Resource Slicing for eMBB and URLLC Services in Radio Access Network Using Hierarchical Deep Learning

Mehdi Setayesh, *Graduate Student Member, IEEE*, Shahab Bahrami, *Member, IEEE*, and Vincent W.S. Wong, *Fellow, IEEE*

***Abstract*—Network slicing is a promising technique for wireless service providers to support enhanced mobile broadband (eMBB) and ultra-reliable low-latency communication (URLLC) services in a shared radio access network (RAN) infrastructure. In this paper, we apply numerology, mini-slot based transmission, and punctured scheduling techniques to support eMBB and URLLC network slices. For efficient allocation of radio resources (e.g., physical resource blocks, transmit power) to the users, we formulate RAN slicing problem as a multi-timescale problem. To solve this problem and address the dynamics of the traffic, we propose a hierarchical deep learning framework. Specifically, in each long time slot, the service provider employs a deep reinforcement learning (DRL) algorithm to determine the slice configuration parameters. The eMBB and URLLC schedulers use their own attention-based deep neural network (DNN) algorithm to allocate radio resources to their corresponding users in each short and mini time slot, respectively. Simulation results show that the proposed framework can achieve a higher aggregate throughput and a higher service level agreement (SLA) satisfaction ratio compared to some other RAN slicing approaches, including the resource proportional placement algorithm, decomposition and relaxation based resource allocation algorithm, and distributed bandwidth optimization algorithm.**

***Index Terms*—Attention mechanism, deep reinforcement learning (DRL), enhanced mobile broadband (eMBB), network slicing, radio access network (RAN), ultra-reliable low-latency communication (URLLC).**

I. Introduction

THE fifth generation (5G) New Radio (NR) wireless systems are envisioned to accommodate a wide range of services with diverse quality of service (QoS) requirements in terms of data rate, latency, reliability, and security [1]. The 5G NR supports three major use cases; namely, (a) enhanced mobile broadband (eMBB) with high transmission data rate for human-type communications, (b) ultra-reliable low-latency communication (URLLC), which targets mission-critical communications with stringent latency requirement, and (c) massive machine-type communication (mMTC) to support a large number of Internet of things (IoT) devices within a geographical area [2]. Network slicing is a promising technique to support eMBB, URLLC, and mMTC use cases over a shared physical infrastructure by partitioning the physical network into multiple virtual and isolated network slices [3].

Recently, the coexistence of eMBB and URLLC traffic in a shared radio access network (RAN) has received considerable attention [4]–[13]. Given the limited radio resources (e.g., physical resource blocks (PRBs), transmit power) in a RAN, an efficient resource allocation among eMBB and URLLC slices is crucial to satisfy the QoS requirements of the users. To facilitate the support of eMBB and URLLC network slices, 5G NR standardized the techniques of numerology [14], mini-slot based transmission [15], and punctured scheduling [16] to be used for service multiplexing in a RAN. The numerology provides multiple frequency domain subcarrier spacings (SCSs) and time domain symbol lengths in the 5G NR time-frequency orthogonal frequency division multiplexing (OFDM) grid [14]. The flexibility in the numerology enables efficient scheduling of eMBB and URLLC users by selecting SCS and OFDM symbol length which satisfy the service requirements. Meanwhile, the mini-slot based transmission in 5G NR enables packet transmission over a short period of time (referred to as mini-slot in [15]) for URLLC users with stringent delay requirements. Thus, different transmission time intervals (TTIs) can be supported by using the numerology and mini-slot based transmission. Moreover, the punctured scheduling enables non-orthogonal slicing of radio resources and facilitates the URLLC traffic to preempt resources which have already been allocated to the eMBB users [16]. Taking into account these three techniques, the RAN slicing becomes a multi-timescale problem.

Both model-based and model-free approaches have been proposed in the literature to address the RAN slicing problem for the eMBB and URLLC services. In the model-based approach, the users' traffic demand and channel gain distributions are known *a priori*. Hence, the RAN slicing problem can be formulated as an optimization problem with the objective of maximizing the system utility subject to the QoS constraints. Bairagi *et al.* [4] considered the network slicing problem in a downlink orthogonal frequency division multiple access (OFDMA) system by maximizing the spectral efficiency, while guaranteeing the required data rate for the eMBB users and

Manuscript received October 19, 2021; revised March 5, 2022; accepted April 18, 2022. This work was supported by Rogers Communications Canada Inc. and the Natural Sciences and Engineering Research Council of Canada (NSERC). The editor coordinating the review of this paper and approving it for publication was Lingjie Duan. (Corresponding author: Vincent W.S. Wong)

The authors are with the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC, V6T 1Z4, Canada (e-mail: {setayeshm, bahramis, vincentw}@ece.ubc.ca).

Color versions of one or more of the figures in this paper are available online at https://ieeexplore.ieee.org.

latency for the URLLC users. Yang *et al.* in [5] proposed an algorithm based on sample average approximation and alternating direction method of multipliers (ADMM) techniques for a two-timescale RAN slicing problem to support multicast eMBB and bursty URLLC services. Anand *et al.* in [6] considered a joint eMBB/URLLC scheduling problem for various eMBB rate loss models while the URLLC traffic is dynamically multiplexed with the eMBB traffic through punctured scheduling. Alsenwi *et al.* in [7] proposed a risk-sensitive punctured scheduling approach, where the radio resources used by the eMBB users can be reallocated to the URLLC users. In our previous work [8], we proposed an algorithm based on penalized successive convex approximation to determine the allocation of radio resources for eMBB and URLLC users.

Obtaining a global optimal solution for a RAN slicing problem using exact analytical approaches can sometimes be mathematically intractable. Therefore, different assumptions such as known traffic distribution for the URLLC users have been used in the model-based approach to simplify the problem formulation. However, these assumptions may degrade the performance of the obtained solution in practical systems. To relax these assumptions, deep reinforcement learning (DRL) [17], which is a model-free approach, has been applied to learn a policy without prior information about the dynamics of URLLC traffic and channel gain variations. Wu *et al.* in [9] proposed a DRL algorithm to solve a RAN slicing problem for vehicular networks. Hua *et al.* [10] applied DRL to design an online RAN slicing algorithm in a single timescale framework, in which the same TTI is considered for the eMBB and URLLC users. Alsenwi *et al.* in [11] proposed an optimization-aided DRL algorithm for radio resource slicing with punctured scheduling for eMBB and URLLC services. Huang *et al.* in [12] applied the punctured scheduling technique and proposed a DRL algorithm to minimize the loss of eMBB users' data rate due to the URLLC packet transmissions. Liu *et al.* in [13] applied DRL augmented with ADMM to allocate radio resources to the network slices.

The aforementioned related works fall into two main threads. The first line of research pertains to the orthogonal slicing approach, where the wireless service provider reserves a portion of bandwidth for the eMBB users, and another portion of bandwidth for the URLLC users. In this approach, which is considered in [5], [8]–[10], [13], service isolation among network slices is provided. However, the allocated resources to URLLC slice may be underutilized due to the URLLC traffic dynamics. The second line of research uses the non-orthogonal slicing approach with punctured scheduling. This approach, which is used in [4], [6], [7], [11], [12], can provide an efficient use of radio resources for URLLC users. However, punctured scheduling may degrade the performance of eMBB slice due to potential reduction of the eMBB users' data rate. Moreover, the proposed RAN slicing schemes in the aforementioned works consider the same numerology for both eMBB and URLLC slices, which remains unchanged over time. Thus, those algorithms do not consider the impact of numerology selection on the system performance.

To address the aforementioned issues, in this paper, we study the radio resource slicing problem for serving eMBB and URLLC users in a downlink OFDMA-based RAN by leveraging the techniques of numerology, mini-slot based transmission, and punctured scheduling. In our RAN slicing problem formulation, a combination of orthogonal and non-orthogonal slicing approaches can be used. To tackle this multi-timescale problem, we propose a hierarchical deep learning framework, which is modular and contains three different algorithms for the service provider, eMBB scheduler, and URLLC scheduler. The main contributions of this paper are as follows:

- *Selection of the Slice Configuration Parameters:* To ensure service isolation among the network slices, we consider the numerology, bandwidth, and transmit power used by each network slice as its configuration parameters. The service provider determines the configuration parameters for the network slices in the long time slots [18]. We model the selection of slice configuration parameters in each long time slot as a partially observable Markov decision process (POMDP) and propose a DRL algorithm to efficiently determine configuration parameters for the slices. We use a long short-term memory (LSTM) layer in the deep neural network (DNN) architecture to capture the temporal correlation. The DRL algorithm guarantees that the inter-slice constraints (e.g., limits on the total network bandwidth and the total available transmit power) are satisfied.
- *Hybrid RAN Slicing Approach:* We use a combination of orthogonal and non-orthogonal slicing approaches. In particular, portions of the bandwidth are reserved exclusively for the eMBB and URLLC users. Another portion of the bandwidth is shared between the eMBB and URLLC users. Punctured scheduling is used in the shared bandwidth part.
- *Resource Allocation in Schedulers:* Given the slice configuration parameters, the eMBB and URLLC schedulers allocate the radio resources (i.e., PRBs, transmit power) to the eMBB and URLLC users in the short and mini time slots, respectively [6]. We formulate the radio resource allocation performed by the schedulers as mixed-integer nonlinear optimization problems. We propose two algorithms based on DNNs with attention mechanism [19] to learn the stochastic policy for obtaining a near-optimal PRB allocation to eMBB and URLLC users. The two proposed algorithms interact with each other to efficiently utilize the PRBs in the shared bandwidth part. The attention-based DNN algorithms guarantee that the intra-slice constraints (e.g., data rate requirement for the eMBB users and latency requirement for the URLLC users) are satisfied. The learning environment of the DRL algorithm depends on the schedulers' functionality. Hence, we propose a hierarchical deep learning framework that takes into account the coordination among the DRL algorithm for the service provider and the attention-based DNN algorithms for the schedulers.
- *Performance Evaluation:* We evaluate the performance of our proposed hierarchical deep learning framework by comparing the average aggregate throughput for the eMBB users, and the average service level agreement (SLA) satisfaction ratio for the eMBB and URLLC slices

with other RAN slicing approaches. Simulation results show that our proposed framework can achieve an average aggregate throughput for the eMBB users, which is on average 9.03%, 29.29%, and 75.21% higher than that of the resource proportional (RP) placement algorithm [6], decomposition and relaxation based resource allocation (DRRA) algorithm [11], and distributed bandwidth optimization based on ADMM (DBO-ADMM) algorithm [5], respectively. Our case study demonstrates that our proposed framework can well adapt to the network dynamics. It can maintain the SLA satisfaction ratio for the eMBB and URLLC slices above 95% and 99%, respectively, as the URLLC traffic load is increased.

This paper is organized as follows. Section II introduces the POMDP model for the selection of slice configuration parameters. The resource allocation problems for downlink packet scheduling of eMBB and URLLC users are presented in Section III. In Section IV, we propose a hierarchical deep learning framework to solve the formulated problem. In Section V, we evaluate the performance of the proposed framework via simulations. Conclusion is given in Section VI.

II. POMDP Model for Slice Configuration Parameters Selection

Consider a downlink OFDMA system shown in Fig. 1(a) with a base station serving multiple eMBB and URLLC users. Let $\mathcal{U}^{\text{eMBB}}$ and $\mathcal{U}^{\text{URLLC}}$ denote, respectively, the set of U^{eMBB} users in the eMBB slice and the set of U^{URLLC} users in the URLLC slice. The wireless service provider serves both the eMBB and URLLC users using a RAN intelligent controller (RIC) that selects the slice configuration parameters (i.e., numerology, transmit power, reserved bandwidth for each slice, shared bandwidth among the slices). The slice configuration parameters are updated by the RIC in each long time slot. Let $\mathcal{T}^{\text{long}} = \{1, 2, \dots\}$ denote the set of indices corresponding to long time slots. Each long time slot has a duration of ΔT^{long} (e.g., 1 sec) [18], [20]. On the other hand, the radio resources (i.e., PRBs, transmit power) are allocated to the eMBB and URLLC users in the order of milliseconds (ms) and microseconds (μs), respectively [6]. Hence, the eMBB scheduler divides each long time slot $t \in \mathcal{T}^{\text{long}}$ into T^{short} short time slots with equal duration ΔT^{short} (e.g., 1 ms). Let $\mathcal{T}^{\text{short}} = \{1, \dots, T^{\text{short}}\}$ denote the set of indices corresponding to short time slots within each long time slot $t \in \mathcal{T}^{\text{long}}$. The URLLC scheduler divides each long time slot $t \in \mathcal{T}^{\text{long}}$ into T^{mini} mini time slots, each with equal duration ΔT^{mini} (e.g., 143 μs[1]). Let $\mathcal{T}^{\text{mini}} = \{1, \dots, T^{\text{mini}}\}$ denote the set of indices corresponding to mini time slots within each long time slot $t \in \mathcal{T}^{\text{long}}$. Fig. 1(b) shows the relation between the long, short, and mini time slots. The key notations used in this work are listed in Table I. Next, we model the selection of slice configuration parameters as a POMDP.

1) Observation: The RIC's observation in the current long time slot $t \in \mathcal{T}^{\text{long}}$ is obtained according to the system performance in the previous time slot $t-1 \in \mathcal{T}^{\text{long}}$. This

[1]Details about the selection of this value for mini time slot duration is provided in Section III-B.

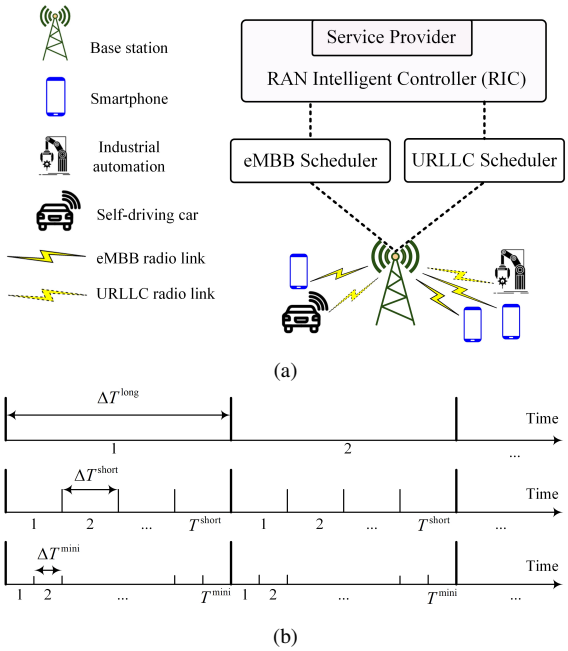


Fig. 1: (a) A downlink OFDMA system with a wireless service provider that uses a RIC to update the slice configuration parameters for the eMBB and URLLC schedulers; (b) The time horizon is divided into long time slots, where each long time slot is divided into T^{short} short time slots and T^{mini} mini time slots.

observation is obtained based on the functionality of the eMBB and URLLC schedulers during the previous long time slot $t-1$. However, the obtained observation may be unreliable for the RIC due to the uncertainty about the users' traffic demand and channel gain. Thus, the RIC has no access to the true system state in the current long time slot t. Instead, it obtains a partial observation of the underlying system state. We use the average aggregate throughput of the eMBB users, the average aggregate traffic demand of the URLLC users, and the average SLA satisfaction ratio (SSR) [21] for the eMBB and URLLC slices in the previous time slot $t-1$ as the RIC's observation in time slot t. Next, we describe the RIC's observation vector in long time slot t.

