ABSRRACT:

Improving the efficiency of traffic signal control is an effective way to alleviate traffic congestion at signalized intersections. To achieve effective management of the system-wide traffic flows, many research tends to focus on applying reinforcement learning (RL) techniques for collaborative traffic signal control in a traffic road network, and most them assume that the signal controllers can collect all instantaneous vehicular features without delay. To fill the gap, we propose an RL-based cooperative traffic signal control scheme considering the data transmission delay issue in a traffic road network. we design our new RL agents to cooperatively control the traffic signals by improving the reward and state representation based on the state-of-the-art max-pressure control theory. propose a traffic state prediction method to address the data transmission delay issue by decreasing the discrepancy between the real-time and delayed traffic conditions.

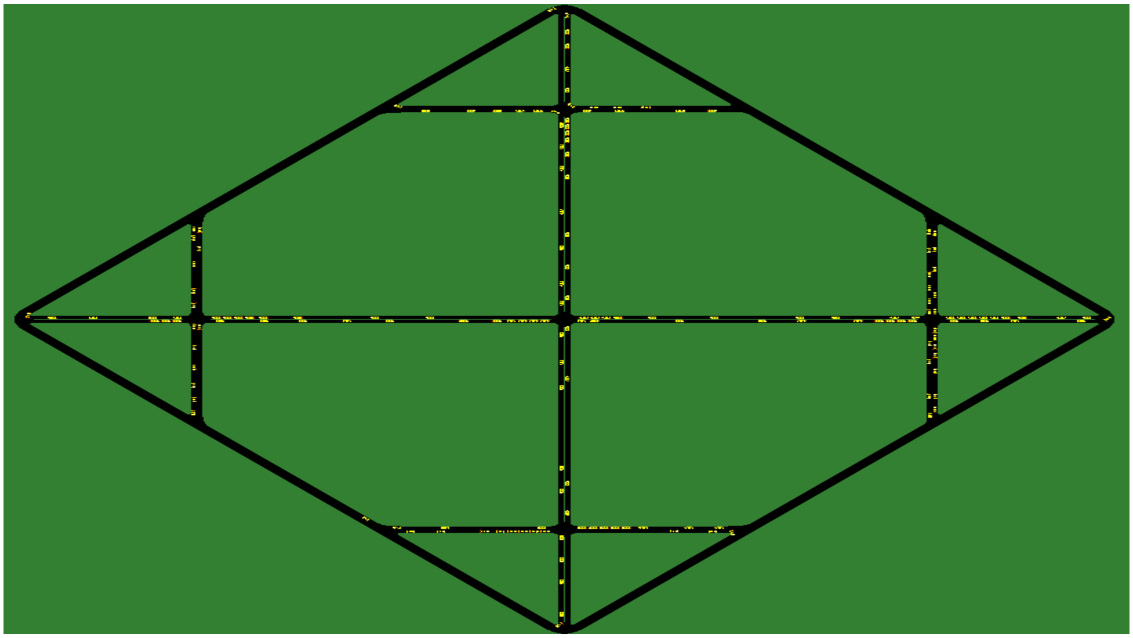
INTRODUCTION:

In this work, we will build a collaborative traffic light control system using reinforcement learning techniques taking into account the delay in transmission of control messages by vehicles. Each vehicle sends its information to the agent controlling the intersection where the vehicle is located. The messages are then aggregated and the state is generated which the agent will use to decide whether to open or close the controlled lanes containing the vehicles. How to design the agent, the environment, and all the other components, we'll go into more detail in the next section, and the section that follows will contain the results of experiments and comparisons

METHODOLOGY:

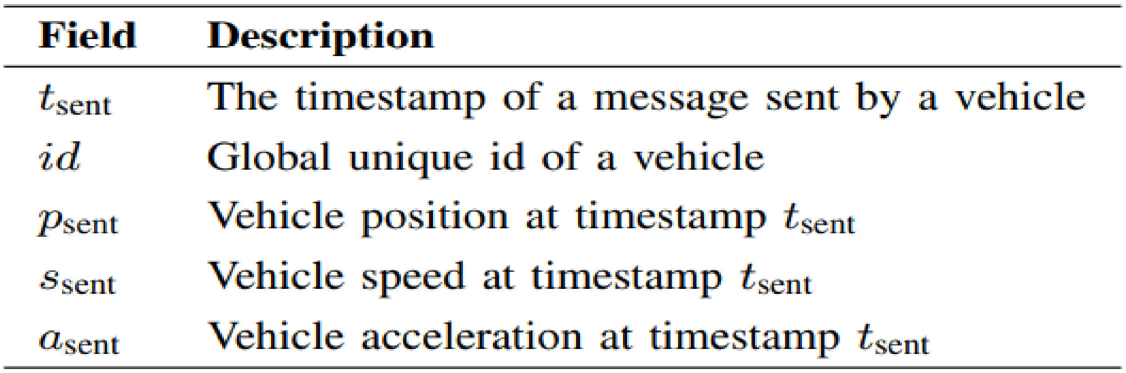
I. Environment design:

The environment consists of five interconnecting intersections, each intersection controlling four lanes, all intersections of left-turn permitted design. As in this design, every two opposite lanes are opened together and the remaining two lanes are closed.



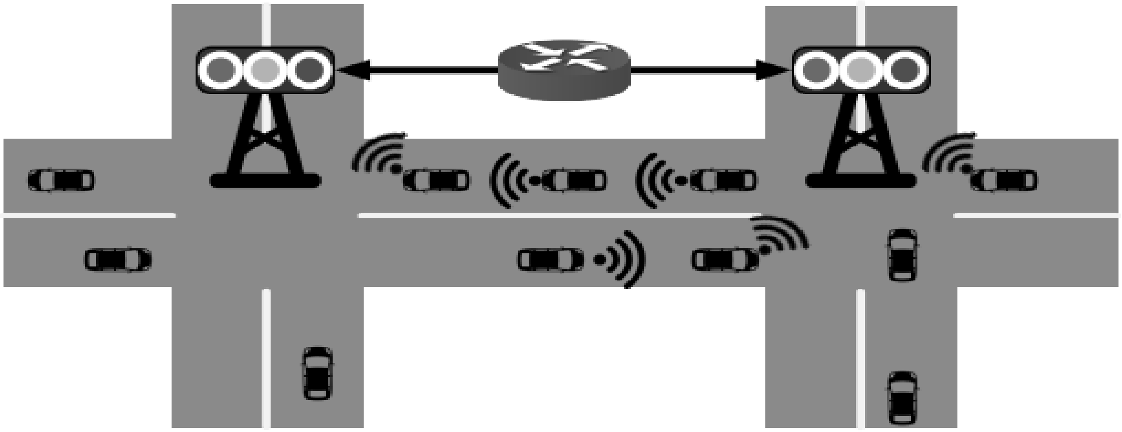
II. Obtaining information from the environment:

Each vehicle sends control messages to the agent responsible for the intersection it is in. The message contains five information about the vehicle, the vehicle ID. Time of sending the message, location of the vehicle at the time of transmission, vehicle speed, vehicle acceleration.



