

A Hard and Soft Hybrid Slicing Framework for Service Level Agreement Guarantee via Deep Reinforcement Learning

Heng Zhang[†], Guangjin Pan[†], Shugong Xu[†], *Fellow, IEEE*, Shunqing Zhang[†], and Zhiyuan Jiang[†]

[†] *School of Communication & Information Engineering,
Shanghai University, Shanghai, 200444, China*

Email: {hengzhang, guangjin_pan, shugong, shunqing, jiangzhiyuan}@shu.edu.cn

Abstract—Network slicing is a critical driver for guaranteeing the diverse service level agreements (SLA) in 5G and future networks. Recently, deep reinforcement learning (DRL) has been widely utilized for resource allocation in network slicing. However, existing related works do not consider the performance loss associated with the initial exploration phase of DRL. This paper proposes a new performance-guaranteed slicing strategy with a soft and hard hybrid slicing setting. Mainly, a common slice setting is applied to guarantee slices' SLA when training the neural network. Moreover, the resource of the common slice tends to precisely redistribute to slices with the training of DRL until it converges. Furthermore, experiment results confirm the effectiveness of our proposed slicing framework: the slices' SLA of the training phase can be guaranteed, and the proposed algorithm can achieve the near-optimal performance in terms of the SLA satisfaction ratio, isolation degree and spectrum maximization after convergence.

Index Terms—Network slicing, service level agreements, deep reinforcement learning, resource allocation.

I. INTRODUCTION

With the emergence of 5G telecommunication technology, cellular networks are envisioned to cater services to a wide variety of innovative vertical applications, such as Cellular Vehicle-to-Everything (C-V2X), augmented/virtual reality (AR/VR), with heterogeneous performance requirements including high data rates, ultra-low latency and high reliability [1]. Network slicing is recognized as a promising technique to guarantee differentiated service QoS and service level agreements (SLAs). Since it can enable multiple logical networks corresponding to different network services run on top of a common physical network infrastructure such that the slices can be customized to satisfy various SLAs through virtualization, isolation techniques [2].

From a perspective of radio resource management, the fundamental challenge of network slicing lies in the trade-off of isolation and resource efficiency. On the one hand, to achieve non-interference between slices, the slicing system intends to ensure complete isolation between network slices. On the other hand, inherent radio spectrum scarcity promotes that all slices share a limited radio resource on-demand to ensure efficient utilization. Therefore, inter-slice

radio resource allocation (IS-RRA) in the radio access network (RAN) becomes an open technical challenge [3].

In order to address the above problem, deep reinforcement learning (DRL) technology is widely applied due to its ability in model-free problems [4]–[8]. [4] investigates the application of DRL in solving radio resource slicing and priority-based core network slicing, and the results exhibit the advantage of DRL in solving model-free resource allocation problems. Based on [4], [5] proposed a faster convergence DRL scheme by integrating discrete normalized advantage functions (DNAF) and the deterministic policy gradient descent (DPGD) algorithm. The authors in [6] propose a hierarchical control strategy to guarantee the long-term QoS of services and spectrum efficiency (SE), where DQN and DDPG networks are applied to solve the long-term and short-term problems, respectively. [7] and [8] develop DRL methods to heterogeneous networks (HetNets) scenarios to solve joint user association and network slicing problems.

However, existing works for IS-RRA focus on purely hard isolation schemes where each slice is allocated with dedicated resources. And the performance loss of such schemes caused by action exploration or network fine-tuning may be unbearable. To minimize the performance loss during exploration phases, we propose a hard and soft hybrid slicing framework by introducing a *Common slice* setting under a specific isolation degree constraint, in which UEs of all slices can utilize the resource of the common slice. Especially, the number of resources of the common slice can be significant in the initial training phase to guarantee slices' SLA. As the network training, the resource of the common slice is gradually adjusted until the DRL network converges to an optimal state.