Let $f_{\tau,t}^{\text{eMBB}}$ denote the aggregate throughput of the eMBB users in short time slot $\tau \in \mathcal{T}^{\text{short}}$ within long time slot t. The average aggregate throughput of the eMBB users $\overline{f}_t^{\text{eMBB}}$ is initialized as zero at $t = 1$, and is obtained at the beginning of each long time slot $t > 1$ as follows:

$$\overline{f}_t^{\text{eMBB}} = \frac{1}{T^{\text{short}}} \sum_{\tau \in \mathcal{T}^{\text{short}}} f_{\tau,\, t-1}^{\text{eMBB}}, \quad t \in \mathcal{T}^{\text{long}} \backslash \{1\}. \tag{1}$$

Let $q_{u,\,m,\,t}$ denote the traffic demand of user $u \in \mathcal{U}^{\text{URLLC}}$ in mini time slot $m \in \mathcal{T}^{\text{mini}}$ within long time slot t, i.e., the amount of data at the base station waiting for transmission to URLLC user u in mini time slot m. The average aggregate traffic demand of URLLC users $\overline{q}_t^{\text{URLLC}}$ is initialized as zero at $t = 1$, and is obtained at the beginning of each long time slot $t > 1$ as follows:

$$\overline{q}_t^{\text{URLLC}} = \frac{1}{T^{\text{mini}}} \sum_{m \in \mathcal{T}^{\text{mini}}} \sum_{u \in \mathcal{U}^{\text{URLLC}}} q_{u,\,m,\,t-1}, \quad t \in \mathcal{T}^{\text{long}} \backslash \{1\}. \tag{2}$$

Table I: Summary of key notations

Notation	Definition
$\mathcal{U}^{\text{eMBB}}$, $\mathcal{U}^{\text{URLLC}}$	Set of eMBB and URLLC users, respectively
$\mathcal{T}^{\text{long}}$, $\mathcal{T}^{\text{short}}$, $\mathcal{T}^{\text{mini}}$	Set of long, short, and mini time slot indices, respectively
ΔT^{long}, ΔT^{short}, ΔT^{mini}	Long, short, and mini time slot duration, respectively
$\boldsymbol{o}_t$	RIC's observation vector in long time slot t
$\boldsymbol{s}_t$	Sequence of actions and observations up to long time slot t
$\boldsymbol{a}(\boldsymbol{s}_t)$	RIC's action vector given sequence $\boldsymbol{s}_t$
$i^{\text{eMBB}}(\boldsymbol{s}_t)$, $i^{\text{URLLC}}(\boldsymbol{s}_t)$	Selected numerologies for the eMBB and URLLC slices
$n^{\text{eMBB}}(\boldsymbol{s}_t)$, $n^{\text{shared}}(\boldsymbol{s}_t)$, $n^{\text{URLLC}}(\boldsymbol{s}_t)$	Number of PRBs with the largest bandwidth allocated to each bandwidth part
$\xi(\boldsymbol{s}_t)$	Portion of power which is allocated to the eMBB slice
$\mathcal{I}$	Set of available choices for the numerologies $i^{\text{eMBB}}(\boldsymbol{s}_t)$, $i^{\text{URLLC}}(\boldsymbol{s}_t)$
$\mathcal{N}$	Set of available choices for tuple $\left(n^{\text{eMBB}}(\boldsymbol{s}_t), n^{\text{shared}}(\boldsymbol{s}_t), n^{\text{URLLC}}(\boldsymbol{s}_t)\right)$
Ξ	Set of possible values for $\xi(\boldsymbol{s}_t)$
$f_{\tau}^{\text{eMBB}}(\boldsymbol{s}_t)$	Aggregate throughput of the eMBB users in short time slot τ within long time slot t
$\vartheta_{\tau}^{\text{eMBB}}(\boldsymbol{s}_t)$	SSR for the eMBB slice in short time slot τ within long time slot t
$\vartheta_{m}^{\text{URLLC}}(\boldsymbol{s}_t)$	SSR for the URLLC slice in mini time slot m within long time slot t
$\mathcal{K}^{\text{eMBB}}(\boldsymbol{s}_t)$, $\mathcal{K}^{\text{URLLC}}(\boldsymbol{s}_t)$	Set of PRB indices for the eMBB and URLLC schedulers given sequence $\boldsymbol{s}_t$, respectively
$\mathcal{J}^{\text{eMBB}}(\boldsymbol{s}_t)$	Set of TTI indices within each short time slot for the eMBB scheduler given sequence $\boldsymbol{s}_t$
$\alpha_{u,k,\tau}(\boldsymbol{s}_t)$	PRB allocation variable for $u \in \mathcal{U}^{\text{eMBB}}$, $k \in \mathcal{K}^{\text{eMBB}}(\boldsymbol{s}_t)$, and $\tau \in \mathcal{T}^{\text{short}}$
$p_{u,k,\tau}(\boldsymbol{s}_t)$	Power allocation variable for $u \in \mathcal{U}^{\text{eMBB}}$, $k \in \mathcal{K}^{\text{eMBB}}(\boldsymbol{s}_t)$, and $\tau \in \mathcal{T}^{\text{short}}$
$\zeta_{k,j,\tau}(\boldsymbol{s}_t)$	Fraction of PRB $k \in \mathcal{K}^{\text{eMBB}}(\boldsymbol{s}_t)$, which is not punctured in TTI j in short time slot τ
$R_u^{\min}$	Minimum data rate requirement for user $u \in \mathcal{U}^{\text{eMBB}}$
$\beta_{u,k,m}(\boldsymbol{s}_t)$	PRB allocation variable for $u \in \mathcal{U}^{\text{URLLC}}$, $k \in \mathcal{K}^{\text{URLLC}}(\boldsymbol{s}_t)$, and $m \in \mathcal{T}^{\text{mini}}$
$\eta_{k,m}(\boldsymbol{s}_t)$	Indication of puncturing for PRB $k \in \mathcal{K}^{\text{URLLC}}(\boldsymbol{s}_t)$ in mini time slot m upon reallocation
$f_{m}^{\text{URLLC}}(\boldsymbol{s}_t)$	System power consumption in mini time slot m for the URLLC slice
$f_{m}^{\text{punc}}(\boldsymbol{s}_t)$	Number of punctured PRBs in the shared bandwidth part in mini time slot m
$\widetilde{\mathcal{K}}^{\text{eMBB}}(\boldsymbol{s}_t)$, $\widetilde{\mathcal{K}}^{\text{URLLC}}(\boldsymbol{s}_t)$	Set of eMBB and URLLC user-PRB pairs, respectively

Let $\vartheta_{\tau,t}^{\text{eMBB}}$ and $\vartheta_{m,t}^{\text{URLLC}}$, respectively, denote the SSR for the eMBB and URLLC slices in short time slot τ and mini time slot m within long time slot t. The SSR for each slice is obtained as the ratio of the number of users, whose QoS requirements are satisfied, to the total number of users in that slice. The average SSRs for the eMBB and URLLC slices are initialized as zero at $t = 1$, and are obtained at the beginning of each long time slot $t > 1$ as follows:

$$\overline{\vartheta}_t^{\text{eMBB}} = \frac{1}{T^{\text{short}}} \sum_{\tau \in \mathcal{T}^{\text{short}}} \vartheta_{\tau,\,t-1}^{\text{eMBB}}, \quad t \in \mathcal{T}^{\text{long}} \backslash \{1\}, \tag{3a}$$

$$\overline{\vartheta}_t^{\text{URLLC}} = \frac{1}{T^{\text{mini}}} \sum_{m \in \mathcal{T}^{\text{mini}}} \vartheta_{m,\,t-1}^{\text{URLLC}}, \quad t \in \mathcal{T}^{\text{long}} \backslash \{1\}. \tag{3b}$$

From (1)–(3), the RIC's observation vector is defined as $\boldsymbol{o}_t = \left(\overline{f}_t^{\text{eMBB}}, \overline{q}_t^{\text{URLLC}}, \overline{\vartheta}_t^{\text{eMBB}}, \overline{\vartheta}_t^{\text{URLLC}}\right)$ in time slot $t \in \mathcal{T}^{\text{long}}$. Since the RIC cannot obtain the system state with complete reliability in the current time slot t only from the observation vector $\boldsymbol{o}_t$, we adopt a POMDP approach. A POMDP model is characterized by the observation space, state space, action space, reward function, state-transition probability function, and observation probability function. By using a history of actions and observations, the RIC can be provided with the sufficient statistic for decision making under uncertainty [17]. Let $\boldsymbol{a}_t$ denote the action taken by the RIC in long time slot t. We denote the sequence of actions and observations up to time slot t by $\boldsymbol{s}_t = (\boldsymbol{o}_1, \boldsymbol{a}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{a}_{t-1}, \boldsymbol{o}_t)$. Let $\mathcal{S}$ denote the set of all sequences of actions and observations.

2) Action: Given the sequence $\boldsymbol{s}_t \in \mathcal{S}$, the action vector of the RIC in long time slot $t \in \mathcal{T}^{\text{long}}$ includes the numerology, transmit power, and the reserved bandwidth for the eMBB and URLLC slices, as well as the shared bandwidth among the slices. By selecting a numerology, the RIC determines the SCS and symbol length of OFDM frames used for data transmission between the base station and users in each slice [22]. The OFDM frame structure in the frequency domain can take values of 15, 30, 60, 120, 240, and 480 kHz, which are the SCSs of different 5G numerologies. For each numerology, a PRB is defined as twelve consecutive SCSs. In the time domain, the duration of each OFDM frame is 10 ms. Each frame has 10 subframes, each with duration of 1 ms. The number of time slots within a subframe is different for each numerology. Hence, each numerology has different OFDM symbol length due to different time slot duration [23]. Table II summarizes the parameters of the 5G NR numerologies. There are two frequency ranges (i.e., FR1 and FR2) for the NR base stations. The numerologies supported by 5G depend on the operating frequency band. Without loss of generality, in this paper, we assume that the base station is operating in the lower frequency range FR1.

Let $\mathcal{I} = \{0, 1, 2\}$ denote the set of choices for the numerologies of each network slice. Given the sequence $\boldsymbol{s}_t \in \mathcal{S}$, let $i^{\text{eMBB}}(\boldsymbol{s}_t)$ and $i^{\text{URLLC}}(\boldsymbol{s}_t) \in \mathcal{I}$, respectively, denote the selected numerologies for the eMBB and URLLC slices. According to the 3rd Generation Partnership Project (3GPP) standard [16], the selected SCS for the eMBB users cannot be greater than the selected SCS for the URLLC users in practical systems. Hence, we have

$$i^{\text{eMBB}}(\boldsymbol{s}_t) \leq i^{\text{URLLC}}(\boldsymbol{s}_t), \quad \boldsymbol{s}_t \in \mathcal{S}, \tag{4a}$$

$$i^{\text{eMBB}}(\boldsymbol{s}_t),\ i^{\text{URLLC}}(\boldsymbol{s}_t) \in \mathcal{I}, \quad \boldsymbol{s}_t \in \mathcal{S}. \tag{4b}$$

Numerology selection enables the corresponding scheduler in each slice to obtain the bandwidth and time duration of the PRBs. Given the sequence $\boldsymbol{s}_t \in \mathcal{S}$, let $b^{\text{eMBB}}(\boldsymbol{s}_t)$ and $\Delta\tau^{\text{eMBB}}(\boldsymbol{s}_t)$ denote the bandwidth and time duration of each PRB in the eMBB slice, respectively. Also, we denote the bandwidth and time duration of each PRB used in the URLLC slice, by $b^{\text{URLLC}}(\boldsymbol{s}_t)$ and $\Delta\tau^{\text{URLLC}}(\boldsymbol{s}_t)$, respectively.

Table II: Parameters of different 5G NR numerologies

Numerology	0	1	2	3	4	5
Subcarrier spacing (SCS)	15 kHz	30 kHz	60 kHz	120 kHz	240 kHz	480 kHz
PRB bandwidth	180 kHz	360 kHz	720 kHz	1.44 MHz	2.88 MHz	5.76 MHz
Number of slots per subframe	1	2	4	8	16	32
Time slot duration	1 ms	0.5 ms	0.25 ms	0.125 ms	0.0625 ms	0.03125 ms
OFDM symbol duration	66.67 μs	33.33 μs	16.67 μs	8.33 μs	4.17 μs	2.08 μs
Frequency range (FR)	FR1	FR1	FR1	FR2	FR2	FR2

By selecting the numerologies $i^{\text{eMBB}}(\boldsymbol{s}_t)$ and $i^{\text{URLLC}}(\boldsymbol{s}_t)$ in time slot $t \in \mathcal{T}^{\text{long}}$, the parameters $b^{\text{eMBB}}(\boldsymbol{s}_t)$ and $\Delta\tau^{\text{eMBB}}(\boldsymbol{s}_t)$ for the eMBB slice, as well as the parameters $b^{\text{URLLC}}(\boldsymbol{s}_t)$ and $\Delta\tau^{\text{URLLC}}(\boldsymbol{s}_t)$ for the URLLC slice can be determined according to Table II. For example, if the service provider selects numerology $i^{\text{eMBB}}(\boldsymbol{s}_t) = 1$ for the eMBB users, then we have $b^{\text{eMBB}}(\boldsymbol{s}_t) = 360$ kHz and $\Delta\tau^{\text{eMBB}}(\boldsymbol{s}_t) = 0.5$ ms.

Using the orthogonal slicing approach, the service provider allocates a portion of the total available bandwidth to each slice. The service provider can also reserve a portion of bandwidth to be shared among the network slices. Since the URLLC traffic has a strict latency requirement, it should be scheduled immediately upon arrival at the base station. Considering the transmission priority of the URLLC traffic in the shared bandwidth part, they can puncture (i.e., override) some of the ongoing eMBB transmissions [24]. That is, a non-orthogonal slicing approach is used in the shared bandwidth part. Since the largest PRB bandwidth is divisible by the PRB bandwidth of all other numerologies, we assume that the granularity of the bandwidth parts is equal to the largest PRB bandwidth among all possible numerologies. Hence, based on the selected numerology, we can obtain the number of PRBs for each bandwidth part. Let $b^{\max}$ denote the largest PRB bandwidth (i.e., 720 kHz for numerology 2 in Table II). Given the sequence $\boldsymbol{s}_t \in \mathcal{S}$, let $n^{\text{eMBB}}(\boldsymbol{s}_t)$, $n^{\text{shared}}(\boldsymbol{s}_t)$, and $n^{\text{URLLC}}(\boldsymbol{s}_t)$ denote the number of PRBs with the largest bandwidth allocated to the eMBB, shared, and URLLC bandwidth parts, respectively. Considering B^{tot} as the total available bandwidth in the base station, for $\boldsymbol{s}_t \in \mathcal{S}$, we have

$$\left(n^{\text{eMBB}}(\boldsymbol{s}_t) + n^{\text{shared}}(\boldsymbol{s}_t) + n^{\text{URLLC}}(\boldsymbol{s}_t)\right) b^{\max} \leq B^{\text{tot}}, \quad \text{(5a)}$$

$$\left(n^{\text{eMBB}}(\boldsymbol{s}_t),\ n^{\text{shared}}(\boldsymbol{s}_t),\ n^{\text{URLLC}}(\boldsymbol{s}_t)\right) \in \mathcal{N}, \quad \text{(5b)}$$

where $\mathcal{N}$ is the set of tuples such that each tuple shows the number of PRBs that can be selected for eMBB, shared, and URLLC bandwidth parts. Fig. 2 shows a time-frequency OFDM grid after the selection of numerologies and bandwidth parts by the RIC for the network slices.

Next, we determine the maximum transmit power which can be used for data transmission to the users in each slice. Let $P^{\max}$ denote the maximum transmit power of the base station. The RIC determines the transmit power $p^{\text{eMBB}}(\boldsymbol{s}_t)$ and $p^{\text{URLLC}}(\boldsymbol{s}_t)$ that are reserved for the eMBB and URLLC slices, respectively. Given the sequence $\boldsymbol{s}_t \in \mathcal{S}$, let $\xi(\boldsymbol{s}_t)$, where $0 \leq \xi(\boldsymbol{s}_t) \leq 1$, denote the portion of power which is allocated to the eMBB slice. Hence, we have

$$p^{\text{eMBB}}(\boldsymbol{s}_t) = \xi(\boldsymbol{s}_t) P^{\max}, \quad \boldsymbol{s}_t \in \mathcal{S}, \quad \text{(6a)}$$

$$p^{\text{URLLC}}(\boldsymbol{s}_t) = (1 - \xi(\boldsymbol{s}_t)) P^{\max}, \quad \boldsymbol{s}_t \in \mathcal{S}. \quad \text{(6b)}$$

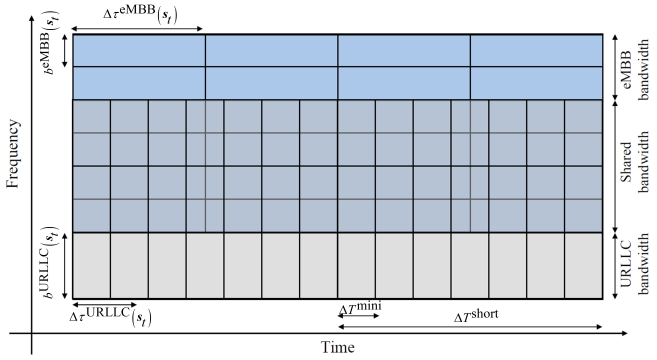


Fig. 2: Time-frequency OFDM grid when numerology 1 is selected for the eMBB slice and numerology 2 is selected for the URLLC slice. A portion of the total available bandwidth has been allocated to each bandwidth part.

We use Ξ to denote the set of possible values for $\xi(\boldsymbol{s}_t)$ selection. Given the sequence $\boldsymbol{s}_t \in \mathcal{S}$, the action vector is defined as $\boldsymbol{a}(\boldsymbol{s}_t) = (i^{\text{eMBB}}(\boldsymbol{s}_t),\ i^{\text{URLLC}}(\boldsymbol{s}_t),\ n^{\text{eMBB}}(\boldsymbol{s}_t),\ n^{\text{shared}}(\boldsymbol{s}_t),\ n^{\text{URLLC}}(\boldsymbol{s}_t),\ \xi(\boldsymbol{s}_t))$. The feasible action space $\mathcal{A}$ is defined by constraints (4a)−(6b), and $\xi(\boldsymbol{s}_t) \in \Xi$.

