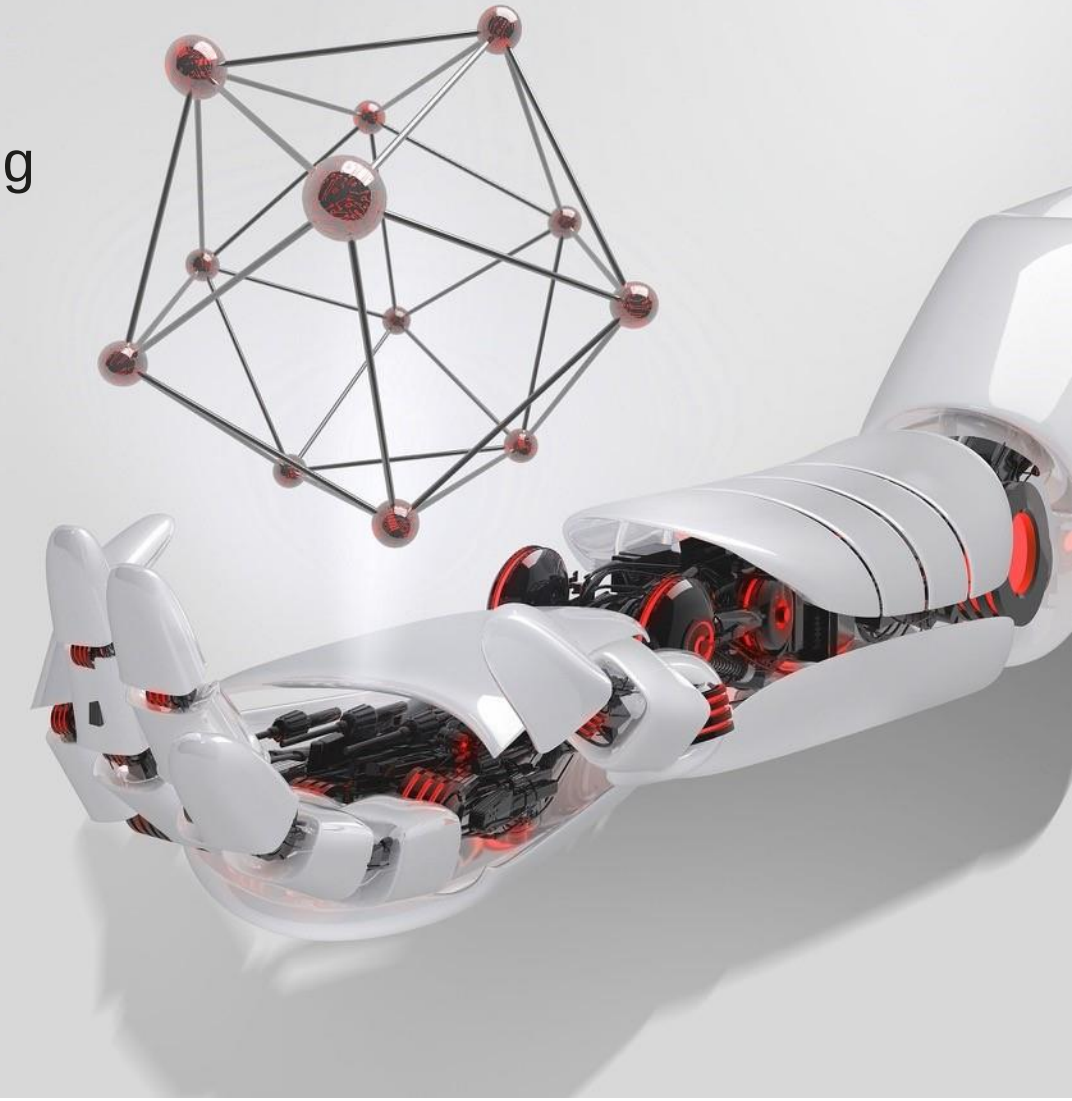
Constrained Policy Optimization for Load Balancing

