Coordinated Slice Management for Non-Stationary Mobile Networks: A Hierarchical Timescale Scheduling Method

Rong Chai, *Graduate Student Member, IEEE*, and Shaowei Wang, *Senior Member, IEEE*

***Abstract*—In this paper, we propose a hierarchical scheduling method for enhanced mobile broadband and ultra-reliable low-latency communication slices in mobile networks with non-stationary traffic arrivals, where resource allocation and traffic scheduling are decoupled into intra-slice and inter-slice processes operating on different timescales. An effective bandwidth estimation algorithm is developed, dynamically triggered by traffic fluctuations to determine transmission rates that ensure ultra-reliable delay guarantees under non-stationary arrivals. The intra-slice scheduling is formulated as a generalized assignment problem, for which we design an efficient algorithm that maximizes throughput under transmission rate constraints and guarantees a provable approximation ratio. We propose an inter-slice online resource allocation algorithm to enhance long-term system utility and establish that it admits a sublinear dynamic regret bound in piecewise-stationary environments. Numerical results demonstrate that the proposed method outperforms alternatives, improving system utility while satisfying slice requirements.**

***Index Terms*—Enhanced mobile broadband, network slicing, ultra-reliable low-latency communication.**

I. Introduction

ENHANCED mobile broadband (eMBB) and ultra-reliable low-latency communication (URLLC) are two pivotal service categories in 5G and beyond mobile networks. The eMBB service supports human-centric multimedia applications, such as high-definition video streaming, virtual reality, and augmented reality, which demand high data rates to provide immersive user experiences [1]. URLLC is designed to deliver extremely high reliability and low latency for time-critical services, including autonomous driving, industrial automation, and remote healthcare, ensuring real-time and error-free data transmission [2], [3].

URLLC and eMBB traffic share time–frequency resources, employing distinct transmission time intervals (TTIs) tailored to their respective requirements. The eMBB transmission typically adopts a 15 kHz subcarrier spacing, with each TTI comprising 14 OFDM symbols, referred to as a slot. Based on the flexible subcarrier spacing technique, URLLC employs larger subcarrier spacings to shorten OFDM symbols. The subcarrier spacing can take values of 15×2^n kHz, where the numerology index n is a non-negative integer typically ranging from 0 to 4. Moreover, URLLC packets use shorter TTIs consisting of 2, 4, or 7 OFDM symbols, referred to as minislots [4]. A key challenge lies in jointly scheduling eMBB and URLLC traffic with different transmission timescales to satisfy their heterogeneous requirements.

As URLLC traffic may arrive during an ongoing eMBB transmission, a feasible scheduling strategy involves preempting the eMBB transmission in the next minislot and immediately transmitting URLLC packets, a mechanism known as puncturing. In [5], the authors propose a puncturing strategy that maximizes eMBB utility while satisfying URLLC delay constraints, applicable to linear, convex, and threshold models of the relationship between eMBB rate loss and puncturing ratio. In [6], a deep learning model is employed to predict block error rate in puncturing scenarios, guiding adaptive modulation and coding adjustments to enhance eMBB link reliability. In [7], a pairing algorithm is designed for eMBB and URLLC users to make puncturing decisions, reducing URLLC packet loss while limiting eMBB rate degradation.

The aforementioned studies adopt immediate scheduling for URLLC, discarding packets that cannot be transmitted in the subsequent minislot. Introducing a queuing mechanism enables buffering of URLLC packets that cannot be served immediately, reducing packet loss, alleviating congestion caused by burst traffic, and preventing excessive puncturing from significantly degrading eMBB throughput. In [8], Bayesian optimization is used to determine puncturing and queue scheduling decisions, enhancing eMBB throughput while mitigating URLLC packet loss. A backlog-aware resource allocation algorithm for radio access networks is proposed in [9], reducing power consumption and providing reliable delay guarantees. In [10], safe reinforcement learning is employed for queue scheduling to ensure transmission reliability and improve long-term system utility.

While preemptive puncturing prioritizes URLLC to satisfy strict delay constraints, it inevitably degrades eMBB throughput and reliability. Specifically, when URLLC packets puncture an eMBB transmission, the block error rate at the eMBB receiver increases, raising the probability of decoding failure. Enhancing the quality of service (QoS) for coexisting

Received 16 September 2025; revised 7 January 2026, 27 February 2026, and 22 April 2026; accepted 31 May 2026. Date of publication 9 June 2026; date of current version 23 June 2026. The associate editor coordinating the review of this article and approving it for publication was J. B. Seo. *(Corresponding author: Shaowei Wang.)*

The authors are with the School of Electronic Science and Engineering, Nanjing University, Nanjing 210023, China (e-mail: rongchai@smail.nju.edu.cn; wangsw@nju.edu.cn).

Digital Object Identifier 10.1109/TCOMM.2026.3701861

eMBB and URLLC users has consequently attracted significant research interest. Network slicing offers a promising solution by enabling multiple subnetworks, i.e., network slices, tailored to different services to operate over a shared physical infrastructure, while slice isolation mitigates the impact of variations in one slice on others [11]. A slice-orchestration framework is proposed in [12], which jointly schedules delay-tolerant and delay-sensitive service requests, reducing data loss and response time in a 5G testbed implementation.

Joint scheduling of eMBB and URLLC slices is typically decoupled into intra-slice and inter-slice components, executed at different timescales [13]. Intra-slice scheduling allocates dedicated slice resources to user equipment (UE) within each TTI, satisfying packet-level delay constraints. In [14], Lyapunov optimization is employed to coordinate intra-slice resource allocation, stabilizing queues through negative feedback and providing packet-level delay guarantees. A joint scheduling method for multi-user downlink transmissions is proposed in [15], ensuring fairness among eMBB users while satisfying URLLC delay requirements. Inter-slice scheduling allocates resources within preset windows based on the QoS requirements of each slice, enhancing long-term system utility while ensuring fairness between slices. In [16], and [17], inter-slice resource allocation methods based on online convex optimization and stochastic network calculus are developed to improve system throughput and reduce packet loss.

Coordinated slice management is inherently shaped by several coupled factors, including the heterogeneous service requirements of eMBB and URLLC, the cross-timescale interaction between inter-slice and intra-slice decisions, and the competition for shared radio resources. These challenges become even more pronounced in practical wireless systems with non-stationary traffic and channel uncertainty. While analyzing either inter-slice or intra-slice scheduling in isolation facilitates tractable modeling, investigating hierarchical scheduling mechanisms that integrate both operations enables a more comprehensive evaluation of joint scheduling performance. In [18], the authors use stochastic network calculus to estimate delay bounds and perform hierarchical resource allocation across timescales based on arrival traffic. In [19], multi-slice scheduling with rate-splitting multiple access is formulated as a multi-objective optimization problem and solved via differential evolution to enhance worst-case eMBB throughput and reduce URLLC packet error rate. In [20], and [21], hierarchical scheduling methods based on deep reinforcement learning and game theory are proposed to optimize multi-slice deployment and resource allocation in air-ground networks, improving slice QoS and overall system throughput. In [22], a hierarchical network slicing framework is proposed, combining reinforcement learning for slice configuration with attention-based networks for user-level resource allocation, improving the fraction of users meeting performance guarantees.

Slice scheduling operates in a non-stationary environment due to time-varying traffic and the stochastic nature of wireless channels. Existing studies often assume a stationary environment with strong priors on traffic arrivals (e.g., stationary Poisson flows [8], [10]), which may result in degraded performance under non-stationary traffic or rapidly varying system conditions. Prior work has derived theoretical bounds on slice performance [17], [18]. These results generally assume stationarity and do not account for performance degradation under non-stationary conditions, such as increased delay violations and reduced throughput. The non-stationary characteristics of wireless communication scenarios complicate the provision of performance guarantees for both eMBB and URLLC slices, rendering their joint scheduling a significant challenge.