3) Reward: By performing action $\boldsymbol{a}(\boldsymbol{s}_t) \in \mathcal{A}$ in time slot $t \in \mathcal{T}^{\text{long}}$, based on the eMBB scheduler functionality, the eMBB slice can obtain the aggregate throughput $f_\tau^{\text{eMBB}}(\boldsymbol{s}_t)$, and SSR $\vartheta_\tau^{\text{eMBB}}(\boldsymbol{s}_t)$ in short time slot $\tau \in \mathcal{T}^{\text{short}}$ within long time slot t. Similarly, based on the URLLC scheduler functionality, the URLLC slice can obtain the SSR $\vartheta_m^{\text{URLLC}}(\boldsymbol{s}_t)$ in mini time slot $m \in \mathcal{T}^{\text{mini}}$ within long time slot t. Thus, the service provider obtains the following reward $R(\boldsymbol{s}_t, \boldsymbol{a}(\boldsymbol{s}_t))$ at the end of long time slot $t \in \mathcal{T}^{\text{long}}$:

$$\begin{aligned} R(\boldsymbol{s}_t, \boldsymbol{a}(\boldsymbol{s}_t)) = \\ \frac{\lambda^{\text{sp}}}{T^{\text{short}}} \sum_{\tau \in \mathcal{T}^{\text{short}}} f_\tau^{\text{eMBB}}(\boldsymbol{s}_t) \\ - \lambda^{\text{eMBB}} \left[\upsilon^{\text{eMBB}} - \frac{1}{T^{\text{short}}} \sum_{\tau \in \mathcal{T}^{\text{short}}} \vartheta_\tau^{\text{eMBB}}(\boldsymbol{s}_t) \right]^+ \\ - \lambda^{\text{URLLC}} \left[\upsilon^{\text{URLLC}} - \frac{1}{T^{\text{mini}}} \sum_{m \in \mathcal{T}^{\text{mini}}} \vartheta_m^{\text{URLLC}}(\boldsymbol{s}_t) \right]^+, \end{aligned} \quad \text{(7)}$$

where $[z]^+ = \max\{0, z\}$. υ^{eMBB} and $\upsilon^{\text{URLLC}} \in [0, 1]$ are thresholds for the average SSR of the eMBB and URLLC slices, respectively. The last two terms in (7) are the penalties that may incur due to the violation of the data rate requirement for the eMBB users, or the reliability and latency requirements for the URLLC users. All three terms in the reward function are affected by the traffic demand of the URLLC users. λ^{sp} is a weighting coefficient for the average aggregate throughput of the eMBB users. λ^{eMBB} and λ^{URLLC} are the weighting

coefficients for the penalty terms in the reward function. To guarantee the average SSR threshold for the slices over long time slots, λ^{eMBB} and λ^{URLLC} should take a large value relative to other values in (7).

4) Stationary policy and value function: Given the sequence $\boldsymbol{s}_t = \boldsymbol{s}$ for any $\boldsymbol{s} \in \mathcal{S}$, a policy is defined as a probability distribution $\boldsymbol{\pi}(\boldsymbol{s}) = (\pi(\boldsymbol{a} \,|\, \boldsymbol{s}),\ \boldsymbol{a} \in \mathcal{A})$, where $\pi(\boldsymbol{a} \,|\, \boldsymbol{s})$ denotes the probability of choosing action $\boldsymbol{a}(\boldsymbol{s}_t) = \boldsymbol{a} \in \mathcal{A}$ given the sequence $\boldsymbol{s}_t = \boldsymbol{s} \in \mathcal{S}$. We define $\boldsymbol{\pi} = (\boldsymbol{\pi}(\boldsymbol{s}),\ \boldsymbol{s} \in \mathcal{S})$ as the stationary randomized policy. Given the discount factor $\gamma \in [0, 1]$, the value function $V_{\boldsymbol{\pi}} : \mathcal{S} \to \mathbb{R}$ for sequence $\boldsymbol{s}$ returns the expected discounted reward when starting from $\boldsymbol{s}_t = \boldsymbol{s}$ in time slot t and following policy $\boldsymbol{\pi}$ in the upcoming time slots. We have

$$V_{\boldsymbol{\pi}}(\boldsymbol{s}) = \mathbb{E}_{\boldsymbol{\pi}}\Big\{ \sum_{t'=t}^{\infty} \gamma^{t'-t} R(\boldsymbol{s}_{t'}, \boldsymbol{a}(\boldsymbol{s}_{t'})) \;\Big|\; \boldsymbol{s}_t = \boldsymbol{s} \Big\}, \quad (8)$$

where $\mathbb{E}_{\boldsymbol{\pi}}\{\cdot\}$ is the expectation over choosing feasible actions with policy $\boldsymbol{\pi}$. The service provider aims to obtain policy $\boldsymbol{\pi}^*$ such that the value function is maximized for all sequences $\boldsymbol{s} \in \mathcal{S}$. This is equivalent to solving the following Bellman optimality equations [25]:

$$\mathcal{P}^{\text{sp}}:\ V_{\boldsymbol{\pi}^*}(\boldsymbol{s}) = \underset{\boldsymbol{a} \in \mathcal{A}}{\text{maximize}}\ \ R(\boldsymbol{s}, \boldsymbol{a}) + \gamma \sum_{\boldsymbol{s}' \in \mathcal{S}} \Pr(\boldsymbol{s}' \,|\, \boldsymbol{s}, \boldsymbol{a}) V_{\boldsymbol{\pi}^*}(\boldsymbol{s}'),\ \forall \boldsymbol{s} \in \mathcal{S}, \quad (9)$$

where $\Pr(\boldsymbol{s}' \,|\, \boldsymbol{s}, \boldsymbol{a})$ denotes the probability that the sequence in the next long time slot be $\boldsymbol{s}'$, given the current sequence $\boldsymbol{s}$ and the chosen action $\boldsymbol{a}$. Problem $\mathcal{P}^{\text{sp}}$ is a recursive optimization problem, which is difficult to be solved. Moreover, the transition probabilities are not available to the service provider. To address this issue, in Section IV-A, we develop a DRL algorithm based on the actor-critic method to gradually update the value function and policy without any knowledge of the transition probabilities. The RIC obtains the reward at the end of each time slot $t \in \mathcal{T}^{\text{long}}$, when the required system performance values are provided based on the eMBB and URLLC schedulers' functionality. In the next section, we present the problem formulation for the resource allocation performed by the schedulers.

III. Resource Allocation Problem Formulation

Given the sequence $\boldsymbol{s}_t$, when the RIC selects action $\boldsymbol{a}(\boldsymbol{s}_t)$, the eMBB and URLLC schedulers get informed about the amount of radio resources that can be allocated to their corresponding users. The eMBB and URLLC schedulers provide the RIC with the values of $\frac{1}{T^{\text{short}}} \sum_{\tau \in \mathcal{T}^{\text{short}}} f_{\tau}^{\text{eMBB}}(\boldsymbol{s}_t)$, $\frac{1}{T^{\text{short}}} \sum_{\tau \in \mathcal{T}^{\text{short}}} \vartheta_{\tau}^{\text{eMBB}}(\boldsymbol{s}_t)$, and $\frac{1}{T^{\text{mini}}} \sum_{m \in \mathcal{T}^{\text{mini}}} \vartheta_{m}^{\text{URLLC}}(\boldsymbol{s}_t)$ at the end of each long time slot. Then, the service provider computes the reward $R(\boldsymbol{s}_t, \boldsymbol{a}(\boldsymbol{s}_t))$ in (7). In this section, we present the resource allocation problems for the schedulers.

A. eMBB Scheduler Model

The eMBB scheduler determines the joint PRB and power allocation for the eMBB users in each short time slot $\tau \in \mathcal{T}^{\text{short}}$. We assume that the channel coherence time is larger than the short time slot duration ΔT^{short}. Hence, the channel gain between the base station and the eMBB users can be assumed to be unchanged for each PRB over short time slot τ. Considering the selected numerology, $i^{\text{eMBB}}(\boldsymbol{s}_t)$, and the allocated bandwidth, $B^{\text{eMBB}}(\boldsymbol{s}_t) = \big(n^{\text{eMBB}}(\boldsymbol{s}_t) + n^{\text{shared}}(\boldsymbol{s}_t)\big)\, b^{\max}$, for the eMBB slice, the eMBB scheduler can allocate at most $K^{\text{eMBB}}(\boldsymbol{s}_t) = \left\lfloor \frac{B^{\text{eMBB}}(\boldsymbol{s}_t)}{b^{\text{eMBB}}(\boldsymbol{s}_t)} \right\rfloor$ PRBs to the eMBB users. We denote the set of PRB indices by $\mathcal{K}^{\text{eMBB}}(\boldsymbol{s}_t) = \big\{1, \ldots, K^{\text{eMBB}}(\boldsymbol{s}_t)\big\}$. Since each PRB has a time duration of $\Delta\tau^{\text{eMBB}}(\boldsymbol{s}_t)$, there are at most $J^{\text{eMBB}}(\boldsymbol{s}_t) = \left\lfloor \frac{\Delta T^{\text{short}}}{\Delta\tau^{\text{eMBB}}(\boldsymbol{s}_t)} \right\rfloor$ TTIs within each short time slot. We denote the set of TTI indices by $\mathcal{J}^{\text{eMBB}}(\boldsymbol{s}_t) = \big\{1, \ldots, J^{\text{eMBB}}(\boldsymbol{s}_t)\big\}$.

Given $\boldsymbol{s}_t$, let binary decision variable $\alpha_{u,k,\tau}(\boldsymbol{s}_t)$ denote whether PRB $k \in \mathcal{K}^{\text{eMBB}}(\boldsymbol{s}_t)$ is allocated to user $u \in \mathcal{U}^{\text{eMBB}}$ in time slot $\tau \in \mathcal{T}^{\text{short}}$ (i.e., $\alpha_{u,k,\tau}(\boldsymbol{s}_t) = 1$) or not (i.e., $\alpha_{u,k,\tau}(\boldsymbol{s}_t) = 0$). If $\alpha_{u,k,\tau}(\boldsymbol{s}_t) = 1$, then PRB $k \in \mathcal{K}^{\text{eMBB}}(\boldsymbol{s}_t)$ should be allocated to user $u \in \mathcal{U}^{\text{eMBB}}$ for all the TTI indices $j \in \mathcal{J}^{\text{eMBB}}(\boldsymbol{s}_t)$ within short time slot τ. Each PRB $k \in \mathcal{K}^{\text{eMBB}}(\boldsymbol{s}_t)$ can be allocated to at most one eMBB user. Hence, we have

$$\sum_{u \in \mathcal{U}^{\text{eMBB}}} \alpha_{u,k,\tau}(\boldsymbol{s}_t) \le 1,\ \ k \in \mathcal{K}^{\text{eMBB}}(\boldsymbol{s}_t),\ \tau \in \mathcal{T}^{\text{short}},\ \boldsymbol{s}_t \in \mathcal{S}. \quad (10)$$

Given $\boldsymbol{s}_t$, let $p_{u,k,\tau}(\boldsymbol{s}_t)$ denote the allocated transmission power to user $u \in \mathcal{U}^{\text{eMBB}}$ using PRB $k \in \mathcal{K}^{\text{eMBB}}(\boldsymbol{s}_t)$ in time slot $\tau \in \mathcal{T}^{\text{short}}$. Considering the allocated power $p^{\text{eMBB}}(\boldsymbol{s}_t)$ to the eMBB slice by the RIC in long time slot t, we have

$$\sum_{u \in \mathcal{U}^{\text{eMBB}}} \sum_{k \in \mathcal{K}^{\text{eMBB}}(\boldsymbol{s}_t)} \alpha_{u,k,\tau}(\boldsymbol{s}_t)\, p_{u,k,\tau}(\boldsymbol{s}_t) \le p^{\text{eMBB}}(\boldsymbol{s}_t), \quad \tau \in \mathcal{T}^{\text{short}},\ \boldsymbol{s}_t \in \mathcal{S}. \quad (11)$$

Let $\Gamma_{u,k,\tau}(\boldsymbol{s}_t) = \frac{p_{u,k,\tau}(\boldsymbol{s}_t) |g_{u,k,\tau}(\boldsymbol{s}_t)|^2}{\sigma^2}$ denote the received signal-to-noise ratio (SNR) for user $u \in \mathcal{U}^{\text{eMBB}}$ using PRB $k \in \mathcal{K}^{\text{eMBB}}(\boldsymbol{s}_t)$ in time slot $\tau \in \mathcal{T}^{\text{short}}$, where $g_{u,k,\tau}(\boldsymbol{s}_t) \in \mathbb{C}$ denotes the channel gain between the base station and user u on PRB k, and σ^2 is the variance of the additive white Gaussian noise. Given the sequence $\boldsymbol{s}_t$, we denote the data rate for eMBB user u using PRB k in short time slot τ by $R_{u,k,\tau}(\boldsymbol{s}_t) = b^{\text{eMBB}}(\boldsymbol{s}_t) \log_2\big(1 + \Gamma_{u,k,\tau}(\boldsymbol{s}_t)\big)$.

In the shared bandwidth part, an arriving URLLC data packet cannot be delayed due to its stringent low latency requirement. Hence, a URLLC packet will be transmitted immediately by puncturing the ongoing eMBB transmissions. We use parameter $\zeta_{k,j,\tau}(\boldsymbol{s}_t)$ to denote the fraction of PRB $k \in \mathcal{K}^{\text{eMBB}}(\boldsymbol{s}_t)$, which is not punctured by the URLLC users in TTI $j \in \mathcal{J}^{\text{eMBB}}(\boldsymbol{s}_t)$ in time slot $\tau \in \mathcal{T}^{\text{short}}$, where $0 \le \zeta_{k,j,\tau}(\boldsymbol{s}_t) \le 1$. Parameter $\zeta_{k,j,\tau}(\boldsymbol{s}_t)$ depends on the allocation of the PRBs in the shared bandwidth part. For the PRBs in the eMBB bandwidth part, we have $\zeta_{k,j,\tau}(\boldsymbol{s}_t) = 1$. Fig. 3 shows how the eMBB and URLLC schedulers allocate PRBs to their corresponding users. Considering the punctured PRBs, the data rate for user $u \in \mathcal{U}^{\text{eMBB}}$ in TTI $j \in \mathcal{J}^{\text{eMBB}}(\boldsymbol{s}_t)$ in time slot $\tau \in \mathcal{T}^{\text{short}}$ is equal to $R_{u,j,\tau}(\boldsymbol{s}_t) = \sum_{k \in \mathcal{K}^{\text{eMBB}}(\boldsymbol{s}_t)} \zeta_{k,j,\tau}(\boldsymbol{s}_t) \alpha_{u,k,\tau}(\boldsymbol{s}_t) R_{u,k,\tau}(\boldsymbol{s}_t)$.

Given the sequence $\boldsymbol{s}_t \in \mathcal{S}$, the following constraint guarantees the minimum data rate $R_u^{\min}$ for user $u \in \mathcal{U}^{\text{eMBB}}$

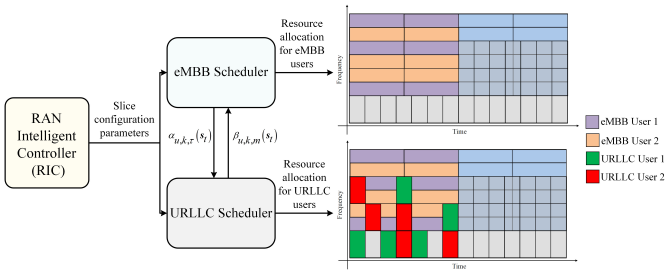


Fig. 3: The eMBB scheduler allocates PRBs in the eMBB and shared bandwidth parts to the eMBB users in each short time slot. The URLLC scheduler allocates PRBs in the URLLC and shared bandwidth parts to the URLLC users in each mini time slot. The schedulers interact with each other to efficiently utilize the PRBs in the shared bandwidth part.

in TTI $j \in \mathcal{J}^{\text{eMBB}}(\boldsymbol{s}_t)$ in time slot $\tau \in \mathcal{T}^{\text{short}}$:

$$R_{u,j,\tau}(\boldsymbol{s}_t) \geq R_u^{\min},\ u \in \mathcal{U}^{\text{eMBB}},\ j \in \mathcal{J}^{\text{eMBB}}(\boldsymbol{s}_t), \\ \tau \in \mathcal{T}^{\text{short}},\ \boldsymbol{s}_t \in \mathcal{S}. \quad (12)$$

To maximize the aggregate throughput of the eMBB users in each short time slot τ, the objective function for the eMBB scheduler is $f_\tau^{\text{eMBB}}(\boldsymbol{s}_t) = \frac{1}{J^{\text{eMBB}}(\boldsymbol{s}_t)} \sum_{j \in \mathcal{J}^{\text{eMBB}}(\boldsymbol{s}_t)} \sum_{u \in \mathcal{U}^{\text{eMBB}}} R_{u,j,\tau}(\boldsymbol{s}_t)$.