Field	**Description**
t_{sent}	The timestamp of a message sent by a vehicle
id	Global unique id of a vehicle
p_{sent}	Vehicle position at timestamp t_{sent}
s_{sent}	Vehicle speed at timestamp t_{sent}
a_{sent}	Vehicle acceleration at timestamp t_{sent}

After the messages from all the vehicles arrive, the stage of converting them into information about the lanes that contain the vehicles comes. First, the messages of each lane will be sorted according to the position of the vehicle, then the messages of each lane will be processed, as each lane will be divided into three parts, for each part we calculate the number of cars in it, average speed, average acceleration. But this is not all the information that the agent needs during the learning process, so we calculate for each lane the number of available places in it and the number of vehicles in the lanes leading to this lane.



III. Agent design:

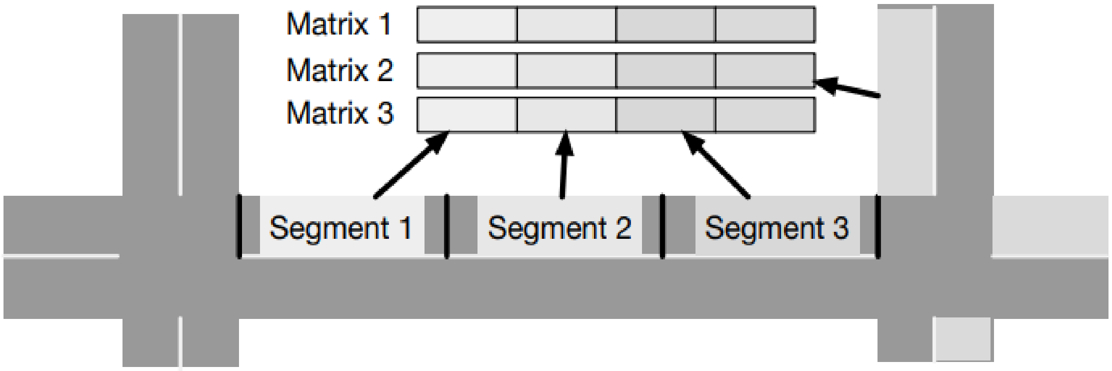
The agent learns and decides using the DQN algorithm. This algorithm combines the Q-learning algorithm and deep neural networks (DNNs). As is well known in the field of artificial intelligence, DNNs are great non-linear function approximators. Thus, DNNs are used to approximate the Q-function, replacing the need for a table to store the Q-values.

Bellman's Equation for DQN:

$$Q(s, a; \theta) = r + \gamma \max_{a'} Q(s', a'; \theta')$$

- ✓ *State:*

  Each lane we divide into three segment, for each section we calculate the average speeds of the cars in it, the average accelerations and the number of cars, then we put these calculations in three matrices M1, M2, M3 each matrix of dimension 4×L where L indicates the number of lanes and the fourth dimension in each line contains the same information calculated for each segment of the current lane, but for the adjacent corridor leading to the current lane.



Then for each of the three matrices we flatten and then concatenate, So we get the state.

Taking into account the delay in the arrival of the message by the vehicle is an important part in order to determine the exact position of the vehicle in order to know to which segment of the corridor the information of this vehicle will be attributed to use this information later in establishing the state. Assuming that at time step t, an agent has received a sequence of messages M_V of upcoming vehicles V from adjacent intersection j in the previous control cycle, the delay of the message M_{V_i} from vehicle V_i to the intersection can be calculated as follows:

$$d(V_i) = t - \ M_{V_i}(t_{sent})$$

Since the speed of vehicle V_i is constantly changing, the control agent does not know the instantaneous speed of the vehicle V_i during the delay period $d(V_i)$. Hence, we assume that the received vehicle speed s_{sent} is the average speed of the vehicle V_i in the delay period. Based on the speed assumption, the expected moving distance of V_i can be estimated as follows:

$$E(dis(V_i)) = d(V_i) \times \ M_{V_i}(s_{sent})$$

We must consider the angle of the lane with the coordinate axis, since the distance traveled during the delay must be added to the position on the x-axis or to the position on the y-axis or subtracted from the position on the x-axis or from the position on the y-axis depending on the angle. If the angle is 90 we add the distance to the position on the x axis, if it is 270 we subtract, and if the angle is 0 we add the distance to the position on the y axis, if it is

180 we subtract. However, if the lane signal is red and the vehicle is the leading vehicle, then its position must be ascertained if it is standing at the end of the lane or has not reached its end yet, in this case there is no distance traveled because the signal is red and the vehicle is at the end of the lane, so it is standing waiting for the signal to turn to complete its path But if it has not yet reached the end of the lane, its position must be predicted after the delay, while in the event that the vehicle is not the leading vehicle, it must be ensured that its new position does not exceed the position of the vehicle before it.

$$x_{new}|y_{new} = x_{old}|y_{old} \mp E(dis(V_i))$$

After predicting the positions (the positions of the vehicles after taking into account the delay), and after we divided each lane into three segments, we attribute each vehicle to the segment whose new position is within its limits (the starting position and the ending position).

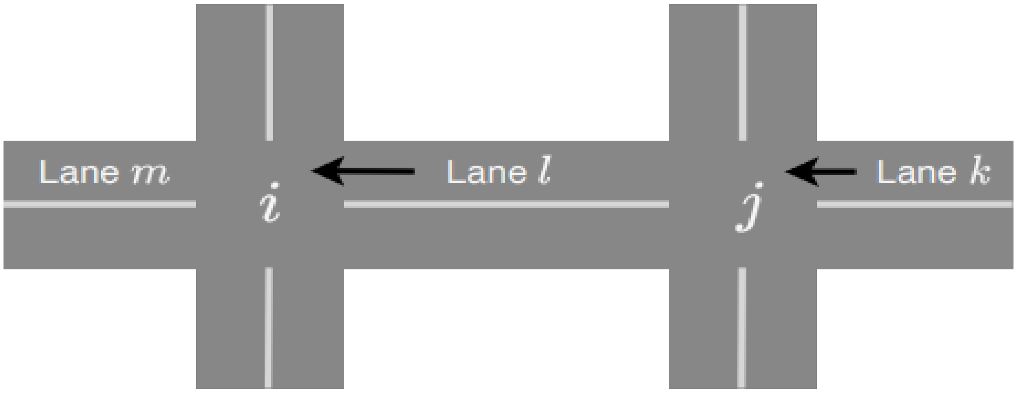
- ✓ *Action:*

Since the design of the intersections is of type left-turn permitted, we have two action (1, 0) or (0, 1), Where 1 indicates the green signal and 0 indicates the red signal. Every 30 seconds the agent giving a new action and calculating the reward for the previous action. also in this design the action applies to both north and south or east and west together.

