# Adaptive Traffic Signal Control Using Multi-Agent RL

Sonu Dixit

# Problem Formulation

- Environment = Whole traffic Network
- Every junction is an Agent, working in a common environment.
- Every agent can experience a part(local data) of environment.

## Agent Description

- State:
  Congestion at each incoming lane
  Congestion at immediate Neighbors incoming lanes
  Current Phase
- Action : Green signal duration {20,25,30,….65,70}
- Reward : Change in congestion at its lanes and neighbors lanes after performing action
- Objective : Maximize discounted sum of rewards

Data For Training:

- PTV Vissim Simulation for modelling Traffic behaviour
- Queue Counters for getting congestion in a lane at particular time
- Self Designed Network

# Algorithm Used

- PPO, Advantage Actor critic
- Entropy Loss included as well
- Buffer of 100 latest samples

$$\mathrm{r}_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$$

$$\mathrm{L}^{CLIP}(\theta) = E_t\left[min(r_t(\theta)\hat{A}_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)\right]$$

$$\mathrm{H}(\theta) = \sum_a -\pi_\theta(a_t|s_t) log(\pi_\theta(a_t|s_t))$$

$$\max\ [\mathrm{L}^{CLIP}(\theta) + \alpha H(\theta)]$$

# Algorithm

Start Simulation

For every Agent that needs to take action

1. Get s,a,r,s’. put into buffer
2. if buffer has enough samples

3. K times:

4. Sample minibatch(32) from buffer

5. Train actor(current_policy) on this data

6. Train Critic on whole buffer

7. Set old_policy = current_policy

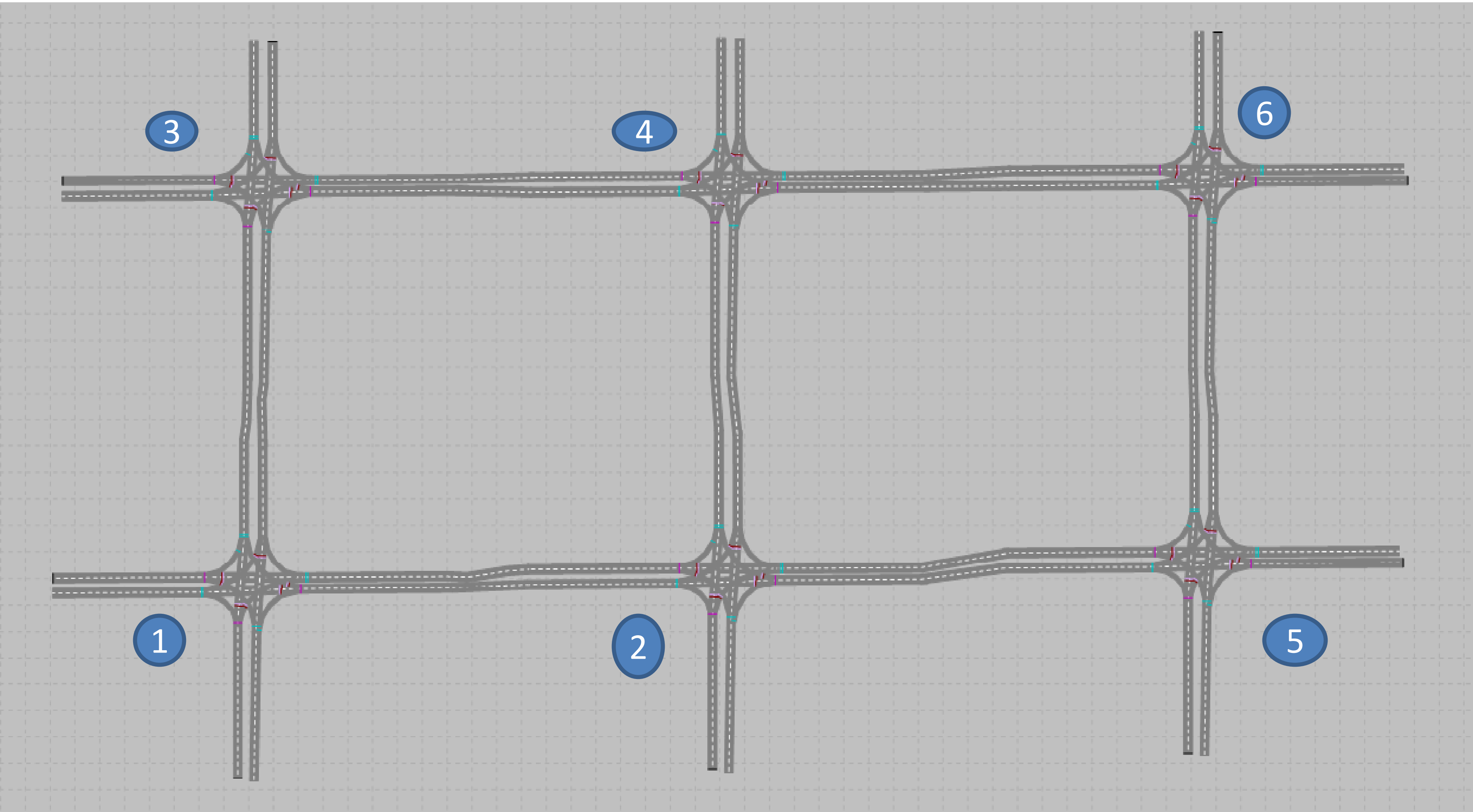
8. Empty buffer

# Actor Critic Network Details

- Both are simple Dense Network, 2 Hidden layers (64 nodes each)
- PPO loss epsilon = 0.2
- Entropy loss coefficient = 0.01
- Two actor networks( old, new), 1 critic network for every agent

