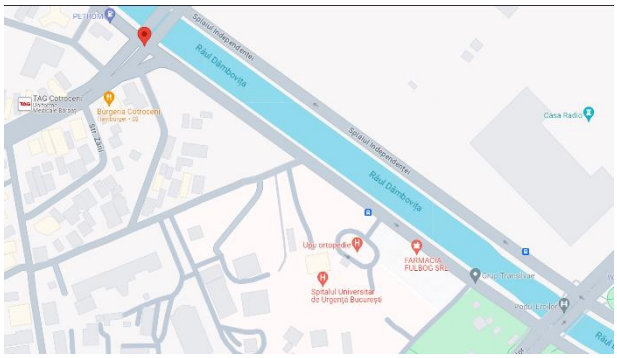

Figure 1.3 Connection Street

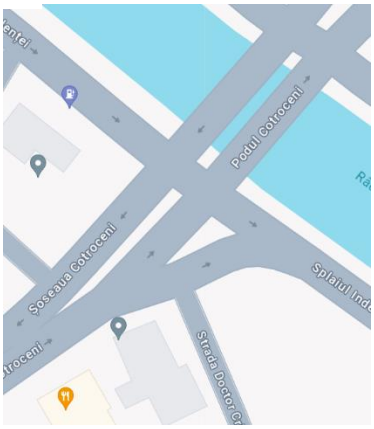

Figure 1.4 Intersection 2 (Pin2)

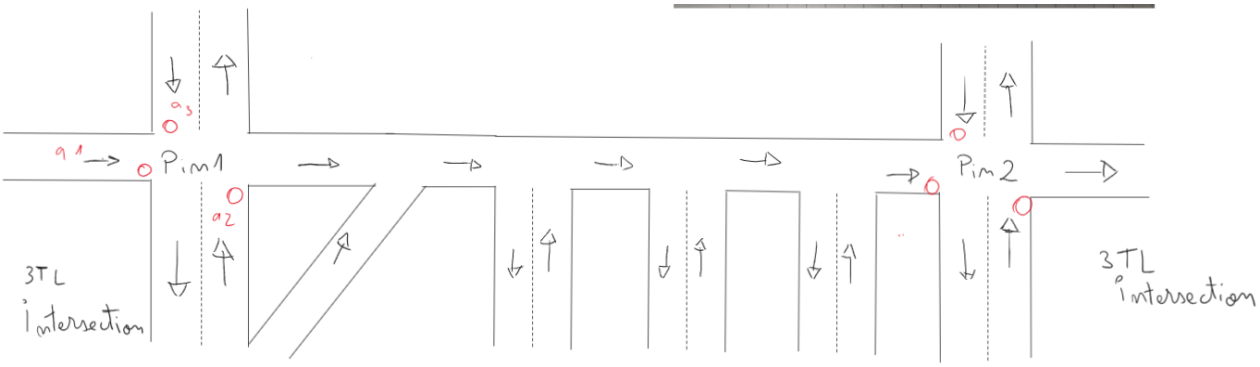

Figure 1.5 Illustrated Intersection System

Based on the given Intersections, the prospect of the project is the representation, modelling and control of the intersection using the OETPN_OERTPN java framework. Below the system is simplified through illustration fig1.5, the red dots representing the position of control places TL (traffic lights) which will dictate the flow of traffic inside the intersections. Both intersection have similar forms and require 3 control points. The traffic lights of the connection street are ignored.

2. Design

2.1 Intersections - Transitions

Since both intersections are extremely similar, their transitions are the same, with the exception of transition t_4n and place P_4N present only for Intersection1.