Overall speaking, this paper proposes a hard and soft hybrid slicing framework to guarantee the slices' SLA and maximize the SE as much as possible under a specific isolation constraint. Compared with purely hard algorithms based on DRL, the proposed scheme is capable of guaranteeing slices' SLA all the time, even in the initial training phase. Moreover, it achieves near-optimal performance in terms of SLA satisfaction, SE and isolation.

II. SYSTEM MODEL

A. Communication Model

We consider a typical OFDMA based downlink cellular network consisting of a single base station (BS), where there exist multiple users denoted as $\mathcal{N} = \{1, 2, \dots, N\}$. Assume that the cellular network consists of a set of network slices denoted as $\mathcal{M} = \{1, 2, \dots, M\}$ and \mathcal{N}_m denotes the UEs that belongs to slice m . Radio resource is divided into Transmission Time Intervals (TTIs) denoted by $t \in \{1, 2, \dots\}$ in time domain. The bandwidth is partitioned into W resource blocks (RBs). The duration of a slicing window, where the resource allocated to each slice remains constant, is called *epoch*, denoted by $k \in \{1, 2, \dots\}$, and each epoch contains T consecutive TTIs. Consider a equal power allocation, the SINR of user n at time t is given as $\gamma_{n,t} = \frac{PH_{n,t}}{WN_0}$, where P is the transmit power of BS and $H_{n,t}$ is the channel gain of user n . N_0 is the power of additive white Gaussian noise.

For the traditional traffic with a large packet size, e.g. eMBB traffic, the achievable rate of the user n can be directly estimated according to Shannon's capacity. For the short-sized packet transmission, such as uRLLC and MTC services, the data rate falls in the finite blocklength channel coding regime [9]. Therefore, the data rate for are modeled as (1), where Δt is the time duration of one TTI and $W_{n,t}$ is the allocated RBs to UE n within t -th TTI. ϵ is the transmission error probability, and $Q^{-1}(\cdot)$ is the inverse of the Gaussian Q-function, and $l_{n,t}$ represents the the length of codeword block in symbols, and $C_{n,t}$ is channel dispersion, given by $C_{n,t} = 1 - \frac{1}{(1+\gamma_{n,t})^2}$.

B. SLA Model

Generally speaking, classical QoS metrics for slices' SLA include throughput, packet latency and transmission reliability. For the throughput, it can be easily derived by aggregating the amount of data that is successfully transmitted over time. For the packet delay, a detailed queuing model of UEs' packets needs to be clarified.

In this paper, the arrival distribution of traffic is characterised by the pattern of service, and there is no prior knowledge of volatile demand. The arriving packets of UEs are cached in the BS's buffer and are delivered according to the first-come-first-serve (FCFS) policy. Assume that each UE is corresponding one data queue at BS. The packet delay consists of two parts, i.e., queuing time and transmission time, where the former is influenced by scheduling policy and the latter is decided by instantaneous data rate.

From the perspective of the network, the packet is dropped if its delay exceeds the predefined maximum packet latency [10]. The reliability is determined by the percentage of packets that are successfully delivered. Therefore, the transmission reliability of UE n is expressed as

$$\theta_n = \Pr\{D_{n,i} \leq D_m^{max}\}, n \in \mathcal{N}_m, \quad (2)$$

where $D_{n,i}$ is the delay of the i -th packet of UE n , and D_m^{max} corresponds to the maximum packet delay of UEs in slice m .

For the throughput, the SLA satisfaction ratio of slice m within one epoch k is defined as follows

$$Q_{m,k}^{rate} = \frac{1}{|\mathcal{N}_m|} \sum_{n \in \mathcal{N}_m} \min \left(\frac{\sum_{t=(k-1)T+1}^{kT} r_{n,t}}{R_m^{th}}, 1 \right), \quad (3)$$

where R_m^{th} is the minimum data rate requirement.