Ahmed Yassine Kamri, Pham Tran Anh Quang,
Nicolas Huin, Jeremie Leguay

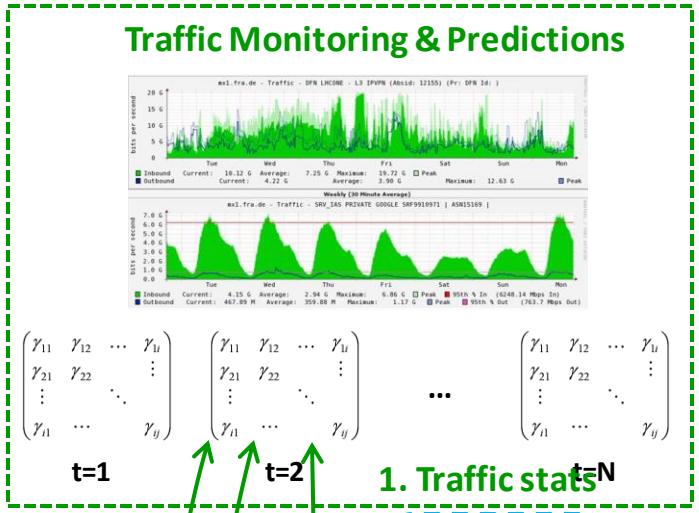
AlgoTel 2021, La Rochelle

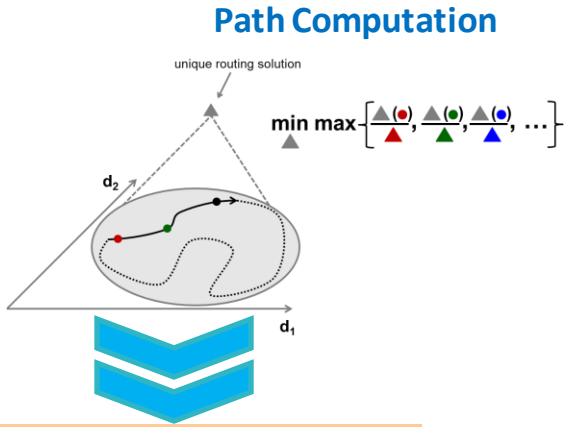


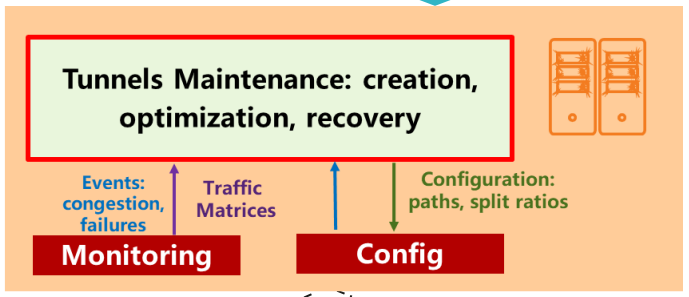
HUAWEI

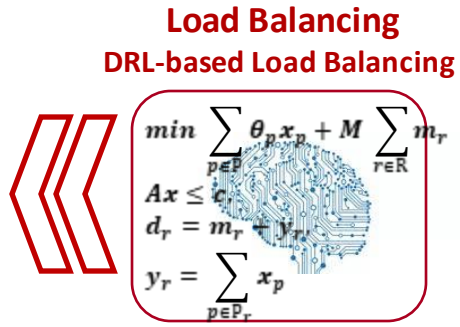
Centralized TE architecture

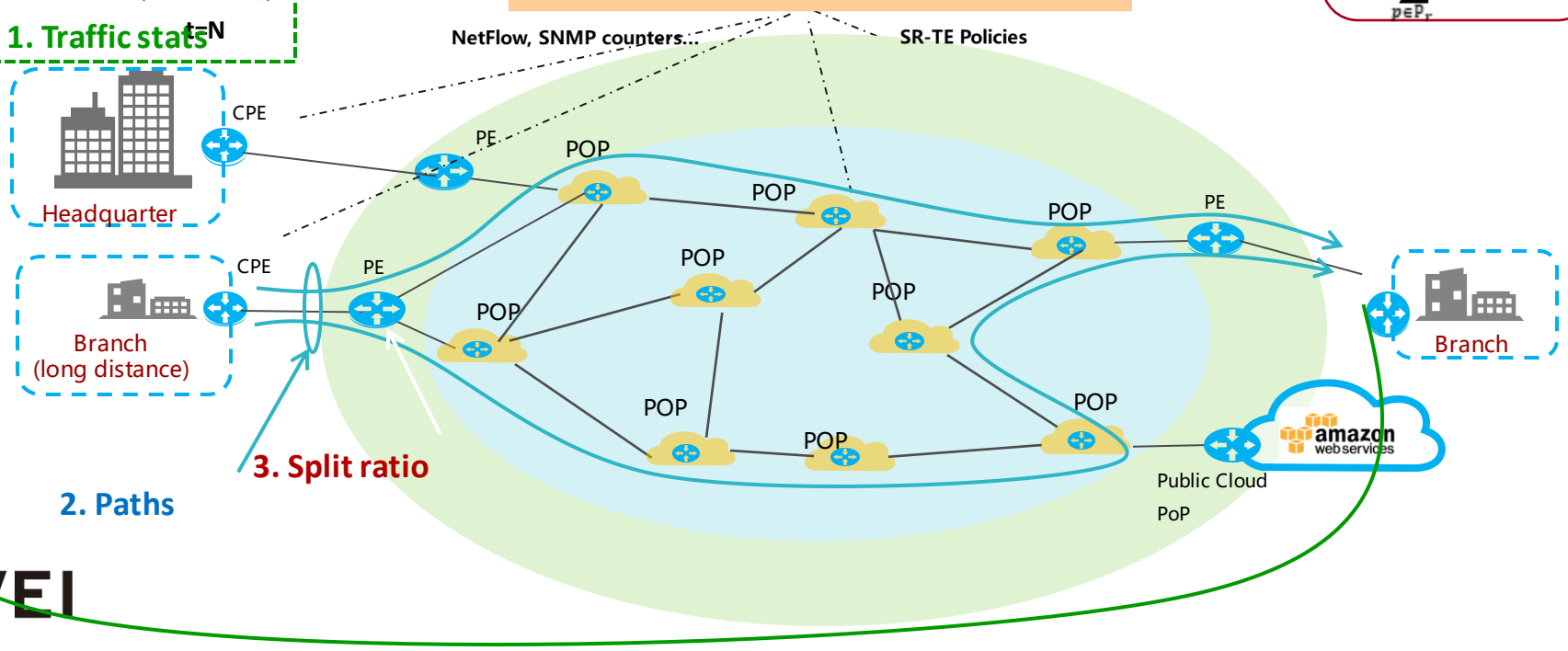
Slow and Fast Control Loops







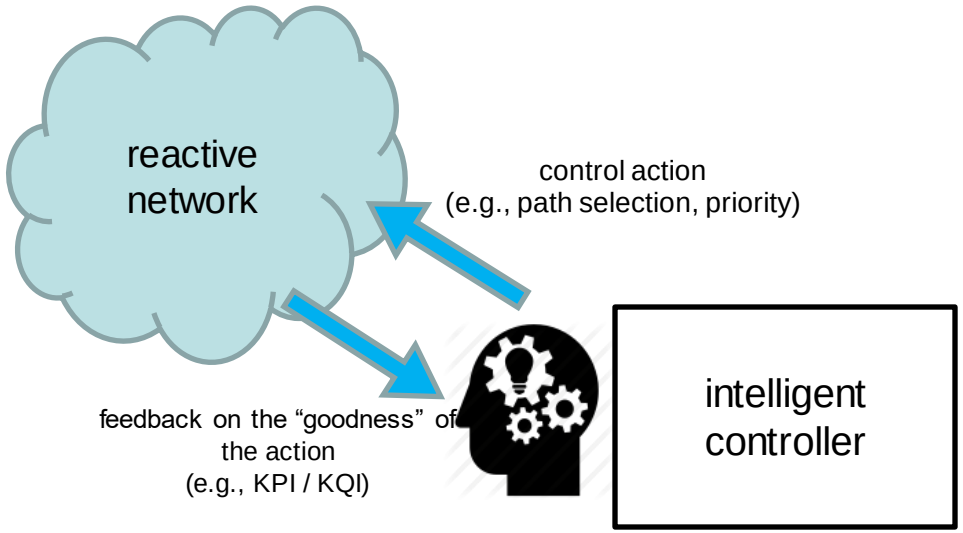




Load Balancing: A good use case for RL