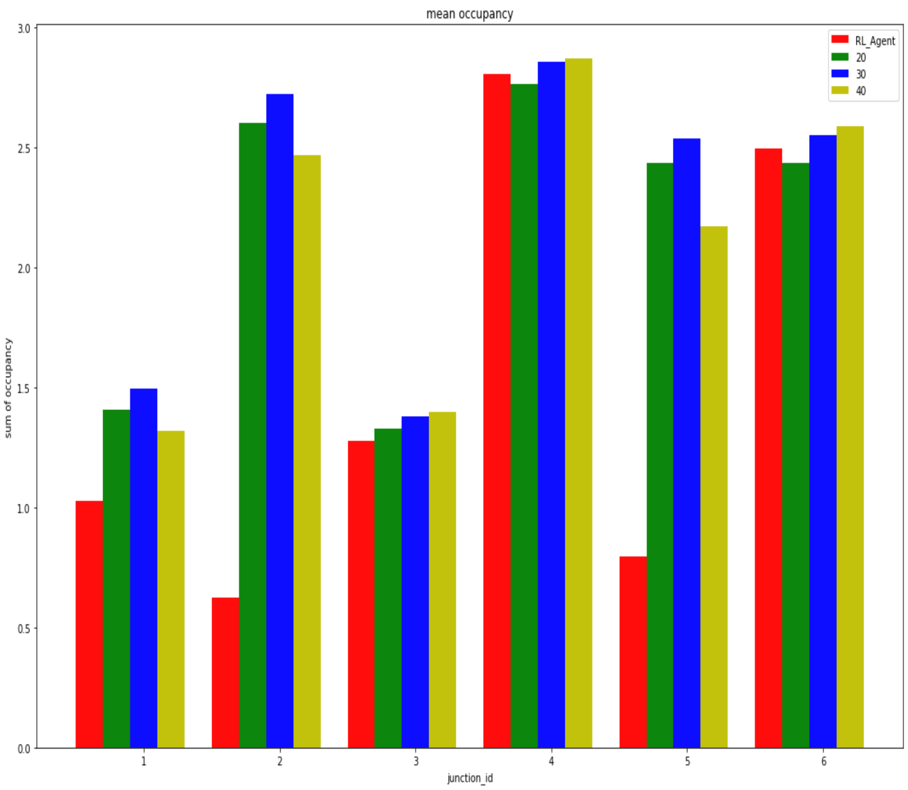
# Traffic Network for experimentation

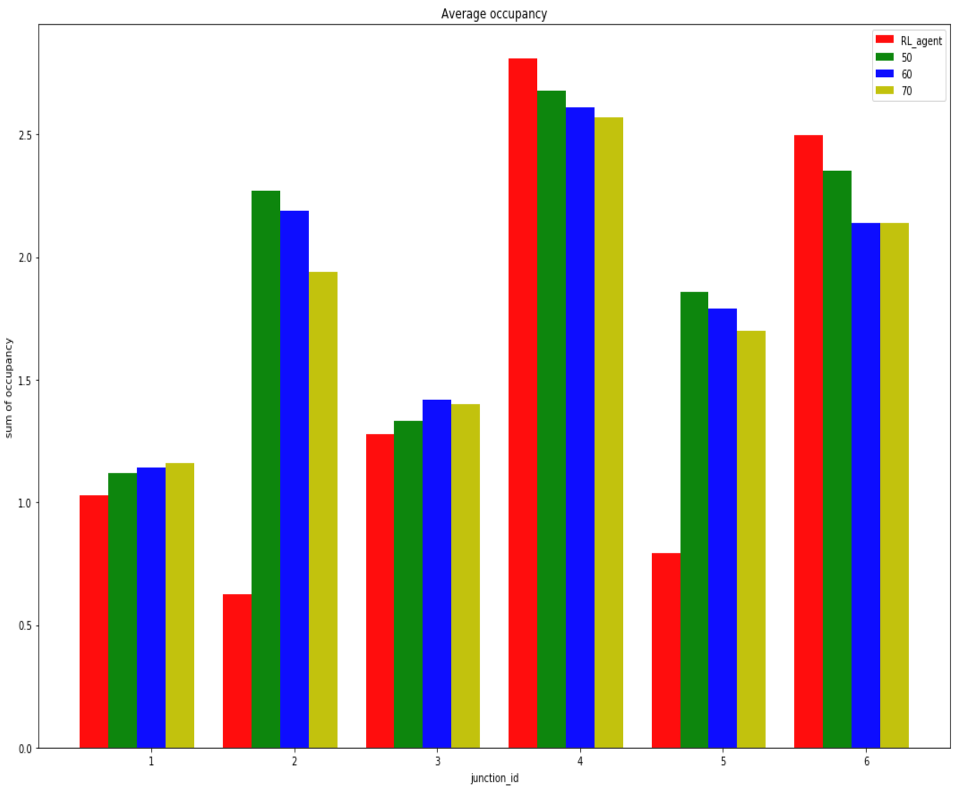
- 6 junctions, 8 signals(que_counters) per junction
- 10 Vehicle Input points
- Left, Straight, Right = 0.25, 0.50, 0.25



# Experimentation and Results

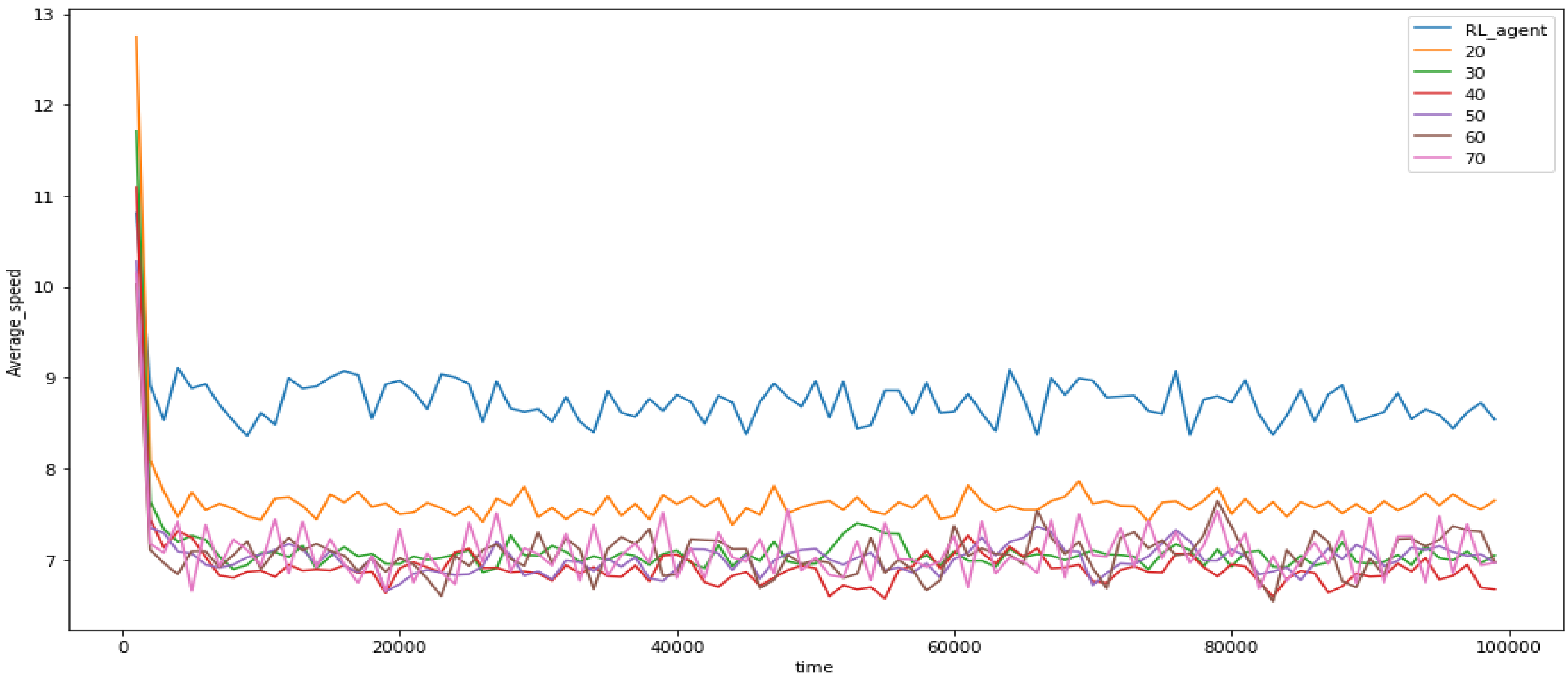
- 1k vehicles per hour at every incoming lane
- Avg Occupancy Comparison with fixed time algorithms