- ✓ *Reward:*

we design a reward model based on MP that measures the pressure of vehicular movements at an intersection based on incoming and outgoing queue length measurements.

We assume that a traffic environment contains an intersection i and an adjacent intersection j ∈ NB_i , where the NB_i denotes the collection of neighbouring intersections of the intersection i.



Then, we define the pressure of a vehicular movement from incoming lane l to outgoing lane m at time step t as follows:

$$P_t(l,m) = \sum_{l\in L_j} q_{j,t}(l,m) - \; c_t(m) + \sum_{k\in L_j} s_{j,t}(k,l).\, q_t(k,l)$$

where $q_t(l,m)$ denotes the number of vehicles moving from lane l to lane m, l_i denotes the incoming lanes of adjacent intersection j. $s_{i,t}(k,l)$ = {1, 0} is a binary variable denoting the traffic signals of the vehicular movements from lane k to lane l. The value 1 means that the signal is green for the vehicular movement (k, l) at time step t; otherwise, it returns 0. $c_t(m)$ denotes the vacancy of exiting lane m that is available for incoming vehicles at time step t. The values of $c_t(m)$ describe the capability of reducing the intersection's pressure. When $c_t(m)$ = 0, the lane m is full of vehicles and no space for feeding vehicles from upstream lanes. $c_t(m)$ > 0 denotes the number of vehicles that can go through the intersection and enter into lane m. Based on the pressure measurements described above, the pressure of intersection i at time step t is defined as:

$$P_t = \sum_{l,m\in L_i} P_t(l,m)$$

The objective of the traffic signal controller is to minimize the total pressure of the intersection. Therefore, we define the reward r_t at time step t as follows:

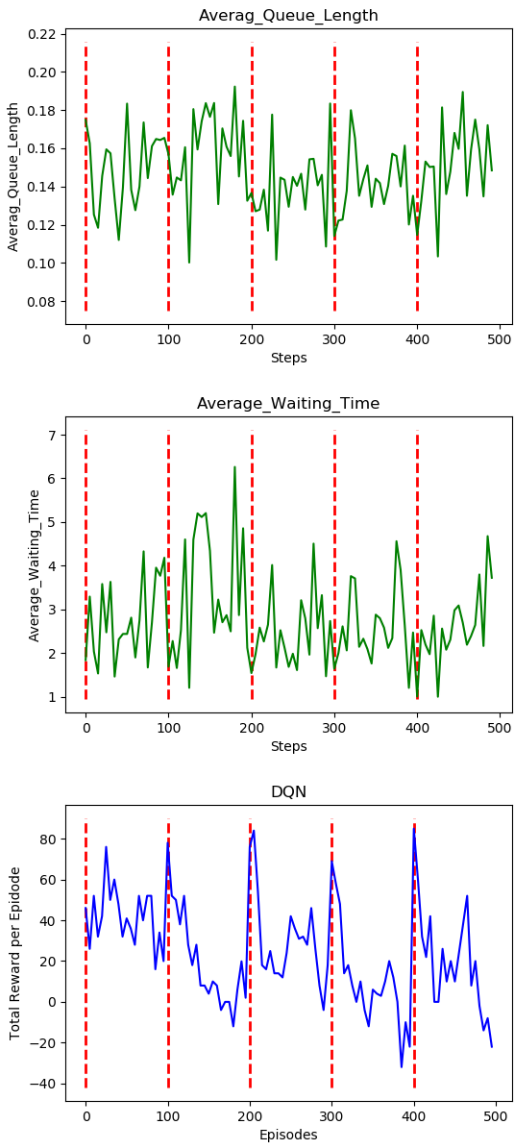
$$r_t = -P_t$$

IV. Results and Experiments:

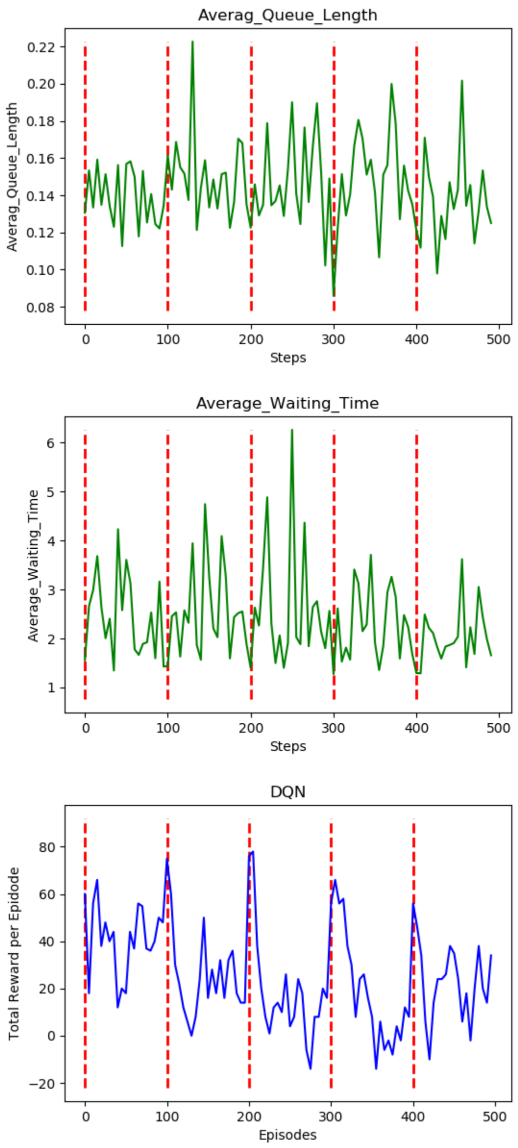
To test our method and ensure that it is effective and has a positive effect in controlling traffic congestion, we compared our method with different methods.

✓ *without regard for the delay:*

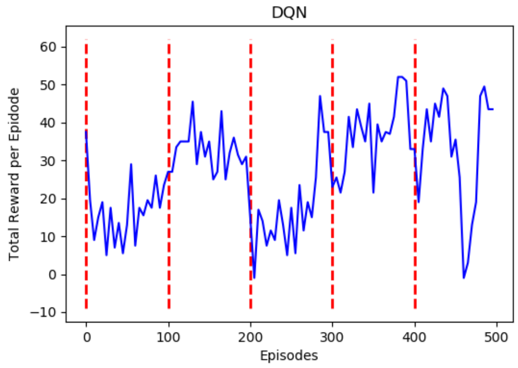
Out method but without taking into account the delay in arrival time	Our method