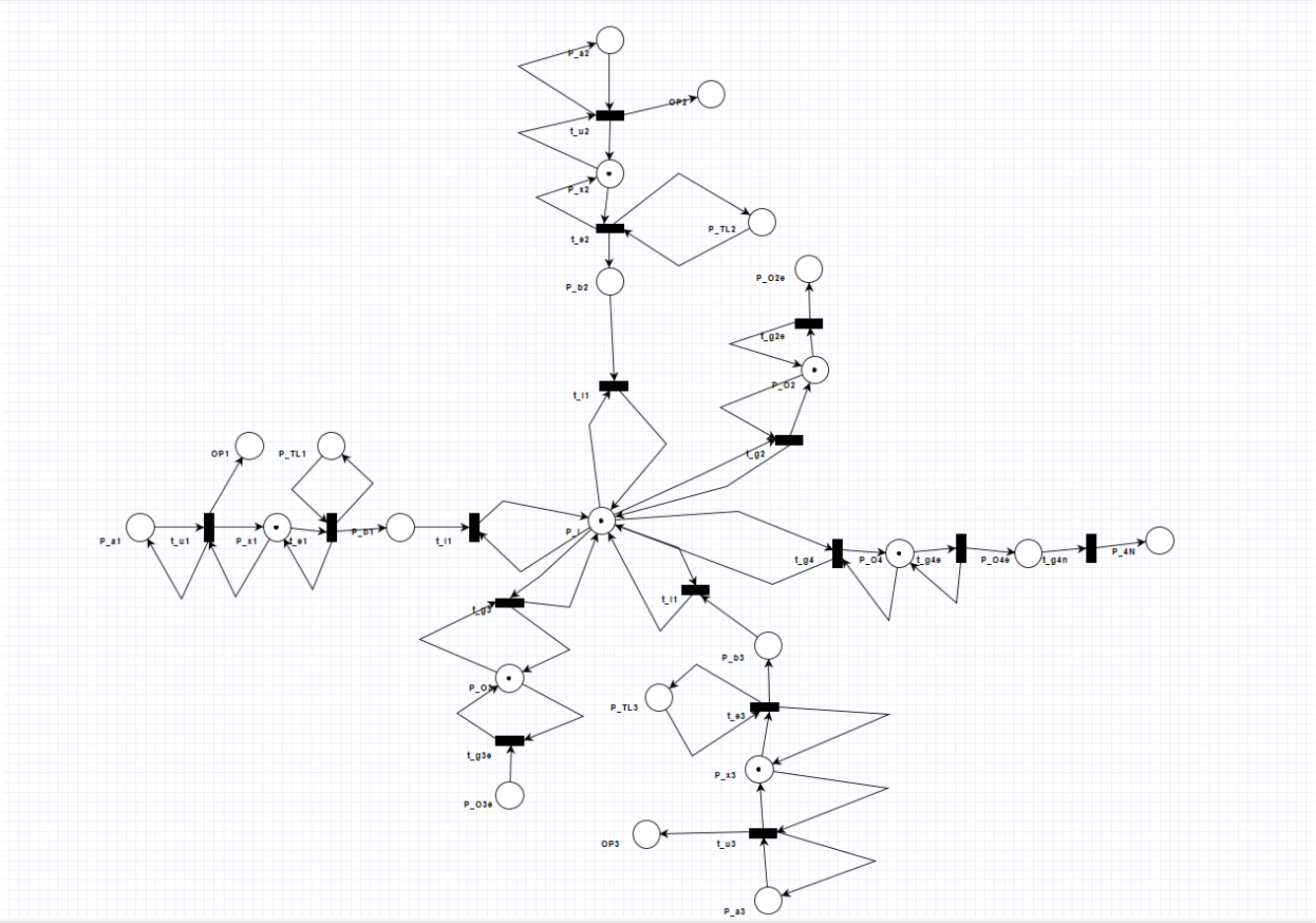

Figure 2.1.1 Intersection 1 Petri Net

PLACES:

- ➔ P__a1, P_a2, P_a3, P_b1,P_b2, P_b3, P_o2Exit, P_o3Exit, P_o4Exit = DataCar type
- ➔ OP1, OP2, OP3, P_4N = DataTransfer type
- ➔ P_x1, P_x2, P_x3, P_I, P_o2, P_o3, P_o4 = DataCarQueue type
- ➔ P_TL1, P_TL2, P_TL3 = DataString type

TRANSITIONS:

- t_u1: input place: P_a1, P_x1
  grd1: (m(P_a1) ≠ ϕ And m(P_x1).CanAddCars));
  map1: m(P_a1).addElement() (m(P_x1))
  grd2: (m(P_a1) ≠ ϕ And m(P_x1).CanNotAddCars));

map2: m(P_a1).Move() (m(P_a1)) ; m(OP1) .SendOverNetwork(full)

same logic applies to t_u2 and t_u3

- t_e1: input place: P_x1, P_TL1
  grd: (m(P_x1).HaveCar And m(P_TL1)=green);

  map: m(P_x1).PopElementWithoutTarget() (m(P_b1)); m(P_TL1).Move() m(P_TL1)

same logic applies to t_e2 and t_e3

- t_i1: input place: P__b1, P_I
  grd: (m(P_b1) ≠ ϕ And m(P_I).CanAddCars));

  map: m(P_b1).AddElement() (m(P_I))

same logic applies to t_i2 and t_i3

- t_g2: input place: P_I, P_o2
  grd: (m(P_I).HaveCar And m(P_o2).CanAddCars));

  map: m(P_I).PopELementWithTargetToQueue() (m(P_o2))

same logic applies to t_g3 and t_g4

- t_g2e: input place: P_o2
  grd: (m(P_o2).HaveCar));

  map: m(P_o2).PopElementWithoutTarget() (m(P_o2Exit))

same logic applies to t_g3e and t_g4e

- t_4N: input place: P_04e
  grd: (m(P_04e) ≠ ϕ);
  map: m(P_04e).SendOverNetwork() (m(P_4N) = m(P_4N)