Reinforcement Learning is ideal for non-explicit model

- Traffic patterns are difficult to characterize
- The impact of routing decisions on KPIs (delay, jitter, packet loss) is difficult to anticipate
- The impact of routing decisions on KQI (e.g., QoE) is difficult to anticipate

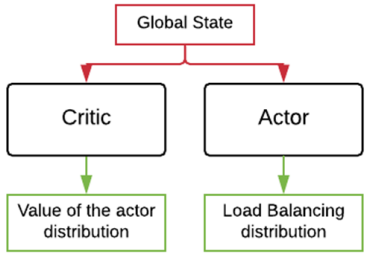


Load Balancing: A good use case for RL

RILNET

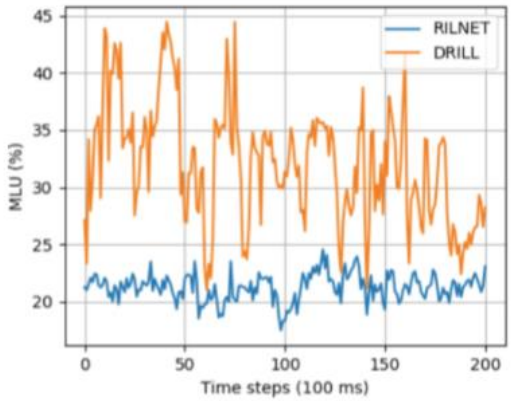
RILNET: A Reinforcement Learning Based Load Balancing Approach for Datacenter Networks. MLN 2018.