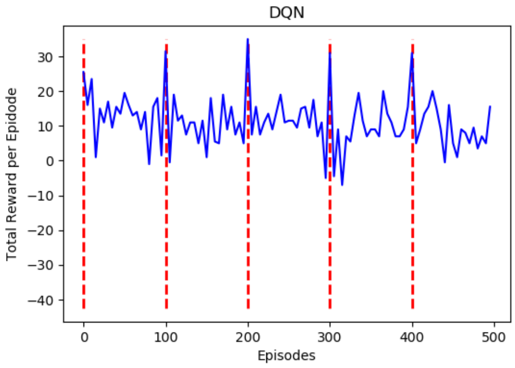
Reward curve for agent number 0



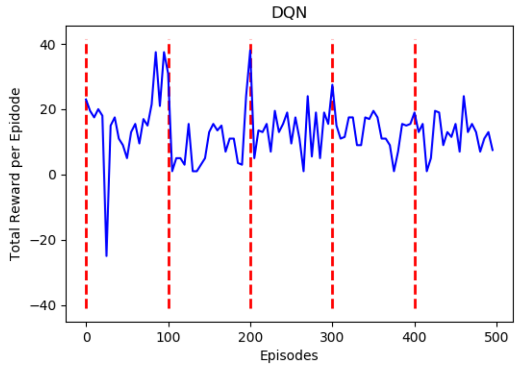
Reward curve for agent number 0



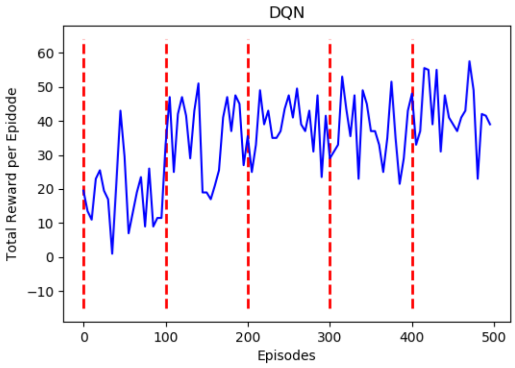
Reward curve for agent number 1



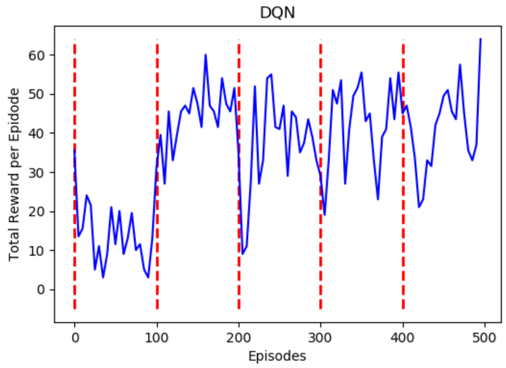
Reward curve for agent number 1



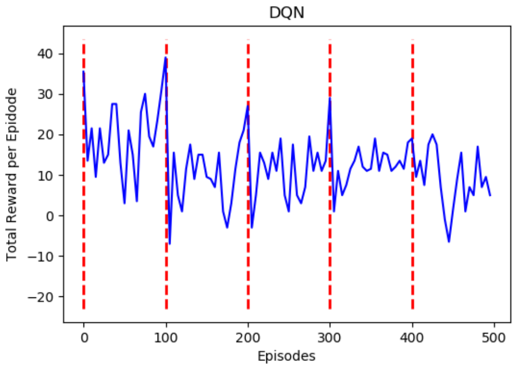
Reward curve for agent number 2



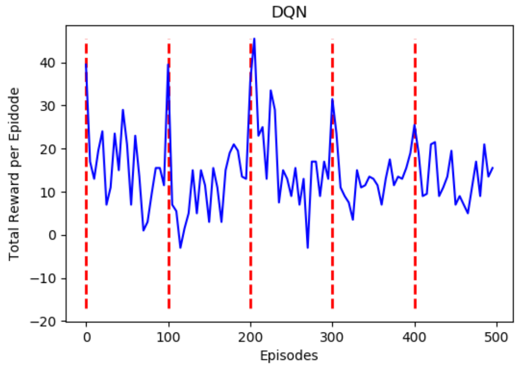
Reward curve for agent number 2



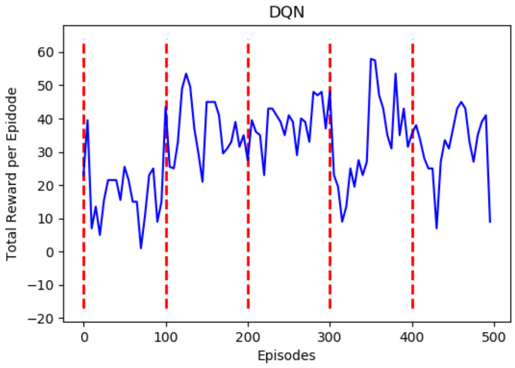
Reward curve for agent number 3



Reward curve for agent number 3



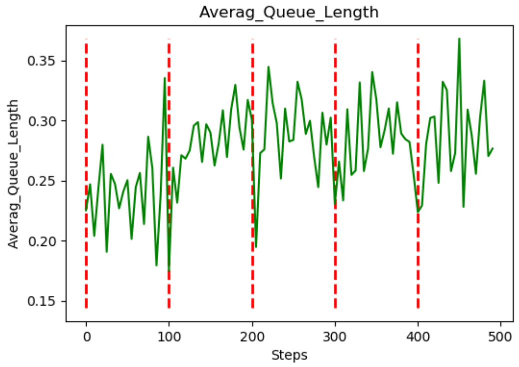
Reward curve for agent number 4

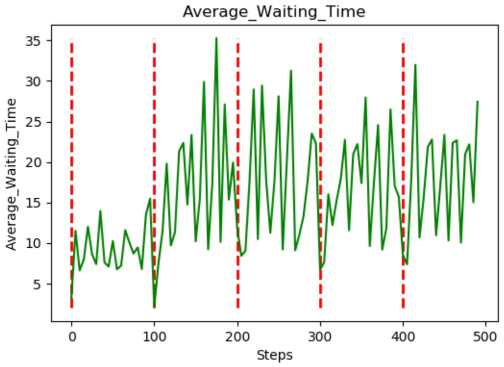