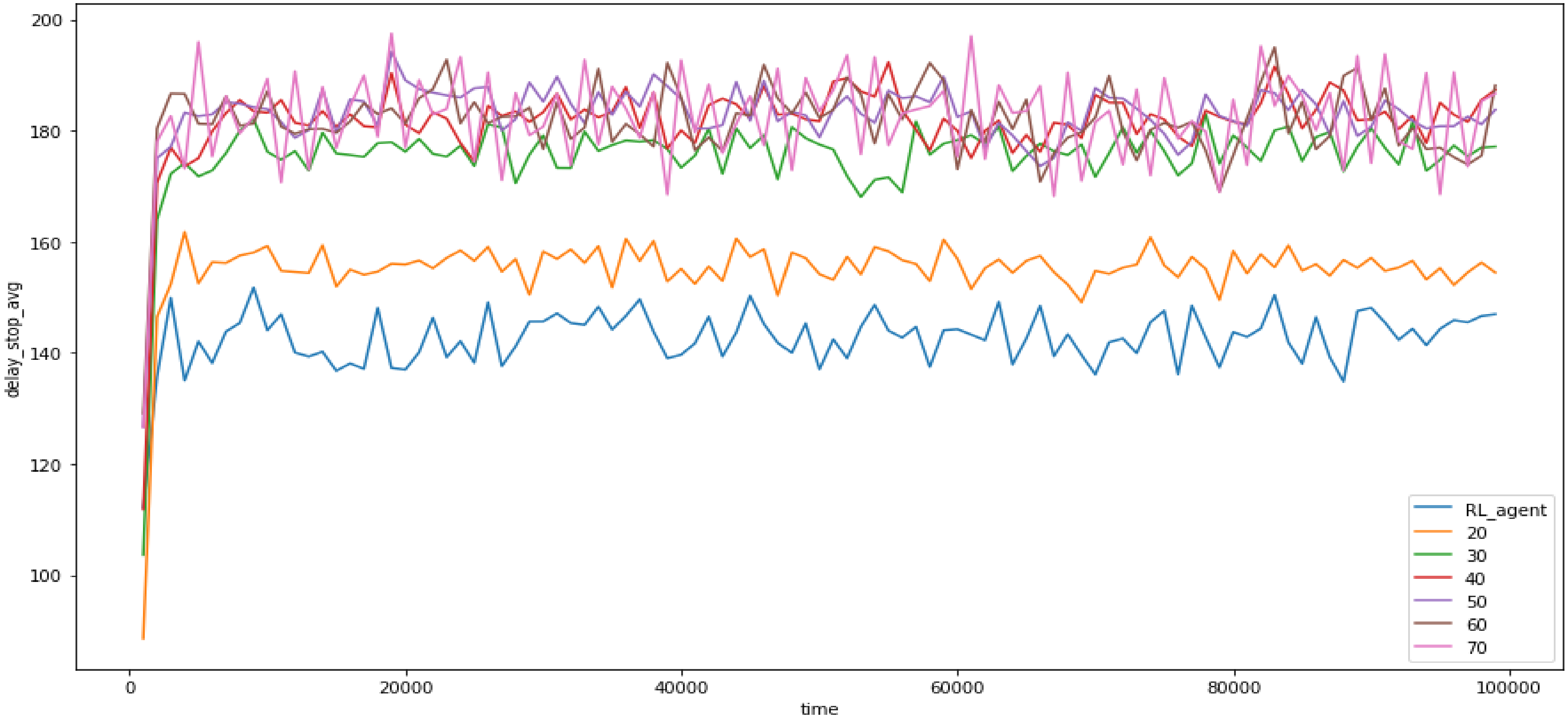
# Experimentation and Results

- 1k vehicles per hour at every incoming lane,
- Avg Speed Comparison, calculated in 1000secs interval



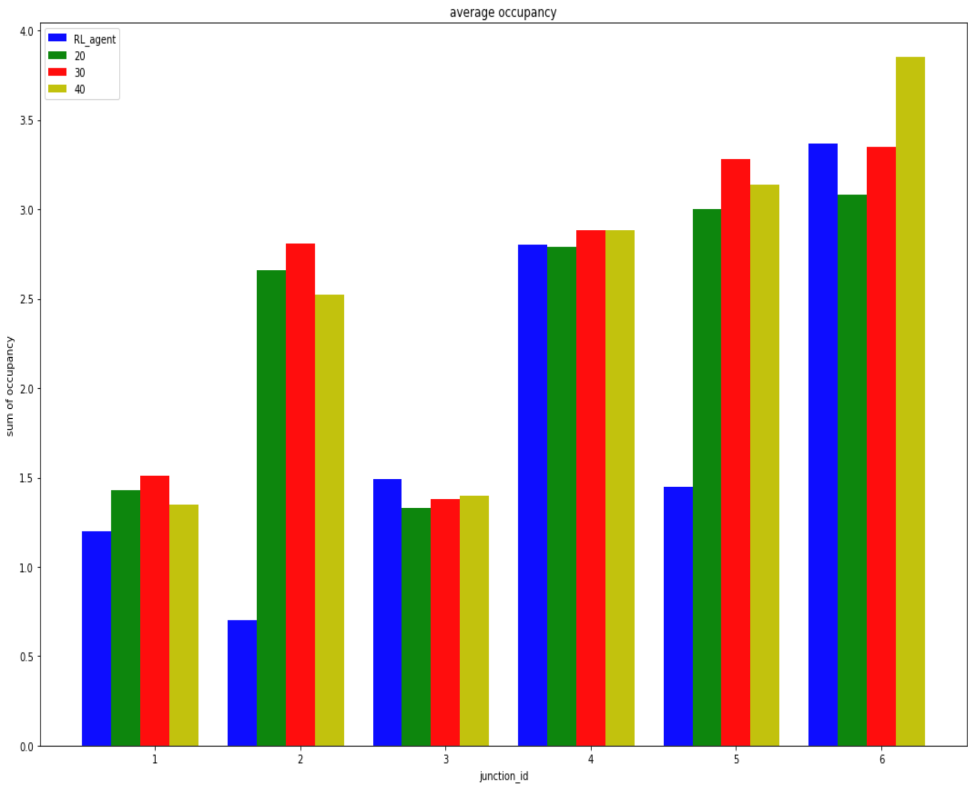
# Experimentation and Results

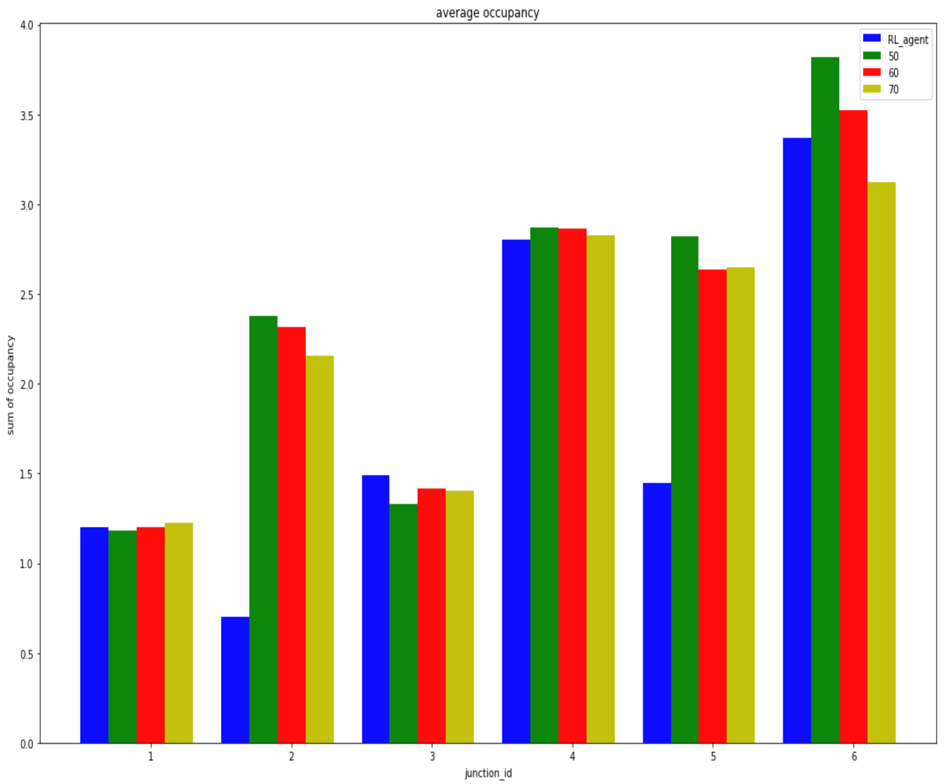
- 1k vehicles per hour at every incoming lane,
- Delay Stop Avg, calculated in 1000secs interval



