Resource Allocation Method for Network Slicing Using Constrained Reinforcement Learning

Yongshuai Liu*, Jiaxin Ding†, Xin Liu*
*Computer Science Department, University of California, Davis
†John Hopcroft Center for Computer Science, Shanghai Jiao Tong University
yshliu@ucdavis.edu, jiaxinding@sjtu.edu.cn, xinliu@ucdavis.edu

2021 IFIP Networking Conference (IFIP Networking) | 978-3-9031-7639-3/21/$31.00 ©2021 IFIP | DOI: 10.23919/IFIPNETWORKING52078.2021.9472202

***Abstract*—With the proliferation of mobile networks, we face strong diversification of services, demanding the network to be more flexible. To satisfy this dire need, network slicing is embraced as a promising solution for resource utilization in 5G and future networks. However, this process is complicated that the traditional approaches cannot effectively perform resource orchestration due to the lack of accurate models and the existence of dynamic hidden structures. We formulate the resource allocation problem as a Constrained Markov Decision Process and solve it using constrained reinforcement learning. Specifically, we use the adaptive interior-point policy optimization and projection layers to handle cumulative and instantaneous constraints. Our evaluations show that our method is effective in resource allocation and outperforms baselines.**

***Index Terms*—Resource Allocation, Network Slicing, 5G, Constraints, Deep Reinforcement Learning**

I. Introduction

With the proliferation of mobile networks, we face strong diversification of services. These services demand the network to embed more flexibility. In 5G and future networks, network slicing [1], enabled by network function virtualization (NFV) and software defined networking (SDN), is embraced as a promising solution for flexible resource provisioning.

Network slicing is a generalized resource allocation problem in compliance with the complex network dynamics, in the long run. However, resource allocation is a highly complicated that the traditional approaches cannot solve effectively and efficiently. First, traditional approaches require accurate mathematical models with parameters known, which is often difficult to achieve in practice. Constraints from the physical systems or service demands are prevalent and complex, which further adds to the difficulty. Second, traditional methods do not adapt to epistemic uncertainty, exhibited as hidden structures in networks, due to a lack of knowledge and ability to explore and learn from the studied system.

We propose a resource allocation method for network slicing using constrained reinforcement learning (RL). The learning-based methods are beneficial because they explore and learn from the network without knowing those prior knowledge. We model the problem as a Constrained Markov Decision Process and develop efficient RL algorithms for network slicing under both cumulative and instantaneous constraints. To the best of our knowledge, we are the first one to apply RL for network slicing with constraints. To deal with cumulative constraints, we propose adaptive Interior-point Policy Optimization (IPO) [2]. For instantaneous constraints, we project a resource allocation decision generated by the RL policy to its nearest feasible decision [3].

Y. Liu and X. Liu acknowledge supports from NSF CNS-1718901, IIS-1838207, CNS 1901218, OIA-2040680, and USDA-020-67021-32855. J. Ding acknowledges support from Shanghai Sailing Program 20YF1421300.
Annex to ISBN 978-3-903176-39-3© 2021 IFIP

II. System Description

We describe the network slicing architecture with constrained RL, as shown in Fig. 1a. It is developed from the ETSI reference model [4] colored in blue, and our contribution is highlighted with green. The Virtual Network Functions (VNF) consist of the virtual resources. They provide different service functionalities that make up the Network Slice. The Network Slicing Manager (NSM) is responsible for the initialization, configuration, and managing the life cycles of the network slices. A policy trained with constrained RL algorithms, is in charge of providing solutions for the resource allocation on network slices to the NSM and the Orchestrator, and receive feedback to improve the policy on further decisions.