- Centralized Actor / Critic approach
- Offline training based on an abstract model (training speedup)
- Fixed policy
- State of art algorithm : DDPG (Deep Deterministic Policy Gradients)

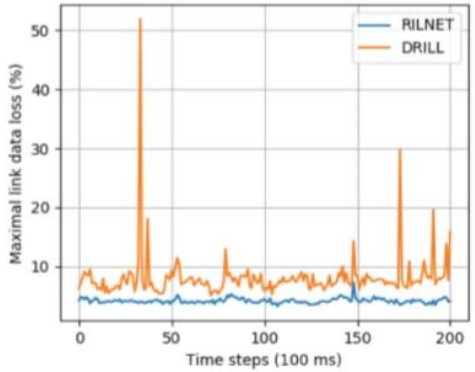


Results

- Compared to DRILL in DCN
  - Improvement of the MLU by 30% and link delay / latency by 40%



(a) MLU



(c) Maximal link data loss

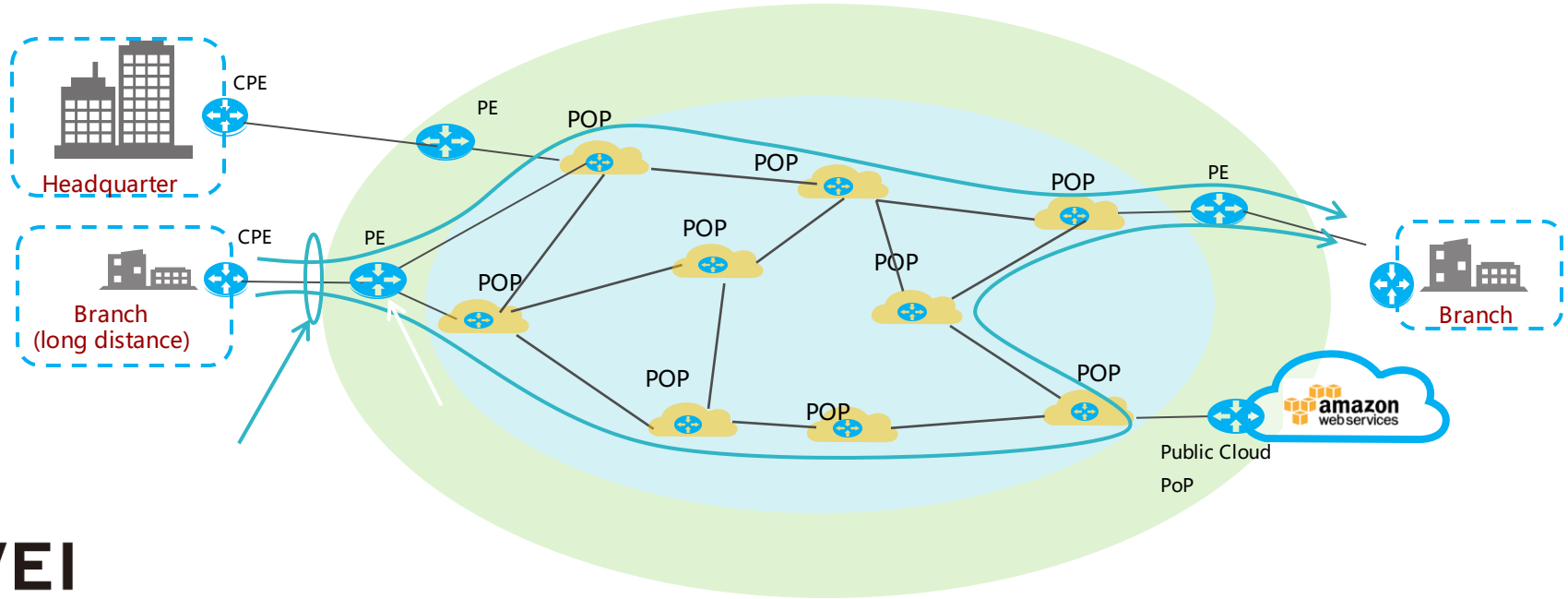
HUAWEI

Outline

- Centralized TE
- Load balancing model
- Constrained MDP & RCPO
- Simulation results
- Conclusions

Load balancing problem

Time-based slot
Pre-defined paths
Only split ratio are changed
Minimize delay without too much drops



HUAWEI

MM1 delay model

Markovian arrival process:

Packet arrivals occur at rate λ according to Poisson process

Markovian service process:

Service times have exponential distribution μ

1 server

Packet size

Propagation delay

$$\frac{S}{C_e - \sum_k d^k(t) \sum_{\substack{p \in P^k \\ e \in p}} x_p^k\ (t)} + \lambda_e$$

Capacity (bps)

Load (bps)

http://www.pitt.edu/~dtipper/2120/mm1_tutorial.pdf

NLP-based approach

Delay variable for tunnels

$$\min \sum_{k \in K} d_k$$

Linking constraints

$$D_k y_{pk} \geq x_{pk} \qquad \forall k \in K, p \in \mathbb{P}_k$$

$$D_k x_{pk} \geq y_{pk} \qquad \forall k \in K, p \in \mathbb{P}_k$$

Delay variable of edges

MM1 constraints

$$z_e \geq \frac{1}{c_e - \sum_{k \in K} \sum_{p \in \mathbb{P}_k : e \in p} x_{pk}} \qquad \forall e \in E$$

Capacity constraints

$$\sum_{k \in K} \sum_{p \in \mathbb{P}_k : e \in p} x_{pk} \leq c_e, \qquad \forall e \in E$$

Ratio variable for tunnels

Delay constraints

$$\sum_{e \in p} z_e y_{pk} \leq d_k \qquad \forall k \in K, p \in \mathbb{P}_k$$

HUAWEI

LearnQueue [Bouacida19]

Proposed for wireless network
Two-part reward (delay and loss)
Manual configuration for priority

Learn Queue reward $r_t = r_d^t + r_{enq}^t$
Delay related part: $r_d^t = -\delta \sum_k d_k(t)$
Loss related part: $r_{enq}^t = (1-\delta) er^t$

[Bouacida19] N. Bouacida and B. Shihada, "Practical and Dynamic Buffer Sizing Using LearnQueue," in IEEE Transactions on Mobile Computing, vol. 18, no. 8, pp. 1885-1897, 1 Aug. 2019

Constrained MDP & Reward Constrained Policy Optimisation

State $s \in \mathbb{S}$
Action $a \in \mathbb{A}$
Transition model P(s' | s, a): probability of going to s' from s depends only on s and action a
Reward $r(s_t, a_t)$
Penalty $c(s_t, a_t)$
μ: initial state distribution
Constraint $C(s_t) = F(c(s_t, a_t), \ldots, c(s_N, a_N))$, e.g.
F: discounted sum or average

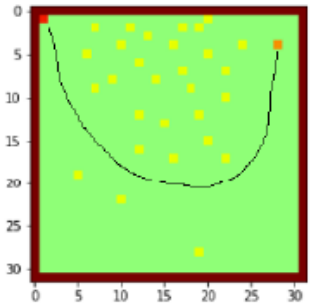
$$\max_{\pi \in \Pi} J_R^\pi$$

s.t. $J_C^\pi \leq \alpha$

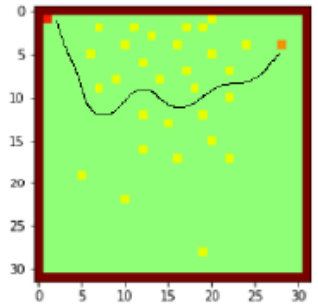
$\alpha \in [0,1]$

$$J_R^\pi = \mathbb{E}_{s_0 \sim \mu}^\pi \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right]$$

$$J_C^\pi = \mathbb{E}_{s_{0 \sim \mu}}^\pi [C(s)]$$



(a) $\alpha = 0.01$



(b) $\alpha = 0.5$

Lagrangian relaxation

$$\min_{\lambda \geq 0} \max_{\theta} L(\lambda, \theta) = \left[J_R^{\pi_\theta} - \lambda \left(J_C^{\pi_\theta} - \alpha \right) \right]$$