# Experimentation and Results

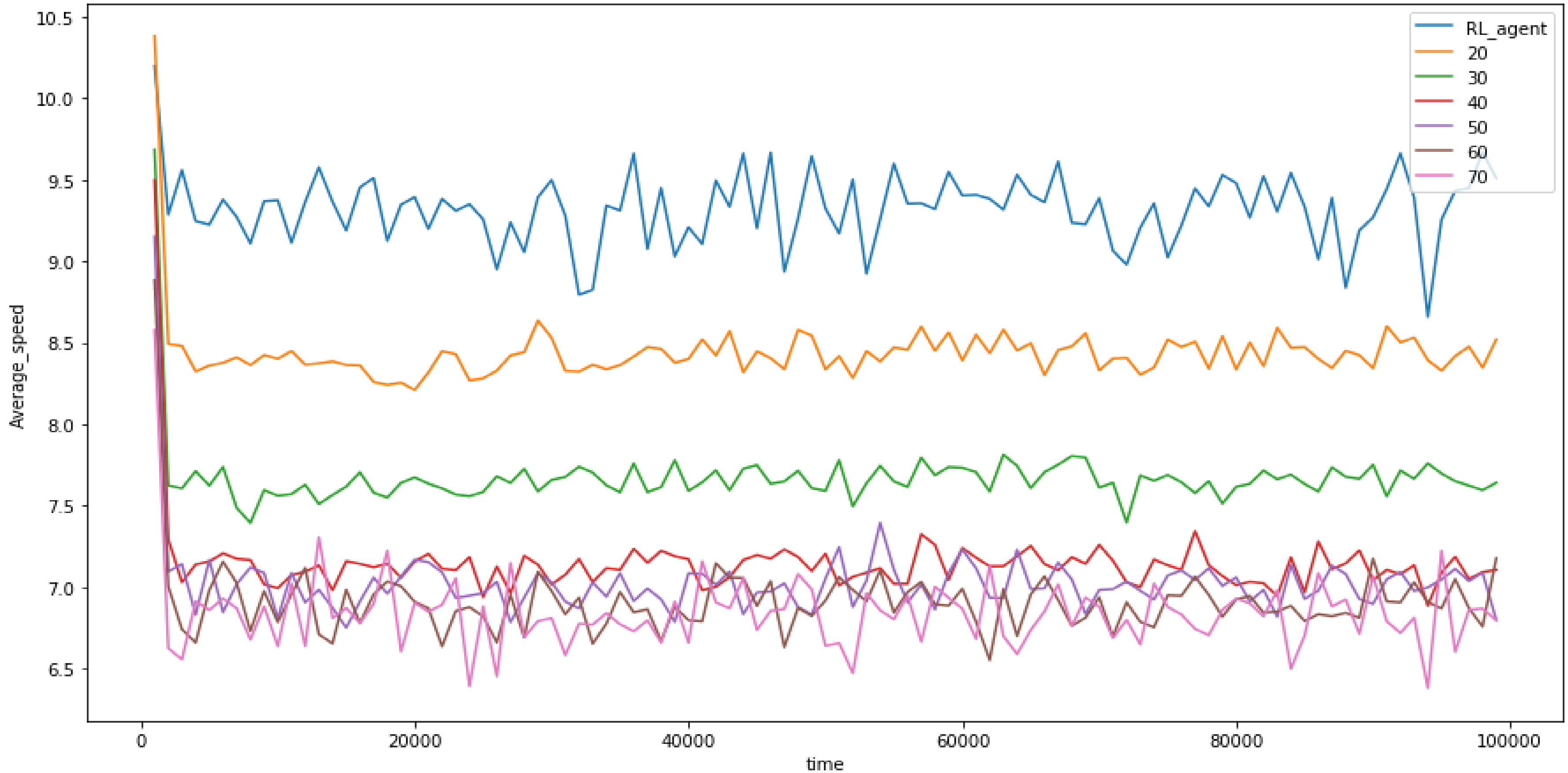
- 2k vehicles per hour at every incoming lane
- Avg Occupancy Comparison with fixed time algorithms



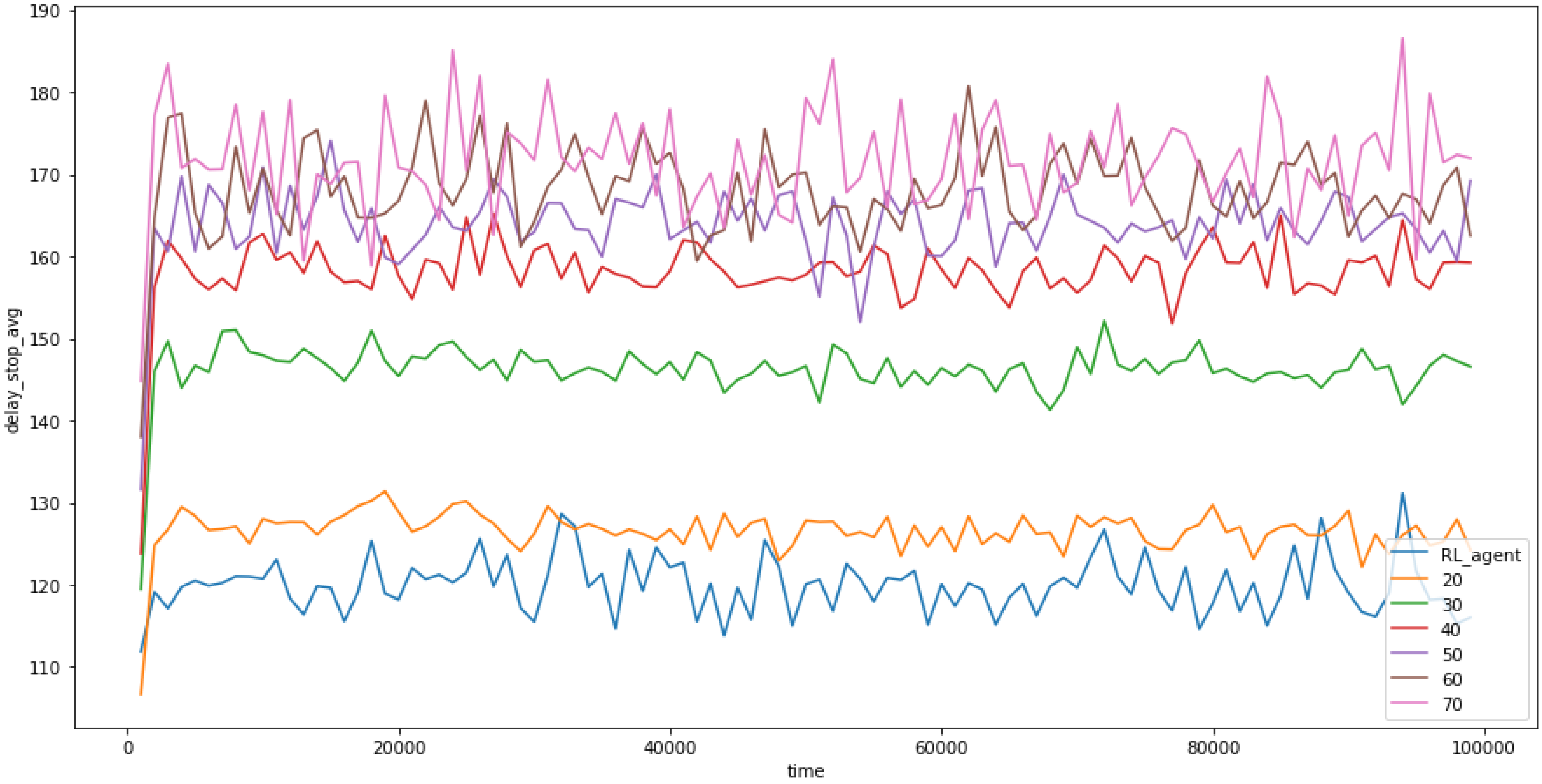
# Experimentation and Results

- 2k vehicles per hour at every incoming lane,
- Avg Speed Comparison, calculated in 1000secs interval