In this paper, we investigate resource allocation and traffic scheduling for eMBB and URLLC slices in non-stationary mobile networks, decoupling joint scheduling into intra-slice and inter-slice components executed at different timescales. Intra-slice schedulers allocate dedicated slice resources to UEs at each TTI, ensuring eMBB transmission rate requirements and providing ultra-reliable packet-level delay guarantees for URLLC. Based on slice demand feedback and historical utility, the inter-slice scheduler allocates resources online at a coarse timescale to optimize long-term system utility. Our contributions can be summarized as follows:

- A hierarchical scheduling strategy is proposed to support coexisting eMBB and URLLC services, enabling resource allocation and traffic management across two timescales, while guaranteeing per-slice performance, optimizing utility in non-stationary environments, and enhancing scalability.
- We introduce a queuing mechanism for URLLC and propose an algorithm to estimate the transmission rate required to satisfy delay constraints. Unlike most existing studies that assume prior traffic distributions for tractability, our algorithm relies solely on observed traffic samples, without assuming any specific distribution. Triggered by traffic variation detection, the estimation algorithm guides intra-slice scheduling to ensure ultra-reliable delay guarantees under non-stationary traffic conditions.
- Intra-slice resource allocation and traffic scheduling are formulated as a generalized assignment problem, which is NP-hard. We propose an efficient local search algorithm to solve this problem, analyze its worst-case performance, and derive the corresponding approximation ratio.
- We design an inter-slice resource allocation method that balances exploration and exploitation by selecting decisions that maximize the sum of expected utility and confidence width, improving long-term system utility. In piecewise stationary environments, dynamic regret is defined as the cumulative difference in expected utility between the per-segment optimal decision and the algorithm decision. We establish an $\mathcal{O}\left(L^{\frac{5}{6}}\ln L \cdot \Gamma^{\frac{1}{6}}\right)$ upper bound on the dynamic regret of the proposed method, where L denotes the time horizon and Γ the number of stationary segments.

The rest of the paper is organized as follows. Section II describes the system model. Section III presents the transmission-rate estimation algorithm that ensures ultra-reliable delay guarantees. Sections IV and V introduce the

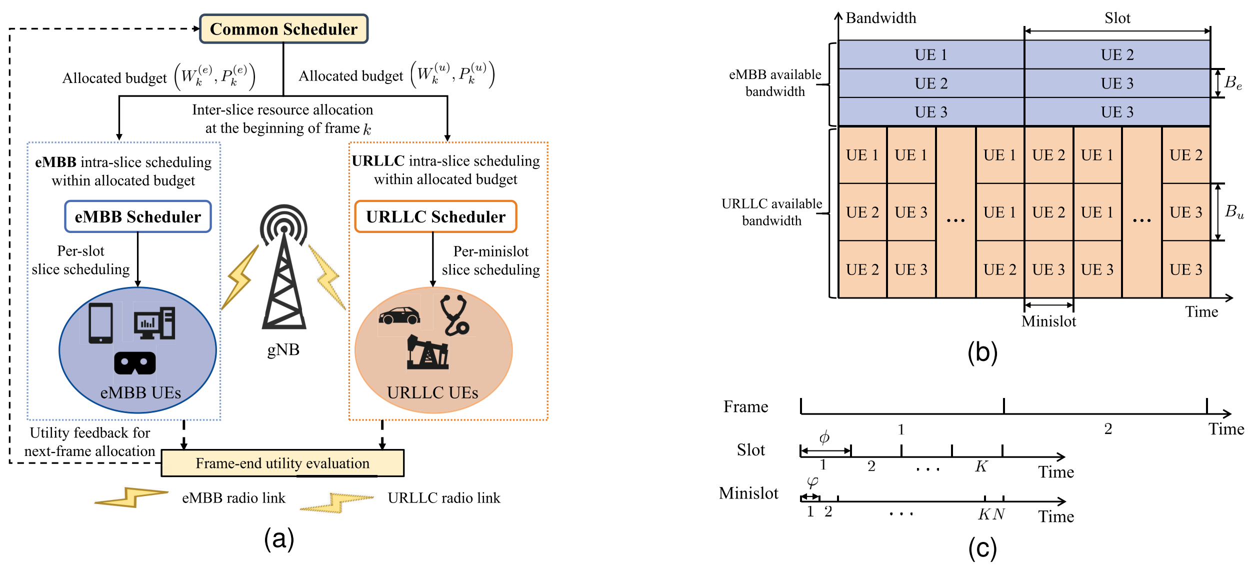

Fig. 1. System model: (a) hierarchical scheduling for eMBB and URLLC slices; (b) time–frequency resource structure; (c) timescale hierarchy.

scheduling algorithms for eMBB and URLLC slices, respectively. Section VI discusses inter-slice resource allocation. Numerical results and analysis are given in Section VII. Section VIII concludes the paper. Proofs of some results are deferred to the Appendices.

II. System Model

Consider downlink transmissions in a cellular network consisting of URLLC and eMBB slices, where $\mathcal{U}$ and $\mathcal{E}$ denote the sets of URLLC and eMBB UEs, respectively. The eMBB TTI has a duration of ϕ, referred to as a slot. URLLC traffic consists of smaller data packets, with a TTI of duration φ, referred to as a minislot. Without loss of generality, we assume that ϕ is divisible by φ,[1] i.e., there exists $N \in \mathbb{N}^+$ such that $\phi = N\varphi$.

As illustrated in Fig. 1(a), the proposed hierarchical scheduling architecture coordinates inter-slice resource allocation, intra-slice scheduling, and performance feedback over multiple timescales. Specifically, we define a frame as K consecutive slots, serving as the decision period for inter-slice resource allocation. Let W_0 denote the available spectrum bandwidth and P_0 the maximum transmit power. At the beginning of frame k, the common scheduler determines the spectrum and power budgets for the URLLC and eMBB slices based on historical slice-utility feedback, denoted by $(W_k^{(u)}, P_k^{(u)})$ and $(W_k^{(e)}, P_k^{(e)})$, respectively. Once the inter-slice resource allocation is determined, the URLLC and eMBB slices perform intra-slice scheduling independently and in parallel under their respective allocated resource budgets, with the eMBB scheduler operating at the slot timescale and the URLLC scheduler operating at the minislot timescale. At the end of frame k, the slice utilities are evaluated according to the scheduling outcomes and service performance of the two slices over the frame, and the resulting feedback is used by the common scheduler to update the inter-slice resource allocation for the next frame.

5G and beyond mobile networks enable slices to adopt flexible subcarrier spacings to accommodate their diverse requirements [4], [6]. Consequently, resource blocks (RBs) of URLLC and eMBB slices have different bandwidths, denoted by B_u and B_e, respectively, as each RB occupies 12 subcarriers in the frequency domain, with the subcarrier spacing determined by the slice numerology index. As depicted in Fig. 1(b), each slice scheduler partitions its time–frequency resources according to its respective TTI and RB bandwidth and allocates them to the intra-slice UEs. Fig. 1(c) illustrates the relative timescales of frames, slots, and minislots. The commonly used notations are summarized in Table I.

Let b_u and b_e denote the packet sizes of URLLC and eMBB slices, respectively, and m_u and m_e the corresponding numbers of RBs required to transmit a packet. An eMBB subchannel is defined as a group of m_e RBs, with the set of subchannels in frame k given by $\mathcal{M}_{k,e} = \{1, \cdots, \lfloor W_k^{(e)}/m_e B_e \rfloor\}$. Consider the downlink transmission to UE $e \in \mathcal{E}$ over subchannel $m \in \mathcal{M}_{k,e}$, with transmit power $p_{e,m,t}$ and channel gain $h_{e,m,t}$ obtained via pilot-based channel estimation. The achievable transmission rate is given by

$$\frac{b_e}{\phi} = m_e B_e \log_2(1 + \gamma_{e,m,t}), \tag{1}$$

where $\gamma_{e,m,t} = \frac{p_{e,m,t} h_{e,m,t}}{N_0 m_e B_e}$ denotes signal-to-noise ratio, and N_0 is the power spectral density of Gaussian noise. We assume that interference cancellation techniques are employed to keep inter-cell interference at a negligible level [23].

URLLC packets with short code blocks and small payloads are transmitted within a minislot to reduce latency, with their achievable rates characterized by finite blocklength channel coding theory. The achievable rate for UE $u \in \mathcal{U}$ during minislot τ over subchannel $m \in \mathcal{M}_{k,u} = \{1, \cdots, \lfloor W_k^{(u)}/m_u B_u \rfloor\}$

[1]For typical combinations of eMBB (numerology index 0 or 1) and URLLC (numerology index 2 or 3) subcarrier spacings, the slot duration can be approximated as an integer multiple of the minislot duration [4].

