Intelligent traffic signal control

Kranti Shingate 101652
Komal Jagdale 101669
Yohann Dias 101667

Roadmap

1. Problem Statement
2. Existing system
3. Implemented traffic control system using Reinforcement learning
4. Abstract
5. Proposed system
6. S/W and H/W requirements
7. Design
8. Implementation
9. Conclusion
10. References

Problem Statement

- Vehicle delay
- Pollution
- Fuel wastage
- Congestion level as high as **65 per cent**

Causes

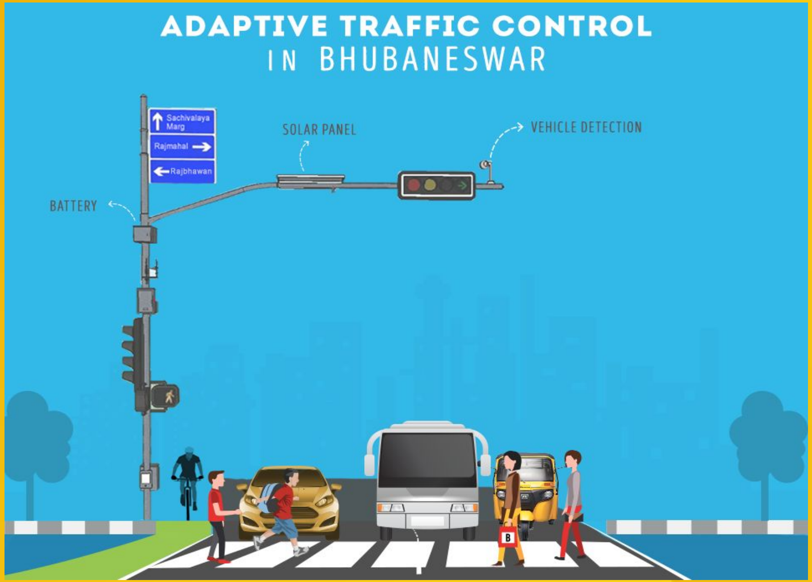
- Lack of road space
- Improper Infrastructure
- High population
- Inadequate green time
- No priority for public transport
- Traffic signals aren’t optimized