Given constraints (10)−(12), the eMBB scheduler can find a feasible PRB and power allocation for all the eMBB users in each short time slot τ if the RIC provides the eMBB slice with sufficient radio resources (i.e., $B^{\text{eMBB}}(\boldsymbol{s}_t)$ and $p^{\text{eMBB}}(\boldsymbol{s}_t)$), as well as a proper choice of numerology $i^{\text{eMBB}}(\boldsymbol{s}_t)$. We consider the following penalized optimization problem:

$$\mathcal{P}_\tau^{\text{eMBB-Pen}}(\boldsymbol{s}_t): \\ \underset{\substack{\alpha_{u,k,\tau}(\boldsymbol{s}_t),\\ p_{u,k,\tau}(\boldsymbol{s}_t), \Delta_{u,j,\tau}(\boldsymbol{s}_t),\\ u \in \mathcal{U}^{\text{eMBB}}, k \in \mathcal{K}^{\text{eMBB}}(\boldsymbol{s}_t), j \in \mathcal{J}^{\text{eMBB}}(\boldsymbol{s}_t)}}{\text{maximize}} \quad f_\tau^{\text{eMBB}}(\boldsymbol{s}_t) - \lambda^{\text{eMBB-Pen}} \sum_{j \in \mathcal{J}^{\text{eMBB}}(\boldsymbol{s}_t)} \sum_{u \in \mathcal{U}^{\text{eMBB}}} \Delta_{u,j,\tau}^2(\boldsymbol{s}_t)$$

subject to constraints (10)−(11),

$$R_{u,j,\tau}(\boldsymbol{s}_t) + \Delta_{u,j,\tau}(\boldsymbol{s}_t) \geq R_u^{\min}, \\ u \in \mathcal{U}^{\text{eMBB}},\ j \in \mathcal{J}^{\text{eMBB}}(\boldsymbol{s}_t), \quad (13)$$

where $\lambda^{\text{eMBB-Pen}} \gg 1$ is the penalizing coefficient and $\Delta_{u,j,\tau}(\boldsymbol{s}_t)$ is a slack variable to penalize the objective function due to the minimum data rate constraint (12) violation for user $u \in \mathcal{U}^{\text{eMBB}}$ in TTI $j \in \mathcal{J}^{\text{eMBB}}(\boldsymbol{s}_t)$ in time slot $\tau \in \mathcal{T}^{\text{short}}$, when the radio resources provided by the RIC are not sufficient for the eMBB slice. At the end of long time slot t, the eMBB scheduler computes $\frac{1}{T^{\text{short}}} \sum_{\tau \in \mathcal{T}^{\text{short}}} f_\tau^{\text{eMBB}}(\boldsymbol{s}_t)$ based on the joint PRB and power allocation obtained by solving problem $\mathcal{P}_\tau^{\text{eMBB-Pen}}(\boldsymbol{s}_t)$. Furthermore, the value of $\frac{1}{T^{\text{short}}} \sum_{\tau \in \mathcal{T}^{\text{short}}} \vartheta_\tau^{\text{eMBB}}(\boldsymbol{s}_t)$ can be determined based on the obtained solution for penalizing slack variables $\Delta_{u,j,\tau}(\boldsymbol{s}_t)$:

$$\vartheta_\tau^{\text{eMBB}}(\boldsymbol{s}_t) = \\ \frac{1}{J^{\text{eMBB}}(\boldsymbol{s}_t) U^{\text{eMBB}}} \sum_{j \in \mathcal{J}^{\text{eMBB}}(\boldsymbol{s}_t)} \sum_{u \in \mathcal{U}^{\text{eMBB}}} \mathbb{1}(\Delta_{u,j,\tau}(\boldsymbol{s}_t) = 0), \\ \tau \in \mathcal{T}^{\text{short}},\ \boldsymbol{s}_t \in \mathcal{S}, \quad (14)$$

where the indicator function $\mathbb{1}(z \in \mathcal{Z})$ is equal to 1 if $z \in \mathcal{Z}$, and is equal to zero otherwise. The eMBB scheduler informs the URLLC scheduler about how the shared bandwidth part will be used by providing the obtained solution for $\alpha_{u,k,\tau}(\boldsymbol{s}_t)$, $u \in \mathcal{U}^{\text{eMBB}}$, $k \in \mathcal{K}^{\text{eMBB}}(\boldsymbol{s}_t)$.

Table III: 5G numerologies and the considered URLLC transmission duration

Numerology	URLLC transmission duration	Blocklength per PRB
0	2 OFDM symbols	24 OFDM symbols
1	4 OFDM symbols	48 OFDM symbols
2	8 OFDM symbols	96 OFDM symbols

B. URLLC Scheduler Model

Given the sequence $\boldsymbol{s}_t$, the solution of problem $\mathcal{P}_\tau^{\text{eMBB-Pen}}(\boldsymbol{s}_t)$ in each short time slot $\tau \in \mathcal{T}^{\text{short}}$ depends on the parameters $\zeta_{k,j,\tau}(\boldsymbol{s}_t)$ obtained based on the PRB allocation for URLLC users in the shared bandwidth part. To meet the latency requirement of URLLC users, the URLLC scheduler uses mini time slots for scheduling. When data arrives at the base station to be transmitted for a URLLC user, the data will first be placed in a buffer corresponding to that URLLC user. The URLLC scheduler aims to guarantee that the waiting time for the URLLC traffic in the buffer does not exceed ΔT^{mini}. To achieve this goal, first, the URLLC scheduler sends *all* the arrived data in the buffer corresponding to each URLLC user at the beginning of each mini time slot. Second, the URLLC TTI is considered to be equal to the mini time slot duration ΔT^{mini}. Based on [26], we can configure the transmission start time and mini-slot length for different numerologies such that URLLC transmissions start at the beginning of each mini time slot with TTI of one mini time slot duration. Table III shows the considered transmission time for each possible numerology. From Tables II and III, we have $\Delta T^{\text{mini}} = 143\ \mu$s, which is sufficient to meet the latency constraint for URLLC traffic. Considering the allocated bandwidth, $B^{\text{URLLC}}(\boldsymbol{s}_t) = \left(n^{\text{URLLC}}(\boldsymbol{s}_t) + n^{\text{shared}}(\boldsymbol{s}_t)\right) b^{\max}$ and the selected numerology, $i^{\text{URLLC}}(\boldsymbol{s}_t)$, the URLLC scheduler can allocate at most $K^{\text{URLLC}}(\boldsymbol{s}_t) = \left\lfloor \frac{B^{\text{URLLC}}(\boldsymbol{s}_t)}{b^{\text{URLLC}}(\boldsymbol{s}_t)} \right\rfloor$ PRBs to the URLLC users. We denote the set of PRB indices by $\mathcal{K}^{\text{URLLC}}(\boldsymbol{s}_t) = \{1, \ldots, K^{\text{URLLC}}(\boldsymbol{s}_t)\}$.

Given the sequence $\boldsymbol{s}_t$, we use binary decision variable $\beta_{u,k,m}(\boldsymbol{s}_t)$ to indicate whether PRB $k \in \mathcal{K}^{\text{URLLC}}(\boldsymbol{s}_t)$ is allocated to user $u \in \mathcal{U}^{\text{URLLC}}$ in time slot $m \in \mathcal{T}^{\text{mini}}$ (i.e., $\beta_{u,k,m}(\boldsymbol{s}_t) = 1$) or not (i.e., $\beta_{u,k,m}(\boldsymbol{s}_t) = 0$). Each PRB k can be allocated to at most one URLLC user. We have

$$\sum_{u \in \mathcal{U}^{\text{URLLC}}} \beta_{u,k,m}(\boldsymbol{s}_t) \leq 1,\ k \in \mathcal{K}^{\text{URLLC}}(\boldsymbol{s}_t), \\ m \in \mathcal{T}^{\text{mini}},\ \boldsymbol{s}_t \in \mathcal{S}. \quad (15)$$

Let $\Gamma_{u,k,m}(\boldsymbol{s}_t) = \frac{p_{u,k,m}(\boldsymbol{s}_t) |g_{u,k,m}(\boldsymbol{s}_t)|^2}{\sigma^2}$ denote the SNR for user $u \in \mathcal{U}^{\text{URLLC}}$ using PRB $k \in \mathcal{K}^{\text{URLLC}}(\boldsymbol{s}_t)$ in mini time slot $m \in \mathcal{T}^{\text{mini}}$, where $p_{u,k,m}(\boldsymbol{s}_t)$ is the allocated transmission power to URLLC user u using PRB k in mini time slot m. Given the sequence $\boldsymbol{s}_t$, we denote the number of transmitted bits for user u according to the PRB allocation in mini time slot m by $R_{u,m}(\boldsymbol{s}_t)$. Due to the finite blocklength coding for the URLLC traffic, the short-packet transmission regime is

used to approximate $R_{u,m}(\boldsymbol{s}_t)$ as follows [27]:

$$R_{u,m}(\boldsymbol{s}_t) \approx N^{\mathrm{B}}(\boldsymbol{s}_t) \sum_{k\in\mathcal{K}^{\text{URLLC}}(\boldsymbol{s}_t)} \beta_{u,k,m}(\boldsymbol{s}_t)\log_2\left(1+\Gamma_{u,k,m}(\boldsymbol{s}_t)\right) - \log_2 e\, Q^{-1}(\epsilon^{\mathrm{B}})\times \sqrt{N^{\mathrm{B}}(\boldsymbol{s}_t)\sum_{k\in\mathcal{K}^{\text{URLLC}}(\boldsymbol{s}_t)}\beta_{u,k,m}(\boldsymbol{s}_t)V_{u,k,m}(\boldsymbol{s}_t)}, \quad (16)$$

where $N^{\mathrm{B}}(\boldsymbol{s}_t)$ is the blocklength and can be obtained according to Table III based on the selected numerology for the URLLC slice. ϵ^{B} is the decoding error probability, $Q^{-1}(\cdot)$ is the inverse of the Gaussian Q-function, and $V_{u,k,m}(\boldsymbol{s}_t) = 1 - \frac{1}{(1+\Gamma_{u,k,m}(\boldsymbol{s}_t))^2}$ is the channel dispersion.

To guarantee $1-\epsilon^{\mathrm{B}}$ reliability for transmission of specific $R_{u,m}(\boldsymbol{s}_t)$ bits per mini time slot, it is required to assign sufficient PRBs with a large SNR to user $u \in \mathcal{U}^{\text{URLLC}}$. We use a binary SNR model for the URLLC users [28]. We consider that PRB $k \in \mathcal{K}^{\text{URLLC}}(\boldsymbol{s}_t)$ is active for user u if $\Gamma_{u,k,m}(\boldsymbol{s}_t)$ is not less than the SNR threshold Γ^{THR}. The allocated power for transmission from the base station to URLLC user u on active PRB k is given by

$$p_{u,k,m}(\boldsymbol{s}_t) = \frac{\Gamma^{\text{THR}}\sigma^2}{|g_{u,k,m}(\boldsymbol{s}_t)|^2},\ u\in\mathcal{U}^{\text{URLLC}},\ k\in\mathcal{K}^{\text{URLLC}}(\boldsymbol{s}_t), m\in\mathcal{T}^{\text{mini}},\ \boldsymbol{s}_t\in\mathcal{S}. \quad (17)$$

Given the allocated power $p^{\text{URLLC}}(\boldsymbol{s}_t)$ to URLLC slice by the RIC in long time slot $t \in \mathcal{T}^{\text{long}}$, we have the following power allocation constraint:

$$\sum_{u\in\mathcal{U}^{\text{URLLC}}}\sum_{k\in\mathcal{K}^{\text{URLLC}}(\boldsymbol{s}_t)} \beta_{u,k,m}(\boldsymbol{s}_t)\, p_{u,k,m}(\boldsymbol{s}_t) \le p^{\text{URLLC}}(\boldsymbol{s}_t), \quad m\in\mathcal{T}^{\text{mini}},\ \boldsymbol{s}_t\in\mathcal{S}. \quad (18)$$

Given the sequence $\boldsymbol{s}_t$, all the data arrived at the base station for user $u \in \mathcal{U}^{\text{URLLC}}$ before mini time slot $m \in \mathcal{T}^{\text{mini}}$, i.e., $q_{u,m}(\boldsymbol{s}_t)$, should be transmitted during the current mini time slot. Hence, the latency constraint for user u will be satisfied. We have

$$R_{u,m}(\boldsymbol{s}_t) \ge q_{u,m}(\boldsymbol{s}_t),\ \ u\in\mathcal{U}^{\text{URLLC}},\ m\in\mathcal{T}^{\text{mini}},\ \boldsymbol{s}_t\in\mathcal{S}. \quad (19)$$

The URLLC scheduler aims to minimize the system power consumption in each mini time slot $m \in \mathcal{T}^{\text{mini}}$. Hence, we have $f_m^{\text{URLLC}}(\boldsymbol{s}_t) = \sum_{u\in\mathcal{U}^{\text{URLLC}}}\sum_{k\in\mathcal{K}^{\text{URLLC}}(\boldsymbol{s}_t)} \beta_{u,k,m}(\boldsymbol{s}_t)\, p_{u,k,m}(\boldsymbol{s}_t)$. In addition, the URLLC scheduler aims to reduce the number of punctured PRBs in the shared bandwidth part, which can be modeled as $f_m^{\text{punc}}(\boldsymbol{s}_t) = \sum_{k\in\mathcal{K}^{\text{URLLC}}(\boldsymbol{s}_t)} \mathbb{1}\Big(\eta_{k,m}(\boldsymbol{s}_t)\sum_{u\in\mathcal{U}^{\text{URLLC}}}\beta_{u,k,m}(\boldsymbol{s}_t) > 0\Big)$, where the parameter $\eta_{k,m}(\boldsymbol{s}_t) \in \{0,1\}$ denotes whether PRB $k \in \mathcal{K}^{\text{URLLC}}(\boldsymbol{s}_t)$ has already been used by the eMBB users in time slot $m \in \mathcal{T}^{\text{mini}}$ (i.e., $\eta_{k,m}(\boldsymbol{s}_t) = 1$) or not (i.e., $\eta_{k,m}(\boldsymbol{s}_t) = 0$). For the PRBs in the URLLC bandwidth part, we have $\eta_{k,m}(\boldsymbol{s}_t) = 0$.

Given constraints (15)−(19), the URLLC scheduler can find a feasible PRB allocation for all the URLLC users in each mini time slot $m \in \mathcal{T}^{\text{mini}}$ if the RIC provides the URLLC slice with sufficient radio resources (i.e., $B^{\text{URLLC}}(\boldsymbol{s}_t)$ and $p^{\text{URLLC}}(\boldsymbol{s}_t)$) and a proper numerology $i^{\text{URLLC}}(\boldsymbol{s}_t)$. We consider the following penalized optimization problem:

$$\mathcal{P}_m^{\text{URLLC-Pen}}(\boldsymbol{s}_t):$$
$$\underset{\substack{\beta_{u,k,m}(\boldsymbol{s}_t),\,\Delta_{u,m}(\boldsymbol{s}_t),\\ u\in\mathcal{U}^{\text{URLLC}},\,k\in\mathcal{K}^{\text{URLLC}}(\boldsymbol{s}_t)}}{\text{minimize}} \quad f_m^{\text{URLLC}}(\boldsymbol{s}_t) + \lambda^{\text{punc}} f_m^{\text{punc}}(\boldsymbol{s}_t) + \lambda^{\text{URLLC-Pen}}\sum_{u\in\mathcal{U}^{\text{URLLC}}}\mathbb{1}(\Delta_{u,m}(\boldsymbol{s}_t)\neq 0)$$
subject to constraints (15)−(18),
$$R_{u,m}(\boldsymbol{s}_t) + \Delta_{u,m}(\boldsymbol{s}_t) \ge q_{u,m}(\boldsymbol{s}_t),\ u\in\mathcal{U}^{\text{URLLC}}, \quad (20)$$

where $\lambda^{\text{punc}} > 0$ is a weighting coefficient to obtain a tradeoff between minimizing the system power consumption and minimizing the number of punctured PRBs, $\lambda^{\text{URLLC-Pen}} \gg 1$ is the penalty weighting coefficient, and $\Delta_{u,m}(\boldsymbol{s}_t)$ is a slack variable for penalizing the objective function due to the violation from constraint (19) for URLLC user u in mini time slot m. Based on the obtained solution for the slack variables $\Delta_{u,m}(\boldsymbol{s}_t)$, the URLLC scheduler computes $\frac{1}{T^{\text{mini}}}\sum_{m\in\mathcal{T}^{\text{mini}}}\vartheta_m^{\text{URLLC}}(\boldsymbol{s}_t)$ at the end of each long time slot $t \in \mathcal{T}^{\text{long}}$. We have

$$\vartheta_m^{\text{URLLC}}(\boldsymbol{s}_t) = \frac{1}{U^{\text{URLLC}}}\sum_{u\in\mathcal{U}^{\text{URLLC}}}\mathbb{1}(\Delta_{u,m}(\boldsymbol{s}_t)=0), \quad m\in\mathcal{T}^{\text{mini}},\ \boldsymbol{s}_t\in\mathcal{S}. \quad (21)$$

Problems $\mathcal{P}_\tau^{\text{eMBB-Pen}}(\boldsymbol{s}_t)$ and $\mathcal{P}_m^{\text{URLLC-Pen}}(\boldsymbol{s}_t)$ are mixed-integer nonlinear optimization problems, which are NP-hard and difficult to solve. In the next section, we propose a hierarchical framework to solve these two resource allocation problems as well as the RIC optimization problem $\mathcal{P}^{\text{sp}}$.

IV. Algorithm Design

In this section, we develop a hierarchical deep learning framework to support eMBB and URLLC services in a 5G RAN. First, we propose a DRL algorithm that enables the RIC to determine the slice configuration parameters at the beginning of each long time slot. Next, we propose two attention-based DNN algorithms for the eMBB and URLLC schedulers to allocate radio resources to their users at the beginning of each short and mini time slot, respectively.

A. Slice Configuration Parameters Selection Algorithm for the RIC

We develop a DRL-based algorithm for the selection of slice configuration parameters using the actor-critic method [25, Ch. 13]. It enables the RIC to determine the optimal policy and value function. Let $\boldsymbol{\theta}_t$ and $\boldsymbol{\psi}_t$ denote the neural network parameters for the policy and value function in long time slot t, respectively. As shown in Fig. 4, an LSTM layer is used in the DNN structure of the actor and critic networks. By using an LSTM layer, the history sequence of actions and observations, which is required to tackle POMDP problems can be formed implicitly for the actor and critic DNNs due to the internal memory mechanism of the LSTM layer [29], [30]. Thus, the DNN for the policy (i.e., the actor) takes only the previous action $\boldsymbol{a}(\boldsymbol{s}_{t-1})$ and the current observation vector $\boldsymbol{o}_t$ to return a probability distribution over the action space

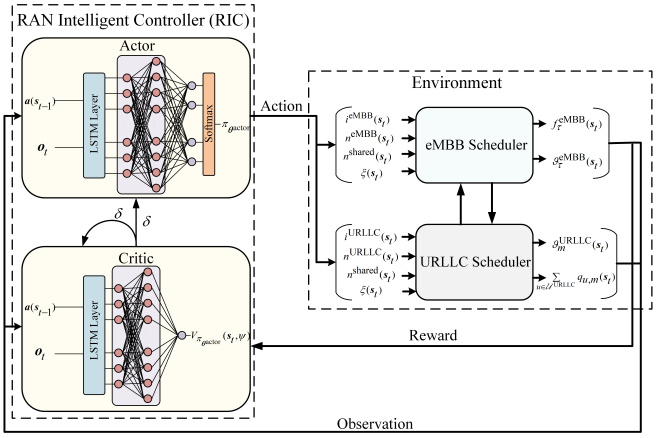


Fig. 4: Illustration of the DNNs structure of the actor-critic networks. The RIC interacts with the eMBB and URLLC schedulers as its environment.