TABLE I
LIST OF MAIN NOTATIONS

Notations	Descriptions
$a_{u,\tau}$	Packet arrivals of UE u at minislot τ
b_u, b_e	Packet sizes of URLLC and eMBB
B_u, B_e	RB bandwidths of URLLC and eMBB
$\widehat{\mathcal{C}}$	Discrete inter-slice resource allocation decision set
d_u	Queueing delay of UE u
$\mathcal{E}$	eMBB UE set
$h_{e,m,t}$	Channel gain of UE e on subchannel m at slot t
m_u, m_e	RBs per subchannel for URLLC and eMBB
$\mathcal{M}_{k,u}, \mathcal{M}_{k,e}$	URLLC and eMBB subchannel sets in frame k
N_0	Power spectral density of Gaussian noise
$p_{e,m,t}$	Transmit power of UE e on subchannel m at slot t
P_0	Maximum transmit power
$P_k^{(u)}, P_k^{(e)}$	URLLC and eMBB power resources in frame k
$q_{u,\tau}$	Backlog of UE u at minislot τ
$s_{u,\tau}$	Packets transmitted of UE u at minislot τ
$\hat{s}_u^*$	Estimated effective bandwidth of UE u
T_d	Delay constraint
T_d'	Queueing delay constraint
$\mathcal{U}$	URLLC UE set
v_k	System utility in frame k
$v_{k,u}, v_{k,e}$	URLLC and eMBB utilities in frame k
W_0	Available spectrum bandwidth
$W_k^{(u)}, W_k^{(e)}$	URLLC and eMBB bandwidths in frame k
$\gamma_{e,m,t}$	Signal-to-noise ratio of UE e on subchannel m at slot t
Γ	Number of piecewise-stationary segments
ε_e	Transmission error probability constraint
ε_d	DVP constraint
$\zeta_k(\boldsymbol{c})$	Confidence width of decision $\boldsymbol{c}$ in frame k
$\mu_k(\boldsymbol{c})$	Expected utility of decision $\boldsymbol{c}$ in frame k
$\hat{\mu}_k(\boldsymbol{c})$	Estimated expected utility of decision $\boldsymbol{c}$ in frame k
ϕ, φ	Duration of slot and minislot

is given by

$$\frac{b_u}{\varphi} = m_u B_u \log_2(1+\gamma_{u,m,\tau}) - \sqrt{\frac{m_u B_u V_{u,m,\tau}}{\varphi}} Q^{-1}(\varepsilon_e), \tag{2}$$

where $V_{u,m,\tau} = (1-(1+\gamma_{u,m,\tau})^{-2})(\log_2 e)^2$, $Q^{-1}(x)$ is the inverse of $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{\xi^2}{2}} \mathrm{d}\xi$, and $\varepsilon_e \in (0,1)$ is the transmission error probability [24].

Arriving packets may not be transmitted immediately due to limited resources and time-varying channel conditions. For each URLLC UE u, let $q_{u,\tau}$ denote the backlog at minislot τ, $a_{u,\tau}$ the number of newly arrived packets, and $s_{u,\tau}$ the number of packets transmitted. The backlog evolves as

$$q_{u,\tau+1} = \max(q_{u,\tau} + a_{u,\tau} - s_{u,\tau}, 0), \tag{3}$$

where packets are transmitted in a first-in-first-out manner. URLLC requires that the probability of any packet experiencing latency exceeding the threshold T_d does not exceed $\varepsilon_d \in (0,1)$. Total latency comprises queuing delay and transmission delay (equal to minislot duration φ), with propagation time considered negligible. Let d_u denote the queuing delay of UE u measured in minislots, which satisfies $\mathrm{P}\{d_u > T_d'\} \le \varepsilon_d$, where $T_d' = \lfloor T_d/\varphi \rfloor - 1$.

III. Reliable Delay Guarantee

We employ the effective bandwidth theory, derived from *Large Deviation Principle* [25], [26], to characterize the constant transmission rate required to satisfy the delay violation probability (DVP) constraint of URLLC slice. Specifically, define cumulative arrival over τ slots as $A_{u,\tau} := \sum_{\tau_1=1}^{\tau} a_{u,\tau_1}$. For any $u \in \mathcal{U}$ and $\theta > 0$, assume that the following limit exists and is differentiable

$$\Lambda_u(\theta) := \lim_{\tau\to\infty} \frac{1}{\tau} \ln \mathbb{E}[e^{\theta A_{u,\tau}}].$$

For UE u assigned a constant transmission rate s_u, let θ_u denote the solution to $\Lambda_u(\theta)/\theta = s_u$. According to [27], for sufficiently large T_d', the probability that the queuing delay exceeds T_d' decays approximately as $e^{-\theta_u s_u T_d'}$. For finite T_d', a more accurate approximation of DVP is given by

$$\mathrm{P}(d_u > T_d') \approx p(s_u) e^{-\theta_u s_u T_d'}, \tag{4}$$

where $p(s_u)$ represents the probability that the queue operating under a hypothetical constant transmission rate s_u is non-empty [28], [29]. Let θ_u^* and s_u^* denote the solution pair to $p(s_u) e^{-\theta_u s_u T_d'} = \varepsilon_d$, such that the DVP of the queue with arrival process $\{a_{u,\tau}\}$ is approximately bounded by ε_d when the transmission rate is no less than s_u^*. Here, s_u^* is referred to as the effective bandwidth and serves as a lower-bound reference transmission rate required to satisfy the DVP constraint. In practice, the URLLC scheduler allocates resources to maintain the transmission rate of each UE no less than its effective bandwidth.

The stochastic and non-stationary nature of arrival traffic makes a closed-form expression for $\Lambda_u(\theta)$ generally intractable, thereby complicating the computation of effective bandwidth. We propose a traffic-sampling-based approach to estimate the effective bandwidth. From (4), the cumulative distribution function and probability density function of queuing delay can be approximated as

$$F_u(x) = 1 - \mathrm{P}(d_u > x) = 1 - p(s_u) e^{-\Theta_u x},$$
$$f_u(x) = \frac{\mathrm{d}F_u(x)}{\mathrm{d}x} = \Theta_u p(s_u) e^{-\Theta_u x},$$

where $\Theta_u = \theta_u s_u$. Consequently, the expected queueing delay is $\mathbb{E}[d_u] = \int_0^\infty x f_u(x) \mathrm{d}x = p(s_u)/\Theta_u$.

As shown in Fig. 2, we construct a virtual queue with a constant reference transmission rate s_u for UE u to enable

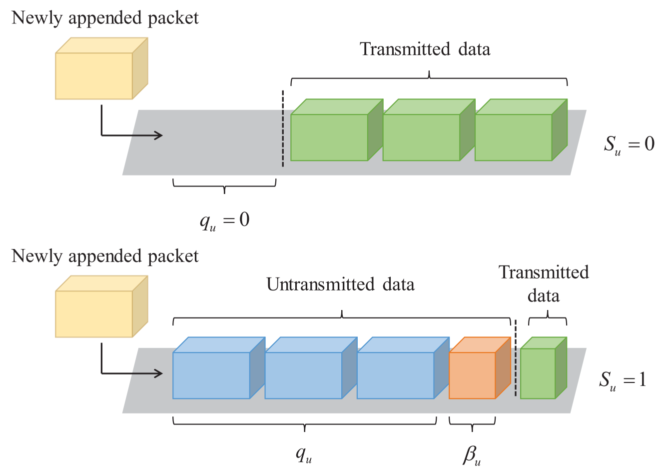

Fig. 2. Two states of virtual queue.

tractable effective bandwidth estimation. This virtual queue is an auxiliary analytical construct that characterizes the queueing behavior under a constant-rate assumption, rather than the actual physical queue whose instantaneous transmission rate is time-varying and determined by channel conditions, resource allocation, and scheduling decisions. Based on the simulated evolution of the virtual queue, we estimate the DVP under the reference transmission rate and determine whether the corresponding delay requirement can be satisfied. At an arbitrary packet arrival, let q_u denote the number of packets in the buffer, and define $S_u \in \{0,1\}$ as the queue state indicator, where $S_u = 1$ if the queue is non-empty and $S_u = 0$ otherwise. Let β_u denote the residual transmission time of the packet being serviced, conditioned on $S_u = 1$. The expected queueing delay can then be expressed as $\mathbb{E}[d_u] = \mathbb{E}[\beta_u]\mathbb{E}[S_u] + \mathbb{E}[q_u]/s_u$. Combining $\mathbb{E}[d_u] = p(s_u)/\Theta_u$, we obtain

$$\begin{aligned} p(s_u) &= \mathbb{E}[S_u], \\ \Theta_u &= \frac{s_u \mathbb{E}[S_u]}{s_u \mathbb{E}[\beta_u]\mathbb{E}[S_u] + \mathbb{E}[q_u]}. \end{aligned} \tag{5}$$