Existing system

1. Traffic signals with preset timers

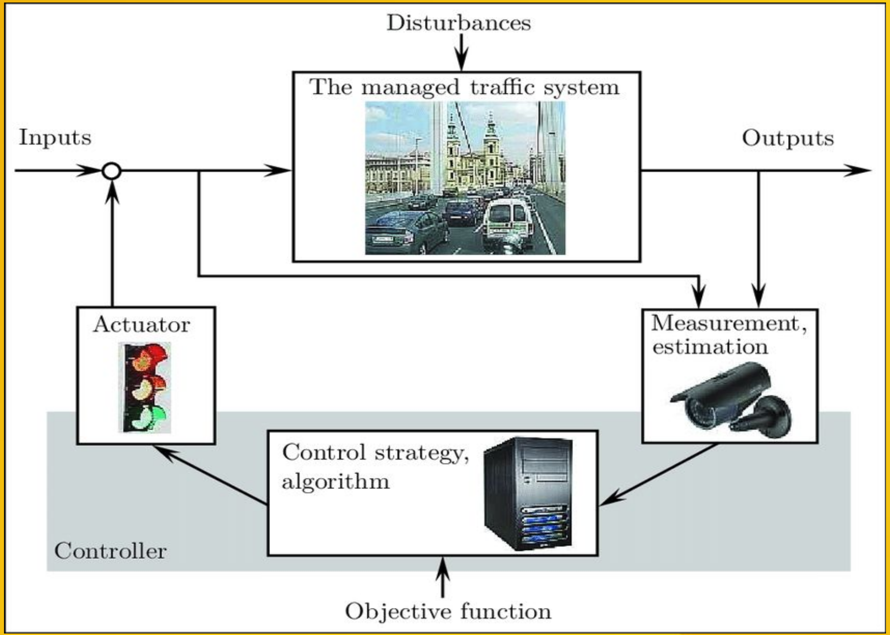
Existing system

2. Adaptive Traffic Signal Control System (Bhubaneswar)



Existing System

3. LQF(longest queue first) scheduling algorithm (North America)

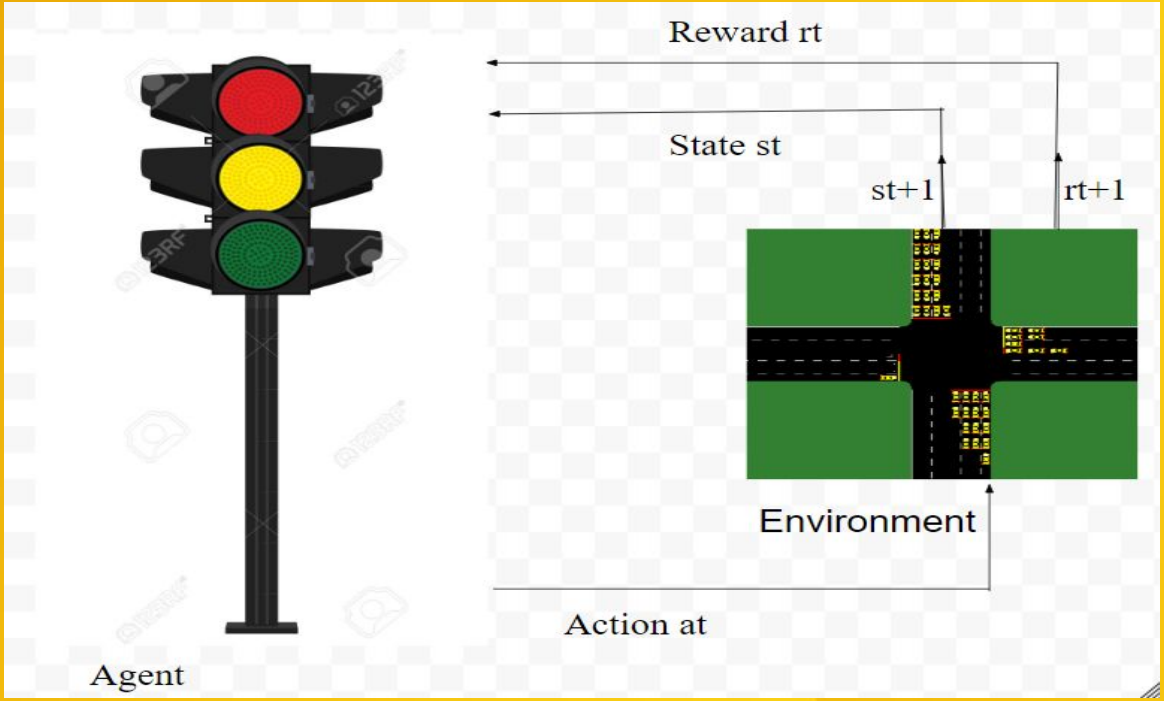


Proposed system

- The proposed system makes use of **reinforcement learning** to obtain optimal traffic signal control.

- Each intersection has an autonomous intelligent agent (traffic signal) which makes the right decisions on the basis of rewards (reduction in delay) given after interacting with the environment (traffic flow).

Proposed system



State, Action and Reward

- STATE : Positions of vehicles inside the environment.
- ACTION : A configuration of the traffic light that implies the green phase for some lanes for a fixed amount of time.
- REWARD : A positive reward is a consequence of a good action, while a negative reward is received after a bad action.

