Multi-agent meta reinforcement learning for packet routing in dynamic network environments

Shan Sun
ssun029@ucr.edu
University of California, Riverside

Mariam Kiran (advisor)
mkiran@es.net
Lawrence Berkeley National Lab

ABSTRACT

Traffic optimization challenges, such as flow scheduling and completion time reducing, are difficult online decision-making problems in wide area networks. Previous works apply heuristics that rely on full knowledge of the system to design optimization algorithms. In this work, we explore building a model-free approach, applying multi-agent meta reinforcement learning to solve complex online control problem that generates optimal paths to reroute traffic. Focusing on decentralized solutions, our experiment aims to efficiently minimize the average packet completion time while reducing packet loss across complex network topologies. To evaluate, we test with a static topology and dynamically changing network topologies and compare results to the classical shorted path algorithm.

KEYWORDS

meta learning, multi-agent reinforcement learning, packet routing, dynamic environment, decentralized solution

ACM Reference Format:
Shan Sun and Mariam Kiran (advisor). 2020. Multi-agent meta reinforcement learning for packet routing in dynamic network environments. In *SC '20: ACM/IEEE Supercomputing Conference, November 17–19, 2020, Online.* ACM, 2 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Packet routing is one of the fundamental problems in computer networks, where a router determines the next hop for each packet to its destination. Modern packet routing is very challenging, mainly due to two reasons. First, communication networks are highly dynamic, which makes them hard to model and control. Second, as networks scale, a central controller may be costly to control all devices. It is critical to develop innovative ways where flows can self-manage in real-time and distributed manner. In this work, we address this with a multi-agent meta reinforcement learning algorithm (MAMRL) by using policy optimization and learning a well-generalized policy initialization. This is shown to be quickly adapted to different scenarios with fewer steps.

SC '20, November 17–19, 2020, Online

ACM ISBN 978-1-4503-XXXX-X/18/06...$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 PROBLEM FORMULATION

Network: We consider a possibly time-varying communication network, characterized by an undirected graph $\mathcal{G}_t = (\mathcal{V}, \mathcal{E}_t)$, where $\mathcal{V} = \{1, \cdots, n\}$ is a set of routers and $\mathcal{E}_t \subseteq \mathcal{V} \times \mathcal{V}$ are transmission links between the routers at time t.

Routing: Each packet is routed from router to router via links. Routers process packets using first-in first-out criterion.

Objective: The algorithm aims at finding optimal paths between sources and destinations, minimizing average packet completion time, given by sum of queuing and transmission time, while reducing packet loss.

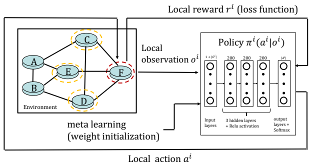

Figure 1: The framework of the packet routing problem.

3 METHODOLOGY

3.1 Policy optimization

We use multi-agent policy optimization, particularly trust region policy optimization, to solve packet routing. As shown in Figure 1, at each timestep t, each router^i observes a local observation o_t^i and takes an action a_t^i, and receives feedback reward estimate r_t^i from the environment. The goal is to learn a parameterized policy $\pi_{\theta^i}^i(a^i|o^i)$, i.e., a mapping from observation to action, with respect to the expected return (long-term cumulative reward). Those are defined below,

Observation o^i: 1) destination router of first packet in local queue; 2) Latest five step actions taken by router$_i$; 3) router with longest queue among all router^i's neighbors[1].

Action a^i: next hop of current packet in the queue.

Reward r^i: a global reward estimate. Let x^i denote the negative average completion time of all packets delivered to router$_i$ plus negative number of packet loss occurred in router^i. Then r^i is an estimate of $\frac{1}{n}\sum_{j \in \mathcal{V}} x^j$. [2].

---

[1] Router^i and router^j are neighbors, if there is a link connecting them. For example, in Figure 1 routers C, D and E are neighbors of F.

[2] The packet routing aims to minimize the average packet completion time of the whole network while reducing packet loss. In order to achieve this goal cooperatively, each router needs to know the global reward function which is the average of all the local reward [4].