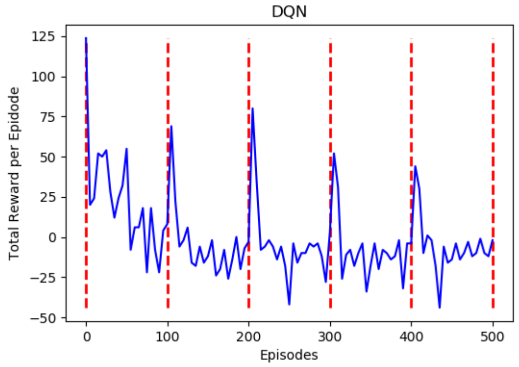
Reward curve for agent number 4

✓ *Rely on sumo decisions:*

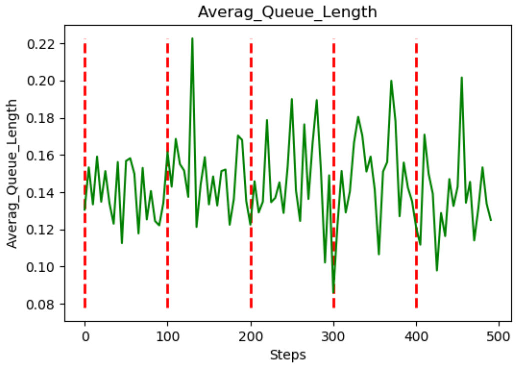
Automatic sumo decisions	Our method

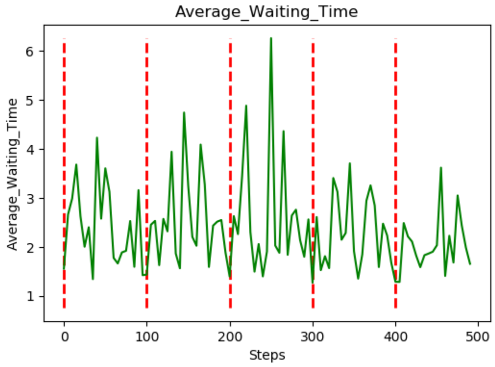


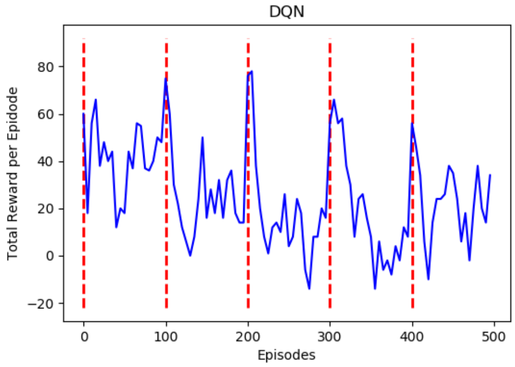




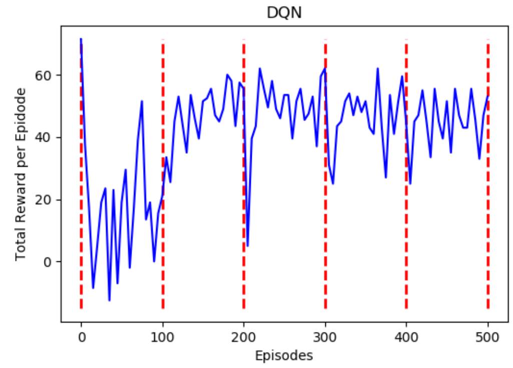
Reward curve for agent number 0



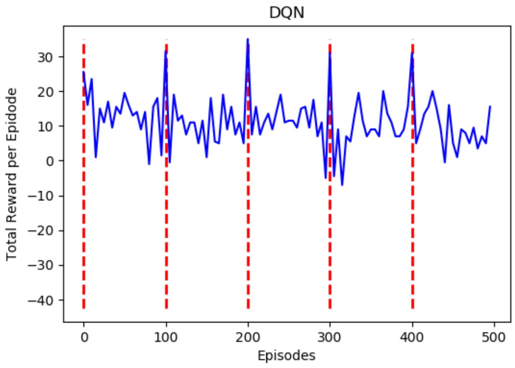




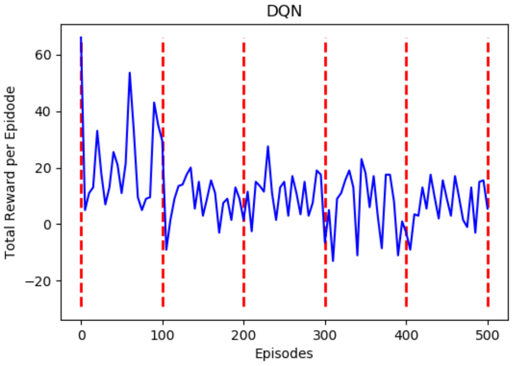
Reward curve for agent number 0



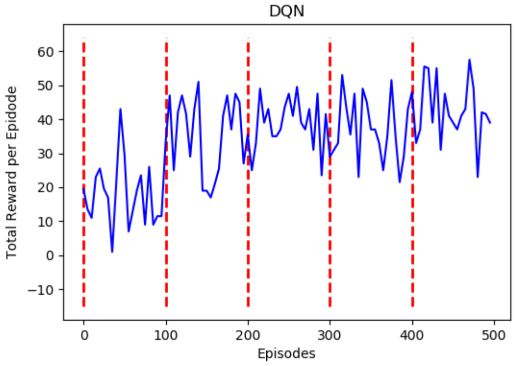
Reward curve for agent number 1



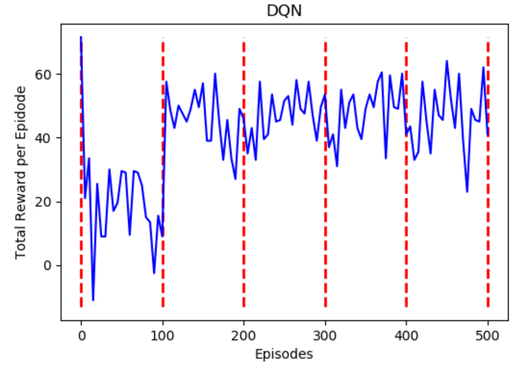
Reward curve for agent number 1



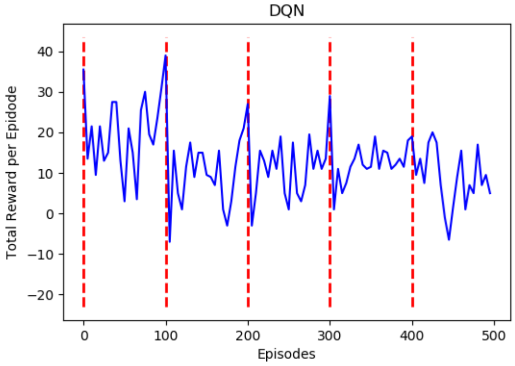
Reward curve for agent number 2