S/W And H/W Requirements

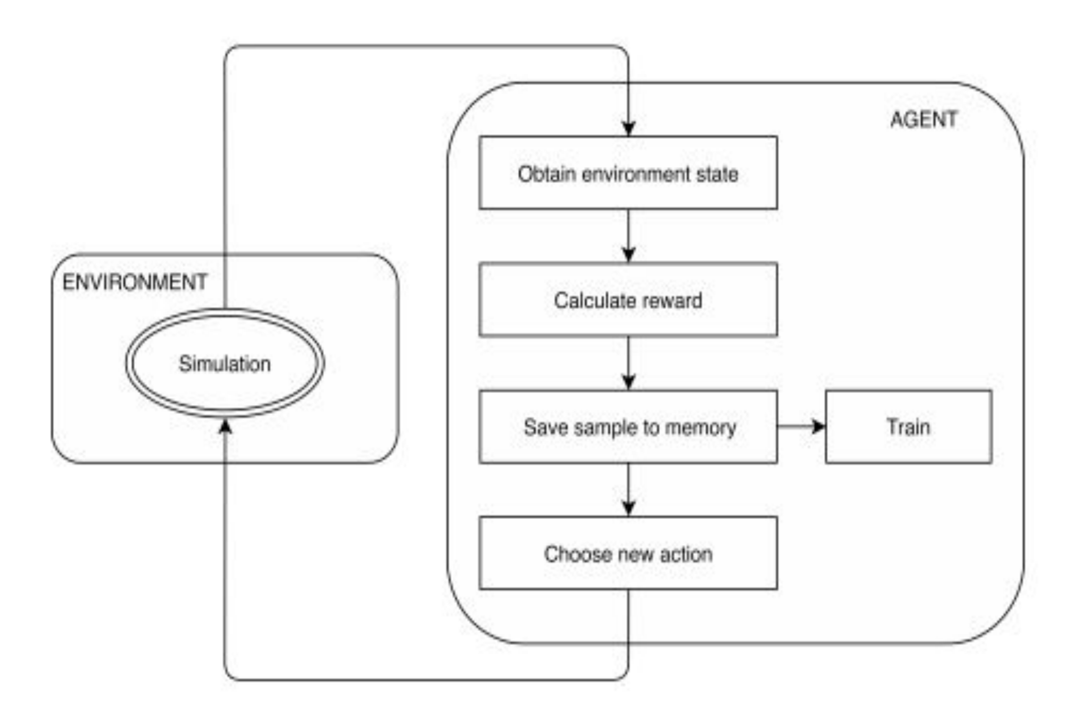
Software Requirements

- Operating System: Windows 7, Windows 8 or Windows 10
- Python 3.6
- SUMO Traffic simulator 1.0.1
- Tensorflow 1.11.0