Algorithm 1: Slice Configuration Parameters Selection Algorithm

1: Initialize the actor learning rate ν^{actor}, the critic learning rate ν^{critic}, $t := 1$, and $\epsilon := 10^{-6}$.
2: Initialize randomly the learnable parameters $\boldsymbol{\theta}_1^{\text{actor}}$ and $\boldsymbol{\psi}_1^{\text{critic}}$ for the actor and critic networks, respectively.
3: Initialize the observation $\boldsymbol{o}_1$.
4: **Repeat**
5: Obtain the slice configuration parameters $\boldsymbol{a}(\boldsymbol{s}_t)$ by sampling from the probability distribution $\pi_{\boldsymbol{\theta}_t^{\text{actor}}}(\cdot \mid \boldsymbol{s}_t)$.
6: Update the slice configuration parameters for the eMBB and URLLC network slices.
7: Obtain $\frac{1}{T^{\text{short}}} \sum_{\tau \in \mathcal{T}^{\text{short}}} f_\tau^{\text{eMBB}}(\boldsymbol{s}_t)$ and $\frac{1}{T^{\text{short}}} \sum_{\tau \in \mathcal{T}^{\text{short}}} \vartheta_\tau^{\text{eMBB}}(\boldsymbol{s}_t)$ from the eMBB slice, and $\frac{1}{T^{\text{mini}}} \sum_{m \in \mathcal{T}^{\text{mini}}} \sum_{u \in \mathcal{U}^{\text{URLLC}}} q_{u,m}(\boldsymbol{s}_t)$ and $\frac{1}{T^{\text{mini}}} \sum_{m \in \mathcal{T}^{\text{mini}}} \vartheta_m^{\text{URLLC}}(\boldsymbol{s}_t)$ from the URLLC slice.
8: Receive reward $R(\boldsymbol{s}_t, \boldsymbol{a}(\boldsymbol{s}_t))$.
9: Obtain the new observation vector $\boldsymbol{o}_{t+1}$.
10: Obtain the TD error $\delta(\boldsymbol{\psi}_t^{\text{critic}})$ corresponding to long time slot t.
11: Update the learnable parameters $\boldsymbol{\theta}_{t+1}^{\text{actor}}$ and $\boldsymbol{\psi}_{t+1}^{\text{critic}}$ according to (22) and (23), respectively.
12: $t := t + 1$.
13: **Until** $||\boldsymbol{\theta}_{t-1}^{\text{actor}} - \boldsymbol{\theta}_{t-2}^{\text{actor}}|| < \epsilon$ and $||\boldsymbol{\psi}_{t-1}^{\text{critic}} - \boldsymbol{\psi}_{t-2}^{\text{critic}}|| < \epsilon$.
14: Outputs are the learned parameters for the actor network.

$\mathcal{A}$. Also, the DNN for the value function (i.e., the critic) takes $\{\boldsymbol{a}(\boldsymbol{s}_{t-1}), \boldsymbol{o}_t\}$ to return the value function for the sequence $\boldsymbol{s}_t$. In addition, Fig. 4 illustrates how the actor and critic networks interact with each other and with the environment, i.e., the eMBB and URLLC schedulers.

Algorithm 1 describes our proposed algorithm to obtain the slice configuration parameters by the RIC. Line 1 describes the initialization for ν^{actor} and ν^{critic} as the learning rates for the actor and critic networks, respectively. In Line 2, the neural network parameters $\boldsymbol{\theta}_t^{\text{actor}}$ and $\boldsymbol{\psi}_t^{\text{critic}}$ are initialized. In Line 3, we set the initial observation vector $\boldsymbol{o}_t$ in long time slot $t = 1$. The loop within Lines 4 to 13 involves the RIC's learning process. At the beginning of each long time slot t, the RIC selects an action based on the output of the actor network in Line 5. Then, it updates the configuration parameters of the eMBB and URLLC network slices by providing the corresponding schedulers with the selected action $\boldsymbol{a}(\boldsymbol{s}_t)$ in Line 6. Using the pre-trained DNNs for the eMBB scheduler, $f_\tau^{\text{eMBB}}(\boldsymbol{s}_t)$ and $\vartheta_\tau^{\text{eMBB}}(\boldsymbol{s}_t)$ are determined in each short time slot $\tau \in \mathcal{T}^{\text{short}}$ within long time slot t. Using the pre-trained DNNs for the URLLC scheduler, $\vartheta_m^{\text{URLLC}}(\boldsymbol{s}_t)$, and $\sum_{u \in \mathcal{U}^{\text{URLLC}}} q_{u,m}(\boldsymbol{s}_t)$ are computed in each mini time slot $m \in \mathcal{T}^{\text{mini}}$ within long time slot t. In Line 7, the average values for the aggregate throughput of the eMBB users and SSR for the eMBB slice are computed. The aggregate traffic demand of the URLLC users and average SSR for the URLLC slice are obtained. The RIC receives the reward $R(\boldsymbol{s}_t, \boldsymbol{a}(\boldsymbol{s}_t))$ and obtains the next observation $\boldsymbol{o}_{t+1}$ at the end of long time slot t, in Lines 8 and 9, respectively. In Line 10, the RIC computes the temporal difference (TD) error $\delta(\boldsymbol{\psi}_t^{\text{critic}}) = R(\boldsymbol{s}_t, \boldsymbol{a}(\boldsymbol{s}_t)) + \gamma V_{\boldsymbol{\pi}_{\boldsymbol{\theta}_t^{\text{actor}}}}(\boldsymbol{s}_{t+1}, \boldsymbol{\psi}_t^{\text{critic}}) - V_{\boldsymbol{\pi}_{\boldsymbol{\theta}_t^{\text{actor}}}}(\boldsymbol{s}_t, \boldsymbol{\psi}_t^{\text{critic}})$. In Line 11, using the stochastic gradient descent (SGD) approach, the updated neural network parameters in the actor DNN are obtained as follows:

$$\boldsymbol{\theta}_{t+1}^{\text{actor}} = \boldsymbol{\theta}_t^{\text{actor}} + \nu^{\text{actor}}\, \delta(\boldsymbol{\psi}_t^{\text{critic}}) \times \nabla_{\boldsymbol{\theta}^{\text{actor}}} \ln\left(\pi_{\boldsymbol{\theta}^{\text{actor}}}(\boldsymbol{a}(\boldsymbol{s}_t) \mid \boldsymbol{s}_t)\right)\Big|_{\boldsymbol{\theta}^{\text{actor}} = \boldsymbol{\theta}_t^{\text{actor}}}, \tag{22}$$

and the neural network parameters in the critic DNN are updated as follows:

$$\boldsymbol{\psi}_{t+1}^{\text{critic}} = \boldsymbol{\psi}_t^{\text{critic}} - \nu^{\text{critic}} \delta(\boldsymbol{\psi}_t^{\text{critic}}) \nabla_{\boldsymbol{\psi}^{\text{critic}}} \delta(\boldsymbol{\psi}^{\text{critic}})\Big|_{\boldsymbol{\psi}^{\text{critic}} = \boldsymbol{\psi}_t^{\text{critic}}}. \tag{23}$$

The gradients in (22) and (23) are computed using the back-propagation algorithm [31, Ch. 6]. The next long time slot begins in Line 12. In Line 13, the stopping criteria are given. After reaching the stopping criteria, the RIC employs the output of the trained actor DNN to obtain a near-optimal action for the selection of the network slice configuration parameters.

B. Resource Allocation Algorithm for the eMBB and URLLC Schedulers

The PRB allocation in problems $\mathcal{P}_\tau^{\text{eMBB-Pen}}(\boldsymbol{s}_t)$ and $\mathcal{P}_m^{\text{URLLC-Pen}}(\boldsymbol{s}_t)$ for the eMBB and URLLC users in their corresponding network slices are equivalent to solving combinatorial optimization problems, where a subset of user-PRB pairs should be selected as the optimal solution to problems $\mathcal{P}_\tau^{\text{eMBB-Pen}}(\boldsymbol{s}_t)$ and $\mathcal{P}_m^{\text{URLLC-Pen}}(\boldsymbol{s}_t)$, respectively. Recently, DNN with attention mechanism [19] has shown as a promise technique to obtain a near-optimal solution for combinatorial optimization problems. Inspired by the work in [32], we develop an encoder-decoder DNN architecture shown in Fig. 5 to obtain a near-optimal solution for PRB allocation to the eMBB and URLLC users in their corresponding network slices. As an input, the encoder receives a sequence of feature vectors for all possible user-PRB pairs. Let $\widetilde{\mathcal{K}}$ denote the set of user-PRB pairs. We denote the feature vector for user-PRB pair $\tilde{k} \in \widetilde{\mathcal{K}}$ by $\boldsymbol{\phi}_{\tilde{k}}$. Let $\Phi = \left\{\boldsymbol{\phi}_{\tilde{k}} \mid \tilde{k} \in \widetilde{\mathcal{K}}\right\}$ denote the input sequence of the encoder. As shown in Fig. 5(a), the input sequence passes through linear projection layers, an attention layer, and feedforward layers to generate the encoder embedding sequence $\boldsymbol{h}^{\text{enc}} = \{\boldsymbol{h}_{\tilde{k}}^{\text{enc}}, \tilde{k} \in \widetilde{\mathcal{K}}\}$, which is a high-dimensional representation of the input sequence Φ.

The decoder is invoked D times to produce output sequence $\Omega = \{\omega_1, \ldots, \omega_D\} \subseteq \widetilde{\mathcal{K}}$ as the selected user-PRB pairs. Let $\mathcal{D}$

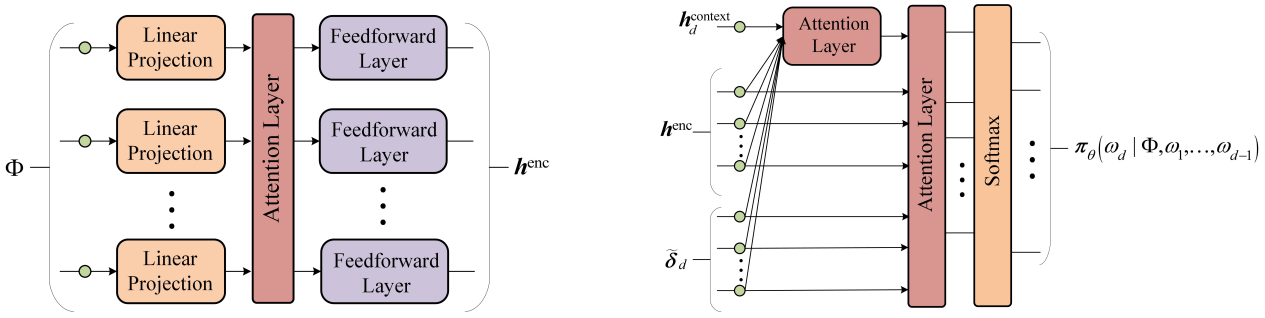


Fig. 5: Illustration of the DNN structure for (a) the encoder module and (b) the decoder module.

denote the set of decoding timesteps. The decoder at decoding timestep $d \in \mathcal{D}$ is responsible for selecting a user-PRB pair ω_d from set $\widetilde{\mathcal{K}}$. Fig. 5(b) shows the architecture of the decoder. The input of the decoder at decoding timestep d consists of three parts. The first part is the context embedding $\boldsymbol{h}_d^{\text{context}}$, which depends on the selected user-PRB pairs $\omega_1, \ldots, \omega_{d-1}$ in decoding timesteps $1, \ldots, d-1$. We will provide details about the elements of the context embedding $\boldsymbol{h}_d^{\text{context}}$ for the URLLC and eMBB schedulers in Sections IV-B1 and IV-B2, respectively. The second part is the encoder embedding sequence $\boldsymbol{h}^{\text{enc}}$. The third part is sequence $\tilde{\boldsymbol{\delta}}_d = \{\tilde{\delta}_{\tilde{k},d},\ \tilde{k} \in \widetilde{\mathcal{K}}\}$, where $\tilde{\delta}_{\tilde{k},d}$ denotes the remaining traffic demand of the user specified by the user-PRB pair $\tilde{k}$ at decoding timestep d. As shown in Fig. 5(b), the input of the decoder passes through attention layers and a softmax layer to generate a conditional probability distribution $\boldsymbol{\pi_\theta}(\omega_d \,|\, \Phi,\, \omega_1, \ldots, \omega_{d-1}) = \Big(\pi_{\boldsymbol{\theta}}(\omega_d = \tilde{k} \,|\, \Phi,\, \omega_1, \ldots, \omega_{d-1}),\ \tilde{k} \in \widetilde{\mathcal{K}}\Big)$, where $\boldsymbol{\theta}$ denotes the neural network parameters for the underlying encoder-decoder DNN architecture.

1) Algorithm Design for the URLLC Scheduler: The URLLC scheduler aims to solve problem $\mathcal{P}_m^{\text{URLLC-Pen}}(\boldsymbol{s}_t)$. We develop a resource allocation algorithm using an encoder-decoder DNN with attention mechanism. Given the sequence $\boldsymbol{s}_t$, the set of user-PRB pairs is obtained as $\widetilde{\mathcal{K}}^{\text{URLLC}}(\boldsymbol{s}_t) = \mathcal{U}^{\text{URLLC}} \times \mathcal{K}^{\text{URLLC}}(\boldsymbol{s}_t)$. For the encoder, $\boldsymbol{\phi}_{\tilde{k}} = \big(|g_{u,k,m}(\boldsymbol{s}_t)|^2,\ q_{u,m}(\boldsymbol{s}_t),\ \eta_{k,m}(\boldsymbol{s}_t)\big)$ is considered as the feature vector of user-PRB pair $\tilde{k} \in \widetilde{\mathcal{K}}^{\text{URLLC}}(\boldsymbol{s}_t)$ in mini time slot $m \in \mathcal{T}^{\text{mini}}$.

For the decoder, the maximum number of decoding timesteps is set to be the number of available PRBs for the URLLC slice, i.e., $D = |\mathcal{K}^{\text{URLLC}}(\boldsymbol{s}_t)|$. At decoding timestep $d \in \mathcal{D}$, the context embedding is defined as $\boldsymbol{h}_d^{\text{context}} = \Big[\boldsymbol{h}^{\text{mean}},\ \boldsymbol{h}^{\text{enc}}_{\omega_{d-1}},\ \boldsymbol{C}_d\Big]$, where $\boldsymbol{h}^{\text{mean}}$ is the mean of the encoder embedding sequence $\boldsymbol{h}^{\text{enc}}$, $\boldsymbol{h}^{\text{enc}}_{\omega_{d-1}}$ is the encoder embedding of the selected user-PRB pair at the previous decoding timestep, and $\boldsymbol{C}_d$ denotes the available resources at decoding timestep d. At decoding timestep $d = 1$, we set $\boldsymbol{h}^{\text{enc}}_{\omega_{d-1}} = 0$ and initialize $\boldsymbol{C}_d = \big(p^{\text{URLLC}}(\boldsymbol{s}_t),\ K^{\text{URLLC}}(\boldsymbol{s}_t)\big)$, where the first component shows the available transmission power and the second component shows the available number of PRBs for the URLLC users. Given the output of the previous decoding timestep $d-1$, we set $\beta_{u,k,m}(\boldsymbol{s}_t) = 1$ for $u \in \mathcal{U}^{\text{URLLC}}$ and $k \in \mathcal{K}^{\text{URLLC}}(\boldsymbol{s}_t)$, which are related to the selected user-PRB pair $\omega_{d-1} \in \widetilde{\mathcal{K}}^{\text{URLLC}}(\boldsymbol{s}_t)$. By assigning PRB k to user u, the allocated transmit power $p_{u,k,m}(\boldsymbol{s}_t)$ for user u on PRB k is obtained based on (17). Hence, $\boldsymbol{C}_d$ at decoding timestep $d > 1$ is updated as $\boldsymbol{C}_d = \boldsymbol{C}_{d-1} - (p_{u,k,m}(\boldsymbol{s}_t),\ 1)$.

Parameter $\hat{\delta}_{\tilde{k},d}$ is the remaining traffic demand, which is initialized by $\hat{\delta}_{\tilde{k},d} = q_{u,m}(\boldsymbol{s}_t)$ for each user-PRB pair $\tilde{k} \in \widetilde{\mathcal{K}}^{\text{URLLC}}(\boldsymbol{s}_t)$ at decoding timestep $d = 1$. We update $\hat{\delta}_{\tilde{k},d}$ for $d > 1$ as follows:

$$\hat{\delta}_{\tilde{k},d} = \begin{cases} \max(0, q_{u,m}(\boldsymbol{s}_t) - R_{u,m}(\boldsymbol{s}_t)), & \text{if } \tilde{k} \in \widetilde{\mathcal{K}}_{\omega_{d-1}}, \\ \hat{\delta}_{\tilde{k},d-1}, & \text{if } \tilde{k} \notin \widetilde{\mathcal{K}}_{\omega_{d-1}}, \end{cases} \tag{24}$$

where $\widetilde{\mathcal{K}}_{\omega_{d-1}} = \{(u,k) \,|\, (u,k') = \omega_{d-1},\ k \in \mathcal{K}^{\text{URLLC}}(\boldsymbol{s}_t) \backslash \{k'\}\}$ denotes the set of (u,k) pairs that u is specified by the selected user-PRB pair ω_{d-1}, and k is specified by the PRB indices. Since the remaining traffic demand $\hat{\delta}_{\tilde{k},d}$ depends on the total number of transmitted bits for user u (i.e., $R_{u,m}(\boldsymbol{s}_t)$), after each selection of ω_{d-1}, we should update $\hat{\delta}_{\tilde{k},d}$ for the selected user and all the remaining PRBs. At decoding timestep d, to satisfy constraint (15), we mask the user-PRB pairs corresponding to the previously selected user-PRB pairs $\omega_1, \ldots, \omega_{d-1}$. To satisfy constraint (18), we mask all the user-PRB pairs in set $\widetilde{\mathcal{K}}^{\text{URLLC}}(\boldsymbol{s}_t)$ that require more transmission power than the remaining transmission power, i.e., first component of $\boldsymbol{C}_d$. Finally, given constraint (20), we mask all the user-PRB pairs with zero remaining traffic demand, i.e., $\hat{\delta}_{\tilde{k},d} = 0$, $\tilde{k} \in \widetilde{\mathcal{K}}^{\text{URLLC}}(\boldsymbol{s}_t)$.