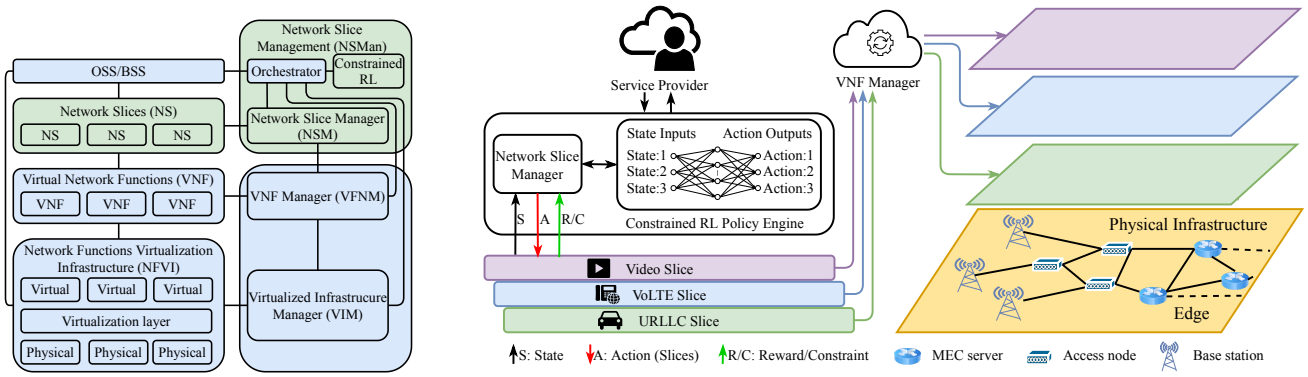
Fig. 1b shows in details. The service providers submit the requirements to the RL engine. The NSM monitors the current system state and sends it to the RL engine. Based on the state and the requirements, the RL engine proposes the resource allocation plan to the NSM. The NSM configure the network slices with the proposed plan. This plan is passed to the VNF manager to further map the virtual resources to physical resources. For each configuration in a decision time slot, the NSM monitors the system and network slices to measure the rewards(constraints) and send the rewards(constraints) to the RL engine for further policy improvement.

III. Problem Formulation

A. Radio Access Network Slicing

To be clear, we formulate how to apply constrained RL for radio access scenario with hidden dynamics, as suggested in [5]. We simulate a scenario containing a Base Station (BS) with three types of services (i.e., Video, VoLTE, URLLC). Each service has a random number of users. The total bandwidth of the BS is fixed and given (100 Mbps).

The network slicing problem is to allocate bandwidth to each type of user (a slice). At the beginning of each decision time slot, the BS decides the bandwidth allocation b_i (i is user



(a) The network slicing architecture with Constrained RL

(b) A use case of network slicing with Constrained RL

Fig. 1: Network slicing with Constrained Reinforcement Learning

type) based on the observation of number of users in each slice. Let t_i be the actual traffic demand for each slice, then the throughput is $\min(b_i, t_i)$. There is a dissatisfaction ratio $1 - \frac{\min(b_i,t_i)}{t_i}$, representing their dissatisfaction with respect to the service received. The latency l_i for each type of user, which does not yield easy mathematical formulation, is decided based on a queue maintained at the BS. Moreover, dynamic hidden structures exist. Each user arrives and departs the network following a Poisson distribution with mean λ_i and μ_i. The arrival rate λ_i adapts based on the satisfaction ratio of the last time slot. Furthermore, users may depart early if the service is unsatisfactory. **We assume these user traffic patterns and mobility are unknown to the slicing algorithms. This is one of the reasons why learning-based approaches, which incorporate exploration, perform better than the traditional methods based only on observed states.**

B. Constrained Markov Decision Process (CMDP)

We formulate the network slicing problem as a CMDP, which is represented with the tuple (S, A, R, C, γ). The network observations constitute the state set S. The resource allocation decisions constitute the action set A. The reward and costs of taking action a under state s is defined as R and C_i, separately. There are m cost functions and each is under a constraint. The discount factor is γ. The actions are constrained by two types of constraints. A cumulative constraint requires that the cumulatively sum of a cost is within a limit, while an instantaneous constraint requires that a cost needs to satisfy a limit in each time slot. Instantaneous constraints can be further divided into explicit and implicit instantaneous constraints. An explicit constraint has a closed-form expression that can be numerically checked. An implicit constraint does not have an accurate closed-form formulation.

Mapping to the radio access scenario, the state $s = (n_{Video}, n_{VoLTE}, n_{URLLC})$ is the number of users observed at the beginning of each time slot. The action $a = (b_{Video}, b_{VoLTE}, b_{URLLC})$ is the bandwidth allocation for each type of users. The reward $R(s, a)$ is the total throughput $\min(b_{Video}, t_{Video}) + \min(b_{VoLTE}, t_{VoLTE}) + \min(b_{URLLC}, t_{URLLC})$. Moreover, each type of users has a cumulative constraint, which is the expectation of cumulative dissatisfaction ratio, where $C_i(s, a) = 1 - \frac{\min(b_i,t_i)}{t_i}$. The explicit instantaneous constraint is the sum of allocated bandwidth, $b_{Video}+b_{VoLTE}+b_{URLLC}$, which must be less or equal to the total bandwidth (100 Mbps). The implicit instantaneous constraint is on the average latency of each type of user, which we cannot get a closed-form solution and needs to be learned.