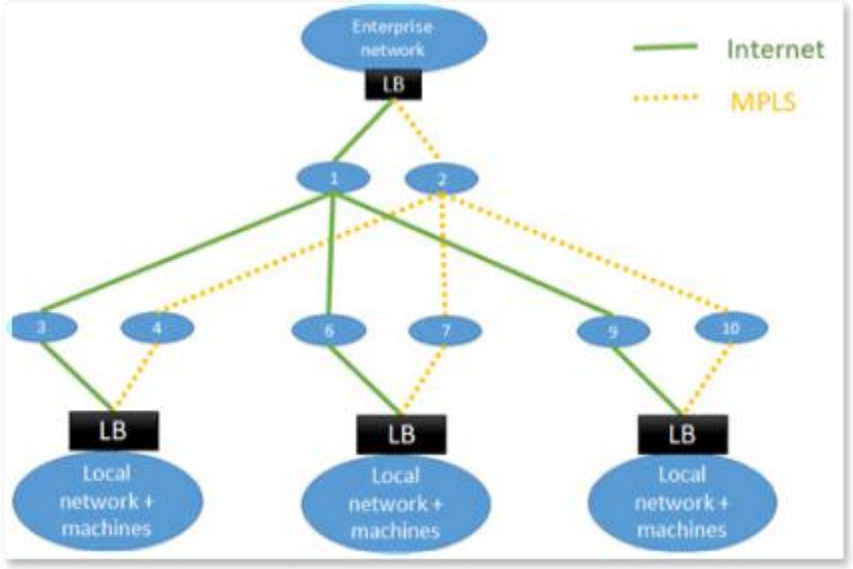
$$\hat{r}(\lambda, s, a) \triangleq r(s, a) - \lambda c(s, a)$$

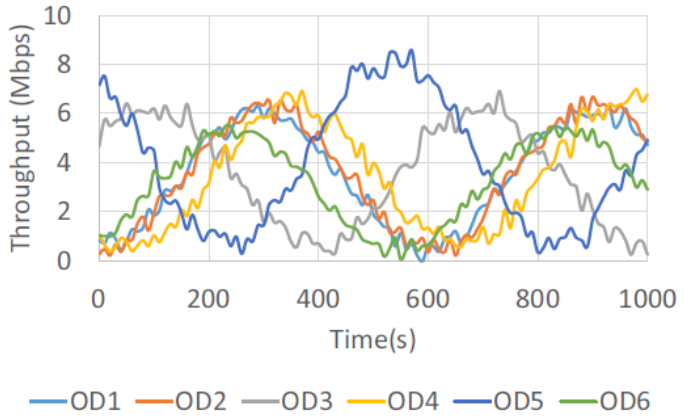
Outline

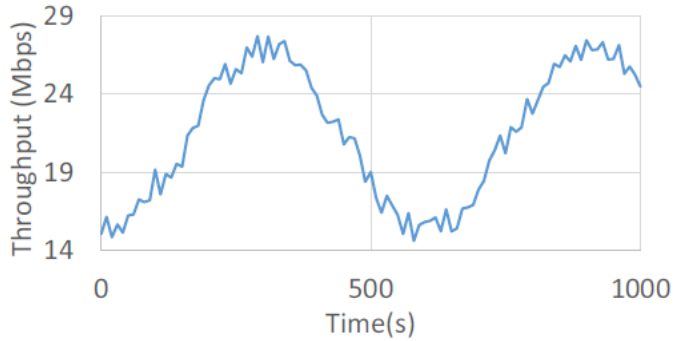
- Centralized TE
- Load balancing model
- Constrained MDP & RCPO
- Simulation results
- Conclusions

Simulation scenario

SD-WAN toy scenario





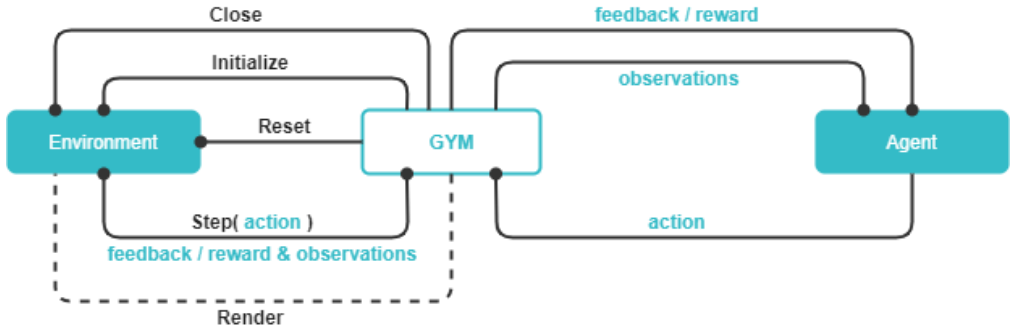


Algorithms			
DDPG-LearnQueue 0.1	DDPG-LearnQueue 0.9	RCPO	Model-based approach
LearnQueue reward with $\delta = 0.1$ (Prefer Traffic Admission)	LearnQueue reward with $\delta = 0.9$ (Prefer low delay)	Penalized reward function	Delay
Stable baseline		Stable baseline + modifications	SCIP
2 hidden layers (128 neurons each) ReLU $\gamma = 0.7$		Similar to DDPG-LQ $\eta_1 = 0.01$	Max time=60s