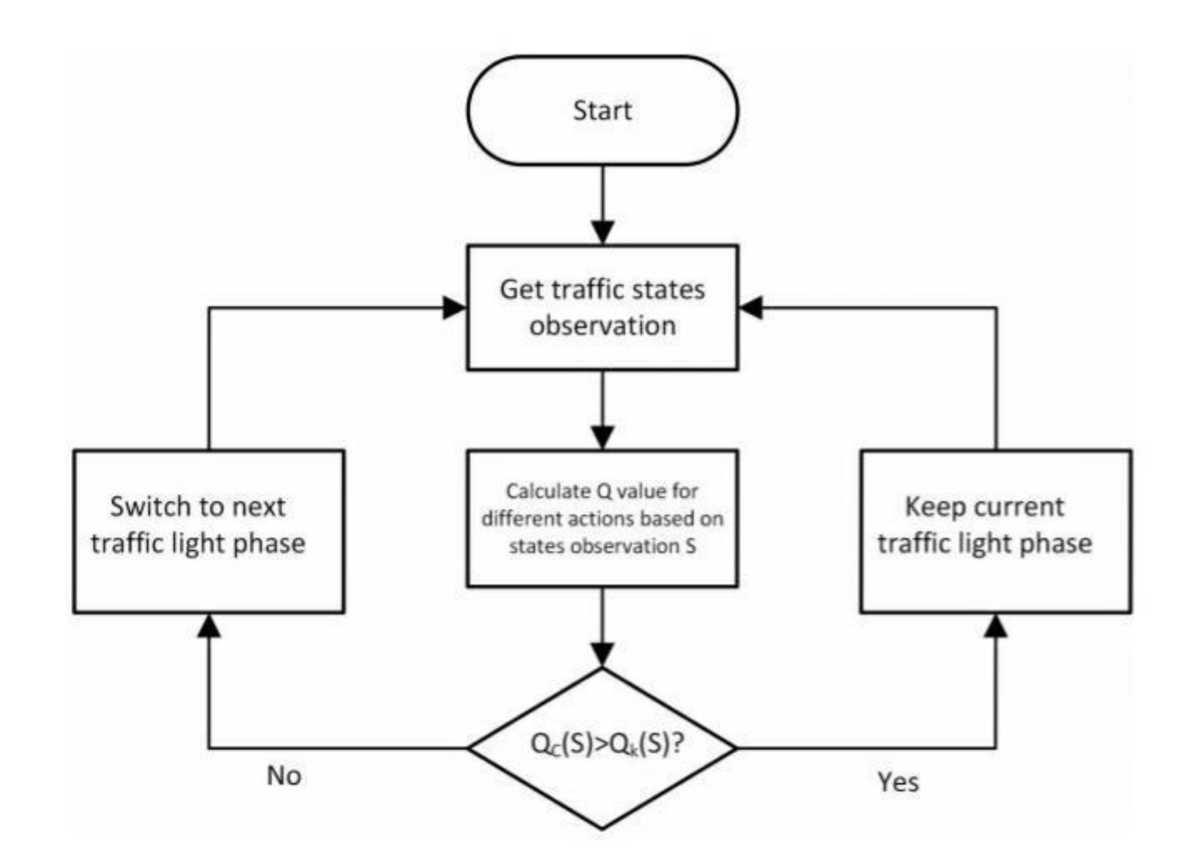
Hardware Requirements

- Processor (CPU) with 2 gigahertz (GHz) frequency or above
- A minimum of 4 GB of RAM
- Monitor Resolution 1024 X 768 or higher
- A minimum of 20 GB of available space on the hard disk

Block Diagram



Q-learning Algorithm



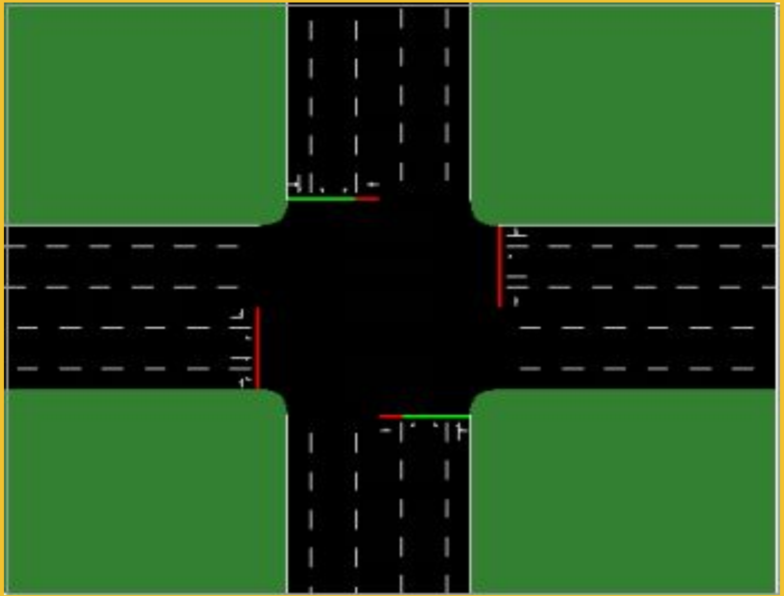
Q-learning equation

$Q(s_t, a_t) = r(s_t, a_t) + \gamma \cdot \max_A Q(s_{t+1}, a_t))$

Where,

$Q(s_t, a_t)$ is value of action at performed in state st
$Q(s_{t+1}, a_t)$ is the Q-value of immediate next step
r_{t+1} is reward agent gets after performing action at
γ is the discount factor (between 0 and 1) focuses on future reward than immediate

