# Learning Model Parameters for Decentralized Schedule-Driven Traffic Control

Carnegie Mellon University The Robotics Institute

Carnegie Mellon Electrical & Computer ENGINEERING

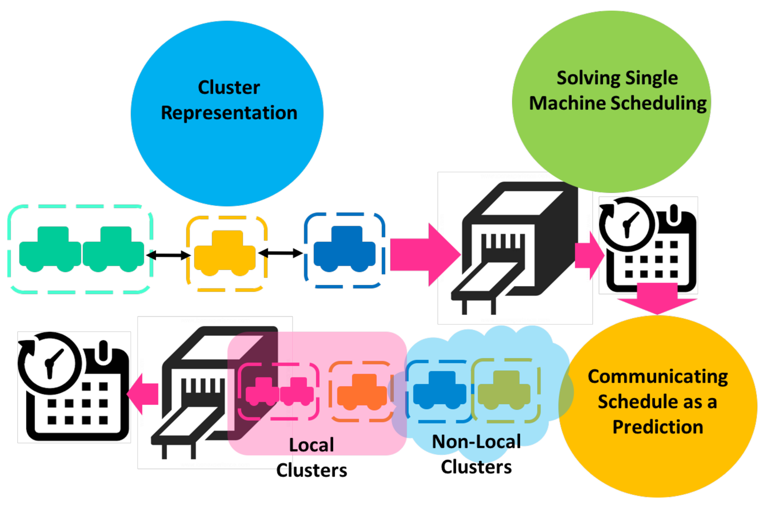
Hsu-Chieh Hu and Stephen F. Smith

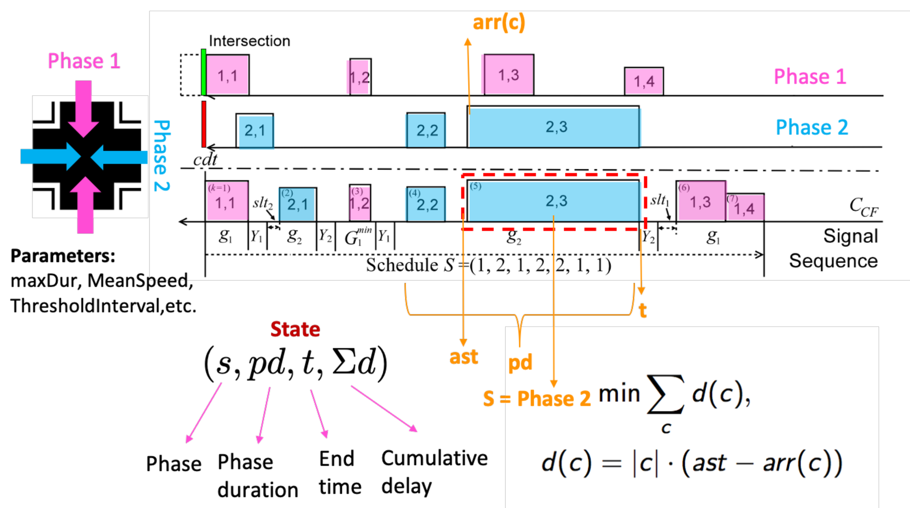
hsuchieh@andrew.cmu.edu, sfs@cs.cmu.edu

## Overview

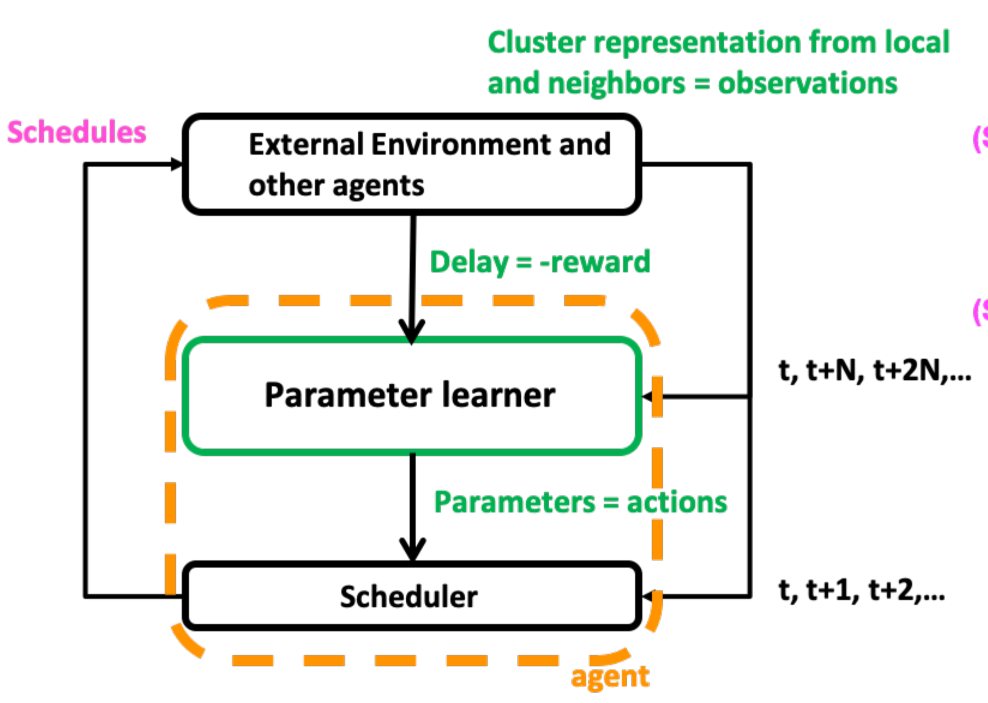
- Schedule-driven traffic control is a decentralized online planning approach for traffic signal control.
- This approach is a model-based intersection optimization that depends on the accuracy of control model and model/setting parameters
