MACCC: Multi-Agent Constrained Congestion Control

Duo Zhang, Yuhao Fan

Introdution

With the rapid and thriving developments of the Internet in recent years, numerous emerging and rising technologies like Internet of Things, VR & AR applications, High-res audio and video streaming, online gaming and so on are influencing people's life and make more demands on the quality of internet services. TCP can become really sluggish and inefficient in some certain circumstances, where the conventional rule-based congestion control scheme fails to adjust with the fluctuating dynamics of the Internet. To handle the incapability of rule-based congestion control methods, [Winstein et al., 2013] provides us a statistical approach of forecasting the threshold of input rate of data(bytes) that prevent a packet from getting queued for too long. Also, [Dong et al., 2018] uses multi-armed bandits based rate-control strategy. However, from the dynamic control theory point of view, deep reinforcement learning seems to be a perfect weapon to handle such a big network with multiple ends to cooperate with each other. [Jay et al., 2019] proposed a RL-based method to dynamically control the input rate in a single linkage network with multiple users. Though this method had reached the SOTA compared with other baselines, it failed to introduce fairness to the system. Additionally, according to the formulation of their reward function, the method failed to build a multi-objective network. Thus, based on [Jay et al., 2019], our team want to propose a new RL-based method to address the congestion and fairness control problem in a network with multiple links and multiple end users. We plan to design a novel reward function with constraints to introduce congestion control and fairness in the same time and make the senders in the network to be multiple agents do cooperate with each other to maximize the reward based on limited state information available to them.

Methodology

Multi-Agents

In the network, each user is an agent(A_i) is a player and interconnected in the simulated network which is comprised of a simple fully connected network. They can receive information of the states s_{t_i} from packet headers. Based on the states, the agents need to explore and probe the dynamics of the whole network by making observations o_{t_i} from s_{t_i}, where o_{t_i} is a prediction about the environment such as per-flow round trip time(rtt_j for flow j) and congestion degree(d_j). According to their own observation, they need to cooperate and make their actions(a_{ij}) to maximize the total reward R of all agents.

Reward Function

Throughput

We plan to use the weighted mean throughout in every link as the main reward, so the cumulative reward function would be:

$$R = \mathbb{E}\left[\sum_{t=0}^{\infty}\sum_{k=1}^{n}\gamma^t\Phi_k \middle| s_0, \pi\right] \tag{1}$$

where n is the number of links, Φ_k is the throughput in k-th link, π is the current policy and s_0 is the initial state of the network. Since the actions are applied to each flow, we can derive that:

$$R = \mathbb{E}\left[\sum_{t=0}^{\infty}\sum_{k=1}^{n}\gamma^t\sum_{j=1}\phi\left(o_{tj}^{(k)}(s_{tj}^{(k)}), a_{tj}^{(k)}\right) \middle| s_0, \pi\right] \tag{2}$$

where j denotes the index of every flow in a link, ϕ, $o_{tj}^{(k)}, s_{tj}^{(k)}$ and $a_{tj}^{(k)}$ are the throughput, observation based on state, state and action on flow j at time t of link k respectively.

Delay

For delay, the purpose is to control the delay below a given threshold, say T, then we can have the first constraint:

$$D = \mathbb{E}\left[\sum_{t=0}^{\infty}\sum_{k=1}^{n}\gamma^t\sum_{j=1}\delta\left(o_{tj}^{(k)}(s_{tj}^{(k)}), a_{tj}^{(k)}\right) \middle| s_0, \pi\right] \leq T \tag{3}$$

where δ is the delay given observation and the action on flow j at time t of link k.

Fairness

As for fairness control, we want to introduce a price-based method. Let's denote the normalized prices for each flow the users are willing to pay as $\mathbf{p} = [p_1, \ldots, p_j]$, then the bandwidth $\mathbf{b} = [p_1, \ldots, p_j]$ should be weighted by $\mathbf{p}$, which leads to

$$\frac{\mathbf{b}\cdot\mathbf{p}}{\|\mathbf{b}\|\cdot\|\mathbf{p}\|} = 1$$

Thus we have,

$$F = \mathbb{E}\left[\sum_{t=0}^{\infty}\sum_{k=1}^{n}\gamma^t\left(\frac{\mathbf{b}_k\cdot\mathbf{p}_k}{\|\mathbf{b}_k\|\cdot\|\mathbf{p}_k\|} - 1\right)^2 \middle| s_0, \pi\right] = 0 \tag{4}$$

Formulation

Now we have the whole optimization formulation:

$$\begin{aligned} max_\pi \quad & \mathbb{E}\left[\sum_{t=0}^{\infty}\sum_{k=1}^{n}\gamma^t\sum_{j=1}\phi\left(o_{tj}^{(k)}(s_{tj}^{(k)}), a_{tj}^{(k)}\right) \middle| s_0, \pi\right] \\ s.t. \quad & \mathbb{E}\left[\sum_{t=0}^{\infty}\sum_{k=1}^{n}\gamma^t\sum_{j=1}\delta\left(o_{tj}^{(k)}(s_{tj}^{(k)}), a_{tj}^{(k)}\right) \middle| s_0, \pi\right] \leq T \\ & \mathbb{E}\left[\sum_{t=0}^{\infty}\sum_{k=1}^{n}\gamma^t\left(\frac{\mathbf{b}_k\cdot\mathbf{p}_k}{\|\mathbf{b}_k\|\cdot\|\mathbf{p}_k\|} - 1\right)^2 \middle| s_0, \pi\right] = 0 \end{aligned} \tag{5}$$

Then we can change this constrained formulation into unconstrained Lagrangian Reward function:

$$L(\lambda, \mu, s, \pi) = R(s, \pi) + \lambda D(s, \pi) + \mu F(s, \pi) \tag{6}$$

The formulation above is only our speculation and conjecture for now, the correctness needs to be further proved. Maybe this optimization has closed-form solution so there's no need for RL, but all of this needs to be answered when we dig deeper.

Resources We Need

The platform for Deep RL is PyTorch, OpenAI GYM.
The simulator we plan to use is: Pantheon.
For Model Training, we plan to use Greene cluster of NYU.

Experiment Pipeline

Our whole pipeline includes several steps and checkpoints:

1. Set up and test the environments *2.14-2.28*
2. Implement basic structure and train first model with single link(Two agents) *2.29-3.14*
3. Extend the model to multi-agents *3.15-4.15*
4. Run all the tests and experiments. Finish up the paper. *4.16-5.15*

References

[Dong et al., 2018] Dong, M., Meng, T., Zarchy, D., Arslan, E., Gilad, Y., Godfrey, B., and Schapira, M. (2018). PCC vivace: Online-Learning congestion control. In *15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18)*, pages 343–356, Renton, WA. USENIX Association.

[Jay et al., 2019] Jay, N., Rotman, N., Godfrey, B., Schapira, M., and Tamar, A. (2019). A deep reinforcement learning perspective on internet congestion control. In *International Conference on Machine Learning*, pages 3050–3059. PMLR.

[Winstein et al., 2013] Winstein, K., Sivaraman, A., and Balakrishnan, H. (2013). Stochastic forecasts achieve high throughput and low delay over cellular networks. In *10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13)*, pages 459–471.