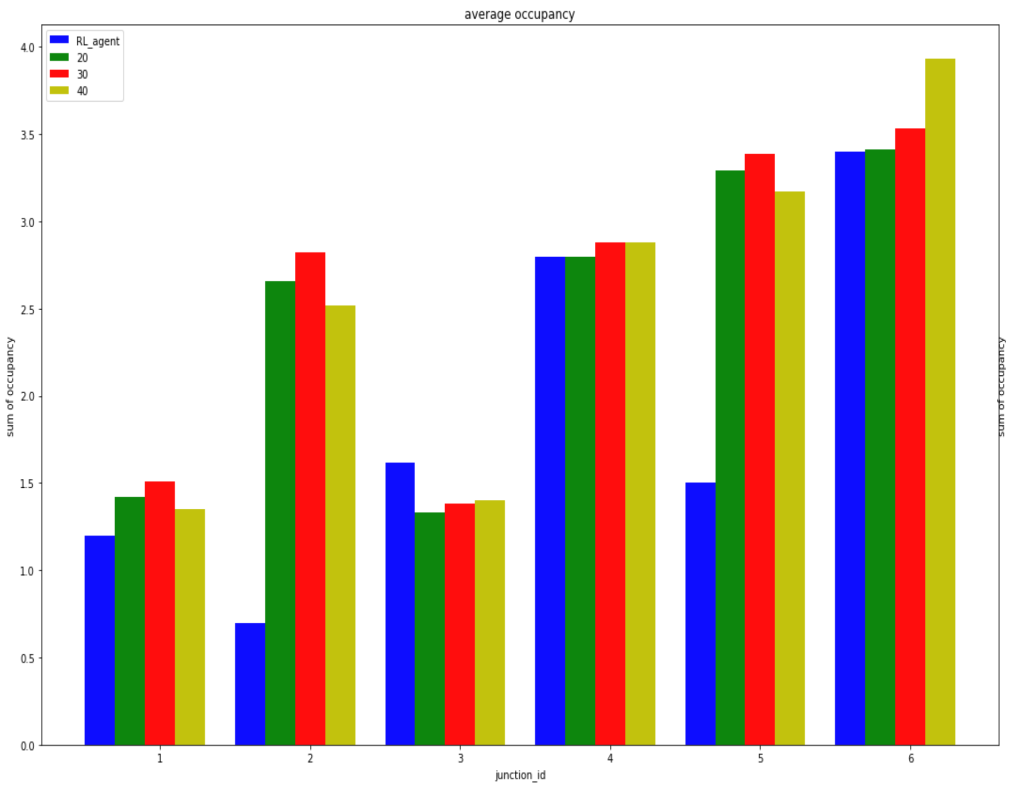
# Experimentation and Results

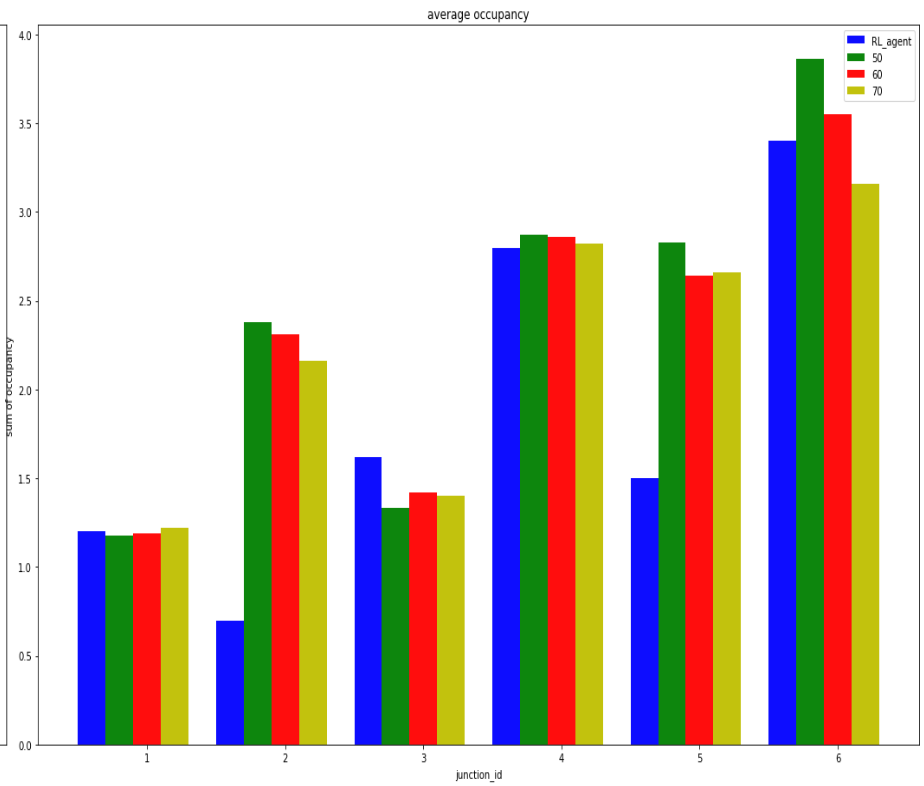
- 2k vehicles per hour at every incoming lane,
- Delay Stop Avg, calculated in 1000secs interval



# Experimentation and Results

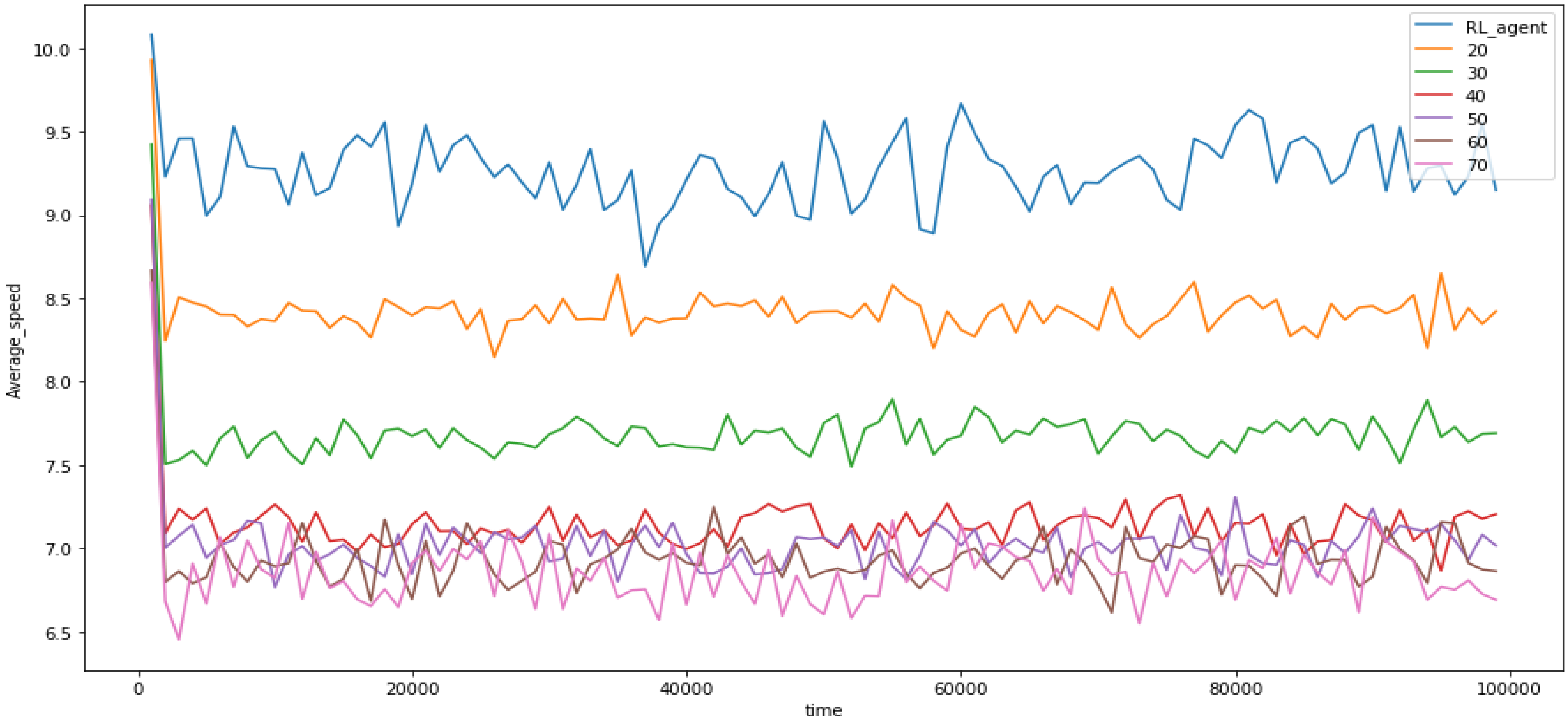
- 3k vehicles per hour at every incoming lane
- Avg Occupancy Comparison with fixed time algorithms