Algorithm 2 describes our proposed training algorithm for the URLLC scheduler. In Line 1, we set the number of training epochs E^{URLLC}, and the batch size κ^{URLLC} for each epoch. In Line 2, the learnable parameters $\boldsymbol{\theta}^{\text{URLLC}}$ are initialized. The loop within Lines 3 to 12 encompasses the learning process of the URLLC scheduler. In Line 4, at each training epoch, we consider κ^{URLLC} input sequences as training samples for that epoch. To generate each training sample, we uniformly select the numerology $i^{\text{URLLC}}(\boldsymbol{s}_t)$, the tuple $\big(n^{\text{eMBB}}(\boldsymbol{s}_t),\ n^{\text{shared}}(\boldsymbol{s}_t),\ n^{\text{URLLC}}(\boldsymbol{s}_t)\big)$, and the power allocation factor $\xi(\boldsymbol{s}_t)$ from the sets $\mathcal{I}$, $\mathcal{N}$, and Ξ, respectively. Then, using different channel gain realizations for user-PRB pairs and different traffic load realizations for the URLLC users, the feature vectors of URLLC user-PRB pairs are determined. In the loop within Lines 5 to 9, for each input sequence, we compute the gradient of the loss function, which is required to update the learnable parameters $\boldsymbol{\theta}^{\text{URLLC}}$. In

Algorithm 2: Training Algorithm for the URLLC Scheduler

1: Set the number of epochs E^{URLLC} and batch size κ^{URLLC}.
2: Initialize randomly the learnable parameters $\boldsymbol{\theta}^{\text{URLLC}}$.
3: **for** each epoch **do**
4: Consider κ^{URLLC} different input sequences $\Phi_{\kappa^{\text{URLLC}}}$ for the URLLC scheduler.
5: **for** each $\Phi \in \Phi_{\kappa^{\text{URLLC}}}$ **do**
6: Feed the sequence Φ into the encoder-decoder DNN modules and obtain Ω using $\pi_{\boldsymbol{\theta}^{\text{URLLC}}}(\Omega \,|\, \Phi)$.
7: Determine $f(\Omega)$ based on the objective value of problem $\mathcal{P}_m^{\text{URLLC-Pen}}(\boldsymbol{s}_t)$.
8: Determine $\nabla \mathcal{L}(\boldsymbol{\theta}^{\text{URLLC}} \,|\, \Phi)$.
9: **end for**
10: Determine the aggregate gradient over the batch as $\nabla \mathcal{L}(\boldsymbol{\theta}^{\text{URLLC}} \,|\, \Phi_{\kappa^{\text{URLLC}}}) := \sum_{\Phi \in \Phi_{\kappa^{\text{URLLC}}}} \nabla \mathcal{L}(\boldsymbol{\theta}^{\text{URLLC}} \,|\, \Phi)$.
11: Update $\boldsymbol{\theta}^{\text{URLLC}}$ using Adam optimizer [33].
12: **end for**
13: Outputs are the learned parameters $\boldsymbol{\theta}^{\text{URLLC}}$.

Line 6, we obtain a probability distribution $\pi_{\boldsymbol{\theta}^{\text{URLLC}}}(\Omega \,|\, \Phi) = \prod_{d=1}^{D} \pi_{\boldsymbol{\theta}^{\text{URLLC}}}(\omega_d \,|\, \Phi, \omega_1, \ldots, \omega_{d-1})$, from which we can sample to determine the allocation of PRBs to URLLC users, i.e., where $\beta_{u,k,m}(\boldsymbol{s}_t)$, $u \in \mathcal{U}^{\text{URLLC}}$, $k \in \mathcal{K}^{\text{URLLC}}(\boldsymbol{s}_t)$ should be equal to one. Considering problem $\mathcal{P}_m^{\text{URLLC-Pen}}(\boldsymbol{s}_t)$, we define $\mathcal{L}(\boldsymbol{\theta}^{\text{URLLC}} \,|\, \Phi) = \mathbb{E}_{\pi_{\boldsymbol{\theta}^{\text{URLLC}}}(\Omega \,|\, \Phi)}[f(\Omega)]$ as the loss function for training our model, where $f(\Omega)$ is the objective function value for problem $\mathcal{P}_m^{\text{URLLC-Pen}}(\boldsymbol{s}_t)$ while all the optimization variables $\beta_{u,k,m}(\boldsymbol{s}_t)$, $\Delta_{u,m}(\boldsymbol{s}_t)$, $u \in \mathcal{U}^{\text{URLLC}}$, $k \in \mathcal{K}^{\text{URLLC}}(\boldsymbol{s}_t)$ have been determined by the output sequence Ω. In Line 7, we obtain $f(\Omega)$. We minimize $\mathcal{L}(\boldsymbol{\theta}^{\text{URLLC}} \,|\, \Phi)$ using Adam optimizer [33]. Hence, in Line 8, we use the REINFORCE gradient estimator to obtain $\nabla \mathcal{L}(\boldsymbol{\theta}^{\text{URLLC}} \,|\, \Phi) = \mathbb{E}_{\pi_{\boldsymbol{\theta}^{\text{URLLC}}}(\Omega \,|\, \Phi)} \left[f(\Omega) \nabla \ln \pi_{\boldsymbol{\theta}^{\text{URLLC}}}(\Omega \,|\, \Phi) \right]$. At the end of each training epoch, we compute the aggregate gradient over the batch in Line 10, and update $\boldsymbol{\theta}^{\text{URLLC}}$ using Adam optimizer in Line 11. The computational complexity of the encoder-decoder DNN architecture with attention mechanism is dominated by the computational complexity of the encoder attention layer [32]. Thus, after training, the computational complexity of the online PRB allocation using the pre-trained DNNs obtained by Algorithm 2 is $O\left(|\widetilde{\mathcal{K}}^{\text{URLLC}}(\boldsymbol{s}_t)| d_{\boldsymbol{h}^{\text{enc}}}^2 + |\widetilde{\mathcal{K}}^{\text{URLLC}}(\boldsymbol{s}_t)|^2 d_{\boldsymbol{h}^{\text{enc}}}\right)$, where $d_{\boldsymbol{h}^{\text{enc}}}$ is the dimension of vector $\boldsymbol{h}^{\text{enc}}$.

2) Algorithm Design for the eMBB Scheduler: The eMBB scheduler aims to solve problem $\mathcal{P}_\tau^{\text{eMBB-Pen}}(\boldsymbol{s}_t)$. The PRB and power allocation is performed at the beginning of each short time slot $\tau \in \mathcal{T}^{\text{short}}$, when the eMBB scheduler has no information about the PRBs that will be punctured by the URLLC traffic in the shared bandwidth part. Hence, the eMBB scheduler considers $\zeta_{k,j,\tau}(\boldsymbol{s}_t) = 1$, $k \in \mathcal{K}^{\text{eMBB}}(\boldsymbol{s}_t)$, $j \in \mathcal{J}^{\text{eMBB}}(\boldsymbol{s}_t)$ at the beginning of short time slot τ, and solves problem $\mathcal{P}_\tau^{\text{eMBB-Pen}}(\boldsymbol{s}_t)$. For solving problem $\mathcal{P}_\tau^{\text{eMBB-Pen}}(\boldsymbol{s}_t)$, the eMBB scheduler determines the PRB allocation, which is a combinatorial problem, using an encoder-decoder DNN with attention mechanism, and obtains $\alpha_{u,k,\tau}(\boldsymbol{s}_t)$, $u \in \mathcal{U}^{\text{eMBB}}$, $k \in \mathcal{K}^{\text{eMBB}}(\boldsymbol{s}_t)$ based on the output of the decoder module. Given the determined $\alpha_{u,k,\tau}(\boldsymbol{s}_t)$, the eMBB scheduler solves the following convex optimization problem to obtain the power allocation for the users:

$$\mathcal{P}_\tau^{\text{eMBB-Power}}(\boldsymbol{s}_t):$$
$$\underset{\substack{p_{u,k,\tau}(\boldsymbol{s}_t),\\ \Delta_{u,j,\tau}(\boldsymbol{s}_t),\\ u \in \mathcal{U}^{\text{eMBB}},\, k \in \mathcal{K}^{\text{eMBB}}(\boldsymbol{s}_t),\, j \in \mathcal{J}^{\text{eMBB}}(\boldsymbol{s}_t)}}{\text{maximize}} \quad f_\tau^{\text{eMBB}}(\boldsymbol{s}_t) - \lambda^{\text{eMBB-Pen}} \sum_{j \in \mathcal{J}^{\text{eMBB}}(\boldsymbol{s}_t)} \sum_{u \in \mathcal{U}^{\text{eMBB}}} \Delta_{u,j,\tau}^2(\boldsymbol{s}_t)$$
subject to constraints (11) and (13).

At the end of short time slot τ, the true value of the parameters $\zeta_{k,j,\tau}(\boldsymbol{s}_t)$, $k \in \mathcal{K}^{\text{eMBB}}(\boldsymbol{s}_t)$, $j \in \mathcal{J}^{\text{eMBB}}(\boldsymbol{s}_t)$ is revealed to the eMBB scheduler through its interaction with the URLLC scheduler to be used for training the encoder and decoder DNNs of the eMBB scheduler.

Given the sequence $\boldsymbol{s}_t$, $\widetilde{\mathcal{K}}^{\text{eMBB}}(\boldsymbol{s}_t) = \mathcal{U}^{\text{eMBB}} \times \mathcal{K}^{\text{eMBB}}(\boldsymbol{s}_t)$ denotes the set of user-PRB pairs for the eMBB scheduler. For the input sequence of the encoder, based on the available information at the eMBB scheduler, $\boldsymbol{\phi}_{\tilde{k}} = \left(|g_{u,k,\tau}(\boldsymbol{s}_t)|^2, \tilde{\eta}_{k,\tau}(\boldsymbol{s}_t)\right)$ is considered as the feature vector of user-PRB pair $\tilde{k} \in \widetilde{\mathcal{K}}^{\text{eMBB}}(\boldsymbol{s}_t)$ in short time slot $\tau \in \mathcal{T}^{\text{short}}$, where $\tilde{\eta}_{k,\tau}(\boldsymbol{s}_t) \in \{0, 1\}$ is a parameter that indicates whether PRB $k \in \mathcal{K}^{\text{eMBB}}(\boldsymbol{s}_t)$ is in the shared bandwidth part or not.

For the decoder, the number of decoding timesteps is set to the number of available PRBs for the eMBB slice, i.e., $D = |\mathcal{K}^{\text{eMBB}}(\boldsymbol{s}_t)|$. At decoding timestep $d \in \mathcal{D}$, the context embedding is defined as $\boldsymbol{h}_d^{\text{context}} = \left[\boldsymbol{h}^{\text{mean}}, \boldsymbol{h}_{\omega_{d-1}}^{\text{enc}}, C_d\right]$, where C_d denotes the available number of PRBs at decoding timestep d. At decoding timestep $d = 1$, we initialize $C_d = K^{\text{eMBB}}(\boldsymbol{s}_t)$, and at decoding timestep $d > 1$, we update $C_d = K^{\text{eMBB}}(\boldsymbol{s}_t) - d + 1$. To satisfy constraint (10) in problem $\mathcal{P}_\tau^{\text{eMBB-Pen}}(\boldsymbol{s}_t)$, at decoding timestep d, we mask the user-PRB pairs corresponding to the previously selected user-PRB pairs $\omega_1, \ldots, \omega_{d-1}$. We should also mask all the user-PRB pairs $\tilde{k} \in \widetilde{\mathcal{K}}^{\text{eMBB}}(\boldsymbol{s}_t)$, which correspond to the selected PRBs during the previous decoding timesteps and the other eMBB users. Given the output sequence of the decoder module, other constraints of problem $\mathcal{P}_\tau^{\text{eMBB-Pen}}(\boldsymbol{s}_t)$ are satisfied by solving the convex optimization problem $\mathcal{P}_\tau^{\text{eMBB-Power}}(\boldsymbol{s}_t)$.

Algorithm 3 describes our proposed training algorithm for the eMBB scheduler. To generate each training sample for Algorithm 3, we uniformly select $i^{\text{eMBB}}(\boldsymbol{s}_t)$, $i^{\text{URLLC}}(\boldsymbol{s}_t)$, $(n^{\text{eMBB}}(\boldsymbol{s}_t), n^{\text{shared}}(\boldsymbol{s}_t), n^{\text{URLLC}}(\boldsymbol{s}_t))$, and $\xi(\boldsymbol{s}_t)$ from the corresponding sets. Then, by using different channel gain realizations for the eMBB user-PRB pairs, the feature vector of each pair is generated. Moreover, for each input sequence of the eMBB scheduler, we generate the input sequences of the URLLC scheduler for the mini time slots within one short time slot according to the procedure described in Section IV-B1. Hence, using the pre-trained DNNs for the URLLC scheduler, we can obtain puncturing variables $\zeta_{k,j,\tau}(\boldsymbol{s}_t)$ to be used for training the DNNs of the eMBB scheduler.

To train the DNN modules for the eMBB scheduler, using the objective function of problem $\mathcal{P}_\tau^{\text{eMBB-Pen}}(\boldsymbol{s}_t)$, we define the loss function $\mathcal{L}(\boldsymbol{\theta}^{\text{eMBB}} \,|\, \Phi) = \mathbb{E}_{\pi_{\boldsymbol{\theta}^{\text{eMBB}}}(\Omega \,|\, \Phi)}[-f(\Omega)]$. In the loop within Lines 5 to 12, at the beginning of each short time slot, we feed the input sequence Φ to the encoder module and obtain the probability distribution $\pi_{\boldsymbol{\theta}^{\text{eMBB}}}(\Omega \,|\, \Phi)$ as the decoder output to determine the allocation of PRBs to eMBB users, i.e., where $\alpha_{u,k,\tau}(\boldsymbol{s}_t)$ should be equal to

Algorithm 3: Training Algorithm for the eMBB Scheduler

Set the number of epochs E^{eMBB} and batch size κ^{eMBB}.
Initialize randomly the learnable parameters $\boldsymbol{\theta}^{\text{eMBB}}$.
for each epoch **do**
  Consider κ^{eMBB} different input sequences $\Phi_{\kappa^{\text{eMBB}}}$ for the eMBB scheduler.
  for each $\Phi \in \Phi_{\kappa^{\text{eMBB}}}$ **do**
    Feed the sequence Φ into the encoder-decoder DNN modules and obtain Ω using $\pi_{\boldsymbol{\theta}^{\text{eMBB}}}(\Omega \,|\, \Phi)$.
    Obtain $\alpha_{u,k,\tau}(\boldsymbol{s}_t)$, $u \in \mathcal{U}^{\text{eMBB}}$, $k \in \mathcal{K}^{\text{eMBB}}(\boldsymbol{s}_t)$ based on the selected user-PRB pairs Ω.
    Set $\zeta_{k,j,\tau}(\boldsymbol{s}_t) = 1$, $k \in \mathcal{K}^{\text{eMBB}}(\boldsymbol{s}_t)$, $j \in \mathcal{J}^{\text{eMBB}}(\boldsymbol{s}_t)$ at the beginning of short time slot τ.
    Obtain $p_{u,k,\tau}(\boldsymbol{s}_t)$, $u \in \mathcal{U}^{\text{eMBB}}$, $k \in \mathcal{K}^{\text{eMBB}}(\boldsymbol{s}_t)$ by solving problem $\mathcal{P}_{\tau}^{\text{eMBB-Power}}(\boldsymbol{s}_t)$.
    Obtain true value of $\zeta_{k,j,\tau}(\boldsymbol{s}_t)$, $k \in \mathcal{K}^{\text{eMBB}}(\boldsymbol{s}_t)$, $j \in \mathcal{J}^{\text{eMBB}}(\boldsymbol{s}_t)$ at the end of short time slot τ.
    Determine $f(\Omega)$ based on the objective function of problem $\mathcal{P}_{\tau}^{\text{eMBB-Pen}}(\boldsymbol{s}_t)$, and compute $\nabla \mathcal{L}(\boldsymbol{\theta}^{\text{eMBB}} \,|\, \Phi)$.
  end for
  Determine the aggregate gradient over the batch as $\nabla \mathcal{L}(\boldsymbol{\theta}^{\text{eMBB}} \,|\, \Phi_{\kappa^{\text{eMBB}}}) := \sum_{\Phi \in \Phi_{\kappa^{\text{eMBB}}}} \nabla \mathcal{L}(\boldsymbol{\theta}^{\text{eMBB}} \,|\, \Phi)$.
  Update $\boldsymbol{\theta}^{\text{eMBB}}$ using Adam optimizer [33].
end for
Outputs are the learned parameters $\boldsymbol{\theta}^{\text{eMBB}}$.

one. To obtain the power allocation variables $p_{u,k,\tau}(\boldsymbol{s}_t)$, we set $\zeta_{k,j,\tau}(\boldsymbol{s}_t) = 1$ and solve problem $\mathcal{P}_{\tau}^{\text{eMBB-Power}}(\boldsymbol{s}_t)$ when variables $\alpha_{u,k,\tau}(\boldsymbol{s}_t)$ are considered to be known. At the end of each short time slot, the eMBB scheduler obtains the true value of $\zeta_{k,j,\tau}(\boldsymbol{s}_t)$. Given the determined $\alpha_{u,k,\tau}(\boldsymbol{s}_t)$, $p_{u,k,\tau}(\boldsymbol{s}_t)$, and $\zeta_{k,j,\tau}(\boldsymbol{s}_t)$, we recompute the penalizing variables $\Delta_{u,j,\tau}(\boldsymbol{s}_t)$ and substitute them into the objective function of problem $\mathcal{P}_{\tau}^{\text{eMBB-Pen}}(\boldsymbol{s}_t)$ to obtain $f(\Omega)$ in the loss function. After training, the computational complexity of performing online PRB allocation based on the pre-trained DNNs obtained by Algorithm 3 is $O\left(|\widetilde{\mathcal{K}}^{\text{eMBB}}(\boldsymbol{s}_t)| d^2_{\boldsymbol{h}^{\text{enc}}} + |\widetilde{\mathcal{K}}^{\text{eMBB}}(\boldsymbol{s}_t)|^2 d_{\boldsymbol{h}^{\text{enc}}}\right)$, and solving the convex optimization problem $\mathcal{P}_{\tau}^{\text{eMBB-Power}}(\boldsymbol{s}_t)$ has a polynomial computational complexity.