Let $\{q_{u,w}\}_{w=1}^W$ and $\{\tilde{s}_{u,w}\}_{w=1}^W$ denote the virtual queue backlog and the corresponding transmission volume over W minislots, respectively, where $\tilde{s}_{u,w} = \min(q_{u,w} + a_{u,w}, s_u)$. The sample mean backlog is $\hat{q}_u = \frac{1}{W}\sum_{w=1}^W q_{u,w}$. Since observations are made at arbitrary times, the probability that the queue is non-empty during minislot w is $\frac{\tilde{s}_{u,w}}{s_u}$, which leads to the sample mean $\widehat{S}_u = \frac{1}{W}\sum_{w=1}^W \frac{\tilde{s}_{u,w}}{s_u}$. As each packet takes $1/s_u$ to transmit, it follows that $\mathbb{E}[\beta_u] = \frac{1}{2s_u}$. Combining (5), we obtain

$$\begin{aligned} \hat{p}(s_u) &= \widehat{S}_u, \\ \widehat{\Theta}_u &= \frac{s_u \widehat{S}_u}{0.5\widehat{S}_u + \hat{q}_u}. \end{aligned} \tag{6}$$

We denote the mean and peak arrival volumes as $\overline{a}_u$ and $\tilde{a}_u$, respectively. The estimated effective bandwidth is given by

$$\hat{s}_u^* = \inf\left\{ s_u \in \left(\overline{a}_u, \tilde{a}_u\right] \middle| \hat{p}(s_u) e^{-\widehat{\Theta}_u T_d'} \le \varepsilon_d \right\}.$$

Given that DVP decreases monotonically with transmission rate, we design an efficient bisection-search algorithm to estimate the effective bandwidth. Specifically, given the arrival samples $\{a_{u,w}\}_{w=1}^W$ of UE u over W minislots, we compute its mean $\overline{a}_u = \frac{1}{W}\sum_{w=1}^W a_{u,w}$ and peak value $\tilde{a}_u = \sup\{a_{u,w}\}_{w=1}^W$. Set $\underline{s}_u = \overline{a}_u$ and $\overline{s}_u = \max(\underline{s}_u, \tilde{a}_u)$, specify the iteration tolerance ε_1 and the maximum iteration count $N_{\max}$, and initialize the virtual queue backlog as $q_u = 0$. We update the virtual queue over the W minislots using a constant transmission rate $s_u = \frac{\underline{s}_u + \overline{s}_u}{2}$ according to (3), yielding $\hat{q}_u$ and $\widehat{S}_u$, and subsequently compute $\hat{p}_u$ and $\widehat{\Theta}_u$ using (6). If $\hat{p}_u e^{-\widehat{\Theta}_u T_d'} > \varepsilon_d$, the current transmission rate is insufficient to satisfy the DVP constraint, and we set $\underline{s}_u = s_u$; otherwise, we set $\overline{s}_u = s_u$. The iteration terminates when either the maximum number of iterations is reached or the interval length satisfies $|\overline{s}_u - \underline{s}_u| \le \varepsilon_1$, and the effective bandwidth is then estimated as $\hat{s}_u^* = \lceil s_u \rceil$. The virtual queue is simulated without actual buffer allocation, recording only the backlog and idle states during the iteration. Algorithm 1 outlines the effective bandwidth estimation method we propose.

Algorithm 1 Effective Bandwidth Estimation

Input: $\{a_{u,w}\}_{w=1}^W$, $\varepsilon_1 > 0$, $N_{\max}$.

$\overline{a}_u \leftarrow \frac{1}{W}\sum_{w=1}^W a_{u,w}, \tilde{a}_u \leftarrow \sup\{a_{u,w}\}_{w=1}^W,$ $\underline{s}_u \leftarrow \overline{a}_u, \overline{s}_u \leftarrow \max(\underline{s}_u, \tilde{a}_u)$;
for $n = 1$ *to* $N_{\max}$ **do**
  $s_u \leftarrow (\underline{s}_u + \overline{s}_u)/2$;
  if $\overline{s}_u - \underline{s}_u \le \varepsilon_1$ **then**
    break;
  end
  $q_u \leftarrow 0, \hat{q}_u \leftarrow 0, \widehat{S}_u \leftarrow 0$;
  for $w = 1$ *to* W **do**
    $\tilde{s}_u \leftarrow \min(q_u + a_{u,w}, s_u)$;
    $q_u \leftarrow q_u + a_{u,w} - \tilde{s}_u$;
    $\hat{q}_u \leftarrow \hat{q}_u + q_u, \widehat{S}_u \leftarrow \widehat{S}_u + \tilde{s}_u/s_u$;
  end
  $\hat{q}_u \leftarrow \hat{q}_u/W, \widehat{S}_u \leftarrow \widehat{S}_u/W$;
  $\hat{p}_u \leftarrow \widehat{S}_u, \widehat{\Theta}_u \leftarrow s_u\widehat{S}_u/(0.5\widehat{S}_u + \hat{q}_u)$;
  if $\hat{p}_u e^{-\widehat{\Theta}_u T_d'} > \varepsilon_d$ **then**
    $\underline{s}_u \leftarrow s_u$
  else
    $\overline{s}_u \leftarrow s_u$
  end
end
return $\hat{s}_u^* \leftarrow \lceil s_u \rceil$;

The computational cost of simulating the queue over W minislots is denoted as $\mathcal{O}(W)$, with the number of iterations not exceeding $\min\left(\log_2\left(\frac{\overline{s}_u - \underline{s}_u}{\varepsilon_1}\right), N_{\max}\right)$. Accordingly, the computational complexity of Algorithm 1 is $\mathcal{O}\left(W \cdot \min\left(\log_2\left(\frac{\overline{s}_u - \underline{s}_u}{\varepsilon_1}\right), N_{\max}\right)\right)$.

A two-sided cumulative sum algorithm is employed to detect traffic variations and dynamically trigger effective bandwidth estimation, accounting for potential statistical non-stationarity in arrivals. The average arrival volume over the most recent W_2 minislots at minislot τ is given by $\overline{a}_{u,\tau} = \frac{1}{W_2}\sum_{w=\tau-W_2+1}^{\tau} a_{u,w}$. The forward and backward cumulative statistics are updated as

$$\begin{aligned} C_{u,\tau}^+ &= \max(0, C_{u,\tau-1}^+ + \overline{a}_{u,\tau} - \overline{a}_u - r), \\ C_{u,\tau}^- &= \max(0, C_{u,\tau-1}^- + \overline{a}_u - \overline{a}_{u,\tau} - r), \end{aligned}$$

with $C_{u,0}^+ = C_{u,0}^- = 0$, where $r > 0$ denotes the allowable deviation. Let $h > 0$ be the detection threshold. Once $C_{u,\tau}^+$ or $C_{u,\tau}^-$ exceeds the threshold h, both statistics are immediately reset to zero, and effective bandwidth estimation is reinitiated.

We define the estimated change point as

$$\tau_c := \min\left\{ \tau_1 > \tau_0 \middle| \begin{array}{l} C_{u,\tau}^+ > 0, \ \forall \tau \ge \tau_1 \\ \text{or} \\ C_{u,\tau}^- > 0, \ \forall \tau \ge \tau_1 \end{array} \right\}.$$

where τ_0 denotes the index of the last reset to zero before the detection time. To avoid estimation lag, only post-change samples $\{a_{u,w}\}_{w=\tau_c}^{\tau}$ are considered, with a sampling window of width W_1 satisfying $W_1 > W_2 > 0$. At minislot τ, let

$W = \tau - \tau_c + 1$ be the number of available samples. If $W \leq W_1$, all samples $\{a_{u,w}\}_{w=\tau_c}^{\tau}$ are used to estimate the mean $\overline{a}_u$ and effective bandwidth; otherwise, only the most recent W_1 samples are retained, and the estimation is suspended until a new change is detected.

The URLLC scheduler proactively estimates effective bandwidth at the beginning of each frame to mitigate the risk that slowly varying traffic fails to trigger change detection. The proposed detection mechanism enables timely updates of effective bandwidth upon traffic changes, thereby mitigating the risk of delay violations caused by estimation lag and preventing resource underutilization resulting from overly conservative estimates. When the traffic distribution remains relatively stable, frequent re-estimation is avoided, which further reduces computational overhead.