- We propose to combine planning and learning to set these parameters for different traffic patterns.

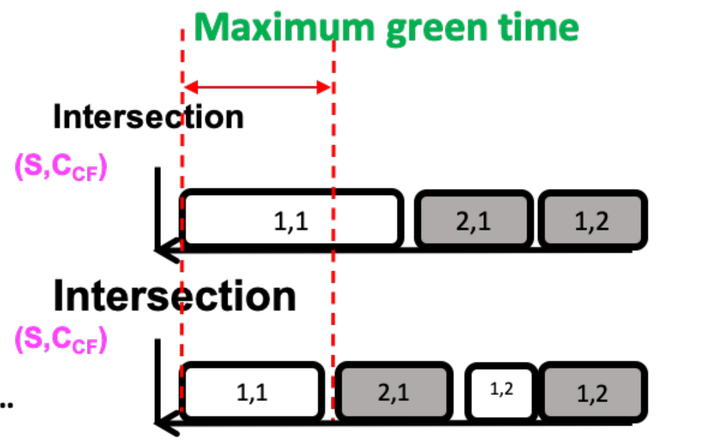
## Schedule-Driven Traffic Control




## A Fully Decentralized Hierarchical Algorithm



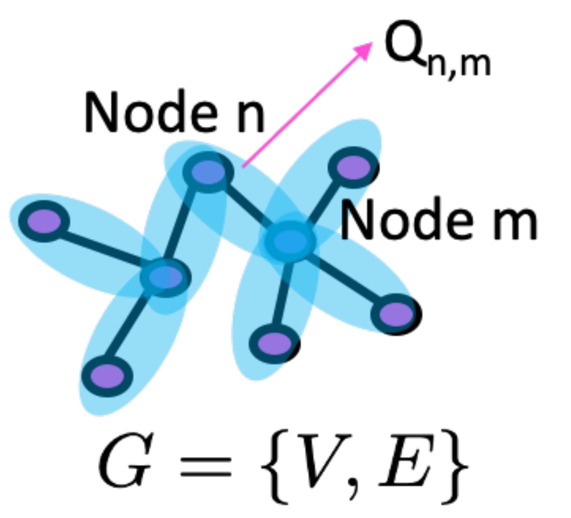

**Observations (State):** traffic data
**Reward:** negative delay
**Actions:** parameter

- Agent (intersection) computes its gradient locally given neighbors' actions and the global state of network.
- The requirement of knowing global state can be relaxed through utilizing neighbor-shared information.