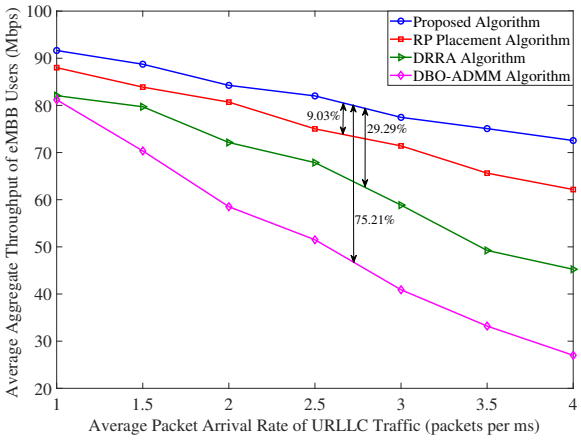
V. Performance Evaluation

In this section, we evaluate the performance of our proposed hierarchical framework. We consider a single-cell RAN, where the base station is located at the center of the cell and its maximum transmit power $P^{\max}$ is set to 1 W. The cell is modeled as a circle of radius 500 m. Unless stated otherwise, we consider six eMBB users and four URLLC users. The eMBB and URLLC users are distributed randomly within the cell. We consider small-scale Rayleigh fading as well as log-normal shadowing path loss model to simulate the wireless channels between the base station and users. Specifically, the path loss exponent is set to 3.76. The path loss at 1 km reference distance is set to 128.1 dB. The log-normal shadowing standard deviation is set to 10 dB. The total system bandwidth is set to 10 MHz. We consider that the RIC selects $\xi(\boldsymbol{s}_t)$ from set $\Xi = \{0.25, 0.5, 0.75\}$. The values in set Ξ indicate that three discrete power levels are used by the RIC for power allocation to the slices. In addition, we consider that the RIC selects tuple $\left(n^{\text{eMBB}}(\boldsymbol{s}_t),\ n^{\text{shared}}(\boldsymbol{s}_t),\ n^{\text{URLLC}}(\boldsymbol{s}_t)\right)$ from set $\mathcal{N} = \{(0,12,0), (6,0,6), (4,4,4), (8,4,0), (8,0,4), (4,0,8), (4,8,0), (0,8,4), (0,4,8), (8,2,2), (2,8,2), (2,2,8)\}$. The elements in set $\mathcal{N}$ correspond to different slicing scenarios. The RIC can perform orthogonal slicing by selecting tuples $(6,0,6)$, $(8,0,4)$, or $(4,0,8)$ from set $\mathcal{N}$. The RIC can perform non-orthogonal slicing by selecting $(0,12,0)$. Moreover, the RIC can use the hybrid slicing approach by selecting other available tuples from set $\mathcal{N}$. Note that one can include other combinations in set $\mathcal{N}$, or consider finer granularity for the discretized power levels. However, a larger number of iterations is required for Algorithm 1 to converge. We consider that the arrival rate of URLLC packets follows the Poisson distribution and the URLLC packet size is set to 32 bytes. We also consider that the eMBB buffers at the base station always have data to send (i.e., full buffer model). Other simulation parameters are summarized in Table IV. For both the actor and critic networks, we consider the neural networks comprising of one LSTM layer with 256 hidden units and one fully connected layer with 512 neurons. We perform simulations using PyTorch library [34] in Python 3.7, and MOSEK solver [35]. Simulation results are obtained by averaging over 50 different simulation trials. We use the following state-of-the-art RAN slicing approaches as the benchmark schemes for performance comparison:

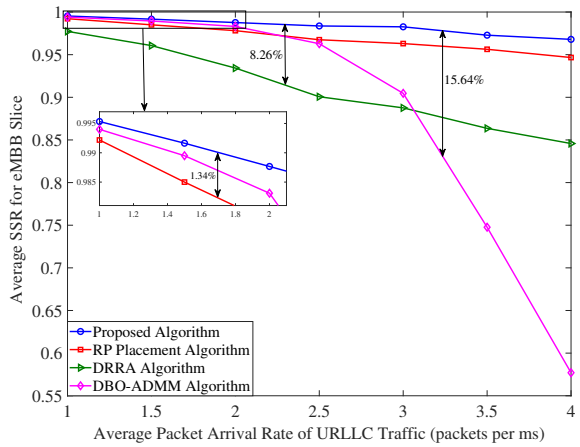
Table IV: Simulation Parameters

Parameter	Value	Parameter	Value	Parameter	Value
ΔT^{mini}	143 μs	ΔT^{short}	1 ms	ΔT^{long}	1 s
ϵ^{B}	10^{-6}	σ^2	-110 dBm	λ^{eMBB}	5
Γ^{THR}	5 dB	γ	0.99	υ^{eMBB}	0.9
λ^{punc}	2	λ^{sp}	0.01	λ^{URLLC}	5
$\lambda^{\text{URLLC-Pen}}$	10	$\lambda^{\text{eMBB-Pen}}$	100	υ^{URLLC}	0.99
E^{URLLC}	100	E^{eMBB}	100	ν^{actor}	10^{-5}
κ^{URLLC}	1280000	κ^{eMBB}	12800	ν^{critic}	10^{-4}

- Resource proportional (RP) placement algorithm proposed in [6]: In this algorithm, the system bandwidth is shared among the eMBB and URLLC users. We use our proposed resource allocation algorithm for the eMBB users. The PRB allocation for URLLC traffic is based on the RP placement algorithm. For URLLC traffic, the PRBs preempted from each eMBB user by this algorithm are proportional to the allocated PRBs to that user.
- Decomposition and relaxation based resource allocation (DRRA) algorithm proposed in [11]: In this algorithm, the system bandwidth is shared among the users. The problem is decomposed into three subproblems: PRB allocation for eMBB users, power allocation for eMBB users, and PRB allocation for URLLC traffic. Integer variables are relaxed to continuous ones. Then, the subproblems are solved iteratively until the algorithm converges.
- Distributed bandwidth optimization based on ADMM (DBO-ADMM) algorithm proposed in [5]: This algorithm employs sample average approximation and ADMM techniques for bandwidth allocation to eMBB and URLLC users. The allocated bandwidth remains the same during the long time slot. Given the allocated bandwidth, a power allocation problem is solved for the eMBB users during each short time slot.

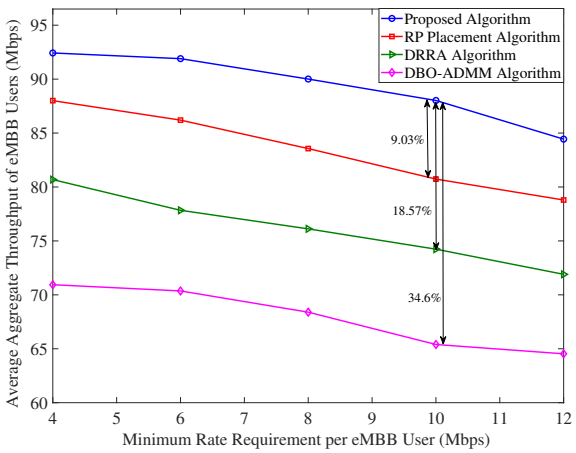


(a)

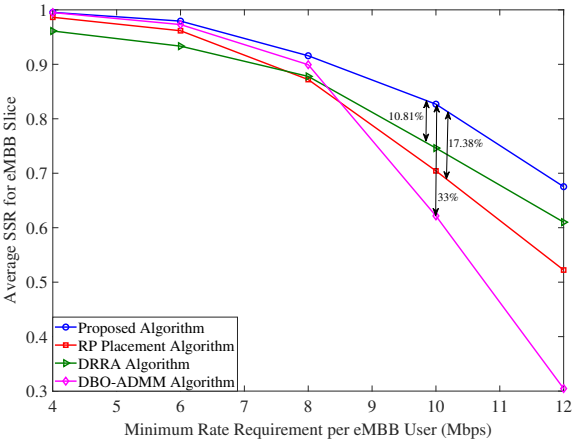


(b)

Fig. 6: (a) Average aggregate throughput and (b) average SSR for eMBB slice versus the average packet arrival rate of URLLC traffic. We set $R_u^{\min} = 4$ Mbps, $u \in \mathcal{U}^{\text{eMBB}}$.



(a)



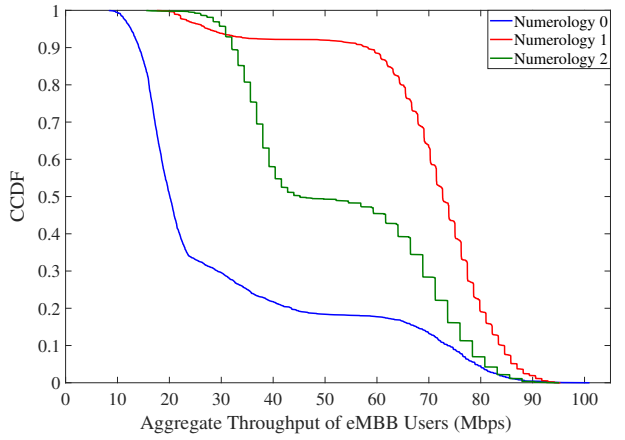
(b)

Fig. 7: (a) Average aggregate throughput and (b) average SSR for eMBB slice versus the minimum rate requirement per eMBB user. We set the average packet arrival rate of each URLLC user to 1.5 packets per ms.

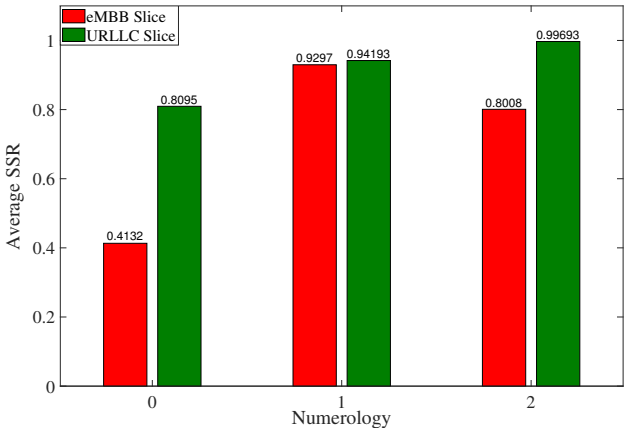
Fig. 6(a) shows the evolution of the average aggregate throughput of eMBB users versus the average packet arrival rate of URLLC traffic. The proposed algorithm achieves an average aggregate throughput, which is 9.03%, 29.29%, and 75.21% higher than that of the RP placement, DRRA, and DBO-ADMM algorithms, respectively. We observe that increasing the average packet arrival rate leads to the average aggregate throughput degradation for eMBB users. However, this degradation is much lower in our proposed algorithm due to the selected slice configuration parameters. Results in Fig. 6(b) show that when the average packet arrival rate for URLLC traffic increases, our proposed algorithm can maintain the SSR for the eMBB slice above 95%. Moreover, as the packet arrival rate increases, our proposed algorithm achieves the SSR, which is on average 1.34%, 8.26%, and 15.64% higher than that of the RP placement, DRRA, and DBO-ADMM algorithms, respectively. Results in Fig. 6 show that using our proposed algorithm, the RIC can provide the schedulers with proper slice configuration parameters based on the network dynamics including the packet arrival rate of URLLC traffic and the channel gain variations.

In Fig. 7(a), we compare the average aggregate throughput of the eMBB users for different algorithms while changing the minimum rate requirement of the eMBB users. The proposed algorithm can achieve an average aggregate throughput that is 9.03%, 18.57%, and 34.6% higher than that of the RP placement, DRRA, and DBO-ADMM algorithms, respectively, when $R_u^{\min} = 10$ Mbps. Fig. 7(b) shows the impact of the minimum rate requirement of the eMBB users on the average SSR for the eMBB slice. All algorithms suffer from a performance degradation as $R_u^{\min}$, $u \in \mathcal{U}^{\text{eMBB}}$ increases. However, in the proposed algorithm, the RIC can dedicate a portion of the bandwidth to be used exclusively by the eMBB users. Thus, when the minimum rate requirement of the eMBB users is large, a higher average SSR for the eMBB slice can be achieved under the proposed algorithm compared to the other benchmarks. Hence, the eMBB rate loss due to the punctured scheduling decreases in our proposed algorithm by reserving a portion of the bandwidth for the eMBB users. In fact, the proposed algorithm can achieve an average SSR for the eMBB slice that is 17.38%, 10.81%, and 33% higher than that of the RP placement, DRRA, and DBO-ADMM algorithms, respectively, when $R_u^{\min} = 10$ Mbps.

In Fig. 8, we study the impact of the numerology selection on the performance of the eMBB and URLLC schedulers.

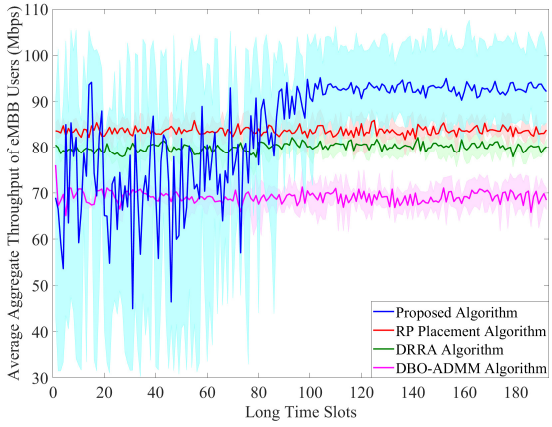


(a)

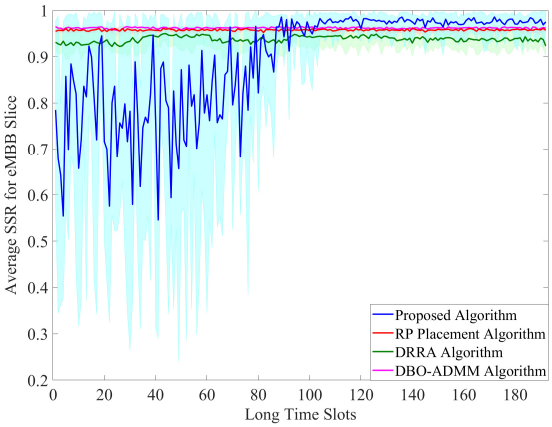


(b)

Fig. 8: (a) CCDF of the aggregate throughput for the eMBB users and (b) average SSR for eMBB and URLLC slices for different numerologies. We set $R_u^{\min} = 4$ Mbps, $u \in \mathcal{U}^{\text{eMBB}}$. The slice configuration parameters are $\xi(\boldsymbol{s}_t) = 0.75$, $n^{\text{eMBB}}(\boldsymbol{s}_t) = 8$, $n^{\text{URLLC}}(\boldsymbol{s}_t) = 4$, and $n^{\text{shared}}(\boldsymbol{s}_t) = 0$. The average packet arrival rate of each URLLC user is set to 2.5 packets per ms.



(a)



(b)

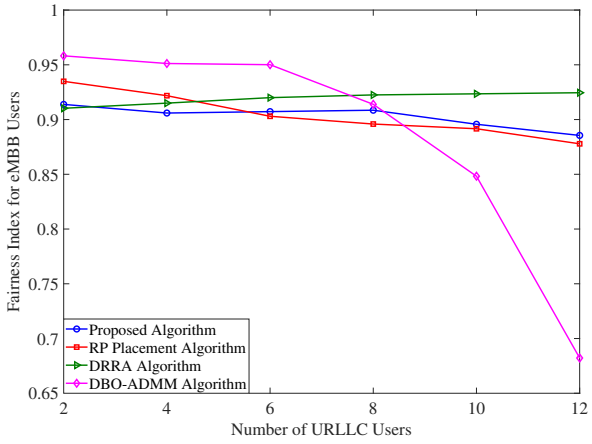
Fig. 9: Convergence of the (a) average aggregate throughput and (b) average SSR for eMBB slice over long time slots. We set $R_u^{\min} = 4$ Mbps, $u \in \mathcal{U}^{\text{eMBB}}$, and the average packet arrival rate of each URLLC user to 1.5 packets per ms.