Constrained RL learns a policy π_θ takes states as input and output actions. Let trajectory $\tau = (s_0, a_0, s_1, a_1...)$ and $\tau \sim \pi_\theta$. The objective is to select a policy π_θ, which maximizes the discounted cumulative reward $J_R^{\pi_\theta} = \mathbb{E}_{\tau\sim\pi_\theta}[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1})]$, while satisfying discounted cumulative constraints $J_{C_i}^{\pi_\theta} = \mathbb{E}_{\tau\sim\pi_\theta}[\sum_{t=0}^{\infty} \gamma^t C_i(s_t, a_t, s_{t+1})]$ and instantaneous constraints. Formally, the problem is

$$\underset{\theta}{\text{maximize}} \quad \max_\theta J_R^{\pi_\theta} \tag{1}$$

$$\text{subject to} \quad J_{C_i}^{\pi_\theta} \le \omega_i, \text{for each i}, \tag{2}$$

$$C_j(s_t, a_t) \le \epsilon_j, \text{for each j and t}, \tag{3}$$

IV. Constrained Reinforcement Learning

A. Cumulative Constraints

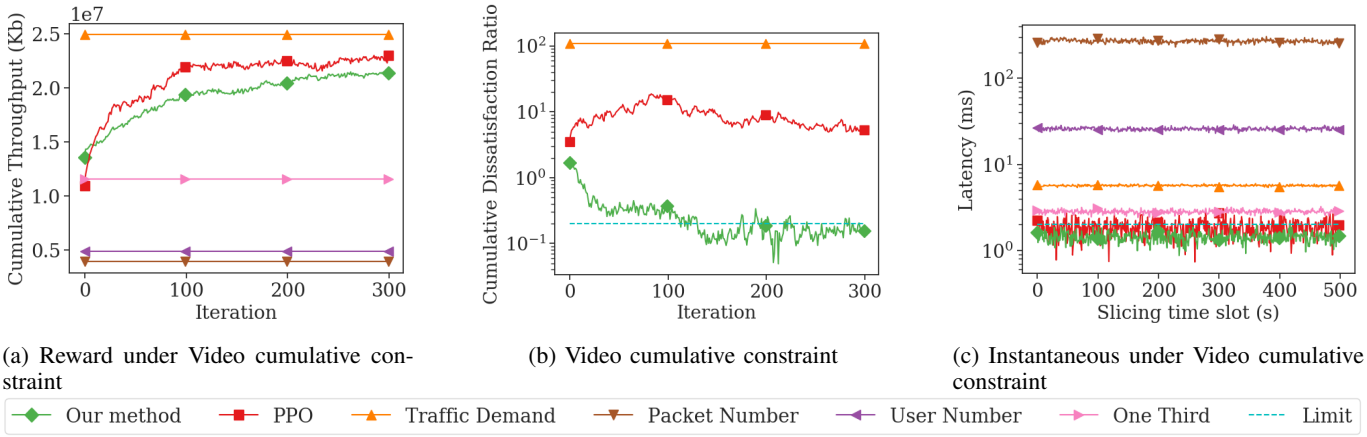
We handle cumulative constraints built on our previous work IPO [2]. IPO augments the objective of PPO $L^{CLIP}(\theta)$ [6] with logarithmic barrier functions $\phi(\widehat{J}_{C_i}^{\pi_\theta}) = \frac{\log(-\widehat{J}_{C_i}^{\pi_\theta})}{t}$, where $\widehat{J}_{C_i}^{\pi_\theta} = J_{C_i}^{\pi_\theta} - \omega_i$. However, IPO is conducted with fixed hyperparameter t for $\phi(\widehat{J}_{C_i}^{\pi_\theta})$, while we propose a method in an adaptive manner. By taking IPO, our objective is

$$\max_\theta \; L^{IPO}(\theta) = L^{CLIP}(\theta) + \sum_{i=1}^{m} \phi(\widehat{J}_{C_i}^{\pi_\theta}). \tag{4}$$

B. Adaptive IPO

We improve IPO in an adaptive manner, to change the hyperparameter t adaptively in the tradeoff of approximation accuracy and algorithm performance [2]. Specifically, we start with a small t to have more stable policy updates, and gradually increase t to achieve better policies on convergence.