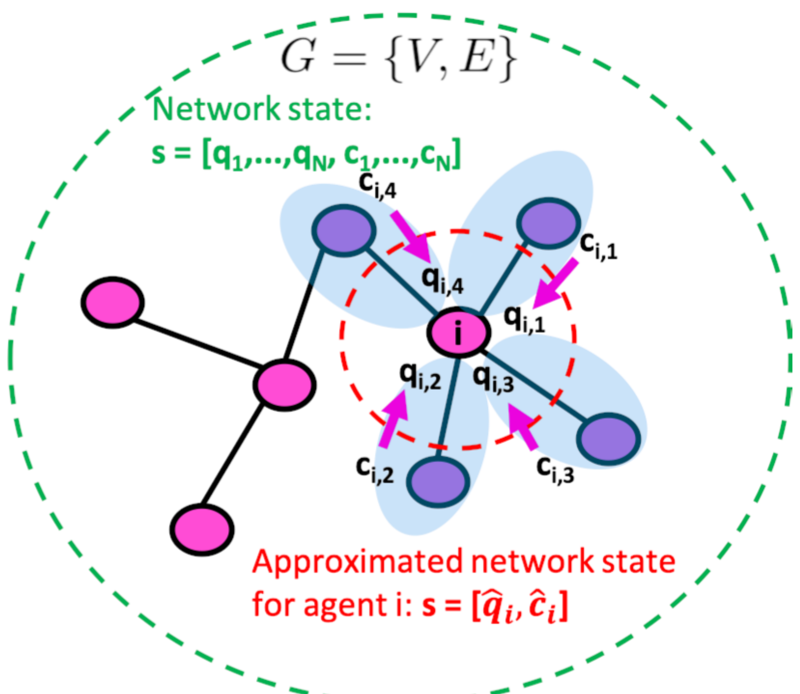
$$\nabla_{\theta_i} J(\boldsymbol{\mu}) = \mathbb{E}_{s\sim\rho^{\mu}}\left[\nabla_{\theta_i} Q^{\boldsymbol{\mu}}(s,\boldsymbol{\mu})\right] = \mathbb{E}_{s\sim\rho^{\mu}}\left[\nabla_{\theta_i}\mu_{\theta_i}(s)\nabla_{a_i} Q^{\boldsymbol{\mu}}(s,\mathbf{a})|_{a_i=\mu_{\theta_i}(s)}\right]$$

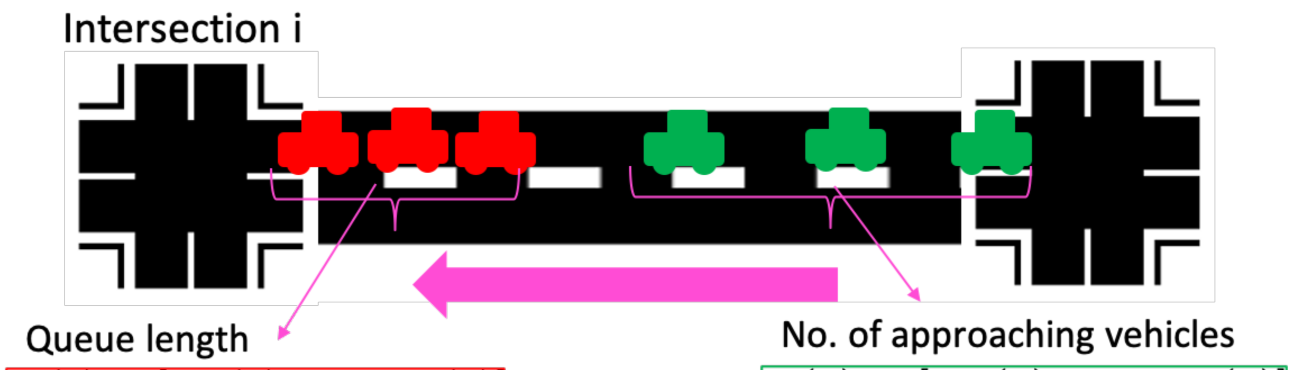
$$Q^{\boldsymbol{\mu}}(s,\mathbf{a}) = \sum_{(n,m)\in E} Q_{n,m}(s,a_n,a_m)$$

$$\nabla_{\theta_i} J(\boldsymbol{\mu}) = \mathbb{E}_{s\sim\rho^{\mu}}\left[\nabla_{\theta_i}\mu_{\theta_i}(s)\nabla_{a_i} \sum_{(n,m)\in E} Q_{n,m}(s,a_n,a_m)|_{a_i=\mu_{\theta_i}(s)}\right] = \mathbb{E}_{s\sim\rho^{\mu}}\left[\nabla_{\theta_i}\mu_{\theta_i}(s)\nabla_{a_i} \sum_{j\in\mathcal{N}_i} Q_{i,j}(s,a_i,a_j)|_{a_i=\mu_{\theta_i}(s)}\right]$$




V: a set of intersections
E: a set of road links




$\mathbf{q}_i(\tau) = [q_{1,i}(\tau), \cdots, q_{p,i}(\tau)]$ $\quad \mathbf{c}_i(\tau) = [c_{1,i}(\tau), \cdots, c_{p,i}(\tau)]$