Generation of Traffic in SUMO Simulator



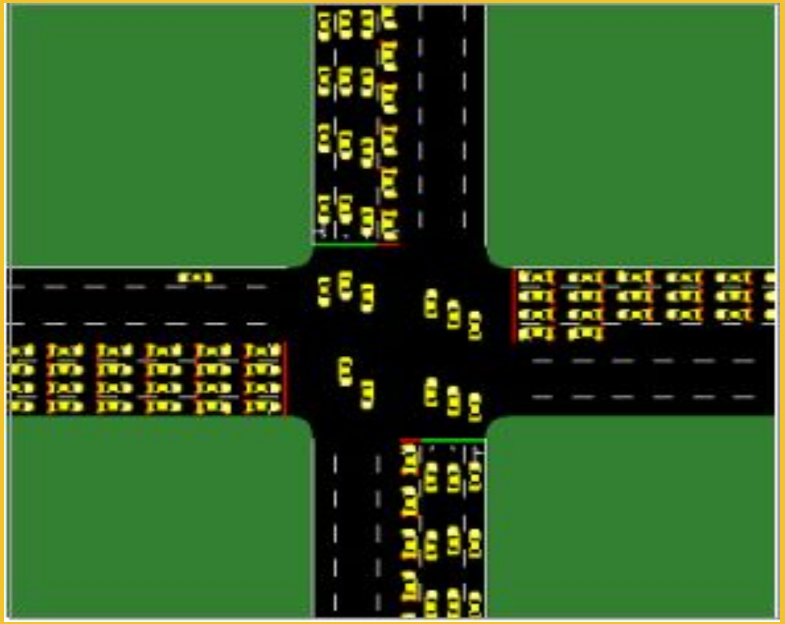
4-Way intersection

Traffic Generation

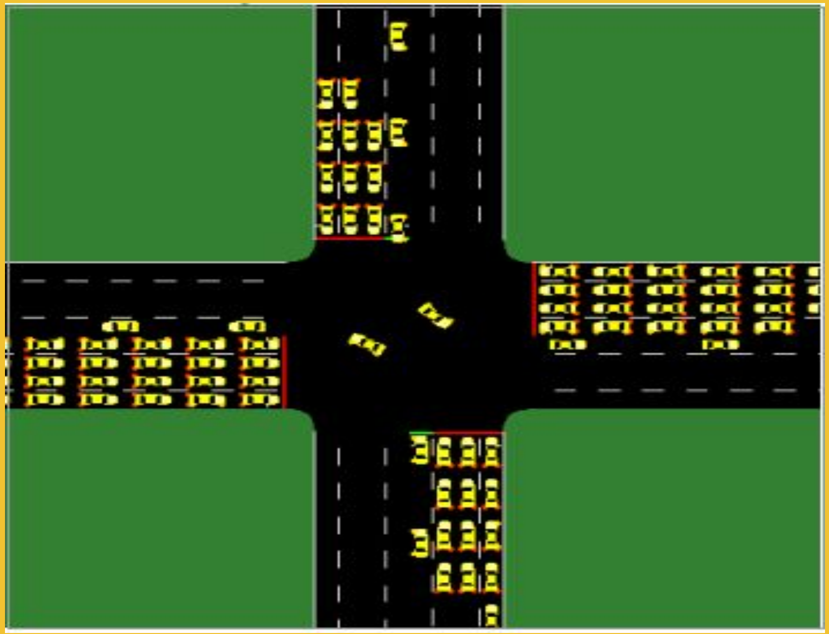
Possible actions taken by agent

ACTION	ARMS OF LANE
North-South Advance	North and south (for straight and right)
North-South Left Advance	North and south(for left turn)
East-West Advance	East and west (for straight and right)
East-West Left Advance	East and west (for left turn)

Possible actions taken by agent



North-south Advance



North-South Left Advance

Possible actions taken by agent

East-West Advance