3.2 Decentralized solution

We aim to find a decentralized solution, such that each router only needs local information: its own and neighboring data. The observation and action are executed locally (Section 3.2). We adapt the following dynamic consensus algorithm [1] to estimate r^i using only local information. It can be proved that the local estimate r^i can converge to the vicinity of global reward $\frac{1}{n}\sum_{j\in\mathcal{V}} x^j$ within a few timesteps as long as the network is connected.

$$\begin{aligned} r_t^i &= x_t^i - y_t^i, \\ y_{t+1}^i &= \sum_{j\in\mathcal{N}_i}(r_t^i - r_t^j) + y_t^i. \end{aligned} \tag{1}$$

3.3 Multi-agent meta reinforcement learning

To address packet routing in dynamic environments, addition to policy optimization, we leverage meta-learning [2]. This helps learn policy parameters θ^i that are close to all optimal parameters in all environments. Therefore θ^i is the best parameter initialization that can quickly adapt to different environments (see Figure 2).

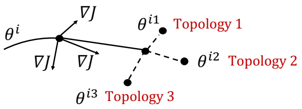

Figure 2: The framework of meta learning.

4 EVALUATION AND RESULTS

Several experiments are conducted in simulators of computer network. Here, the policy model for all the routers is a neural network (see Figure 2).

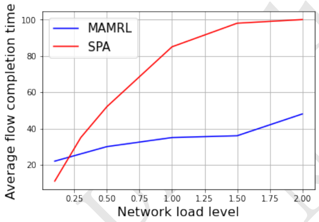

Figure 3: Average packet completion time results in stationary environment.

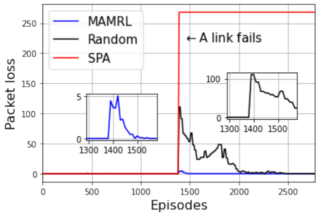

Figure 4: Packet loss results in dynamic environment.

4.1 Stationary environment

We test the MAMRL in packet routing problem with a static topology (see Figure 1). We compare the results to the shorted path algorithm (SPA) at various network load levels. As shown in Figure 3, when the load is vary low, the SPA algorithm performs better than the MAMRL. However, as the load increases, the MAMRL performs much better than the SPA algorithm.

4.2 Dynamic environment

We test the MAMRL in packet routing problem with low network load level and dynamically changing topologies. Let the router encounter different network topologies and return the policy parameter initialization using MAMRL. We compare the results to two baseline controllers: 1) training the policy from randomly initialized weights, and 2) shortest path algorithm (SPA) [3]. At the beginning, routers are in network (Figure 1), then the link between A and B disappears. Since SPA relies on the topology, which changes, the SPA keeps sending packets to the failed link which causes huge packet loss. Reinforcement learning algorithms are model-free controllers, and the policy can be improved by interacting with the environment. In Figures 4, 5, the MAMRL controller adapts to new topologies within **150** episodes, however the non-meta RL controller adapts in about **700** episodes.

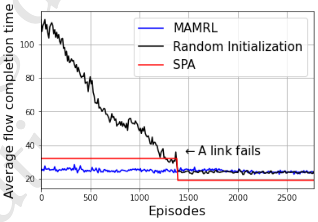

Figure 5: Average packet completion time results in dynamic environment.

5 CONCLUSIONS

In this work, we propose a novel framework MAMRL to improve learning efficiency of deep reinforcement learning in multi-agent packet routing with dynamic network environments. The experiments demonstrate that when the network is static, the MAMRL performs better than SPA at high network load level. Moreover, when there is network topology change, 1) model-free reinforcement learning algorithms can reduce packet loss significantly while offering comparable average packet completion time, and 2) MAMRL can adapt to new topology within fewer episodes.

ACKNOWLEDGMENTS

We would like to express our gratitude to Dr. Bashir Mohammed and Prof. Wei Ren for their useful suggestions. This work was supported by U.S. Department of Energy, Office of Science Early Career Research Program for 'Large-scale Deep Learning for Intelligent Networks' Contract no FP00006145.

REFERENCES

[1] Fei Chen, Yongcan Cao, and Wei Ren. 2012. Distributed average tracking of multiple time-varying reference signals with bounded derivatives. *IEEE Trans. Automat. Control* 57, 12 (2012), 3169–3174.
[2] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv:1703.03400* (2017).
[3] Andrew S Tanenbaum et al. 1996. *Computer networks*. New Jersey.
[4] Kaiqing Zhang, Zhuoran Yang, Han Liu, Tong Zhang, and Tamer Başar. 2018. Fully decentralized multi-agent reinforcement learning with networked agents. *arXiv:1802.08757* (2018).