IV. URLLC Slice Scheduling

A. Problem Formulation

The transmission volume of UE u in minislot τ, denoted by $s_{u,\tau}$, is upper bounded by $\overline{s}_{u,\tau} = a_{u,\tau} + q_{u,\tau}$. According to effective bandwidth theory, the DVP remains below ε_d if the transmission volume satisfies $s_{u,\tau} \geq \underline{s}_{u,\tau} = \min(\hat{s}_u^*, \overline{s}_{u,\tau})$. Let $x_{u,m,\tau}$ be a binary variable, where $x_{u,m,\tau} = 1$ if subchannel m is allocated to UE u at minislot τ, and $x_{u,m,\tau} = 0$ otherwise. Hence, $s_{u,\tau} = \sum_{m\in\mathcal{M}_{k,u}} x_{u,m,\tau}$, where $k = \lceil \tau/KN \rceil$. Assuming a uniform maximum transmission power across all UEs within the same slice, the resource allocation and traffic scheduling problem for URLLC slice is formulated as follows:

$$\begin{aligned}
\max_{\substack{\{x_{u,m,\tau}\},\\ \{y_{u,\tau}\}}} \; & \sum_{u\in\mathcal{U}} \sum_{m\in\mathcal{M}_{k,u}} w_u x_{u,m,\tau} \\
\text{s.t.} \quad & C_1: \sum_{m\in\mathcal{M}_{k,u}} p_{u,m,\tau} x_{u,m,\tau} \leq P_{u,k} y_{u,\tau}, \;\; \forall u, \\
& C_2: \underline{s}_{u,\tau} y_{u,\tau} \leq \sum_{m\in\mathcal{M}_{k,u}} x_{u,m,\tau} \leq \overline{s}_{u,\tau} y_{u,\tau}, \;\; \forall u, \\
& C_3: \sum_{u\in\mathcal{U}} x_{u,m,\tau} \leq 1, \;\; \forall m, \\
& C_4: x_{u,m,\tau} \in \{0,1\}, \;\; \forall u, m, \\
& C_5: y_{u,\tau} \in \{0,1\}, \;\; \forall u,
\end{aligned} \tag{7}$$

where $P_{u,k} = P_k^{(u)}/|\mathcal{U}|$ denotes the maximum transmit power of UE u, and w_u represents the service priority weight for UE u. An auxiliary binary variable $y_{u,\tau}$ is introduced to ensure feasibility, indicating whether UE u is scheduled in minislot τ. C_1 restricts the total transmit power of each UE to the allocated power budget. C_2 constrains the transmission rate of UE u within the feasible range $[\underline{s}_{u,\tau}, \overline{s}_{u,\tau}]$, where the upper bound $\overline{s}_{u,\tau}$ is limited by the actual available data for transmission, and the lower bound $\underline{s}_{u,\tau}$ is derived from the estimated effective bandwidth to satisfy the DVP requirement. C_3 ensures that each subchannel is allocated to at most one UE. We aim to maximize the weighted sum throughput of UEs, subject to DVP and resource constraints. Eq. (7) formulates a generalized assignment problem, which is NP-hard and becomes computationally intractable as the problem size increases [30].

B. Solution and Performance Analysis

Let $\mathcal{I}_u$ denote the family of subchannel sets, including the empty set, that can be assigned to UE u without violating C_1 and the upper bound on the transmission volume, under a relaxation of the lower bound in C_2, i.e.,

$$\mathcal{I}_u := \left\{ \mathcal{S} \subseteq \mathcal{M}_{k,u} \;\middle|\; \begin{array}{l} \sum_{m\in\mathcal{S}} p_{u,m} x_{u,m} \leq P_{u,k}, \\ \sum_{m\in\mathcal{S}} x_{u,m} \leq \overline{s}_u \end{array} \right\}.$$

The minislot index is henceforth omitted for notational simplicity. Define a binary variable $x_{u,\mathcal{S}}$, where $x_{u,\mathcal{S}} = 1$ if all subchannels in $\mathcal{S} \in \mathcal{I}_u$ are allocated to UE u, and $x_{u,\mathcal{S}} = 0$ otherwise. Problem (7) is relaxed as follows:

$$\begin{aligned}
\max_{\{x_{u,\mathcal{S}}\}} \; & \sum_{u\in\mathcal{U}} \sum_{\mathcal{S}\in\mathcal{I}_u} \sum_{m\in\mathcal{S}} w_u x_{u,\mathcal{S}} \\
\text{s.t.} \quad & C_1': \sum_{u\in\mathcal{U}} \sum_{\mathcal{S}\in\mathcal{I}_u} \sum_{m\in\mathcal{S}} x_{u,\mathcal{S}} \leq 1, \;\; \forall m, \\
& C_2': \sum_{\mathcal{S}\in\mathcal{I}_u} x_{u,\mathcal{S}} = 1, \;\; \forall u, \\
& C_3': x_{u,\mathcal{S}} \in \{0,1\}, \;\; \forall u, \mathcal{S} \in \mathcal{I}_u.
\end{aligned} \tag{8}$$

We employ a local search method to determine the intra-slice resource allocation. Let $\widetilde{\mathcal{S}} = \{\mathcal{S}_1, \cdots, \mathcal{S}_{|\mathcal{U}|}\}$ be a feasible solution to (8), with objective value $v(\widetilde{\mathcal{S}})$ and weighted throughput of UE u denoted by $V_u(\widetilde{\mathcal{S}})$. The contribution of subchannel m to the objective under $\widetilde{\mathcal{S}}$ is denoted by $v_m(\widetilde{\mathcal{S}})$, with $v_m(\widetilde{\mathcal{S}}) = 0$ if m is not allocated to any UE.

Given the allocation decision $\widetilde{\mathcal{S}}$, we seek an alternative feasible allocation $\mathcal{S}_u'$ for UE u such that replacing $\mathcal{S}_u$ in $\widetilde{\mathcal{S}}$ with $\mathcal{S}_u'$ yields a higher objective value. A *local search* operation for UE u is defined as follows: (i) For each subchannel m, if m is allocated to some UE $u' \neq u$ in $\widetilde{\mathcal{S}}$, set $w_m(\widetilde{\mathcal{S}}) = w_{u'}$; otherwise, set $w_m(\widetilde{\mathcal{S}}) = 0$. (ii) Define the marginal utility of subchannel m as $\delta_m := w_u - w_m(\widetilde{\mathcal{S}})$. (iii) Select a subchannel set $\mathcal{S}_u'$ for UE u that maximizes the total marginal utility. The single-user allocation problem is formulated as follows:

$$\begin{aligned}
\max_{\{x_m\}} \; & \sum_{m\in\mathcal{M}_{k,u}} \delta_m x_m \\
\text{s.t.} \quad & C_1'': \sum_{m\in\mathcal{M}_{k,u}} p_{u,m} x_m \leq P_{u,k}, \\
& C_2'': \sum_{m\in\mathcal{M}_{k,u}} x_m \leq \overline{s}_u, \\
& C_3'': x_m \in \{0,1\}, \;\; \forall m.
\end{aligned} \tag{9}$$

Eq. (9) formulates a knapsack problem with a cardinality constraint, for which a fully polynomial time approximation scheme (FPTAS) has been established [31], capable of computing a ξ-approximate solution in time polynomial in both the input size and $1/(1-\xi)$ for any approximation ratio $\xi \in (0,1)$. Since the FPTAS is relatively complex, a more efficient rounding-based method is adopted to solve (9) in real-time. We first solve the linear programming (LP) relaxation of (9) to obtain a basic solution $\boldsymbol{x}^* = (x_1, \cdots, x_{|\mathcal{M}_{k,u}|})$. Let $\mathcal{J}_1$ denote the set of subchannels with $x_i = 1$, with total power and total marginal utility denoted by $p(\mathcal{J}_1)$ and $\delta(\mathcal{J}_1)$, respectively.

Lemma 1: $\boldsymbol{x}^*$ contains at most two fractional components. If $\boldsymbol{x}^*$ contains exactly two fractional components, x_i and x_j, then $x_i+x_j=1$ and $p(\mathcal{J}_1)+p_{u,i}x_i+p_{u,j}x_j=P_{u,k}$. Without loss of generality, assume that $p_{u,j}\le p_{u,i}$, then $\delta(\mathcal{J}_1)+\delta_i\ge \tilde{z}_u\ge z_u^*$, where z_u^* and $\tilde{z}_u$ denote the optimal values of problem (9) and its LP relaxation, respectively.

For brevity, the details of Lemma 1 can be found in [32]. Let $\mathcal{J}_F$ denote the set of subchannels corresponding to the fractional components of $\boldsymbol{x}^*$. If $\mathcal{J}_F=\emptyset$, set $\mathcal{S}'_u=\mathcal{J}_1$. If $\mathcal{J}_F=\{i\}$, then set $\mathcal{S}'_u=\mathcal{J}_1$ if $\delta(\mathcal{J}_1)\ge\delta_i$; otherwise, set $\mathcal{S}'_u=\{i\}$. If $\mathcal{J}_F=\{i,j\}$ and $p_{u,j}\le p_{u,i}$, then set $\mathcal{S}'_u=\mathcal{J}_1\cup\{j\}$ if $\delta(\mathcal{J}_1)+\delta_j>\delta_i$; otherwise, set $\mathcal{S}'_u=\{i\}$. The following theorem indicates that by solving the LP relaxation of problem (9) and applying a rounding procedure, the resulting marginal utility reaches at least half of the optimum.