East-West Left Advance

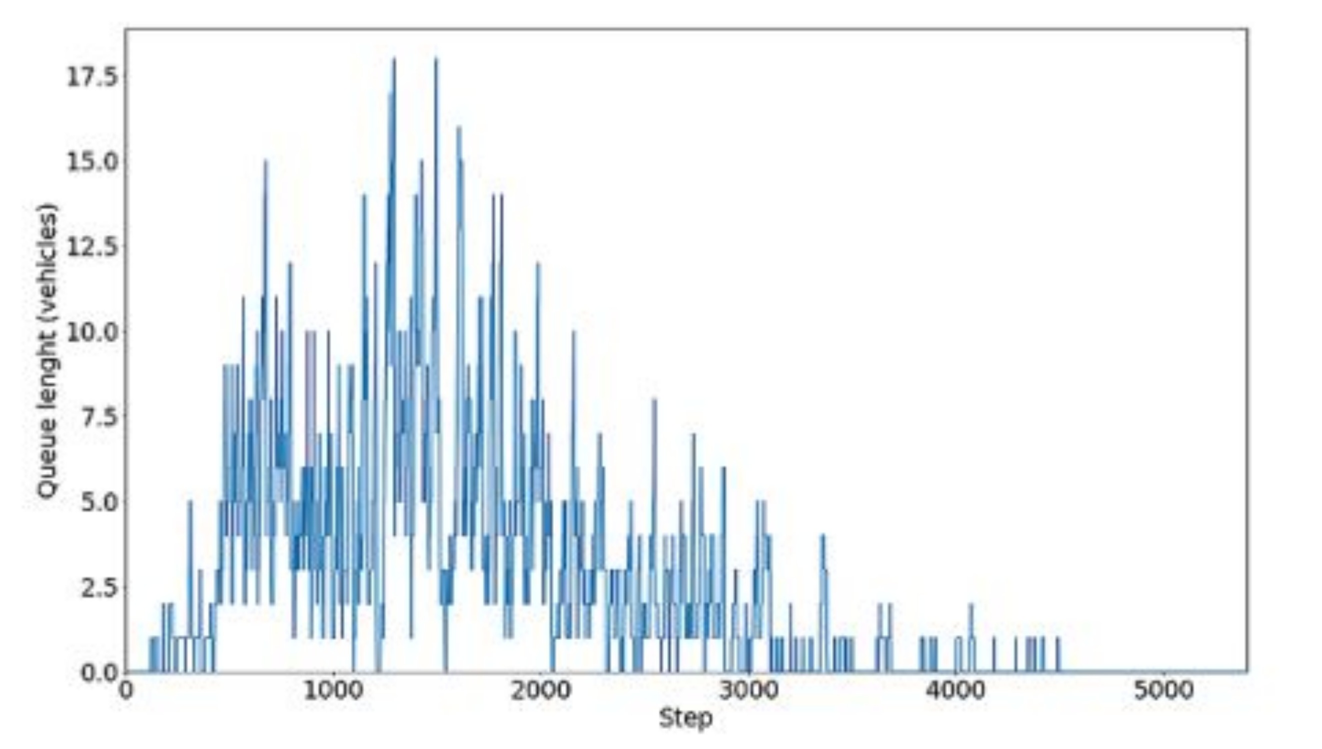
Training Phase

```
The system cannot find the path specified.

(base) C:\Users\user>Cd desktop

(base) C:\Users\user\Desktop>cd TLC

(base) C:\Users\user\Desktop\TLC>python training_main.py

----- Episode 1 of 100
Loading configuration... done.
 Retrying in 1 seconds
Simulating...
Total reward: -23149.0 - Epsilon: 1.0
Training...
Simulation time: 12.2 s - Training time: 0.0 s - Total: 12.2 s

----- Episode 2 of 100
Loading configuration... done.
Simulating...
Total reward: -21195.0 - Epsilon: 0.99
Training...
```