Fig. 8(a) shows the complementary cumulative distribution function (CCDF) of the aggregate throughput for the eMBB users with different numerologies. A higher CCDF indicates a higher probability for obtaining the aggregate throughput above a given threshold value. Results in Fig. 8(a) show that by selecting numerology 1 for the eMBB slice, the aggregate throughput of the eMBB users can be above a threshold value from 31 Mbps to 91 Mbps with a higher probability. Fig. 8(b) shows the average SSR for the eMBB and URLLC slices with different numerologies. The numerology 1 leads to higher average SSR for the eMBB slice. Furthermore, URLLC slice obtains a higher average SSR with numerology 2. Hence, the RIC can improve the system performance by appropriately choosing the numerology.

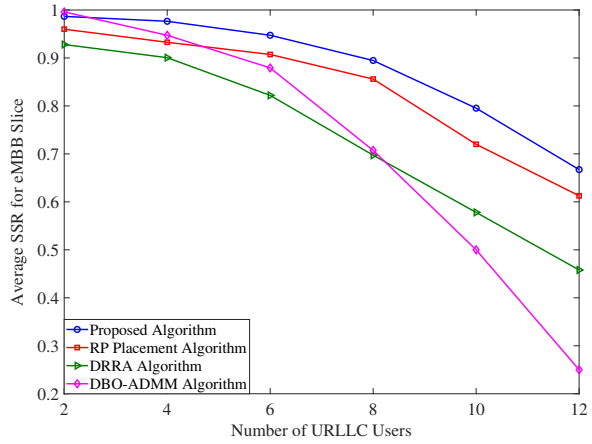
Fig. 9 shows the evolution of the average aggregate throughput and average SSR for the eMBB slice over long time slots. As shown in Fig. 9, our proposed algorithm converges within 100 long time slots. Fig. 9 also shows the convergence region of our proposed algorithm for 50 different trials. We have obtained the convergence region based on the trials with the best and worst performance in each long time slot. Results show that compared to the other benchmarks, our proposed algorithm provides a higher average aggregate throughput and average SSR.

In Fig. 10, we compare the Jain's fairness index and average SSR for the users in the eMBB slice among the considered algorithms, while the number of URLLC users varies from 2 to 12. As Fig. 10(a) illustrates, our proposed algorithm can maintain the fairness index above 0.9. Although our proposed algorithm does not provide the highest fairness index, the fairness index obtained by our proposed algorithm is within the range of the other benchmarks. As shown in Fig. 10(b), our proposed algorithm can obtain a higher average SSR for the eMBB slice compared to the RP placement, DRRA, and DBO-ADMM algorithms. Note that the reward function in (7) and the considered objective function for problem $\mathcal{P}_{\tau}^{\text{eMBB-Pen}}(\boldsymbol{s}_t)$ are based on maximizing the aggregate throughput for the eMBB users. However, one can replace it with the fairness index to maximize the fairness for the eMBB users.

In Fig. 11, we evaluate the effect of choosing the weighting coefficients λ^{eMBB} and λ^{URLLC} used in (7) for penalizing the reward function due to the violation from the SSR threshold values for the eMBB and URLLC slices on the system performance. By choosing small values for λ^{eMBB} and λ^{URLLC} relative to other terms in the reward function, the RIC aims

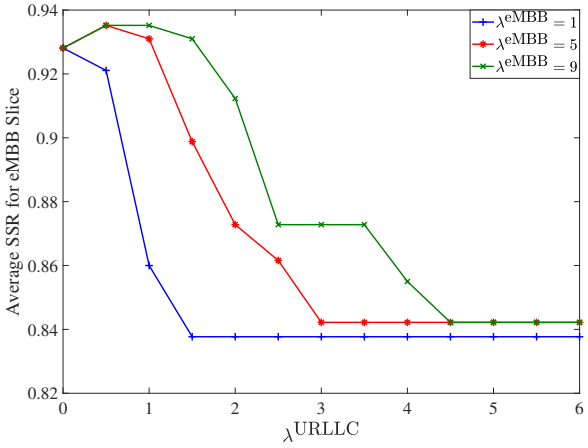


(a)

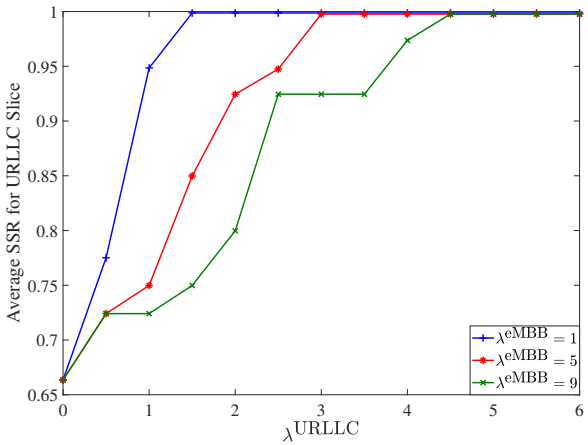


(b)

Fig. 10: (a) Jain's fairness index and (b) average SSR of the users in the eMBB slice for different number of URLLC users. We set $R_u^{\min} = 6$ Mbps, $u \in \mathcal{U}^{\text{eMBB}}$, and the average packet arrival rate of each URLLC user to 1.5 packets per ms.



(a)



(b)

Fig. 11: (a) Average SSR for eMBB slice and (b) average SSR for URLLC slice versus the weighting coefficient λ^{URLLC}. We set $R_u^{\min} = 6$ Mbps, $u \in \mathcal{U}^{\text{eMBB}}$, and the average packet arrival rate of each URLLC user to 1.5 packets per ms. We consider six eMBB users and eight URLLC users in the network.

to only optimize the average aggregate throughput of the eMBB users. If we choose a sufficiently large value for λ^{eMBB}, but not for λ^{URLLC}, the RIC aims to optimize the average aggregate throughput for the eMBB users and average SSR for the eMBB slice. Hence, it may violate the SSR threshold value for the URLLC slice. However, as Fig. 11 shows, by choosing sufficiently large values for λ^{eMBB} and λ^{URLLC}, the SSR threshold can be guaranteed for both the eMBB and URLLC slices.

VI. Conclusion

In this paper, we proposed a hierarchical deep learning framework for resource slicing in an OFDMA-based RAN. To facilitate joint scheduling of eMBB and URLLC traffic in a shared RAN infrastructure, numerology, mini-slot based transmission, and a combination of orthogonal and punctured scheduling approaches are exploited in our RAN slicing problem formulation. We modeled the selection of slice configuration parameters, which are determined by the RIC at the beginning of each long time slot, as a POMDP. We applied a DRL algorithm based on the actor-critic method to determine the optimal slice configuration parameters. Given the slice configuration parameters, we applied DNNs with attention mechanism to develop resource allocation algorithms for the eMBB and URLLC schedulers. Through simulations, we showed that our proposed hierarchical framework can adapt to the network dynamics. When compared with some existing algorithms in the literature, our proposed hierarchical deep learning framework can achieve a higher average aggregate throughput for the eMBB users, and a higher average SSR for the eMBB and URLLC slices. For future work, we plan to study the impact of inter-cell interference in a multi-cell system, as well as the inter-numerology interference, which can arise in a mixed numerologies system [36] on the performance of RAN slicing schemes.

References

[1] K. Katsalis, N. Nikaein, E. Schiller, A. Ksentini, and T. Braun, "Network slices toward 5G communications: Slicing the LTE network," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 146–154, Aug. 2017.

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

[3] P. Rost, C. Mannweiler, D. S. Michalopoulos, C. Sartori, V. Sciancalepore, N. Sastry, O. Holland, S. Tayade, B. Han, D. Bega, D. Aziz, and H. Bakker, "Network slicing to enable scalability and flexibility in 5G

mobile networks," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 72–79, May 2017.
[4] A. K. Bairagi, M. S. Munir, M. Alsenwi, N. H. Tran, S. S. Alshamrani, M. Masud, Z. Han, and C. S. Hong, "Coexistence mechanism between eMBB and uRLLC in 5G wireless networks," *IEEE Trans. Commun.*, vol. 69, no. 3, pp. 1736–1749, Mar. 2021.
[5] P. Yang, X. Xi, T. Q. Quek, J. Chen, X. Cao, and D. Wu, "How should I orchestrate resources of my slices for bursty URLLC service provision?" *IEEE Trans. Commun.*, vol. 69, no. 2, pp. 1134–1146, Feb. 2021.
[6] A. Anand, G. De Veciana, and S. Shakkottai, "Joint scheduling of URLLC and eMBB traffic in 5G wireless networks," *IEEE/ACM Trans. Netw.*, vol. 28, no. 2, pp. 477–490, Apr. 2020.
[7] M. Alsenwi, N. H. Tran, M. Bennis, A. K. Bairagi, and C. S. Hong, "eMBB-URLLC resource slicing: A risk-sensitive approach," *IEEE Commun. Lett.*, vol. 23, no. 4, pp. 740–743, Apr. 2019.
[8] M. Setayesh, S. Bahrami, and V. W.S. Wong, "Joint PRB and power allocation for slicing eMBB and URLLC services in 5G C-RAN," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Taipei, Taiwan, Dec. 2020.
[9] W. Wu, N. Chen, C. Zhou, M. Li, X. Shen, W. Zhuang, and X. Li, "Dynamic RAN slicing for service-oriented vehicular networks via constrained learning," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 7, pp. 2076–2089, Jul. 2021.
[10] Y. Hua, R. Li, Z. Zhao, X. Chen, and H. Zhang, "GAN-powered deep distributional reinforcement learning for resource management in network slicing," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 2, pp. 334–349, Feb. 2020.
[11] M. Alsenwi, N. H. Tran, M. Bennis, S. R. Pandey, A. K. Bairagi, and C. S. Hong, "Intelligent resource slicing for eMBB and URLLC coexistence in 5G and beyond: A deep reinforcement learning based approach," *IEEE Trans. Wireless Commun.*, vol. 20, no. 7, pp. 4585–4600, Jul. 2021.
[12] Y. Huang, S. Li, C. Li, Y. T. Hou, and W. Lou, "A deep-reinforcement-learning-based approach to dynamic eMBB/URLLC multiplexing in 5G NR," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6439–6456, Jul. 2020.
[13] Q. Liu, T. Han, N. Zhang, and Y. Wang, "DeepSlicing: Deep reinforcement learning assisted resource allocation for network slicing," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Taipei, Taiwan, Dec. 2020.
[14] 3GPP TS 38.300 V16.8.0, "Technical specification group radio access network; NR; NR and NG-RAN overall description; Stage 2 (Release 16)," Dec. 2021.
[15] 5G America, "New services and applications with 5G ultra-reliable low latency communications," white paper, Nov. 2018.
[16] 3GPP R1-1700374, "Downlink multiplexing of eMBB and URLLC transmission," Jan. 2017.
[17] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.
[18] O-RAN Alliance, "O-RAN minimum viable plan and acceleration towards commercialization," white paper, Jun. 2021.
[19] W. Kool, H. Van Hoof, and M. Welling, "Attention, learn to solve routing problems!" in *Proc. of Int'l Conf. Learn. Representations (ICLR)*, New Orleans, LA, May 2019.
[20] L. Bonati, S. D'Oro, M. Polese, S. Basagni, and T. Melodia, "Intelligence and learning in O-RAN for data-driven NextG cellular networks," *IEEE Commun. Mag.*, vol. 59, no. 10, pp. 21–27, Oct. 2021.
[21] 3GPP TR 28.809 V17.0.0, "Technical specification group services and system aspects; Management and orchestration; Study on enhancement of management data analytics (MDA) (Release 17)," Mar. 2021.
[22] 3GPP R1-166103, "Discussion on flexible frame structure with different numerologies," Aug. 2016.
[23] 3GPP TR 21.915 V15.0.0, "Technical specification group services and system aspects; Release 15 description; Summary of Rel-15 work items (Release 15)," Sept. 2019.
[24] K. I. Pedersen, G. Pocovi, J. Steiner, and S. R. Khosravirad, "Punctured scheduling for critical low latency data on a shared channel with mobile broadband," in *Proc. IEEE Veh. Technol. Conf. (VTC2017-Fall)*, Toronto, Canada, Sept. 2017.
[25] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. MIT Press, 2018.
[26] 3GPP R1-1700380, "Mini-slot length and start time for URLLC," Jan. 2017.
[27] C. She, C. Sun, Z. Gu, Y. Li, C. Yang, H. V. Poor, and B. Vucetic, "A tutorial on ultrareliable and low-latency communications in 6G: Integrating domain knowledge into deep learning," *Proc. of the IEEE*, vol. 109, no. 3, pp. 204–246, Mar. 2021.
[28] A. Destounis and G. S. Paschos, "Complexity of URLLC scheduling and efficient approximation schemes," in *Proc. Int'l Symp. on Model. and Optim. in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, Avignon, France, Jun. 2019.
[29] M. Hausknecht and P. Stone, "Deep recurrent Q-learning for partially observable MDPs," in *Proc. of AAAI Fall Symp. Series*, Arlington, Virginia, Nov. 2015.
[30] J. Zhang, X. Tao, H. Wu, N. Zhang, and X. Zhang, "Deep reinforcement learning for throughput improvement of the uplink grant-free NOMA system," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6369–6379, Jul. 2020.
[31] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
[32] C. He, Y. Hu, Y. Chen, and B. Zeng, "Joint power allocation and channel assignment for NOMA with deep reinforcement learning," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2200–2210, Oct. 2019.
[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. of Int'l Conf. Learning Representations (ICLR)*, San Diego, CA, May 2015.
[34] PyTorch. (2022) PyTorch documentation [Online]. Available: https://pytorch.org/docs/stable/index.html.
[35] MOSEK ApS, "Mosek modeling cookbook," Available: https://docs.mosek.com/MOSEKModelingCookbook-letter.pdf, Nov. 2021.
[36] J. Mao, L. Zhang, P. Xiao, and K. Nikitopoulos, "Interference analysis and power allocation in the presence of mixed numerologies," *IEEE Trans. Wireless Commun.*, vol. 19, no. 8, pp. 5188–5203, Aug. 2020.

Mehdi Setayesh (S'20) received the B.Sc. and M.Sc. degrees both in Electrical Engineering from the Sharif University of Technology, Tehran, Iran, in 2014 and 2016, respectively. He is currently a Ph.D. Candidate in the Department of Electrical and Computer Engineering, the University of British Columbia (UBC), Vancouver, Canada. He has been a recipient of the Graduate Support Initiative (GSI) Award from the Faculty of Applied Science at UBC (2020−2022). He received the Best Paper Award at the *IEEE GLOBECOM* 2020. His research interests include machine learning, algorithm design, and optimization in wireless communication systems.

Shahab Bahrami (M'17) received the B.A.Sc. and M.A.Sc. degrees both in Electrical Engineering from Sharif University of Technology, Tehran, Iran, in 2010 and 2012, respectively. He received the Ph.D. degree in Electrical & Computer Engineering from the University of British Columbia (UBC), Vancouver, Canada in 2017. Dr. Bahrami has received various prestigious scholarships at UBC, including the distinguished and highly competitive UBC's Four Year Fellowship (2013−2017) and the Graduate Support Initiative Award from the Faculty of Applied Science at UBC (2014−2017). Currently, he works as a postdoctoral research fellow at UBC. His research interests include convex optimization, machine learning, and deep reinforcement learning with applications to 5G wireless communication and smart grid.

Vincent W.S. Wong (S'94, M'00, SM'07, F'16) received the B.Sc. degree from the University of Manitoba, Canada, in 1994, the M.A.Sc. degree from the University of Waterloo, Canada, in 1996, and the Ph.D. degree from the University of British Columbia (UBC), Vancouver, Canada, in 2000. From 2000 to 2001, he worked as a systems engineer at PMC-Sierra Inc. (now Microchip Technology Inc.). He joined the Department of Electrical and Computer Engineering at UBC in 2002 and is currently a Professor. His research areas include protocol design, optimization, and resource management of communication networks, with applications to wireless networks, smart grid, mobile edge computing, and Internet of Things. He received the Best Paper Award at the *IEEE GLOBECOM* 2020. Currently, Dr. Wong is the Chair of the Executive Editorial Committee of *IEEE Transactions on Wireless Communications*, an Area Editor of *IEEE Transactions on Communications* and *IEEE Open Journal of the Communications Society*, and an Associate Editor of *IEEE Transactions on Mobile Computing*. He has served as a Guest Editor of *IEEE Journal on Selected Areas in Communications*, *IEEE Internet of Things Journal*, and *IEEE Wireless Communications*. He has also served on the editorial boards of *IEEE Transactions on Vehicular Technology* and *Journal of Communications and Networks*. He was a Tutorial Co-Chair of *IEEE GLOBECOM*'18, a Technical Program Co-chair of *IEEE VTC*2020-*Fall* and *IEEE SmartGridComm*'14, as well as a Symposium Co-chair of *IEEE ICC*'18, *IEEE SmartGridComm* ('13, '17) and *IEEE GLOBECOM*'13. He is the Chair of the IEEE Vancouver Joint Communications Chapter and has served as the Chair of the IEEE Communications Society Emerging Technical Sub-Committee on Smart Grid Communications. He was an IEEE Communications Society Distinguished Lecturer (2019−2020).