For the latency and reliability, given the the maximum packet delay D_m^{max} , the SLA satisfaction ration can be represented by the reliability. Therefore we have

$$Q_{m,k}^{delay} = \frac{1}{|\mathcal{N}_m|} \sum_{n \in \mathcal{N}_m} \theta_n^k \quad (4)$$

where θ_n^k represents the transmission reliability of packets of UE n under maximum delay constraint. Thus, we use the throughput, latency and reliability as the QoS metrics to evaluate the SLA satisfaction in the following.

III. HYBRID SLICING FRAMEWORK AND PROBLEM FORMULATION

A. Hybrid Slicing Framework

The purely hard slicing strategy can guarantee full isolation among slices, while it suffers from the dynamic environment and results in SLA deterioration and low resource efficiency. On the contrary, the soft slicing method can maximize resource efficiency while limited by isolation. Therefore, we propose a novel hybrid slicing framework that can take advantage of both hard and soft strategies. Especially, soft decision, i.e. common slice setting, is utilized to guarantee SLA and improve resource efficiency in the exploration phase. The hybrid slicing framework can be understood from the following two aspects.

1) *Common Slice Setting*: In purely hard schemes, resources dedicated to a slice need to be large enough or over-provisioning to fully guarantee the SLAs, even in the worst-case scenario of the entire slicing window. Fig. 1 shows a hybrid scheme, where the resources are divided into two parts, i.e. resources dedicated to slices and resources to common slice, corresponding to hard and soft strategies. All UEs can utilize the resource of the common slice according to their demand and priority. Reasonable resource configuration of the hybrid scheme enables both SLA satisfaction and resource efficiency with a small sacrifice of isolation. For example, 90% resources required in worst-case scenarios can realize SLA guarantee in most cases and resources of the common slice are shared to guarantee the slices' performance of worst cases such that both SLA satisfaction and resource efficiency can be maximized under a specific isolation constraint.

2) *Periodically Adjusting Resource Slicing*: As Fig. 1 shows, radio resources can be periodically allocated to each slice

$$r_{n,t} = \begin{cases} \Delta t \cdot W_{n,t} \log_2(1 + \gamma_{n,t}), & \text{for long packets transmission} \\ \Delta t \cdot W_{n,t} \left[\log(1 + \gamma_{n,t}) - \sqrt{\frac{C_{n,t}}{l_{n,t}}} Q^{-1}(\epsilon) \log e \right], & \text{for short packets transmission} \end{cases} \quad (1)$$

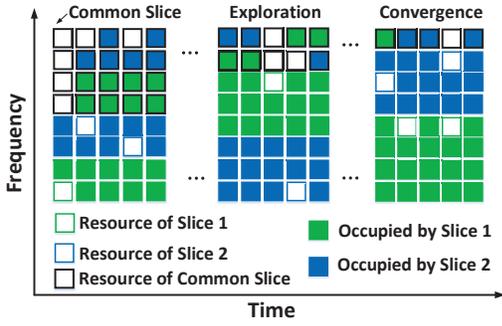


Figure 1: The illustration of the hybrid slicing framework.

to adopt a dynamic wireless environment. For example, the resources of the common slice can be significant in the initial phase to guarantee slices' SLA. As increasing awareness of the environment increases, the slice configuration converges to a precise scheme according to isolation requirement, which corresponds to the last slicing window in Fig. 1.

B. Problem Formulation

For a slice m , the degree of isolation in epoch k is represented by follows

$$o_{m,k} = \frac{w_{m,k}}{w_{m,k} + w_{c,m,k}} \quad (5)$$

where $w_{m,k}$ is the allocated resources of slice m and $w_{c,m,k}$ denotes resources that slice m occupies from the common slice $w_{c,k}$. The objective of the RAN slicing is to guarantee the SLA of diverse slices and simultaneously maximize the SE, which are defined as follows

$$Q_{m,k} = f(d_{m,k}, r_{m,k}, r_{m,c,k}), \quad (6)$$

$$S_k = \sum_{t=(k-1)T+1}^{kT} \sum_{n=1}^N \frac{r_{n,t}}{W} \quad (7)$$

where $d_{m,k}$ is the fluctuation traffic demand of slice m , and function $f(\cdot)$ represents the complicated relationship between the SLA and traffic demand, allocated resources to slices and scheduling algorithms within slices.