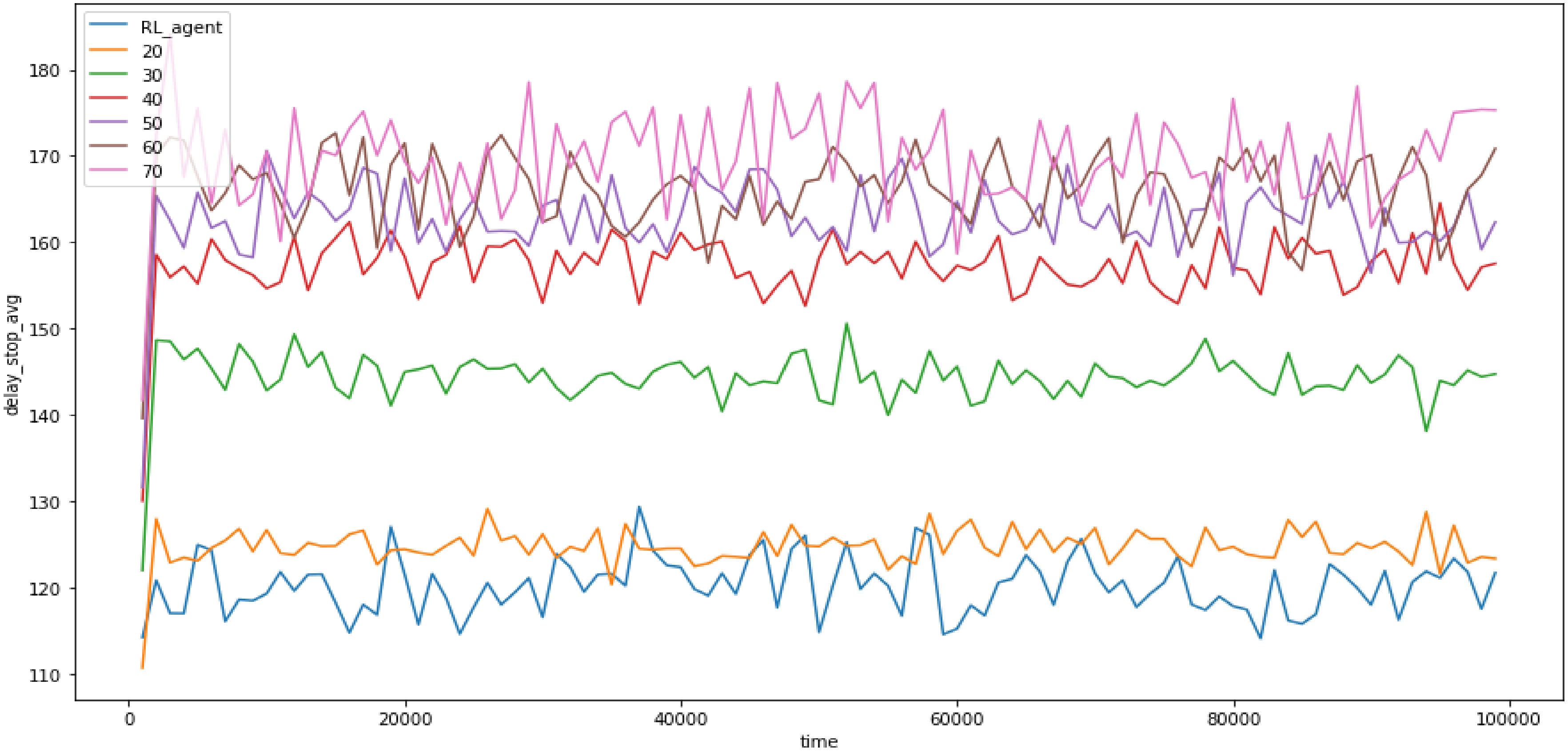
# Experimentation and Results

- 3k vehicles per hour at every incoming lane,
- Avg Speed Comparison, calculated in 1000secs interval