Theorem 1: The marginal utility associated with $\mathcal{S}'_u$, denoted by z_u, satisfies $z_u\ge\frac{1}{2}z_u^*$.

Proof: When $\mathcal{J}_F=\emptyset$, $\mathcal{J}_1$ is the optimal solution to (9). When $\mathcal{J}_F=\{i\}$, we have $z_u^*\le\tilde{z}_u\le\delta(\mathcal{J}_1)+\delta_i\le 2\,z_u$. If $\mathcal{J}_F=\{i,j\}$, then $z_u^*\le\tilde{z}_u\le\delta(\mathcal{J}_1)+\delta_i+\delta_j\le 2\,z_u$, which completes the proof. ■

For each $u\in\mathcal{U}$, initialize its subchannel assignment as $\mathcal{S}_u=\emptyset$, and construct the initial resource allocation state $\widetilde{\mathcal{S}}=(\mathcal{S}_1,\cdots,\mathcal{S}_{|\mathcal{U}|})$. Assuming that problem (9) admits a ξ-approximation solution, for a given accuracy parameter $\epsilon\in(0,1)$, we define $\epsilon'=\frac{\xi+1}{\xi}\epsilon$. At each iteration, a *local search* is conducted for every UE to obtain a candidate subchannel set $\mathcal{S}'_u$, upon which the improvement $\Delta_u=\left(\sum_{m\in\mathcal{S}'_u}\delta_m\right)-V_u(\widetilde{\mathcal{S}})$ is evaluated and the UE yielding the maximum gain is selected, i.e., $u^*=\arg\max_{u\in\mathcal{U}}\Delta_u$. If $\Delta_{u^*}>0$, then $\mathcal{S}'_{u^*}$ replaces $\mathcal{S}_{u^*}$ in $\widetilde{\mathcal{S}}$. Iterate the procedure for $\lceil\frac{|\mathcal{U}|}{\xi}\ln\left(\frac{1}{\epsilon'}\right)\rceil$ rounds, and return $\widetilde{\mathcal{S}}$ as the final resource-allocation solution. Algorithm 2 outlines the proposed local search method for solving (8).

Algorithm 2 Local Search

1. $\widetilde{\mathcal{S}}=(\mathcal{S}_1,\cdots,\mathcal{S}_{|\mathcal{U}|})$, $\mathcal{S}_u\leftarrow\emptyset$, $\forall u\in\mathcal{U}$;
2. **for** $n=1$ *to* $\lceil\frac{|\mathcal{U}|}{\xi}\ln(\frac{1}{\epsilon'})\rceil$ **do**
3. For each UE u, perform *local search* to obtain $\mathcal{S}'_u$, $\Delta_u\leftarrow(\sum_{m\in\mathcal{S}'_u}\delta_m)-V_u(\widetilde{\mathcal{S}})$;
4. $u^*\leftarrow\arg\max_{u\in\mathcal{U}}\Delta_u$;
5. **if** $\Delta_{u^*}>0$ **then**
6. Replace $\mathcal{S}_{u^*}$ with $\mathcal{S}'_{u^*}$;
7. **end**
8. **end**
9. **return** $\widetilde{\mathcal{S}}$;

By employing a hash table to directly index the subchannel occupancy states, the marginal utility can be computed in $\mathcal{O}(|\mathcal{M}_{k,u}|)$ time. Problem (9) is addressed by solving its LP relaxation followed by a rounding procedure, which guarantees an approximation ratio of $\frac{1}{2}$. According to [33], the LP relaxation of the cardinality-constrained knapsack problem can be solved in $\mathcal{O}(|\mathcal{M}_{k,u}|)$ time. The computational complexity of a single iteration across all UEs is $\mathcal{O}(|\mathcal{U}||\mathcal{M}_{k,u}|)$. Combining the iteration count $\lceil 2|\mathcal{U}|\ln\left(\frac{1}{3\epsilon}\right)\rceil$, the computational complexity of Algorithm 2 is $\mathcal{O}\left(|\mathcal{U}|^2|\mathcal{M}_{k,u}|\ln\left(\frac{1}{\epsilon}\right)\right)$.

We establish a provable approximation bound showing that Algorithm 2 achieves a solution with performance close to the optimum of the NP-hard problem (8).

Theorem 2: Given $\epsilon\in\left(0,\frac{\xi}{\xi+1}\right)$, if problem (9) is solved using a ξ-approximation algorithm, then Algorithm 2 achieves an approximation ratio of at least $\frac{\xi}{\xi+1}-\epsilon$.

Proof: Please refer to Appendix A. □

Let $\mathcal{U}_1$ be the set of UEs whose transmission volumes under $\widetilde{\mathcal{S}}$ fall below $\underline{s}_u$, sorted in ascending order of their deficits. Define $\mathcal{U}_2=\mathcal{U}\setminus\mathcal{U}_1$. We traverse $\mathcal{U}_1$, attempting to compensate each UE by reallocating subchannels currently occupied by UEs in $\mathcal{U}_2$. For each $u\in\mathcal{U}_1$, we evaluate its required transmission power over candidate subchannels and prioritize those with lower power consumption for reallocation. A reallocation is performed only if the UE originally occupying the subchannel remains no less than its transmission volume lower bound after release. The procedure terminates for UE u once its transmission volume reaches $\underline{s}_u$ or no eligible subchannel remains.

V. eMBB Slice Scheduling

A. Problem Formulation

The eMBB service is characterized by high throughput requirements, with the average transmission rate of each UE $e\in\mathcal{E}$ constrained to be no less than $\underline{s}_e$. The resource allocation and traffic scheduling problem for eMBB slice is formulated as follows:

$$
\begin{aligned}
\max_{\substack{\{x_{e,m,t}\}\\ \{y_{e,t}\}}}\ & \lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\sum_{e\in\mathcal{E}}\sum_{m\in\mathcal{M}_{k,e}} w_e x_{e,m,t}\\
\text{s.t.}\ \ & C_6:\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\sum_{m\in\mathcal{M}_{k,e}} x_{e,m,t}\ge\underline{s}_e,\ \ \forall e,\\
& C_7:\sum_{m\in\mathcal{M}_{k,e}} p_{e,m,t}x_{e,m,t}\le P_{e,k}y_{e,t},\ \ \forall e,t,\\
& C_8:\sum_{e\in\mathcal{E}} x_{e,m,t}\le 1,\ \ \forall m,t,\\
& C_9:x_{e,m,t}\in\{0,1\},\ \ \forall e,m,t,\\
& C_{10}:y_{e,t}\in\{0,1\},\ \ \forall e,t,
\end{aligned}
\tag{10}
$$

where $k=\lceil t/K\rceil$, $P_{e,k}=P_k^{(e)}/|\mathcal{E}|$ represents the per-UE power budget, while w_e denotes the service priority weight for UE e. C_6 guarantees that the average transmission rate of UE e meets the minimum throughput requirement $\underline{s}_e$. C_7 bounds the total transmit power of each UE within the allocated power budget. C_8 mandates that each subchannel serves at most one UE per slot. C_9 and C_{10} enforce the binary nature of the allocation variables. The objective is to maximize the average weighted throughput of the eMBB slice, subject to transmission rate and resource constraints.

B. Solution and Performance Analysis

The transmission volume to UE e in slot t is given by $s_{e,t}=\sum_{m\in\mathcal{M}_{k,e}}x_{e,m,t}$, which is upper-bounded by

$\overline{s}_e = \lfloor \frac{W_0}{m_e B_e} \rfloor$. We introduce an auxiliary variable $z_{e,t}$, initialized as $z_{e,1} = 0$ and updated according to $z_{e,t+1} = \max(z_{e,t} + \underline{s}_e - s_{e,t}, 0)$. Define the Lyapunov function $F_t = \frac{1}{2}\sum_{e\in\mathcal{E}} z_{e,t}^2$, with its temporal difference given by $\Delta F_t = F_{t+1} - F_t$ [34]. Using the inequality $\max(x,0)^2 \le x^2$, we obtain

$$\Delta F_t \le C + \sum_{e\in\mathcal{E}} z_{e,t}(\underline{s}_e - s_{e,t}), \tag{11}$$

where $C \ge \frac{1}{2}\sum_{e\in\mathcal{E}}(\underline{s}_e^2 + \overline{s}_e^2)$ is a constant.