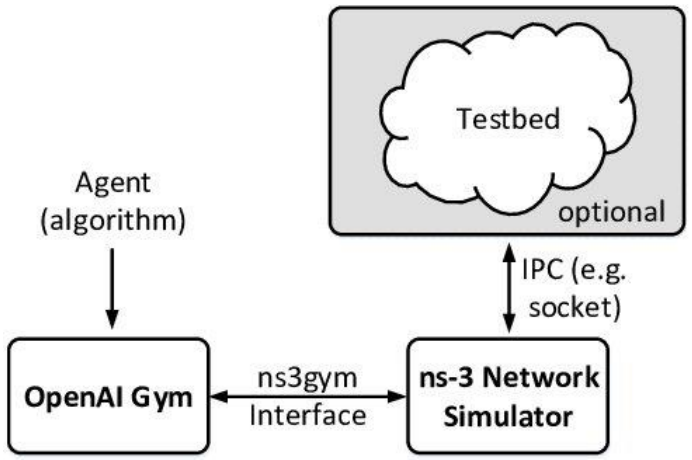
HUAWEI

OpenAI Gym environment

"Simplified environment" with M/M/1 model



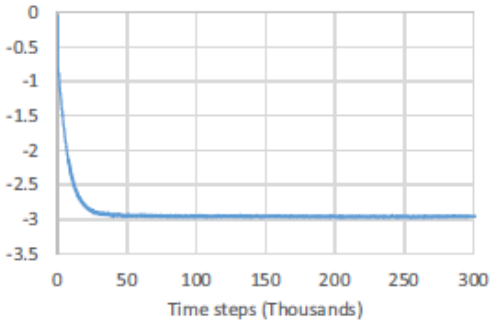
Possible to extend to a packet-level simulator (i.e., ns3) [Gawłowicz18]



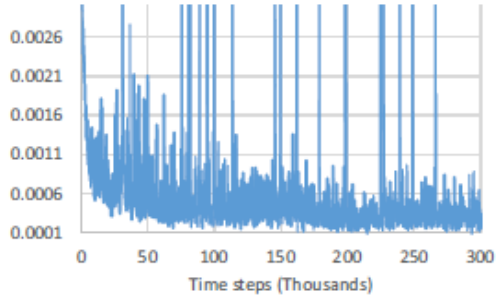
[Gawłowicz18] Gawłowicz, Piotr, and Anatolij Zubow. "ns3-gym: Extending openai gym for networking research." *arXiv preprint arXiv:1810.03943* (2018).