Testing Phase

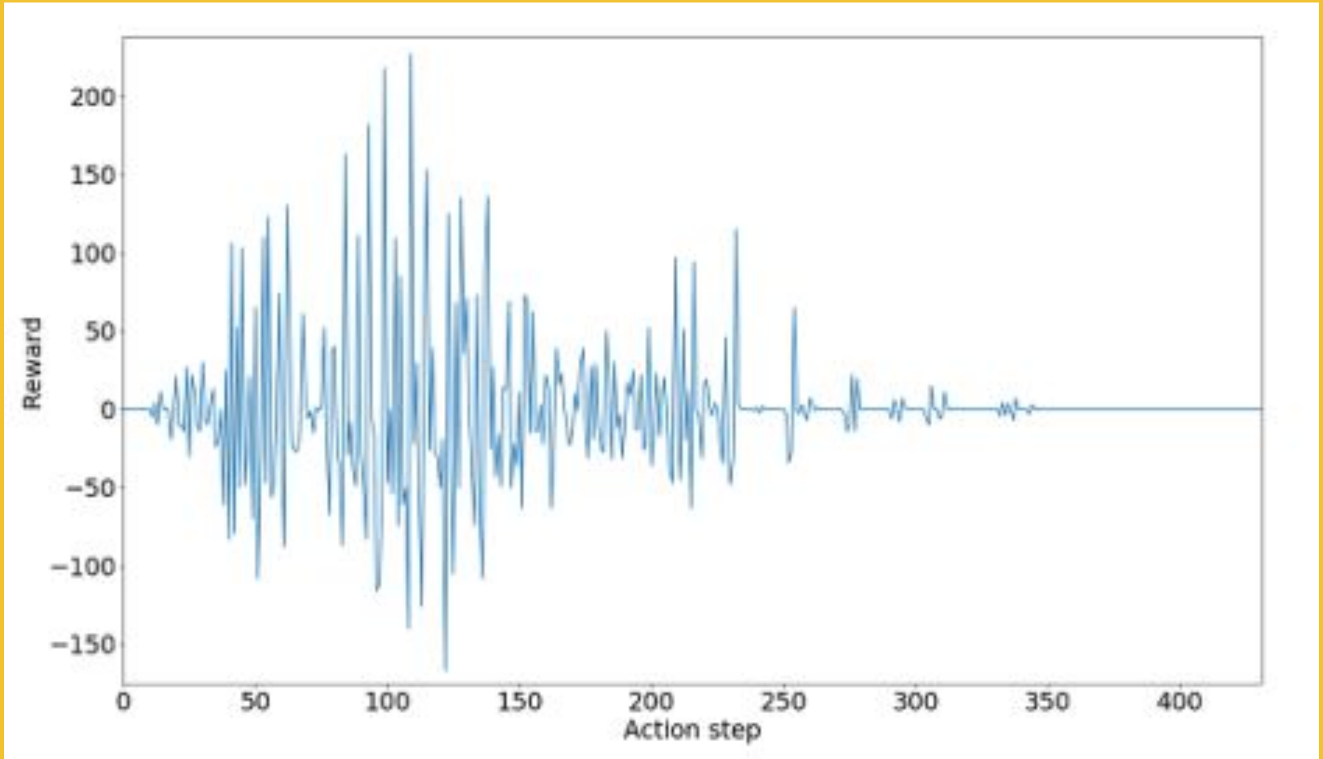
```
(base) C:\Users\user>cd Desktop

(base) C:\Users\user\Desktop>cd TLC

(base) C:\Users\user\Desktop\TLC>python testing_main.py

----- Test episode
Loading configuration... done.
Simulating...
Simulation time: 26.4 s
----- Testing info saved at: C:\Users\user\Desktop\TLC\models\model_13\test\

(base) C:\Users\user\Desktop\TLC>
```