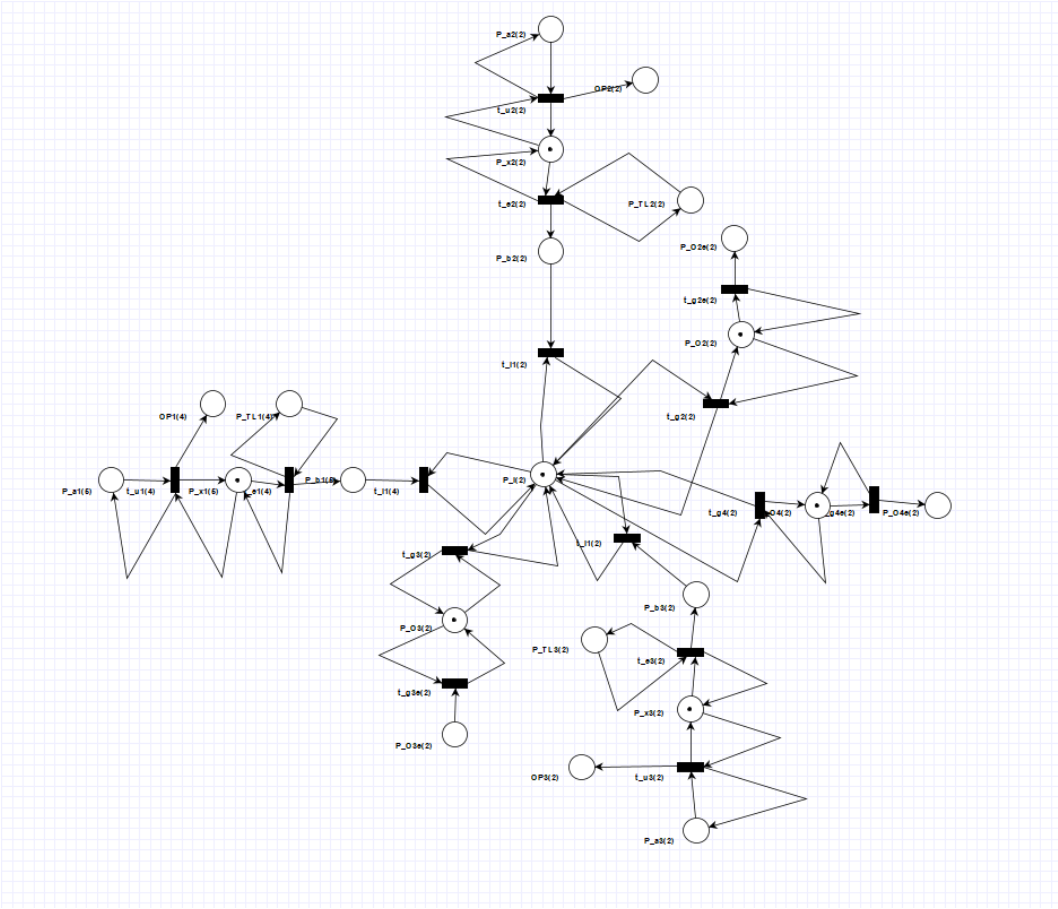

Figure 2.1.2 Intersection 2 Petri Net

Connecting Street – Transitions

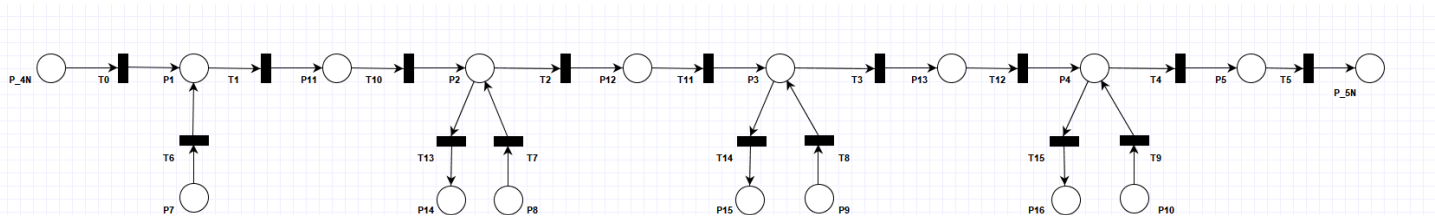

Figure 2.2.1 Connection Street PetriNet

PLACES:

➔ P_4N, P5, P7, P8, ….P15,P16 = DataCar type

➔ P_5N = DataTransfer type

➔ P1, P2, p3, P4 = DataCarQueue type

TRANSITIONS:

➔ First Group of transitions

- t0 : input place: P_4N

grd: (m(P_4N) ≠ ϕ And m(p1).CanAddCars));

map: m(P_4N).addElement() (m(p1))

- t6 : input place: p7
  grd: (m(p7) ≠ ϕ And m(p1).CanAddCars));

  map: m(p7).addElement() (m(p1))

- t7 : input place: p8
  grd: (m(p8) ≠ ϕ And m(p2).CanAddCars));

  map: m(p8).addElement() (m(p2))

- t8 : input place: p9
  grd: (m(p9) ≠ ϕ And m(p3).CanAddCars));

  map: m(p9).addElement() (m(p3))

- t9 : input place :p10
  grd: (m(p10) ≠ ϕ And m(p4).CanAddCars));

  map: m(p10).addElement() (m(p4))

t10, t11 and t12 will follow the same logic as the t0 ->t9 transitions

➔ Second Group

- t1 : input place: p1
  grd: (m(p1).HaveCarForMe);

  map: m(p1).PopElementWithTarget() (m(p11))

- t2 : input place: p2
  grd: (m(p2).HaveCarForMe);

  map: m(p2).PopElementWithTarget() (m(p12))

- t3 : input place: p3
  grd: (m(p3).HaveCarForMe);

  map: m(p3).PopElementWithTarget() (m(p13))

- t13 : input place: p2
  grd: (m(p2).HaveCarForMe);

map: m(p2).PopElementWithTarget() (m(p14))

- t14 : input place: p3
  grd: (m(p3).HaveCarForMe);

  map: m(p3).PopElementWithTarget() (m(p15))

transitions t1,t2,t3,t4 and t13,t14,t15 all have similar logic

➔ Transition t5: input place: p5
grd: (m(p5) ≠ ϕ);
map: m(p5).SendOverNetwork() (m(p5n) = m(p_a1)

2.3 Controllers - Transitions

Since the intersections are so similar, their controllers will have the same logic for their transitions and places

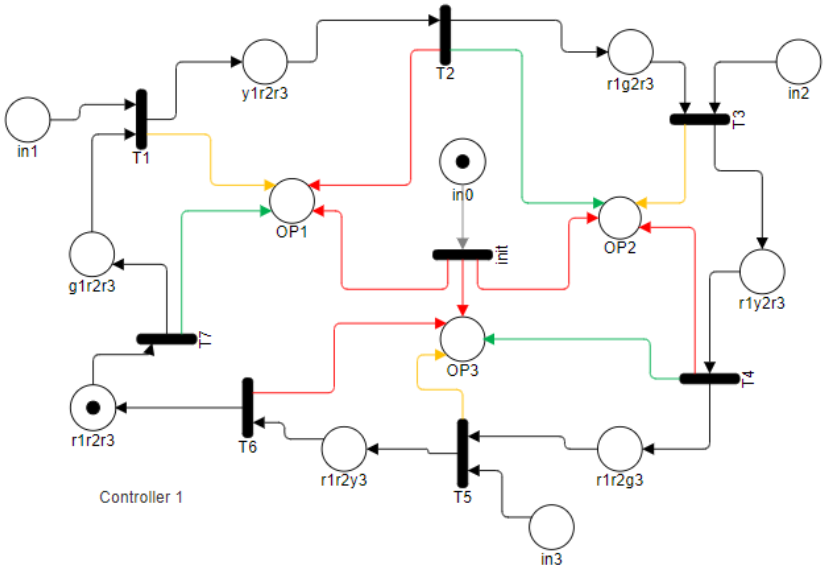

Figure 2.3.1 Controller 1 (same as Controller 2) Petri Net

PLACES:

➔ in0, in1, in2, in3 = DataString type
➔ OP1, OP2, OP3 = DataTransfer type
➔ r1r2r3,g1r2r3,y1r2r3,r1g2r3,r1y2r3,r1r2g3,r1r2y3 = DataString type

TRANSITIONS:

➔ init: input Place: in0
grd: (m(in0) ≠ ϕ);
map: m(in0).MakeNull; m(OP1).SendOverNetwork(in0)

m(OP2).SendOverNetwork(in0)
m(OP3).SendOverNetwork(in0)