The utility function of one epoch is defined as follows

$$U^k = \alpha \sum_{m=1}^M Q_{m,k} + \beta \prod_{m=1}^M \mathbb{1}(Q_{m,k}) \cdot S_k, \quad (8)$$

where α and β are utility coefficients, and $\mathbb{1}(Q_{m,t})$ is the indicator function to denote whether the SLA of slice m is satisfied.

The objective of a slice network is to maximize the long-term utility. A general method to maximize the average utility within a finite time period K , e.g., an hour, a day, or a

week [11]. Hence, the network slice problem is formulated as follows.

$$\mathcal{P} : \max_{w_{m,k}, w_{c,k}} \frac{1}{K} \sum_{k=1}^K (8) \quad (9)$$

$$\text{s. t.} \quad (6), (7)$$

$$o_{m,k} \geq o_m^{th}, \quad (10)$$

$$\sum_{m \in \mathcal{M}} w_{m,k} + w_{c,k} = W, \quad (11)$$

where o_m^{th} represents the threshold of required isolation.

The difficulties of the problem \mathcal{P} is reflected in two aspects. First, the heterogeneous QoS, i.e., throughput, packet delay, reliability, of slices, highly complicates the problem. Second, customized scheduling algorithms within slices and volatile traffic demand make $f(\cdot)$ extremely complex. An analytical model of $f(\cdot)$ in practical networks is almost impossible to derive [8]. Moreover, resource allocation of slicing systems exhibit *Markovian* characteristic, i.e. the allocation strategy affects not only the current SLAs and resource efficiency but also further network state and utility, e.g., the queue of UEs and delay of packets. Therefore, DRL based solution is designed in the following section.

IV. DRL BASED SOLUTION

A. Design of the DRL scheme

As mentioned before, the resource slicing problem can be solved by the DRL technique. In this paper, an initial slice resource allocation, e.g., NVS [12], is first given. Then the DRL agent dynamically adjusts the resource allocated to slices to guarantee the SLA and isolation of slices. To achieve efficient and intelligent slicing, the agent observes the environment, e.g., performance feedback, resource utilization and so on, and makes a decision according to the observed state at the start of each epoch. The states, actions and reward of the DRL scheme is defined as follows.

State: The state is defined as a tuple as follows

$$s_k = \{w_{m,k}, Q_{m,k}, o_{m,k}, \mu_{m,k} | m \in \mathcal{M}\} \quad (12)$$

where $\mu_{m,k}$ is the resource utilization of slice m that is defined as the ration of used resources to the allocated resources.

Action: The agent intelligently adjusts the resource allocation of slices by selecting an action a_k according to the current state s_k . The action for a slice is defined as a set of decreasing, remaining and increasing the allocated resource. It is worth noting that the object of action interaction is the common slice. For example, slice m offloads additional resources to the common slice and slice $m+1$ require more dedicated resources from the common slice at epoch k . And the action set of one slice is defined as $\mathcal{A} = \{-a^j, \dots, -a^1, 0, a^1, \dots, a^j\}$, where $0 < a^1 < \dots < a^j < W$ and j is the positive integer. For example, define the action of slice m is $a_{m,k}$, where $a_{m,k} \in \mathcal{A}$, we have $w_{m,k+1} = w_{m,k} + a_{m,k}$. Therefore, the action of agent at k is defined as follows

$$a_k = \{a_{m,k} | m \in \mathcal{M}, a_{m,k} \in \mathcal{A}\}. \quad (13)$$

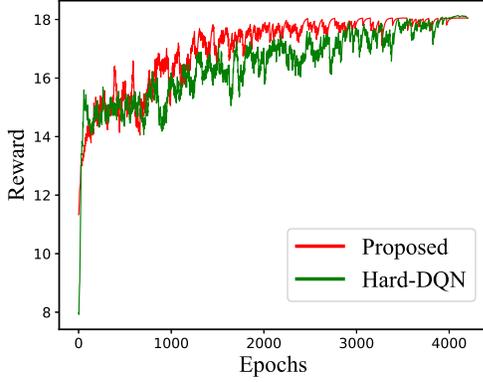


Figure 2: The convergence process of two DRL based algorithms.