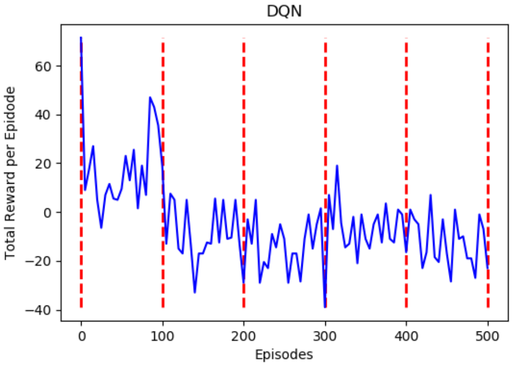
Reward curve for agent number 2



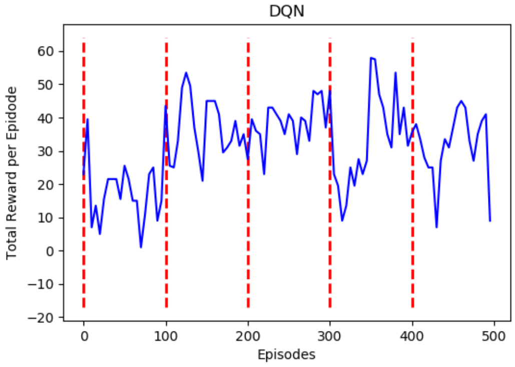
Reward curve for agent number 3



Reward curve for agent number 3

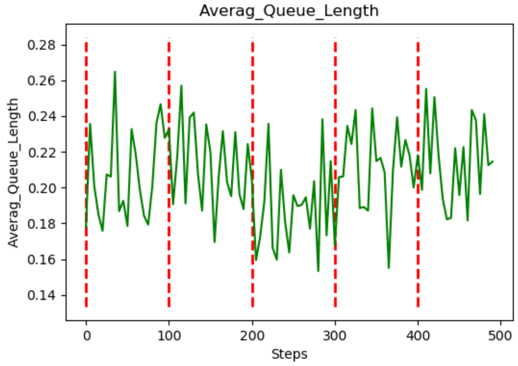
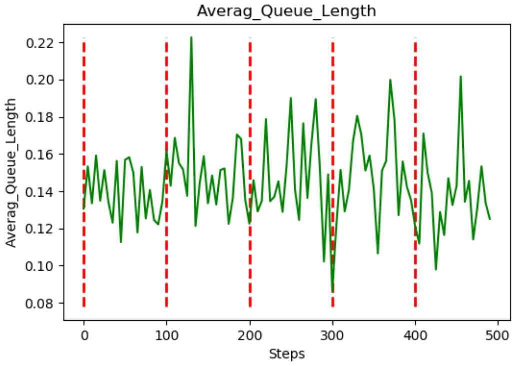
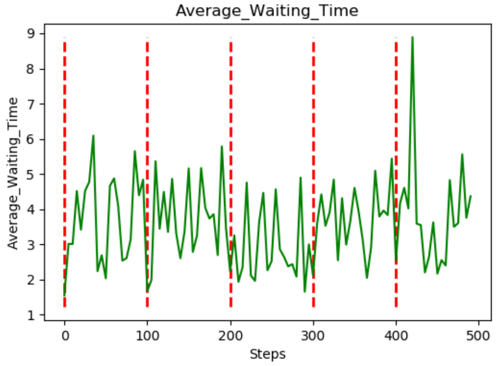
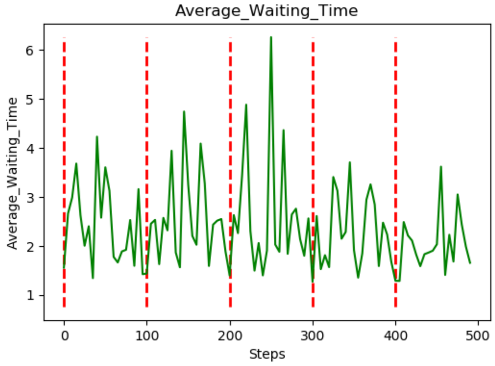
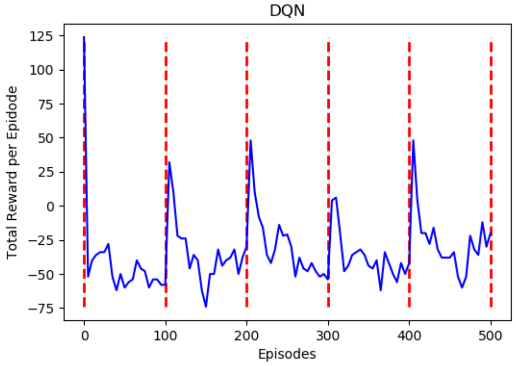
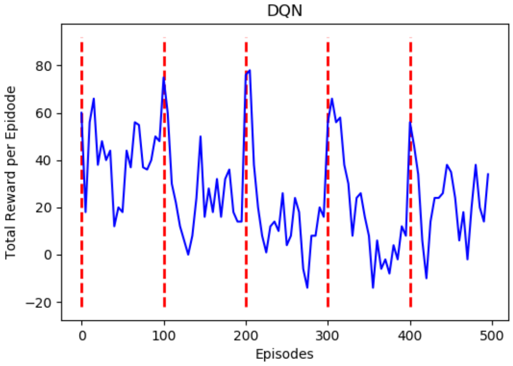


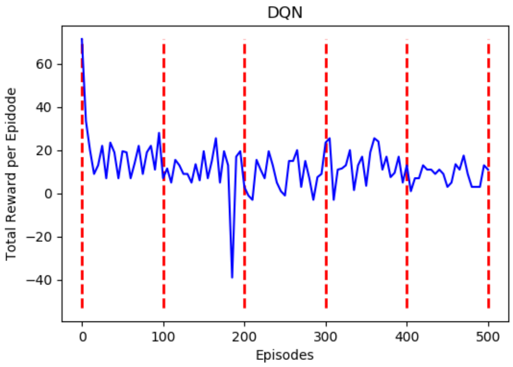
Reward curve for agent number 4



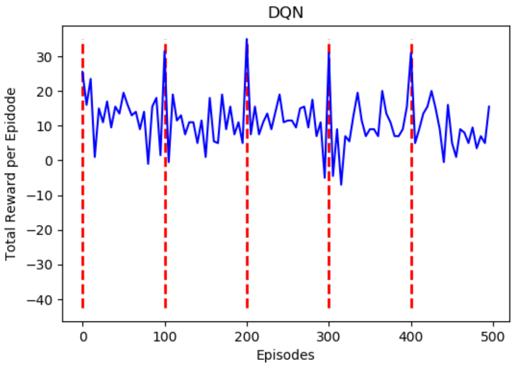
Reward curve for agent number 4

✓ *Random action:*

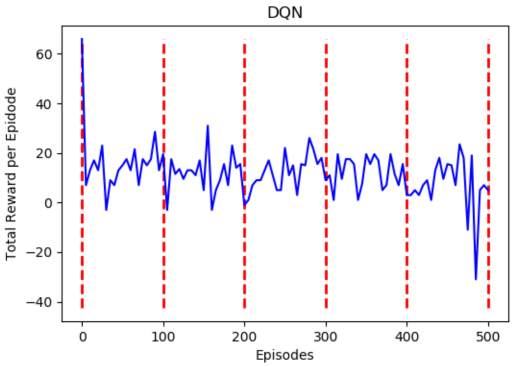
Random action	Our method
	
	
	