➔ First Group of transitions

- t2 : input place: in1, g1r2r3
  grd1: (m(in1)= ϕ And (m(g1r2r3) ≠ ϕ));
  map1: m(g1r2r3).Move() (m(y1r2r3))
  m(OP1).SendOverNetwork(yellow)
  DynamicDelay(Five)

  grd2: (m(in1) ≠ ϕ And (m(g1r2r3) ≠ ϕ));
  map2: m(g1r2r3).Move() (m(y1r2r3))
  m(OP1).SendOverNetwork(yellow)
  DynamicDelay(Ten)

t1, t3 and t5 follow the same logic

➔ Second Group of transitions

➔ t2 : input place: y1r2r3
grd: (m(y1r2r3) ≠ ϕ);
map: m(y1r2r3).Move() (m(r1g2r3));
m(OP1).SendOverNetwork(red)
m(OP2).SendOverNetwork(green)

t2, t4 and t6 follow the same logic

2.4 Component diagram

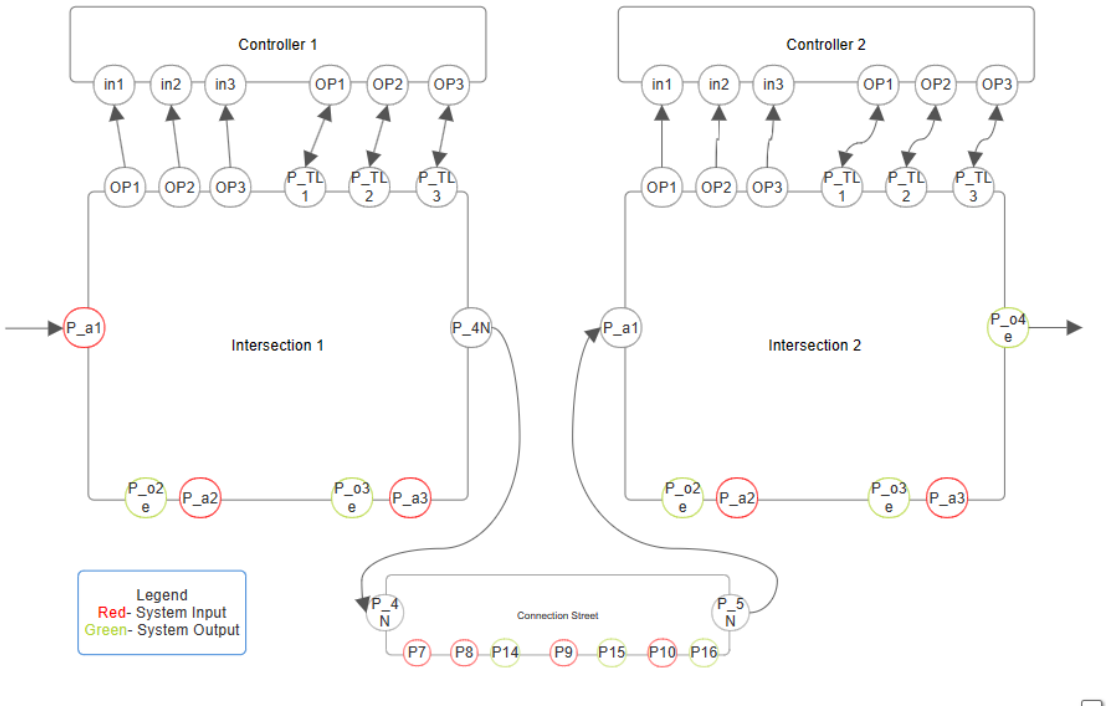

Figure 2.4.1 Component Diagram

As explained in fig.2.4.1, controllers and intersections communicate through a series of input and outputs, sending signals on which the evolution of the system depends. The connection street simply receives cars from Intersection_1 and allows them to travel into Intersection_2.

3. Implementation

- ➔ Inside the Git repository, the location of the project file can be found at the following address: OETPN_OERTPN_Framework/New OETPN/New OETPN/src/ProjectLA
- ➔ Testing Folder offers all images and log files related to the 2 tests required for the project's system
- ➔ The Component Diagram is in the main folder of the project
- ➔ The Drawn_Petri_Nets folder holds all petri nets included in this project, created with diagram tools

4. Testing

The testing sequence includes 2 tests: 1. Car travelling through intersections; 2. Car jam management

4.1 Test 1

For the first test, a DataCar object was sent using the input GUI, entering the system at Intersection_1: place P_a1. The car then travels through the intersection, needing to exit through P_o4exit into the Connection_Street. The car will ignore the exits to its right and will reach P_5N of the Connection_Street, from which it will enter Intersection_2 through place P_a1. Then, on the similar trajectory, it will enter Intersection_2 and exit through P_o4exit.