Multiplying the weighted throughput in Eq. (10) by a penalty factor $\Omega > 0$ and subtracting the upper bound of ΔF_t in Eq. (11), we obtain the following subproblem:

$$\begin{aligned} \max_{\substack{\{x_{e,m,t}\}\\ \{y_{e,t}\}}} \ & \sum_{e\in\mathcal{E}} \sum_{m\in\mathcal{M}_{k,e}} (\Omega w_e + z_{e,t}) x_{e,m,t} \\ \text{s.t.} \ \ & C_7, C_8, C_9, C_{10} \text{ in Eq. (10).} \end{aligned} \tag{12}$$

We employ Algorithm 2 to solve subproblem (12) to obtain per-slot resource allocation decisions. Let ξ' denote the approximation ratio achieved by Algorithm 2, which returns a solution $\{x_{e,m,t}\}, \{y_{e,t}\}$ in slot t with the corresponding weighted throughput $\nu_t = \sum_{e\in\mathcal{E}}\sum_{m\in\mathcal{M}_{k,e}} w_e x_{e,m,t}$.

We further analyze the performance gap between the online decisions obtained by solving subproblem (12) using Algorithm 2 and the per-frame optimal decisions computed with full intra-frame information. As in many online network scheduling models [35], [36], [37], we consider a scenario in which the service provider can increase link capacity by upgrading network infrastructure, assuming that the slice can accommodate user transmission rate demands scaled by a factor of $\frac{1}{\xi'}$. The K-slot planning problem for frame k is formulated as follows:

$$\begin{aligned} \max_{\substack{\{x_{e,m,t}\}\\ \{y_{e,t}\}}} \ & \frac{1}{K}\sum_{t=t_1}^{t_1+K-1} \sum_{e\in\mathcal{E}} \sum_{m\in\mathcal{M}_{k,e}} w_e x_{e,m,t} \\ \text{s.t.} \ \ & C_6' : \frac{\xi'}{K}\sum_{t=t_1}^{t_1+K-1} \sum_{m\in\mathcal{M}_{k,e}} x_{e,m,t} \ge \underline{s}_e, \ \ \forall e, \\ & C_7, C_8, C_9, C_{10} \text{ in Eq. (10),} \end{aligned} \tag{13}$$

where $t_1 = (k-1)K + 1$. Let $\{x_{e,m,t}^*\}, \{y_{e,t}^*\}$ denote the optimal solution of (13), with the corresponding objective value denoted by ν_k^*. Problem (13) involves joint optimization across all slots within a frame and requires full intra-frame information, which is typically unavailable in online settings.

We next establish that, while satisfying the average transmission rate constraints for each eMBB UE, the per-slot resource allocation strategy achieves a provable lower bound on long-term average performance.

Theorem 3: Assume that problem (13) is feasible for each frame $k \in \{1, 2, \cdots, L\}$, and that Algorithm 2, used to solve (12), achieves an approximation ratio of ξ'. The per-slot resource allocation decisions satisfy the transmission rate constraints for all UEs, and the long-term average weighted throughput satisfies

$$\lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \nu_t \ge \lim_{L\to\infty} \frac{1}{L}\sum_{k=1}^{L} \xi' \nu_k^* - \frac{CK}{\Omega}.$$

Proof: Please refer to Appendix B. □

VI. Inter-Slice Resource Allocation

In the proposed hierarchical scheduling framework, inter-slice resource allocation is performed at the beginning of each frame to determine the spectrum and power budgets of the URLLC and eMBB slices based on historical utility feedback. The two slices then carry out intra-slice scheduling independently within their respective allocated budgets during the frame, and generate performance feedback at the frame end for updating the inter-slice resource allocation in the next frame. To characterize this feedback, let $v_{k,u}$ and $v_{k,e}$ denote the utility of URLLC and eMBB slices at frame k, respectively, defined as

$$v_{k,u} := \frac{1}{\sum_{u\in\mathcal{U}} w_u} \sum_{u\in\mathcal{U}} \frac{w_u \sum_{\tau=(k-1)KN+1}^{kKN} s'_{u,\tau}}{\varepsilon \sum_{\tau=(k-1)KN+1}^{kKN} s_{u,\tau}},$$

$$v_{k,e} := \frac{1}{\sum_{e\in\mathcal{E}} w_e} \sum_{e\in\mathcal{E}} \frac{w_e \sum_{t=(k-1)K+1}^{kK} s_{e,t}}{K \underline{s}_e},$$

where $\varepsilon = (1-\varepsilon_e)(1-\varepsilon_d)$, and $s'_{u,\tau}$ denotes the number of packets successfully delivered to UE u in minislot τ that satisfy the delay constraint. Note that $v_{k,u} \ge 1$ when the URLLC slice guarantees lossless transmission within the delay constraint with probability no less than ε, and $v_{k,e} \ge 1$ when the eMBB slice satisfies its transmission rate requirement. The utility values are bounded as $v_{k,u} \in \left[0, \frac{1}{\varepsilon}\right]$ and $v_{k,e} \in [0, \infty)$. To reconcile the disparity between utility scales of two slices, we define increasing utility functions for URLLC and eMBB slices as

$$f_u(v) = \begin{cases} \frac{v}{2}, & \text{if } v \le 1, \\ \frac{v-1}{2(\varepsilon^{-1}-1)} + \frac{1}{2}, & \text{otherwise,} \end{cases} \quad f_e(v) = \frac{v}{v+1},$$

which map slice utilities to $[0,1]$, with $f_u(1) = f_e(1) = \frac{1}{2}$. The system utility at frame k is defined as $v_k := \frac{f_u(v_{k,u}) + f_e(v_{k,e})}{2}$, which satisfies $0 \le v_k \le 1$ for all k.

The inter-slice resource allocation decision at frame k is denoted by $\boldsymbol{c}_k = (c_{k,1}, c_{k,2})$, with the decision space defined as $\mathcal{C} = [0,1] \times [0,1]$. The bandwidth and transmit power allocated to URLLC slice at frame k are given by

$$W_k^{(u)} = \min\left(W_0, \max\left(c_{k,1}W_0, m_u B_u \sum_{u\in\mathcal{U}} \hat{s}_u^*\right)\right),$$

$$P_k^{(u)} = \min\left(P_0, \max\left(c_{k,2}P_0, \sum_{u\in\mathcal{U}} \overline{p}_u \hat{s}_u^*\right)\right),$$

where $\hat{s}_u^*$ and $\overline{p}_u$ denote, respectively, the latest effective bandwidth estimate and the average per-packet transmission power of UE u in the previous frame. The eMBB slice is allocated the remaining resources, i.e., $W_k^{(e)} = W_0 - W_k^{(u)}$, $P_k^{(e)} = P_0 - P_k^{(u)}$.

For each $\boldsymbol{c} \in \mathcal{C}$, let $\mathcal{D}_k(\boldsymbol{c})$ denote the probability distribution of the system utility at frame k, and let $\mu_k(\boldsymbol{c})$ denote its expectation. We make the following mild assumptions: (i) The expected utility is Lipschitz continuous; that is, there exists a constant $\ell > 0$ such that

$$|\mu_k(\boldsymbol{c}) - \mu_k(\boldsymbol{c}')| \leq \ell \cdot \|\boldsymbol{c} - \boldsymbol{c}'\|, \ \forall k, \ \forall \boldsymbol{c}, \boldsymbol{c}' \in \mathcal{C}.$$

(ii) The environment is piecewise stationary, with the utility distributions varying at most $\Gamma - 1$ times over L frames, i.e.,

$$\sum_{k=2}^{L} \mathbb{I}\left[\exists \boldsymbol{c} \in \mathcal{C}, \mathcal{D}_k(\boldsymbol{c}) \neq \mathcal{D}_{k-1}(\boldsymbol{c})\right] \leq \Gamma - 1,$$

where $\mathbb{I}[\cdot]$ is the indicator function. System utilities across frames are assumed to be approximately independent.

Let $\boldsymbol{c}_k^* \in \arg\max_{\boldsymbol{c} \in \mathcal{C}} \mu_k(\boldsymbol{c})$ denote the optimal decision at frame k. As the system environment may vary across frames, the corresponding optimal decision may also evolve over time. Dynamic regret is adopted to characterize the tracking capability of the online inter-slice resource allocation in non-stationary environments. Specifically, it quantifies the cumulative expected utility loss of the online decisions relative to the frame-wise optimal decisions over L frames, i.e.,

$$R_L := \sum_{k=1}^{L} [\mu_k(\boldsymbol{c}_k^*) - \mu_k(\boldsymbol{c}_k)].$$

At frame k, $\mu_k(\boldsymbol{c}_k^*) - \mu_k(\boldsymbol{c}_k)$ represents the expected utility gap between the online decision and the corresponding frame-wise optimal decision, and R_L accumulates this gap over time. Intuitively, a sub-linear growth of R_L with respect to L implies that the average regret R_L/L vanishes asymptotically, indicating that the long-term average performance gap between the online decisions and the frame-wise optimal decisions asymptotically disappears under a non-stationary environment.