Reward: The reward of agent is defined as follows

$$r_k(s_k, a_k) = \alpha_m \sum_{m=1}^M e^{Q_{m,k}} + \beta \prod_{m=1}^M \mathbb{1}(Q_{m,k}) \cdot \frac{S_k}{S_{max}} (14) - \rho \sum_{m=1}^M [o_m^{th} - o_{m,k}]^+,$$

where $[x]^+ = \max(0, x)$, and $\rho > 0$ is a punishment constant. $\frac{S_k}{S_{max}}$ operation normalizes SE by dividing the predefined maximum value S_{max} . The exponential reward function is to train the network more efficiently as $Q_{m,k}$ approaches 1.

B. Training of Agents

A deep Q network (DQN) is applied to design and train the agent, where a neural network (NN) is used to approximate the action-value function, $q(s, a; \theta) \approx Q^*(s, a)$ and θ represents the parameters of NN. The state is input to the DQN, and the network outputs the predicted Q values of each action. With the experience replay and quasistatic target network, the DQN is trained by minimizing the error between the predicted Q values and true Q values as follows,

$$L(\theta) = \frac{1}{B} \sum_k (y_k - q(s_k, a_k; \theta))^2, \quad (15)$$

where B is the batch size. The target value y_k is

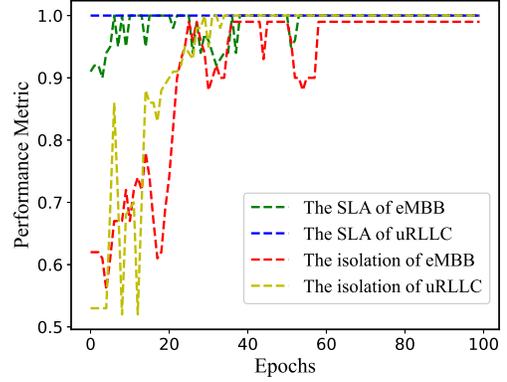
$$y_k = r_{k+1} + \gamma \max_{a'} q(s_{k+1}, a'; \theta'), \quad (16)$$

where θ' represents the parameters of the target network and γ is the discount factor.

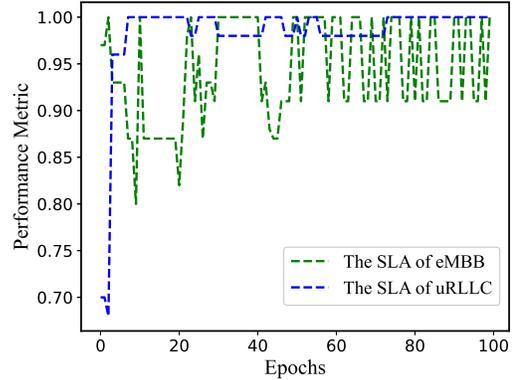
V. NUMERICAL RESULTS

A. Experiments Setup

In a given area of 500×500 m, one BS is located at the center with 43dBm transmission power. 100 RBs with each bandwidth 180kHz are considered as total bandwidth resources. The pathloss model is consistent with [8]. Two slices corresponding to two types of services, i.e. eMBB and uRLLC services, are considered in the simulation. And the



(a) The proposed algorithm



(b) Hard-DQN

Figure 3: The SLA ratio of convergence process for the proposed algorithm and Hard-DQN algorithm.

detailed slice parameters are summarized in Table I. The values of γ , β and ρ are 0.9, 5 and 10, respectively. The network architecture refers to DQN in [13]. The resource allocated to the common slice is 30 RBs in the initial phase, and the action set for one slice is $\{-5, -2, 0, 2, 5\}$. Three baseline algorithms are compared in our experiments:

- **Optimal a Priori (OP):** Given a priori knowledge of traffic and SINR distributions of UEs, the optimal resource slicing is derived by exhaustive search.
- **Hard-DQN [7]:** In this algorithm, a purely hard slicing framework using DQN is utilized.
- **NVS [12]:** NVS considers a static weight-based slicing with the assumption that the channel status of each user in the slice is known in priori.