Our adaptive IPO has two phases. In Phase I, the cumulative costs are successively optimized to obtain a feasible policy. In Phase II, the policy is initialized with the feasible policy



(a) Reward under Video cumulative constraint

(b) Video cumulative constraint

(c) Instantaneous under Video cumulative constraint

Fig. 2: Average performance under Video cumulative constraint.

achieved in Phase I. Then we start with a moderate small t and adaptively increase it with a factor $\mu > 1$ when policy convergence criteria are satisfied. In each iteration, we update the policy by maximizing $L^{IPO}(\theta)$ in Eq. (4). The pseudo-code is shown in our previous workshop paper [7].

C. Instantaneous Constraints

To satisfy instantaneous constraints, one way is to project the infeasible action to the feasible space [3]. One can introduce another additional layer to π_θ, whose role is to solve

$$\begin{aligned} &\min_{a} \frac{1}{2} \|a - \pi_\theta(s)\|^2 \\ &s.t.\ C_j(s,a) \leq \epsilon_j \end{aligned} \tag{5}$$

In other words, we project the action from the policy $\pi_\theta(s)$ to the ℓ_2 nearest feasible action a that satisfies the instantaneous constraint. The projection idea can apply to both explicit and implicit constraints. One challenge for implicit constraints is that the function $C_j(\cdot,\cdot)$ is unknown. To address the problem, we take advantage of another neural network to learn the value of $C_j(s,a)$ simultaneously, as in [3].

V. Experiments

A. Settings

The above formulation and algorithm can be applied to general network slicing problems. Here, we evaluate on the radio resource slicing scenario, as described in Section III-A and it can be extended to more general cases easily.

We apply traditional methods based on observed states, which result in the baselines of One-third equal allocation, User-number-based allocation, Packet-number-based allocation and Traffic-demand-based allocation, as suggested in [5]. For each method, the total bandwidth is sliced weighted by the number of observed states separately. We also choose the most commonly applied RL algorithm, PPO [6] as a baseline.

B. Evaluation Results

We demonstrate results among three network slices, with one cumulative constraint which is selected from cumulative dissatisfaction ratio of Video, VoLTE and URLLC separately. The results show in Fig. 2, where the figures for VoLTE and URLLC are omitted with the same pattern. Fig. 2a, 2b show the results of long term reward (throughput) and cumulative constraints (dissatisfaction ratio) with respect to the iterations of policy updates. For our method and PPO, the rewards and cumulative cost values are updated during the training process, while the traditional baselines does not adapt to the changes in the environment. Even though PPO can get a little higher reward, its cost significantly violates the constraints.

We collect the policy after training and demonstrate the performance on the implicit instantaneous constraints (latency) , shown in Fig. 2c. Our algorithm satisfies the latency requirements best. Moreover, since explicit instantaneous constraints (bandwidth allocation) can be numerically checked. Our method can always make sure that they are satisfied. All above, the final policy learned by our method outperforms all the baselines in either reward or constraint cost, if not both.

VI. Conclusion

We formulate the network slicing problem as a Constrained Markov Decision Process and solve it with constrained reinforcement learning. Our evaluation results show that our method can solve network slicing problems effectively. Much future work exists, including stronger theoretical bounds, improved sample efficiency and real world evaluations.

References

[1] X. Foukas, G. Patounas, A. Elmokashfi, and M. K. Marina, "Network slicing in 5G: Survey and challenges," *IEEE Communications Magazine*, vol. 55, no. 5, pp. 94–100, 2017.

[2] Y. Liu, J. Ding, and X. Liu, "Ipo: Interior-point policy optimization under constraints," in *the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 4940–4947.

[3] G. Dalal, K. Dvijotham, M. Vecerik, T. Hester, C. Paduraru, and Y. Tassa, "Safe exploration in continuous action spaces," *arXiv preprint arXiv:1801.08757*, 2018.

[4] N. ETSI, "Network functions virtualization (nfv) infrastructure overview," *NFV-INF*, vol. 1, p. V1, 2015.

[5] R. Li, Z. Zhao, Q. Sun, I. Chih-Lin, C. Yang, X. Chen, M. Zhao, and H. Zhang, "Deep reinforcement learning for resource management in network slicing," *IEEE Access*, vol. 6, pp. 74 429–74 441, 2018.

[6] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv:1707.06347*, 2017.

[7] Y. Liu, J. Ding, and X. Liu, "A constrained reinforcement learning based approach for network slicing," in *2020 IEEE 28th International Conference on Network Protocols (ICNP)*. IEEE, 2020, pp. 1–6.