Reward curve for agent number 0	Reward curve for agent number 0



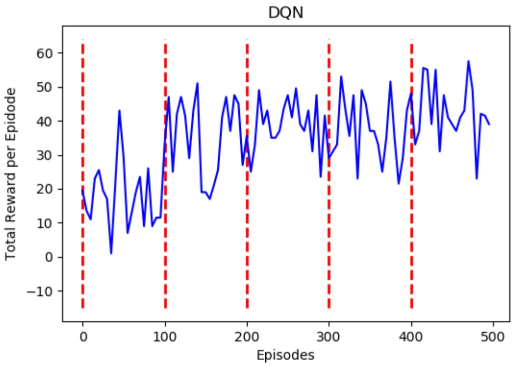
Reward curve for agent number 1



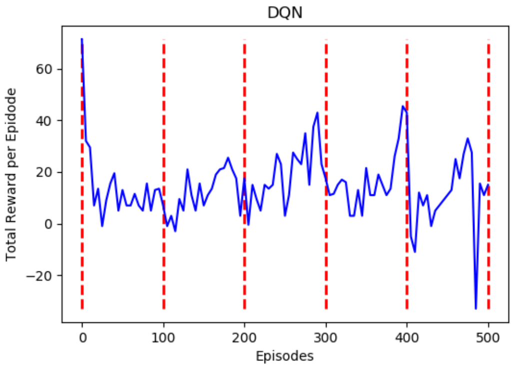
Reward curve for agent number 1



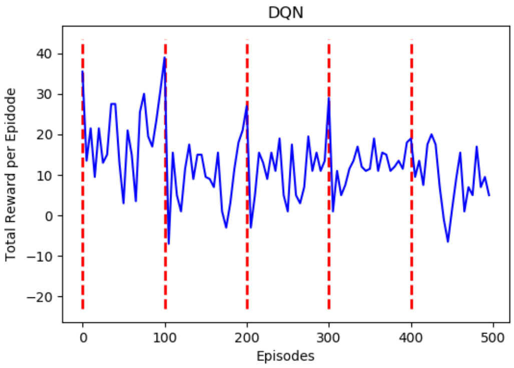
Reward curve for agent number 2



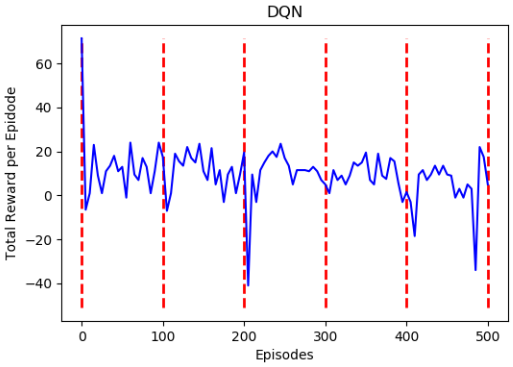
Reward curve for agent number 2



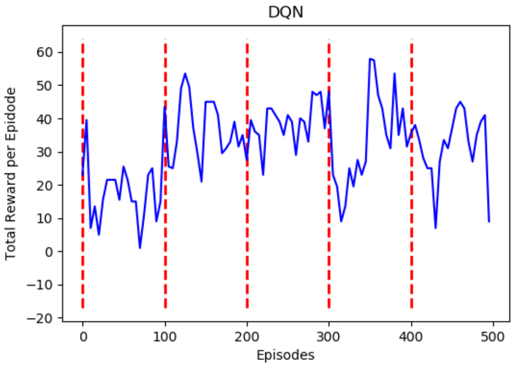
Reward curve for agent number 3



Reward curve for agent number 3



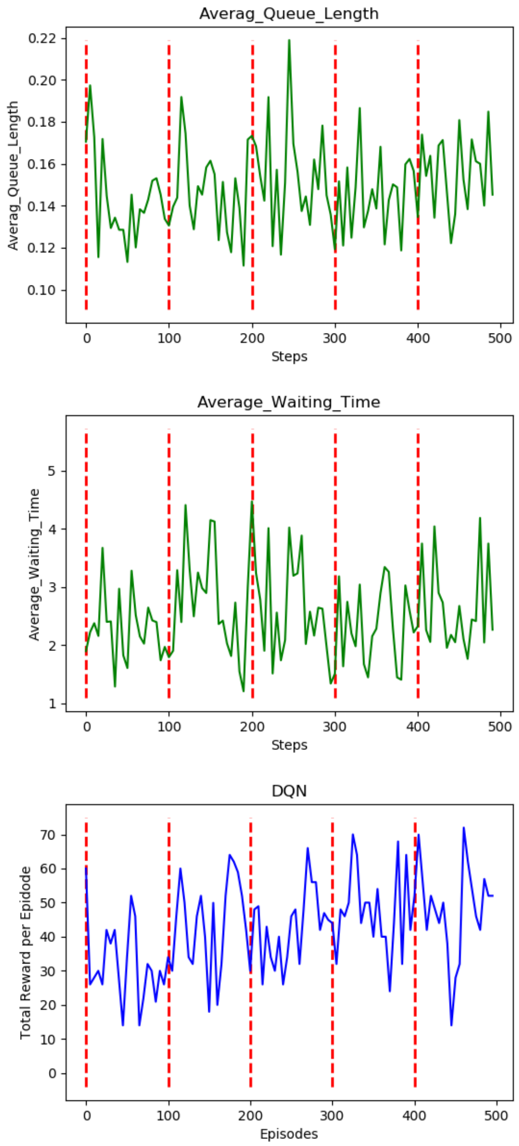
Reward curve for agent number 4



Reward curve for agent number 4

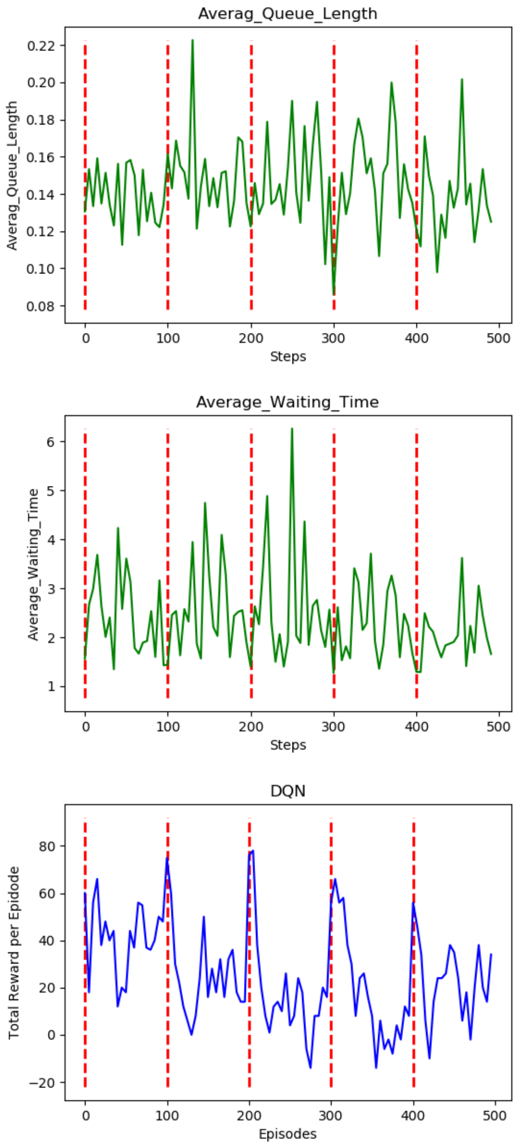
- ✓ *without restart:*

Our method without do restart per 3000 s

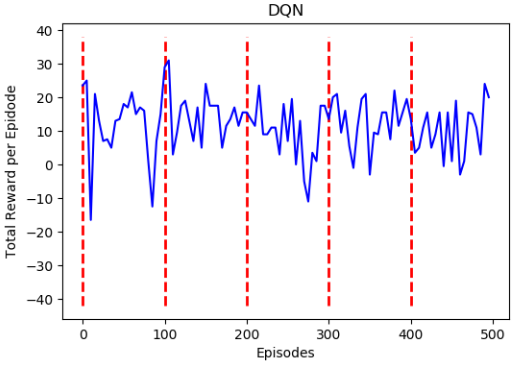


Reward curve for agent number 0

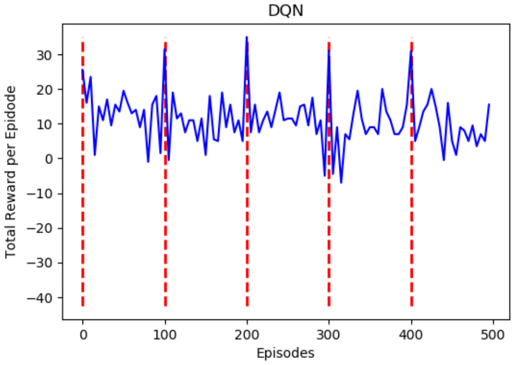
Our method



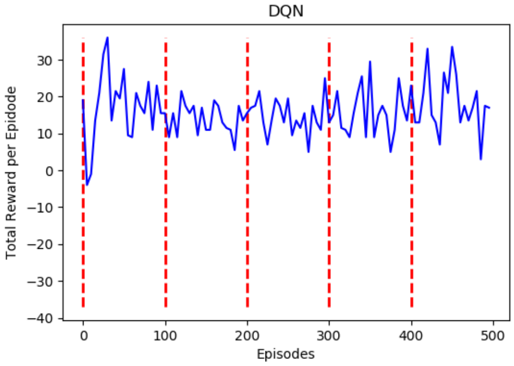
Reward curve for agent number 0



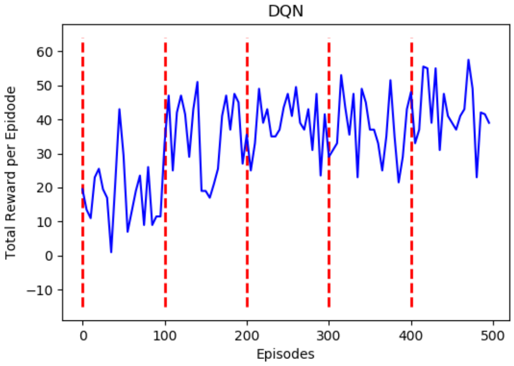
Reward curve for agent number 1



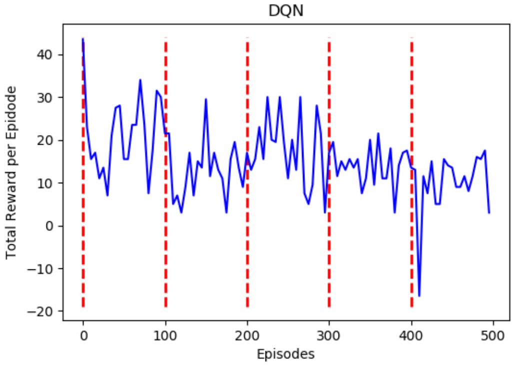
Reward curve for agent number 1



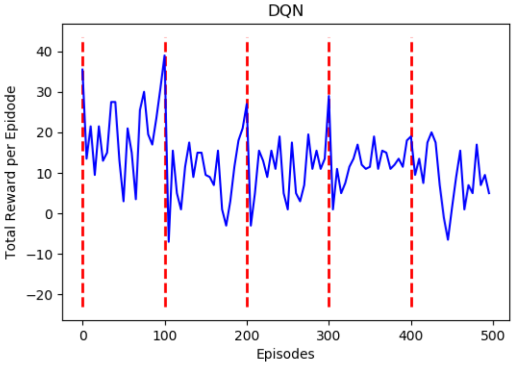
Reward curve for agent number 2



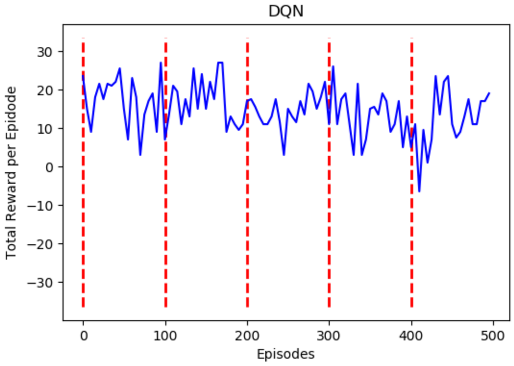
Reward curve for agent number 2



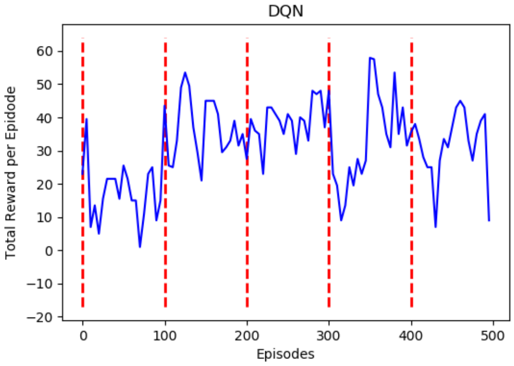
Reward curve for agent number 3



Reward curve for agent number 3



Reward curve for agent number 4



Reward curve for agent number 4

V. Future Prospects:

- ✓ The presence of a vehicle for which the lane must be opened:

  Ambulances, fire engines and other emergency vehicles need to pass quickly and without delay. So we have to get the agent controlling the intersection to decide to open the lane in which one of these vehicles is. A vehicle must send its type in the control message that it sends to the agent, who in turn will use this information to establish the state so that it affects his decisions. Also the agent should get a better reward for opening the lane. But we may face a problem, which is that two vehicles of this type are present on two lanes that conflict with their opening at the same time, here we must have importance for each of these vehicles.

- ✓ Obtaining information through a drone:

  Instead of getting the messages from each vehicle separately and then working on converting these messages into usable information for the agent, the drone enables us to shorten a lot of operations, and we may have the agent himself included in it and the traffic light only receives the action to applies it.

- ✓ Take into account the number of pedestrians standing waiting to close the lane to cross.

- ✓ Making the periods of signals (red, green and yellow) variable according to the status of the lanes and the pressure on them.