Table I: Slices Parameters

	eMBB	uRLLC
Traffic Model	Poisson process	period process
Packet Size	55k bits	256 bits
Arrival Rate	100 packets/s	100 packets/s
SLA	95% {5M bps}	99% {5 ms and 99.99% }
α	2	3
o_m^{th}	80%	90%
Number of UEs	20	50
Schedule	Proportional Fairness	Earliest Deadline First

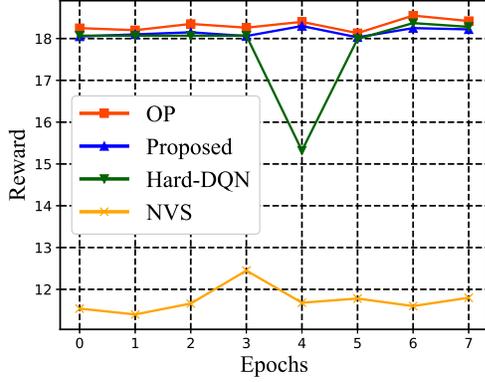


Figure 4: The rewards of four algorithms

B. The Analysis of Convergence Process

As Fig. 2 shows, the rewards of the proposed and the Hard-DQN algorithms are low initially and increase with training until they converge to the same level. It can be observed that the proposed algorithm converges slightly faster than the Hard-DQN algorithm. Since the setting of the common slice increases the SLA satisfaction on the exploration phase if compared with purely hard scheme.

Fig. 3(a) and Fig. 3(b) demonstrate the SLA satisfaction ratio of two algorithms in the first 100 epochs when training the agents. Observing Fig. 3(a), the SLA of uRLLC is always guaranteed. The reasons lie in two aspects. First, the packet size of the uRLLC slice is much smaller than the eMBB slice so that the required resource is lesser than the eMBB slice. Second, the shared resource of common slice prevents extreme scenarios, e.g. most RBs are allocated to eMBB slice. Similarly, the SLA of the eMBB slice is guaranteed after about 50 epochs. Compared with this, the uRLLC slice's SLA of the Hard-DQN algorithm can be guaranteed only after 70 epochs, and the eMBB SLA always fluctuates at the first 100 epochs. Naturally, the isolation degree of two slices of the proposed algorithm cannot approach the required thresholds at the initial phases and the isolation degree of Hard-DQN is always 1. However, it is pointless to discuss isolation when the slices' SLA cannot be guaranteed. Furthermore, the isolation degree of the proposed algorithm can achieve the required thresholds after 60 epochs as shown in 3(a).

C. Performance Comparison

Fig. 4 shows the achievable reward of four algorithms after two DQN-based algorithms converge. First, both the proposed algorithm and Hard-DQN can achieve approximately optimal performance. However, the performance of Hard-DQN fluctuates at 6-th epoch due to a purely hard scheme. Second, the proposed algorithm far outperforms the NVS algorithm. Since NVS considers a static bandwidth provisioning slicing based

VI. CONCLUSION

In this paper, we proposed a hard and soft hybrid slicing framework that introduces the common slice setting. A

on the aggregate throughput, it cannot satisfy the demand of mixed SLAs, e.g. latency and reliability metrics.

DRL-based solution is carefully designed. The comparison experiments indicate the proposed solution can guarantee slices' SLA all the time, even in the initial training phase. Moreover, it achieves near-optimal performance in terms of SLA satisfaction, spectrum efficiency and isolation.