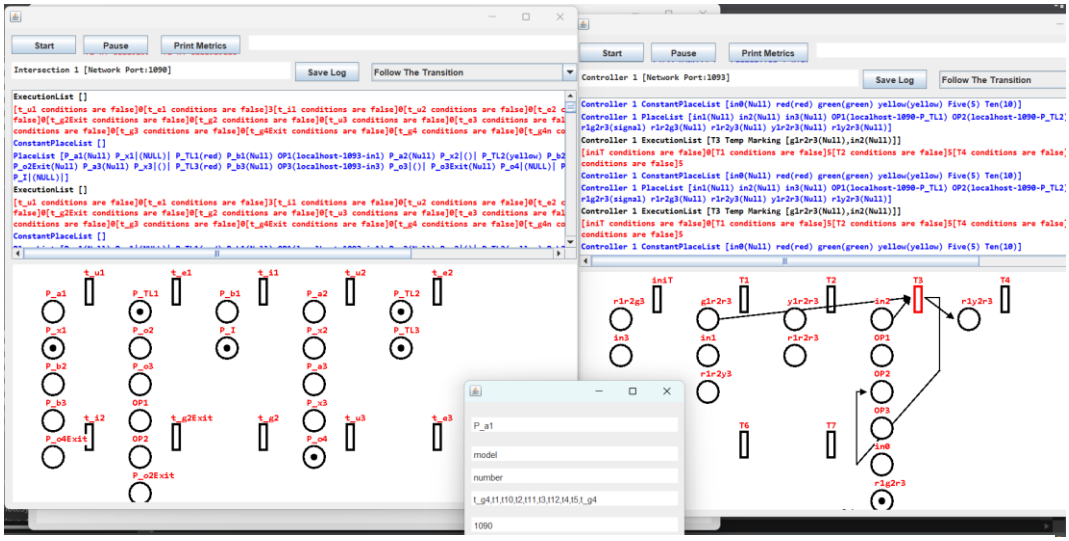

Figure 4.1.1 Test 1 – Intersection 1

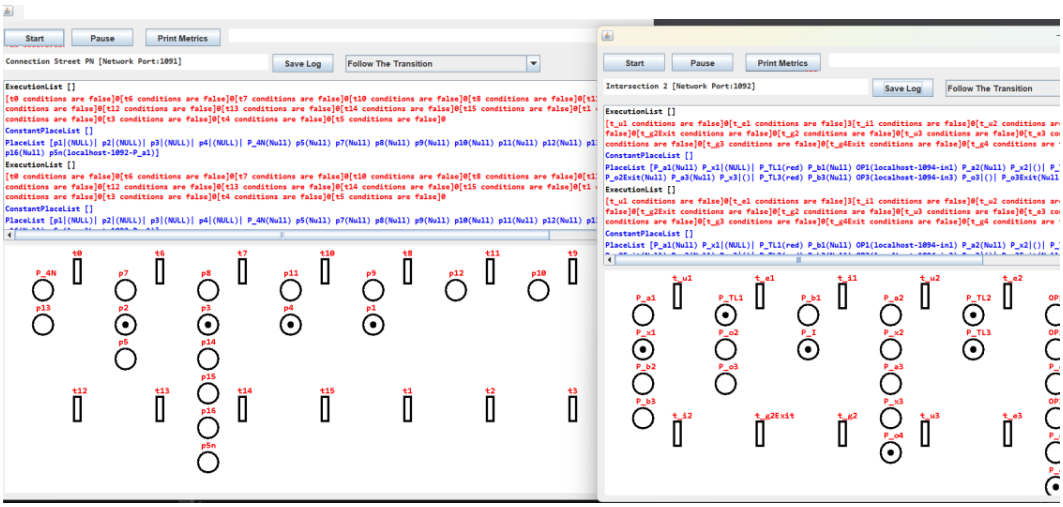

Figure 4.1.2 Test 1 –Connection Street

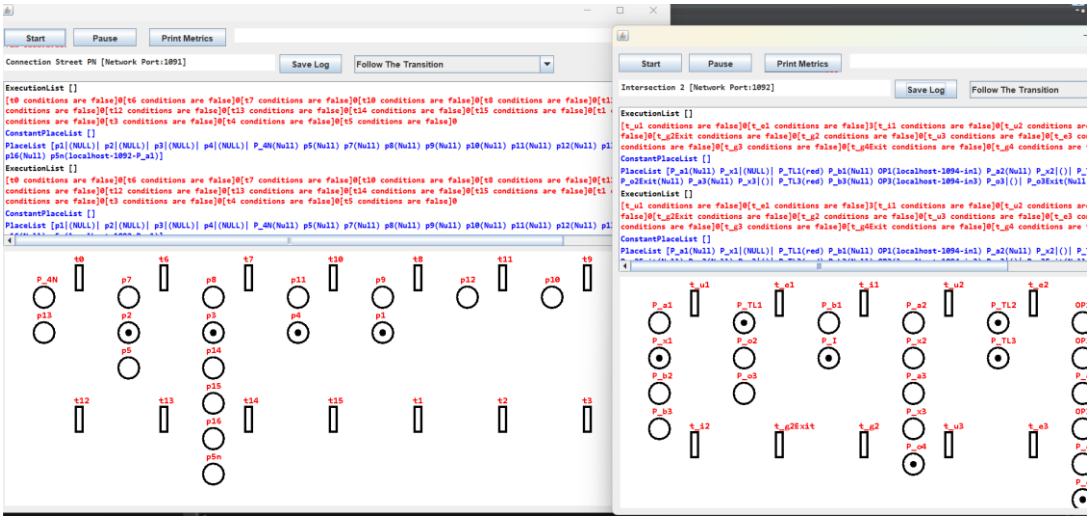

Figure 4.1.3 Test 1 – Intersection 2

4.2 Test 2

A traffic jam is created in both intersections by sending a number of cars higher than the capacity of the DataCarQueue, sending the 'full' signal to the corresponding input of the controller. In turn, the controller will continue to the transition that is responsible for sending a yellow light to that lane where you input the cars to, should have changed the 10 (Ten) seconds delay. After a few loops, it will return the delay to 5 (Five) seconds.

```
Controller 1 ConstantPlaceList [in0(Null)  red(red)  green(green)  yellow(yellow)  Five(5)  Ten(10)]
Controller 1 PlaceList [in2(Null)  in3(Null)  OP1(localhost-1090-P_TL1)  OP2(localhost-1090-P_TL2)  OP3(localhost-1090-P_TL3)  r1r2r3(Null)  g1r2r3(Null)
r1g2r3(Null)  r1r2g3(Null)  r1r2y3(Null)  y1r2r3(Null)  r1y2r3(Null)]
Controller 1 ExecutionList [T2 Temp Marking [y1r2r3(signal)]]
[iniT conditions are false]0[T1 conditions are false]10[T3 conditions are false]5[T4 conditions are false]5[T5 conditions are false]5[T6 conditions are false]5[T7