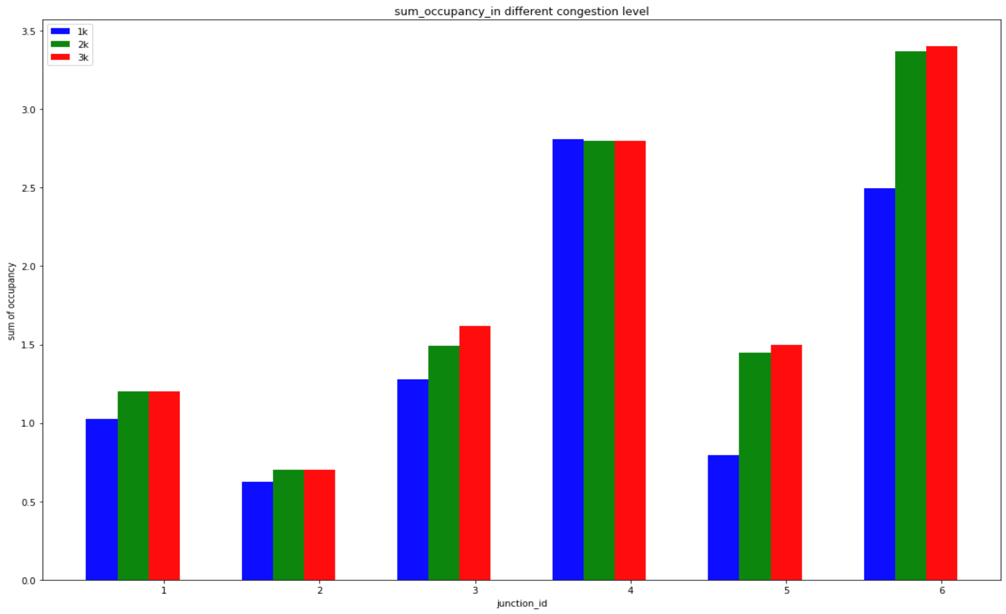


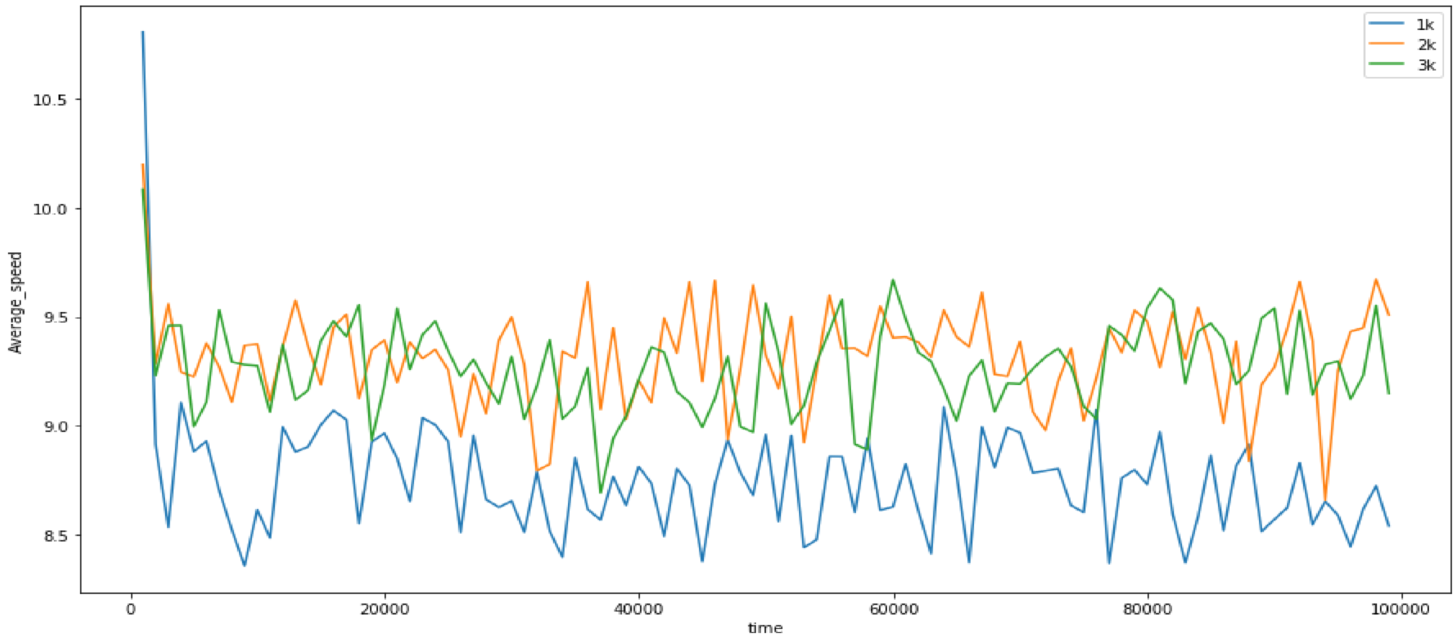
# Experimentation and Results

- 3k vehicles per hour at every incoming lane
- Delay Stop Avg, calculated in 1000secs interval



# RL Agent Performance Comparison at different vehicle incoming density

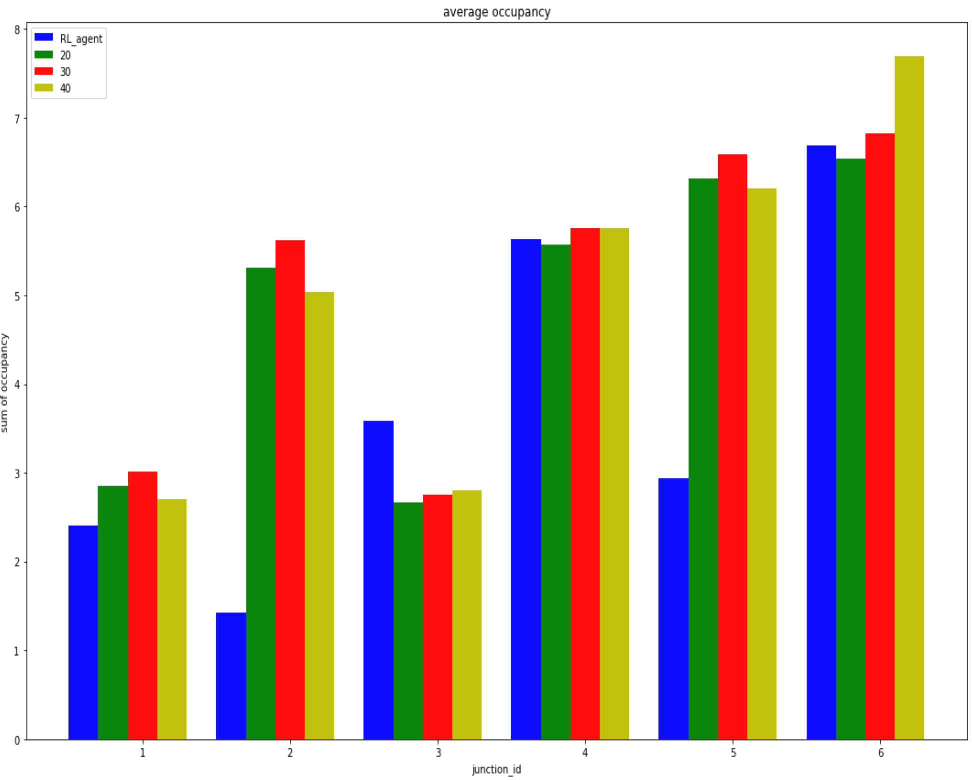


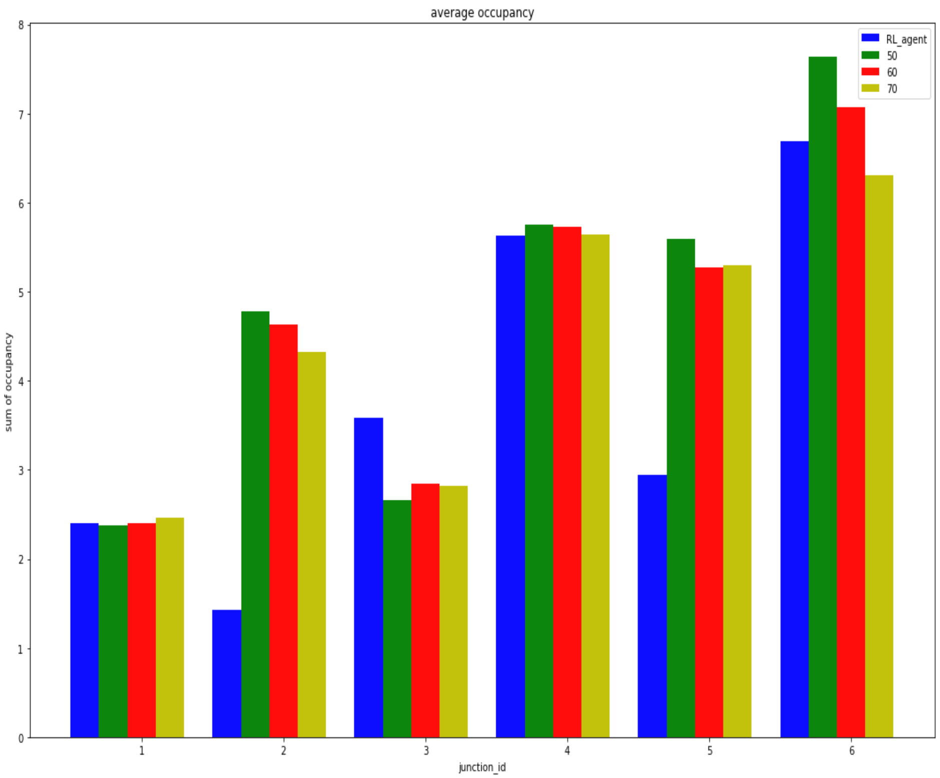



# RL Agent Performance Comparison at different vehicle incoming density

# Experimentation and Results

- Random vehicle input density:
  - Changing after every 50 minutes
  - Selected in range {500,1000,……..,3000,3500} vehicles per hour
  - Asymmetric vehicle incoming density at every junction
- Avg Occupancy Comparison with fixed time algorithms