Convergence

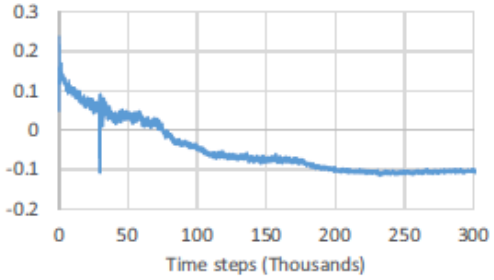
LearnQueue



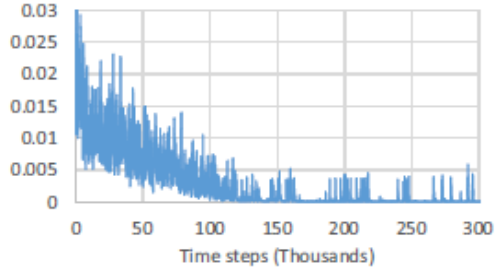
(a) Actor, $\delta = 0.1$



(b) Critic, $\delta = 0.1$

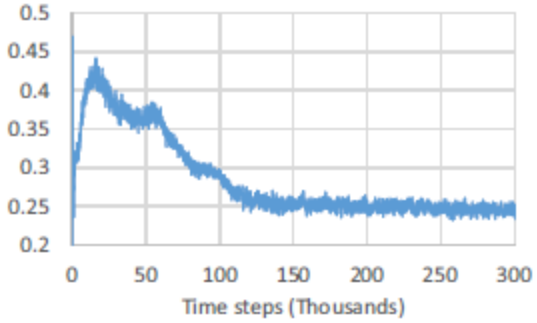


(c) Actor, $\delta = 0.9$

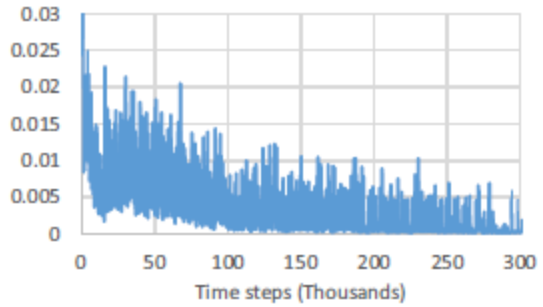


(d) Critic, $\delta = 0.9$

RCPO



(a) Actor loss

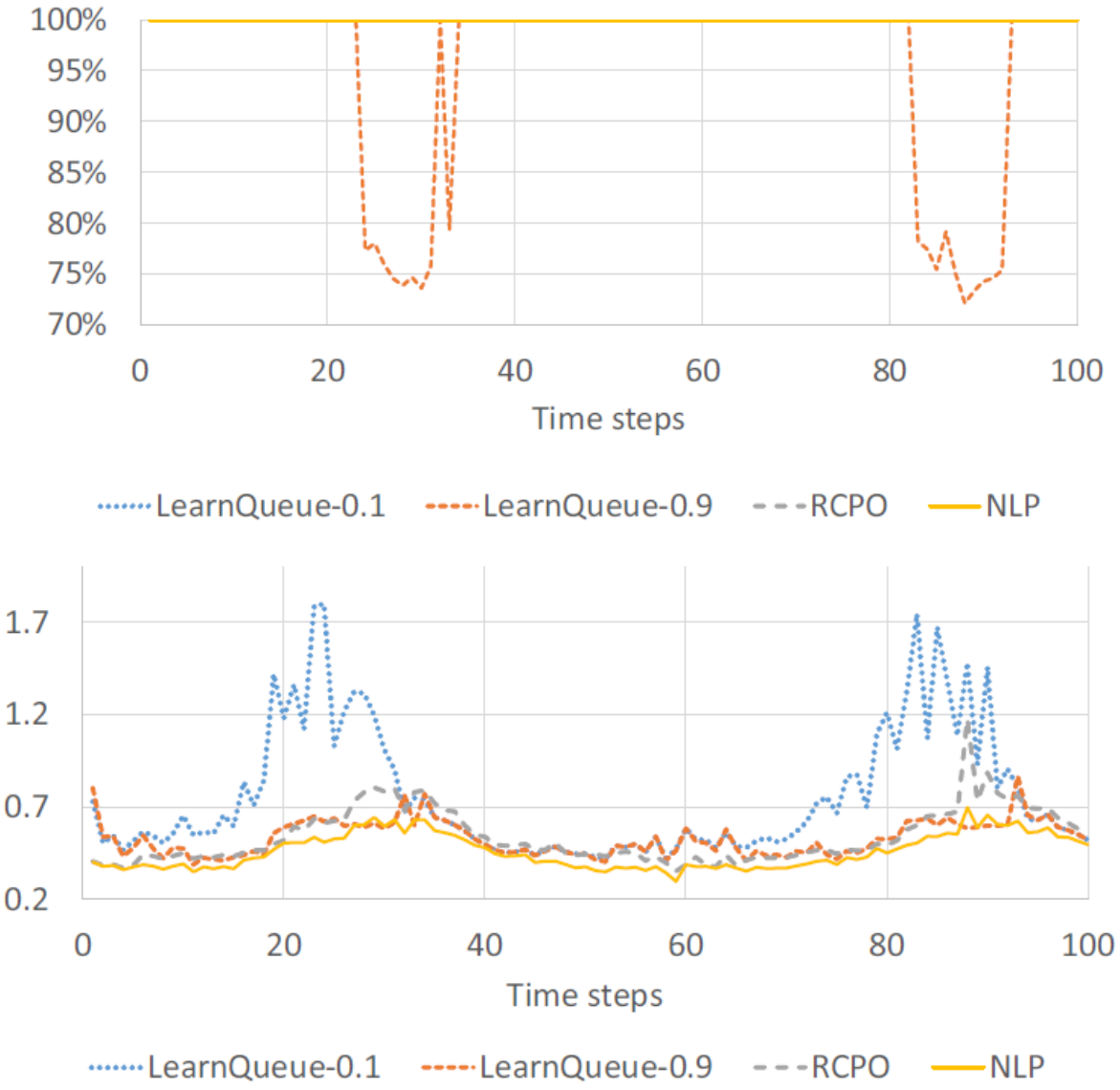


(b) Critic loss

Convergence after 100k time steps

HUAWEI

Simulation results



LearnQueue:

- Prefer delay ($\delta = 0.9$) : RL agent will reject traffic when congestion happens
- Prefer traffic admission ($\delta = 0.1$): RL agent will admit 100% traffic even it increases the delay

RCPO: well balanced between traffic admission and good delay (comparable with NLP)

Conclusions & Perspective

- RCPO minimize constraint violations
- Comparable to optimal solution

- Packet-level simulation
- Comparison with Control Barrier Function

Thank you for
your attention