ACKNOWLEDGEMENT

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61871262, 62071284, and 61901251, the National Key R&D Program of China grants 2017YFE0121400 and 2019YFE0196600, the Innovation Program of Shanghai Municipal Science and Technology Commission grant 20JC1416400, Pudong New Area Science & Technology Development Fund, Key-Area Research and Development Program of Guangdong Province grant 2020B0101130012, Foshan Science and Technology Innovation Team Project grant FS0AA-KJ919-4402-0060, and research funds from Shanghai Institute for Advanced Communication and Data Science (SICS).

REFERENCES

- [1] X. Foukas, G. Patounas, A. Elmokashfi, and M. K. Marina, "Network slicing in 5g: Survey and challenges," *IEEE Commun. Magazine*, vol. 55, no. 5, pp. 94–100, 2017.
- [2] S. Zhang, "An overview of network slicing for 5g," *IEEE Wireless Commun.*, vol. 26, no. 3, pp. 111–117, 2019.
- [3] T. Wang and S. Wang, "Inter-slice radio resource allocation: An online convex optimization approach," *IEEE Wireless Commun.*, vol. 28, no. 5, pp. 171–177, 2021.
- [4] R. Li, Z. Zhao, Q. Sun, C.-L. I, C. Yang, X. Chen, M. Zhao, and H. Zhang, "Deep reinforcement learning for resource management in network slicing," *IEEE Access*, vol. 6, pp. 74429–74441, 2018.
- [5] C. Qi, Y. Hua, R. Li, Z. Zhao, and H. Zhang, "Deep reinforcement learning with discrete normalized advantage functions for resource management in network slicing," *IEEE Commun. Lett.*, vol. 23, no. 8, pp. 1337–1341, 2019.
- [6] J. Mei, X. Wang, K. Zheng, G. Boudreau, A. B. Sediq, and H. Abou-Zeid, "Intelligent radio access network slicing for service provisioning in 6g: A hierarchical deep reinforcement learning approach," *IEEE Trans. Commun.*, vol. 69, no. 9, pp. 6063–6078, 2021.
- [7] G. Sun, K. Xiong, G. O. Boateng, D. Ayepah-Mensah, G. Liu, and W. Jiang, "Autonomous resource provisioning and resource customization for mixed traffics in virtualized radio access network," *IEEE Syst. J.*, vol. 13, no. 3, pp. 2454–2465, 2019.
- [8] H. Zhang, S. Xu, S. Zhang, and Z. Jiang, "Slicing framework for service level agreement guarantee in heterogeneous networks -a deep reinforcement learning approach," *Wireless Commun. Letters*, pp. 1–1, 2021.
- [9] Y. Polyanskiy, H. V. Poor, and S. Verdú, "Channel coding rate in the finite blocklength regime," *IEEE Trans. Inform. Theory*, vol. 56, no. 5, pp. 2307–2359, 2010.
- [10] H. Yang, K. Zheng, K. Zhang, J. Mei, and Y. Qian, "Ultra-reliable and low-latency communications for connected vehicles: Challenges and solutions," *IEEE Netw.*, vol. 34, no. 3, pp. 92–100, 2020.
- [11] H. Chergui and C. Verikoukis, "Offline sla-constrained deep learning for 5g networks reliable and dynamic end-to-end slicing," *IEEE J. Select. Areas Commun.*, vol. 38, no. 2, pp. 350–360, 2020.
- [12] R. Kokku, R. Mahindra, H. Zhang, and S. Rangarajan, "NVS: A substrate for virtualizing wireless resources in cellular networks," *IEEE/ACM Trans. Netw.*, vol. 20, no. 5, pp. 1333–1346, 2011.
- [13] Y. Yu, T. Wang, and S. C. Liew, "Deep-reinforcement learning multiple access for heterogeneous wireless networks," *IEEE J. Select. Areas Commun.*, vol. 37, no. 6, pp. 1277–1290, 2019.