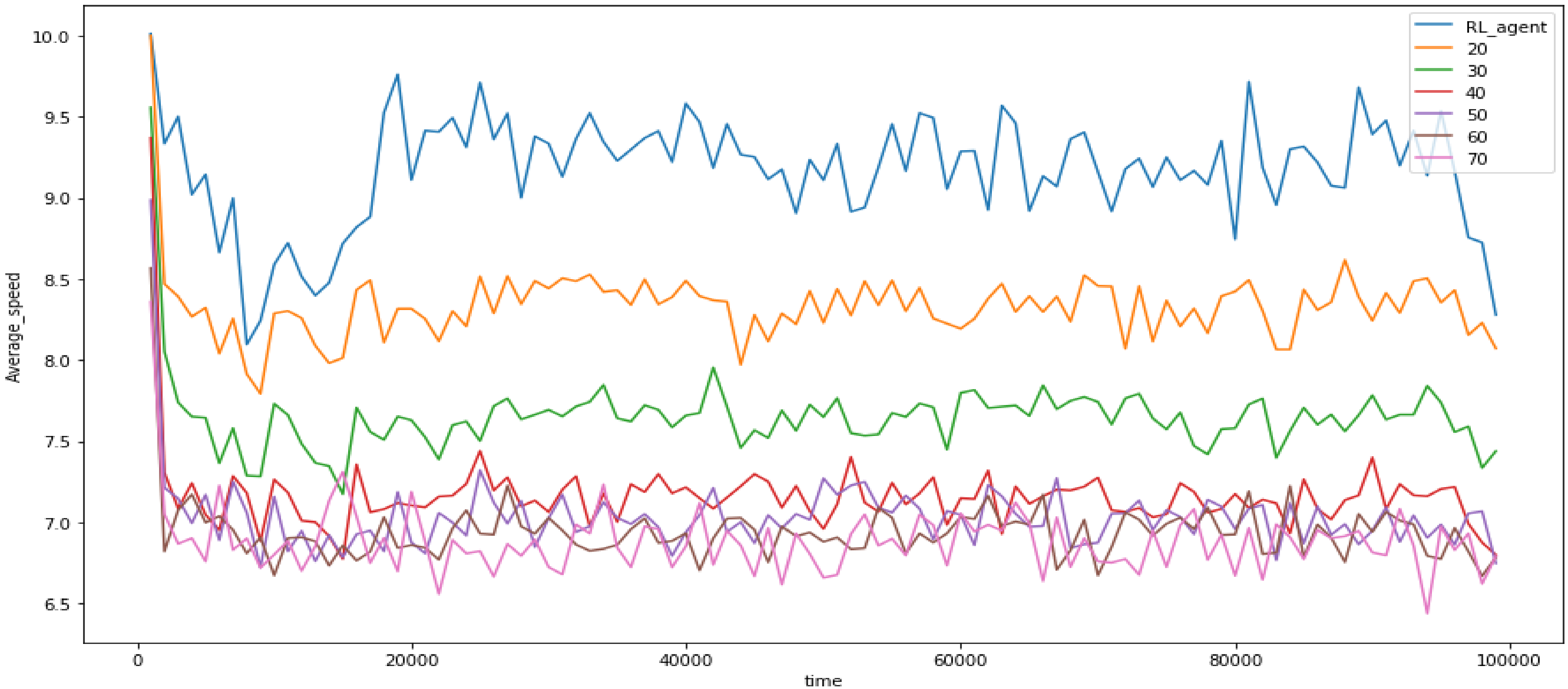



- ➢ Agent was never trained in this kind of changing environment.
- ➢ Still it has generalized reasonably well.

# Experimentation and Results

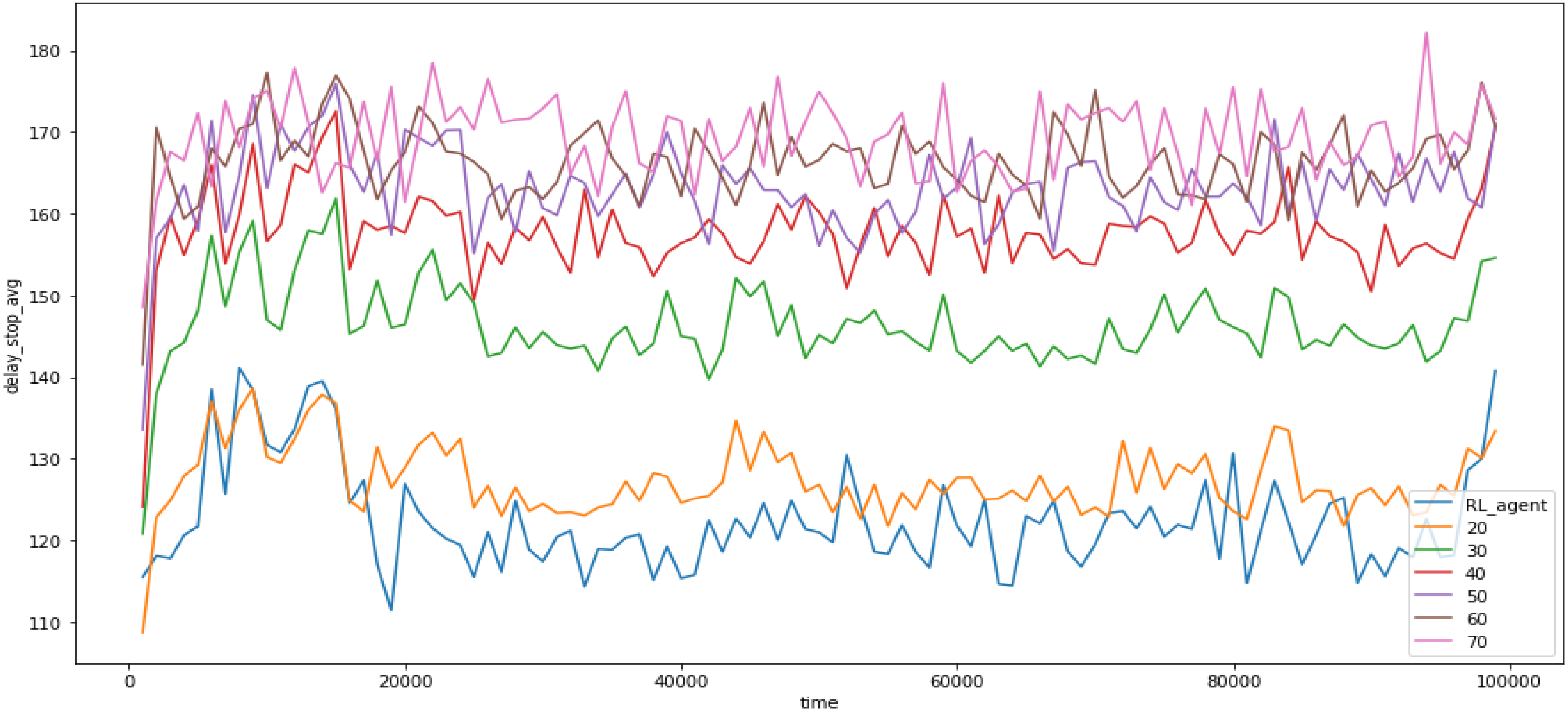
- Random vehicle input density:
  Changing after every 50 minutes
  Selected in range {500,1000,………,3000,3500} vehicles per hour
  Asymmetric vehicle incoming density at every junction
- Avg Speed Comparison with fixed time algorithms




- ➢ Agent was never trained in this kind of changing environment.
- ➢ Still it has outperformed all fixed time algorithms.

# Experimentation and Results

- Random vehicle input density:
  Changing after every 50 minutes
  Selected in range {500,1000,………,3000,3500} vehicles per hour
  Asymmetric vehicle incoming density at every junction
- Avg Delay Stop time Comparison with fixed time algorithms




➢ Agent was never trained in this kind of changing environment.

# Phasing Analysis

- For Asymmetric Network simple phasing performs better.

# References


- **Proximal Policy Optimization Algorithms (**John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, Oleg Klimov )

- **Multi-agent reinforcement learning for traffic signal control**, Prabuchandran K.J. , Hemanth Kumar A.N , Shalabh Bhatnagar