**Features of state** $\hat{\mathbf{q}}_i = [\hat{q}_{1,i}, \cdots, \hat{q}_{P,i}]$ $\quad \hat{\mathbf{c}}_i = [\hat{c}_{1,i}, \cdots, \hat{c}_{P,i}]$

$\hat{q}_{p,i} = \frac{1}{T}(q_{p,i}(t) + \cdots + q_{p,i}(t+T-1))$ Average queue length and flow rate over T

**Actions** $\mathbf{G} = [G_{1,max}, \cdots, G_{P,max}]$ Maximum green time

**-Reward** $\hat{l}_i = \frac{1}{T\times|\mathcal{N}_i|}\sum_{j\in\mathcal{N}_i}\sum_{\tau=t}^{t+T} d_j(\tau)$ Average delay over neighbors and T

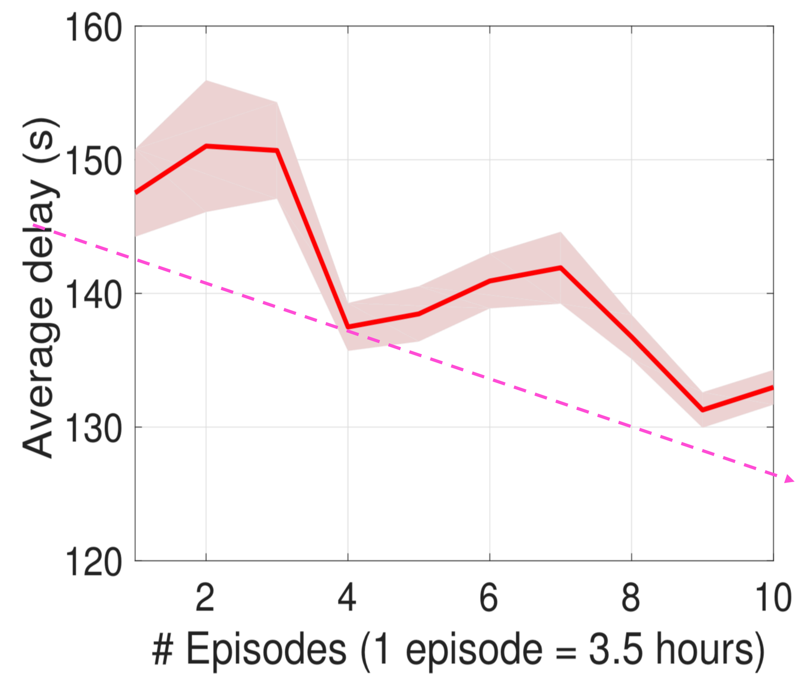
**For agent i,**
$s \approx (\hat{\mathbf{q}}_i, \hat{\mathbf{c}}_i)$ Global state
$(a_i, a_j) = (\mathbf{G}_i, \mathbf{G}_j)$ Parameters as actions

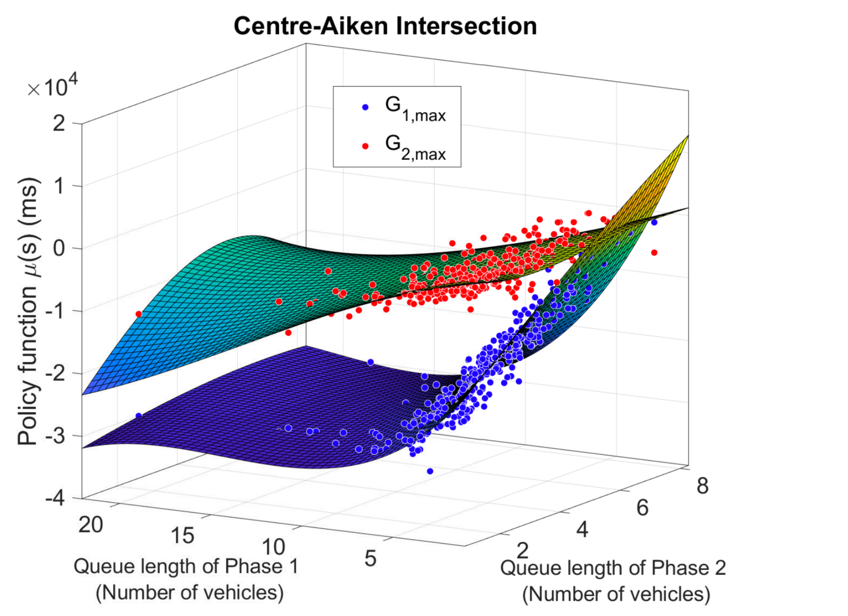
$$\sum_{j\in\mathcal{N}_i} Q_{i,j}^{\mu_{\theta_i}}(s, a_i, a_j) = \sum_{j\in\mathcal{N}_i} Q_i^{\mu_{\theta_i}}(\hat{\mathbf{q}}_i, \hat{\mathbf{c}}_i, \mathbf{G}_i, \mathbf{G}_j)$$