conditions are false]5
Controller 1 ConstantPlaceList [in0(Null)  red(red)  green(green)  yellow(yellow)  Five(5)  Ten(10)]
Controller 1 PlaceList [in2(Null)  in3(Null)  OP1(localhost-1090-P_TL1)  OP2(localhost-1090-P_TL2)  OP3(localhost-1090-P_TL3)  r1r2r3(Null)  g1r2r3(Null)
r1g2r3(Null)  r1r2g3(Null)  r1r2y3(Null)  y1r2r3(Null)  r1y2r3(Null)]
Controller 1 ExecutionList [T2 Temp Marking [y1r2r3(signal)]]
[iniT conditions are false]0[T1 conditions are false]10[T3 conditions are false]5[T4 conditions are false]5[T5 conditions are false]5[T6 conditions are false]5[T7
conditions are false]5
Controller 1 ConstantPlaceList [in0(Null)  red(red)  green(green)  yellow(yellow)  Five(5)  Ten(10)]
Controller 1 PlaceList [in2(Null)  in3(Null)  OP1(localhost-1090-P_TL1)  OP2(localhost-1090-P_TL2)  OP3(localhost-1090-P_TL3)  r1r2r3(Null)  g1r2r3(Null)
r1g2r3(Null)  r1r2g3(Null)  r1r2y3(Null)  y1r2r3(signal)  r1y2r3(Null)]
Controller 1 ExecutionList [T1 Temp Marking [g1r2r3(signal),in1(full)]]
[iniT conditions are false]0[T2 conditions are false]5[T3 conditions are false]5[T4 conditions are false]5[T5 conditions are false]5[T6 conditions are false]5[T7
conditions are false]5
```