Performance of Agent before Training and Testing



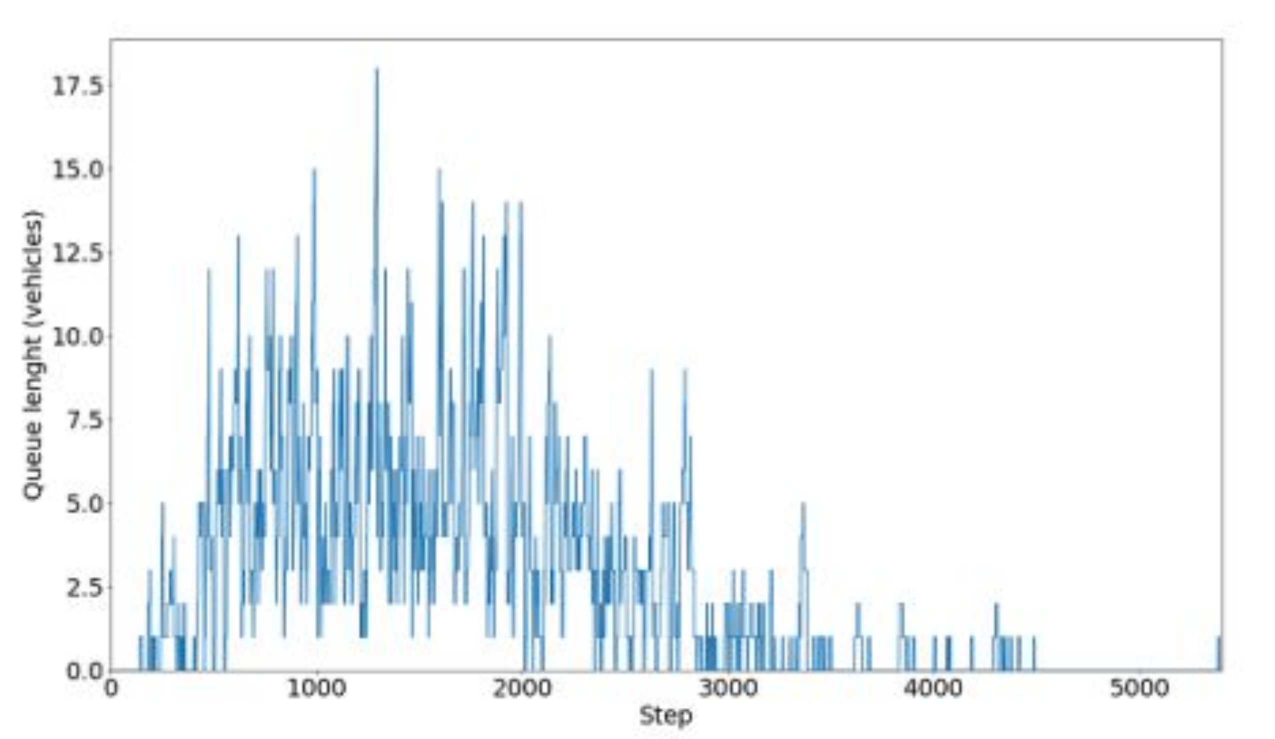
Queue Length

Performance of Agent before Training and Testing



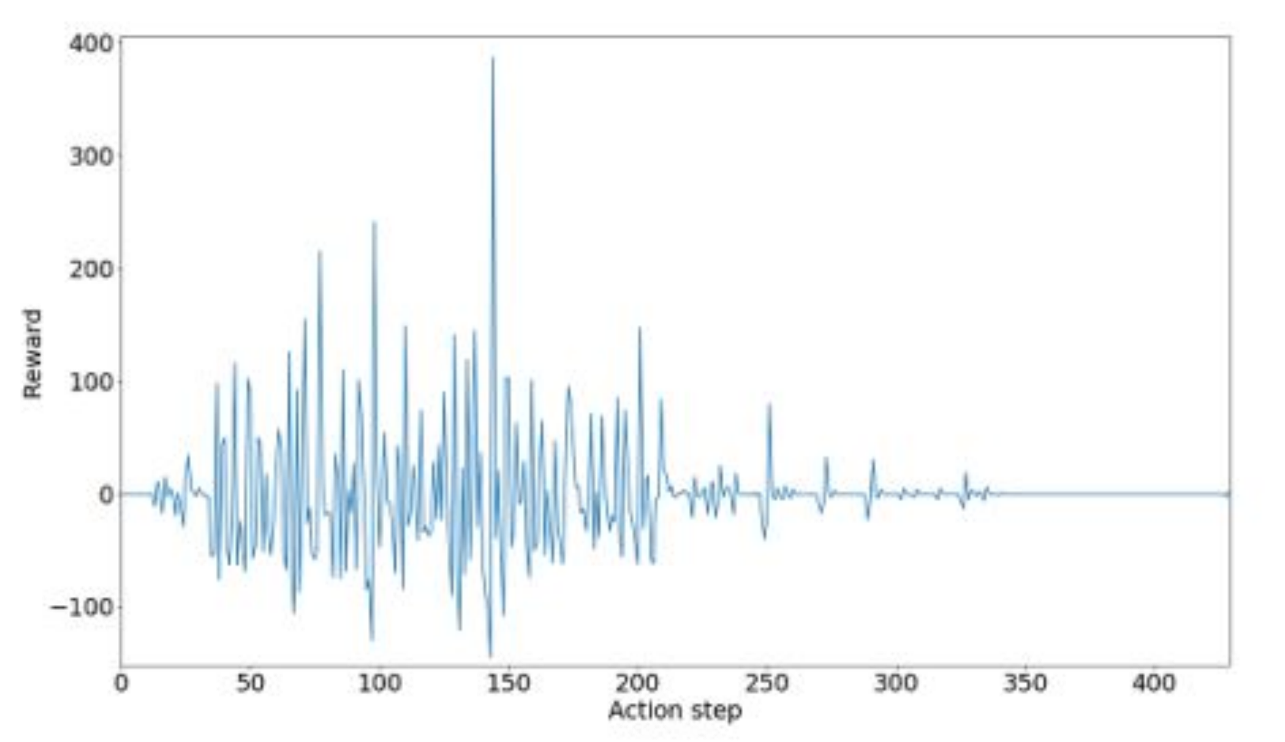
Rewards

Performance of Agent after Training and Testing



Queue Length

Performance of Agent after Training and Testing



Rewards

Conclusion

We propose an intelligent traffic control system using reinforcement learning algorithm.The learning agent of this system is designed with state representation that identifies the position of vehicle in environment and makes decisions according to real time traffic.Based on the decision agent gets reward which is further used by the agent to make appropriate decisions to reduce traffic on the basis of it’s rewards.

References

[1] Rusheng Zhang ,Akihiro Ishikawa,Wenli Wang,Benjamin Striner ,Ozan Tonguz:'Intelligent Traffic Signal Control: Using Reinforcement Learning with Partial Detection'

[2]I. Arel, C. Liu, T. Urbanik, A.G. Kohls : 'Reinforcement learning-based multi-agent system for network traffic signal control'

[3] Juntao Gao, Yulong Shen, Jia Liu, Minoru Ito and Norio Shiratori : 'Adaptive Traffic Signal Control: Deep Reinforcement Learning Algorithm with Experience Replay and Target Network'

[4] Reinforcement Learning for Intelligent Traffic Light Control

[5] Multi-Agent Reinforcement Learning for Intelligent Traffic

Light Control