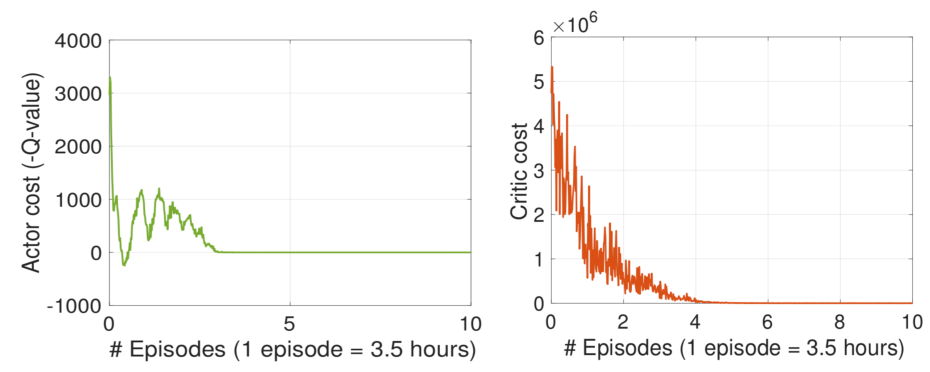
**Replay buffer** $(s', s, a_i, a_{\mathcal{N}_i}, \hat{l}_i) \equiv ([\hat{\mathbf{q}}_i', \hat{\mathbf{c}}_i'], [\hat{\mathbf{q}}_i, \hat{\mathbf{c}}_i], \mathbf{G}_i, \mathbf{G}_{\mathcal{N}_i}, \hat{l}_i) \sim \mathcal{D}_i$

## Experimental Results

| | Average Delay (second) | | |
|---|---|---|---|
| | mean | std. | stop no. |
| DQN | 63.78 | 53.35 | 1.75 |
| DDPG | 57.80 | 47.86 | 1.59 |
| Bench (50s, 60s) | 89.62 | 78.64 | 2.91 |
| Bench (50s, 90s) | 67.66 | 55.50 | 1.91 |
| Bench (50s, 120s) | 73.49 | 62.18 | 2.34 |




(a) Actor cost (b) Critic cost

Apply **episodic RL** here for avoiding "terminate" state (irreducible state transition) that queueing stability cannot be retained

| | Average Delay (second) and Number of Stops | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Benchmark | | | Hierachical | | | Cycle-based Adaptive | | |
| | mean | std. | stop no. | mean | std. | stop no. | mean | std. | stop no. |
| High demand | 212.14 | 361.41 | 9.55 | **132.98** | **92.95** | **6.76** | 230.26 | 279.19 | 12.34 |
| Medium demand | 84.22 | 61.90 | 6.34 | **82.56** | **55.84** | **4.56** | 86.46 | 61.40 | 8.78 |
| Low demand | 71.84 | 54.25 | 6.12 | **72.10** | **49.11** | **4.23** | 73.89 | 56.77 | 8.11 |
| PM rush hour | 147.00 | 177.94 | 8.27 | **113.89** | **88.24** | **5.10** | 169.23 | 265.91 | 10.81 |

- **Model:** Two-way queueing grid network
- **Traffic:** High 1056cars/hour; Medium 708cars/hour; Low 472cars/hour
- **Simulator:** Vissim

RL-based (DDPG):
- layer [30,30]
- Reward discount $\gamma = 0.99$
- Learning rate $\alpha_a = 0.001$, $\alpha_c = 0.002$
- Buffer size = **1000**
- Batch size = 20
- EMA $\tau = 0.01$

## Conclusions

- Self-improving to stabilize a network is realized through learning to configure parameters of the model, and real-time responsiveness is still retained.
- Hierarchical abstraction retains both advantages from learning and planning.
- Sharing real-time information to neighbors can enable decentralized multi-agent learning by approximating global state.

This research was funded in part by the University Transportation Center on Technologies for Safe and Efficient Transportation at Carnegie Mellon University and the CMU Robotics Institute.