Figure 4.2.1 Terminal Log for Controller 1

```
Controller 2 ConstantPlaceList [in0(Null)  red(red)  green(green)  yellow(yello
Controller 2 PlaceList [in2(Null)  in3(Null)  OP1(localhost-1092-P_TL1)  OP2(lo
r1g2r3(Null)  r1r2g3(Null)  r1r2y3(Null)  y1r2r3(Null)  r1y2r3(Null)]
Controller 2 ExecutionList [T2 Temp Marking [y1r2r3(signal)]]
[iniT conditions are false]0[T1 conditions are false]10[T3 conditions are false
conditions are false]5
Controller 2 ConstantPlaceList [in0(Null)  red(red)  green(green)  yellow(yello
Controller 2 PlaceList [in2(Null)  in3(Null)  OP1(localhost-1092-P_TL1)  OP2(lo
r1g2r3(Null)  r1r2g3(Null)  r1r2y3(Null)  y1r2r3(signal)  r1y2r3(Null)]
Controller 2 ExecutionList [T1 Temp Marking [g1r2r3(signal),in1(full)]]
```

```
Controller 2 ConstantPlaceList [in0(Null)  red(red)  green(green)  yellow(yellow)  Five(5)  Ten(10)]
Controller 2 PlaceList [in2(Null)  in3(Null)  OP1(localhost-1092-P_TL1)  OP2(localhost-1092-P_TL2)  OP3(localhost-1092-P_TL3)  r1r2r3(Null)  g1r2r3(Null)
r1g2r3(Null)  r1r2g3(Null)  r1r2y3(Null)  y1r2r3(Null)  r1y2r3(Null)]
Controller 2 ExecutionList [T1 Temp Marking [g1r2r3(signal),in1(Null)]]
[iniT conditions are false]0[T2 conditions are false]5[T3 conditions are false]5[T4 conditions are false]5[T5 conditions are false]5[T6 conditions are false]5[T7
conditions are false]5
Controller 2 ConstantPlaceList [in0(Null)  red(red)  green(green)  yellow(yellow)  Five(5)  Ten(10)]
Controller 2 PlaceList [in2(Null)  in3(Null)  OP1(localhost-1092-P_TL1)  OP2(localhost-1092-P_TL2)  OP3(localhost-1092-P_TL3)  r1r2r3(Null)  g1r2r3(Null)
r1g2r3(Null)  r1r2g3(Null)  r1r2y3(Null)  y1r2r3(Null)  r1y2r3(Null)]
Controller 2 ExecutionList [T1 Temp Marking [g1r2r3(signal),in1(Null)]]
[iniT conditions are false]0[T2 conditions are false]5[T3 conditions are false]5[T4 conditions are false]5[T5 conditions are false]5[T6 conditions are false]5[T7
conditions are false]5
Controller 2 ConstantPlaceList [in0(Null)  red(red)  green(green)  yellow(yellow)  Five(5)  Ten(10)]
Controller 2 PlaceList [in2(Null)  in3(Null)  OP1(localhost-1092-P_TL1)  OP2(localhost-1092-P_TL2)  OP3(localhost-1092-P_TL3)  r1r2r3(Null)  g1r2r3(signal)
r1g2r3(Null)  r1r2g3(Null)  r1r2y3(Null)  y1r2r3(Null)  r1y2r3(Null)]
Controller 2 ExecutionList [T7 Temp Marking [r1r2r3(signal)]]
[iniT conditions are false]0[T1 conditions are false]10[T2 conditions are false]5[T3 conditions are false]5[T4 conditions are false]5[T5 conditions are false]5[T6
conditions are false]5
Controller 2 ConstantPlaceList [in0(Null)  red(red)  green(green)  yellow(yellow)  Five(5)  Ten(10)]
Controller 2 PlaceList [in2(Null)  in3(Null)  OP1(localhost-1092-P_TL1)  OP2(localhost-1092-P_TL2)  OP3(localhost-1092-P_TL3)  r1r2r3(Null)  g1r2r3(Null)
r1g2r3(Null)  r1r2g3(Null)  r1r2y3(Null)  y1r2r3(Null)  r1y2r3(Null)]
Controller 2 ExecutionList [T7 Temp Marking [r1r2r3(signal)]]
[iniT conditions are false]0[T1 conditions are false]10[T2 conditions are false]5[T3 conditions are false]5[T4 conditions are false]5[T5 conditions are false]5[T6
conditions are false]5
Controller 2 ConstantPlaceList [in0(Null)  red(red)  green(green)  yellow(yellow)  Five(5)  Ten(10)]
Controller 2 PlaceList [in2(Null)  in3(Null)  OP1(localhost-1092-P_TL1)  OP2(localhost-1092-P_TL2)  OP3(localhost-1092-P_TL3)  r1r2r3(Null)  g1r2r3(Null)
r1g2r3(Null)  r1r2g3(Null)  r1r2y3(Null)  y1r2r3(Null)  r1y2r3(Null)]
Controller 2 ExecutionList [T7 Temp Marking [r1r2r3(signal)]]
[iniT conditions are false]0[T1 conditions are false]10[T2 conditions are false]5[T3 conditions are false]5[T4 conditions are false]5[T5 conditions are false]5[T6
conditions are false]5
```

Figure 4.2.2. Terminal Log for Controller 2