Let $\boldsymbol{\rho} = (\rho_1, \rho_2)$ denote the discretization resolution, where $\rho_1, \rho_2 \in (0, 1]$. We discretize the decision space into

$$\widehat{\mathcal{C}} = \left\{ (\rho_1 n_1, \rho_2 n_2) \;\middle|\; \begin{array}{l} n_1 = 1, 2, \cdots, \left\lfloor \frac{1}{\rho_1} \right\rfloor, \\ n_2 = 1, 2, \cdots, \left\lfloor \frac{1}{\rho_2} \right\rfloor. \end{array} \right\} \tag{14}$$

At the beginning of frame k, the inter-slice scheduler selects a decision $\boldsymbol{c}_k \in \widehat{\mathcal{C}}$, which determines the resource split for the current frame. After the intra-slice scheduling of the URLLC and eMBB slices is completed, the resulting system utility v_k is observed at the end of frame k and is then used to update the statistics for subsequent decisions.

The expected utility of each candidate decision $\boldsymbol{c} \in \widehat{\mathcal{C}}$ at frame k is estimated using only the observations collected within the most recent W' frames. We introduce the window indicator

$$w_k(s) := \begin{cases} 1, \ s \geq k - W', \\ 0, \ s < k - W'. \end{cases} \tag{15}$$

Define $n_k(\boldsymbol{c}) := \sum_{s=1}^{k-1} w_k(s)\mathbb{I}[\boldsymbol{c}_s = \boldsymbol{c}]$ for each $\boldsymbol{c} \in \widehat{\mathcal{C}}$. The corresponding empirical expected utility is then estimated by

$$\hat{\mu}_k(\boldsymbol{c}) = \frac{\sum_{s=1}^{k-1} w_k(s)\mathbb{I}[\boldsymbol{c}_s = \boldsymbol{c}] v_s}{n_k(\boldsymbol{c})}, \tag{16}$$

where we adopt the convention that $\frac{0}{0} = 0$.

We assume that the environment is approximately stationary over the past W' frames. Since $v_k \in [0, 1]$, for any $\boldsymbol{c} \in \widehat{\mathcal{C}}$ with $n_k(\boldsymbol{c}) > 0$ and any $x > 0$, *Hoeffding's inequality* gives

$$\mathrm{P}\left(|\hat{\mu}_k(\boldsymbol{c}) - \mu_k(\boldsymbol{c})| \leq \frac{x}{n_k(\boldsymbol{c})} \right) \geq 1 - 2\exp\left(-\frac{2x^2}{n_k(\boldsymbol{c})} \right). \tag{17}$$

We define the confidence width $\zeta_k(\boldsymbol{c})$ as

$$\zeta_k(\boldsymbol{c}) = \begin{cases} \sqrt{\frac{\ln(2/\varsigma)}{2n_k(\boldsymbol{c})}}, & n_k(\boldsymbol{c}) > 0, \\ +\infty, & n_k(\boldsymbol{c}) = 0. \end{cases} \tag{18}$$

where $\varsigma \in (0, 1]$. For any $\boldsymbol{c} \in \widehat{\mathcal{C}}$ with $n_k(\boldsymbol{c}) > 0$, substituting $x = \sqrt{\frac{n_k(\boldsymbol{c}) \ln(2/\varsigma)}{2}}$ into (17) yields a confidence interval $[\hat{\mu}_k(\boldsymbol{c}) - \zeta_k(\boldsymbol{c}), \hat{\mu}_k(\boldsymbol{c}) + \zeta_k(\boldsymbol{c})]$ for $\mu_k(\boldsymbol{c})$ with confidence level $1 - \varsigma$. When $n_k(\boldsymbol{c}) = 0$, we set $\zeta_k(\boldsymbol{c}) = +\infty$ to ensure sufficient exploration of unvisited decisions.

Algorithm 3 Inter-Slice Resource Allocation

Input: Discretization resolution $\boldsymbol{\rho} = (\rho_1, \rho_2)$, window width W', confidence parameter ς.

```
Construct Ĉ as defined in (14);
Initialize μ̂_1(c) = 0 and ζ_1(c) = +∞ for each c ∈ Ĉ;
for k = 1 to L do
    Select c_k according to (19);
    Allocate bandwidth and power budgets for URLLC
      and eMBB slices according to c_k;
    Perform intra-slice scheduling for URLLC and
      eMBB slices during frame k;
    Observe the slice feedback and compute the
      resulting system utility v_k;
    Compute w_{k+1}(s) by (15) for s = 1, ..., k;
    foreach c ∈ Ĉ do
        Update n_{k+1}(c) = Σ_{s=1}^{k} w_{k+1}(s) I[c_s = c];
        Update μ̂_{k+1}(c) and ζ_{k+1}(c) by (16) and (18);
    end
end
```

Following the principle of optimism in the face of uncertainty [38], we select the resource allocation decision from $\widehat{\mathcal{C}}$ that maximizes the sum of estimated expected utility and associated confidence width, i.e.,

$$\boldsymbol{c}_k \in \arg\max_{\boldsymbol{c} \in \widehat{\mathcal{C}}} \hat{\mu}_k(\boldsymbol{c}) + \zeta_k(\boldsymbol{c}). \tag{19}$$

After $\boldsymbol{c}_k$ is selected, the corresponding bandwidth and power budgets are assigned to the URLLC and eMBB slices for the current frame according to the resource mapping defined earlier. Once the intra-slice scheduling is completed and the resulting utility v_k is obtained, the quantities $n_{k+1}(\boldsymbol{c})$, $\hat{\mu}_{k+1}(\boldsymbol{c})$, and $\zeta_{k+1}(\boldsymbol{c})$ are updated for all $\boldsymbol{c} \in \widehat{\mathcal{C}}$ using the most recent W' observations. The proposed inter-slice resource allocation method is detailed in Algorithm 3. With incremental maintenance of the sliding-window counts and empirical utility sums, each iteration requires only a linear scan over the discretized decision set, yielding an overall complexity of $\mathcal{O}(L|\widehat{\mathcal{C}}|) = \mathcal{O}\left(\frac{L}{\rho_1 \rho_2}\right)$.

TABLE II
SIMULATION PARAMETERS

Parameter	Symbol	Value
Delay constraint	T_d	0.5 ms
DVP constraint	ε_d	1×10^{-5}
Duration of slot and minislot	ϕ, φ	1 ms, 0.125 ms
Number of slots per frame	K	100
Packet sizes of eMBB and URLLC	b_e, b_u	1008 bit, 168 bit
Power spectral density of Gaussian noise	N_0	−174 dBm/Hz
RB bandwidths of eMBB and URLLC	B_e, B_u	180 kHz, 720 kHz
RBs per subchannel for eMBB and URLLC	m_e, m_u	1, 1
Transmission rate constraint	$\underline{s}_e$	10
Transmission error probability constraint	ε_e	1×10^{-5}

We will show that Algorithm 3 achieves sublinear dynamic regret, demonstrating effective long-term tracking capability in piecewise-stationary environments.

Theorem 4: The dynamic regret of Algorithm 3 with $W' = \lceil (\frac{L}{\Gamma})^{\frac{5}{6}} \rceil$, $\varsigma = L^{-1}$, and $\rho_1 = \rho_2 = \left(\frac{\Gamma}{L}\right)^{\frac{1}{6}}$ satisfies $\mathbb{E}[R_L] \leq \mathcal{O}\left(L^{\frac{5}{6}} \ln L \cdot \Gamma^{\frac{1}{6}}\right)$.

Proof: Please refer to Appendix C. □

VII. Numerical Results

Consider an urban cell with a radius of 150 m operating at a center frequency of 2.6 GHz, where $W_0 = 30$ MHz and $P_0 = 25$ dBm. Let d_i denote the distance between the base station and UE i. Path loss is modeled as $40.74\,\text{dB} + 10\, n \lg\left(\frac{d_i}{d_r}\right) + \xi_\sigma$, where $d_r = 1$ m, $n = 3.19$, and ξ_σ is a zero-mean Gaussian random variable with a standard deviation of $\sigma = 8.2$ dB, representing shadow fading [39]. Small-scale fading is modeled as Rayleigh distributed with a unit second moment. Simulation parameters are summarized in Table II. All simulations were conducted using Python 3.9 on a Windows 11 PC equipped with an Intel Core i9-13900HX processor and 16 GB of RAM. Numerical computations were performed using NumPy and SciPy, with all algorithms executed in single-threaded mode.

A. Effective Bandwidth Estimation

We evaluate the proposed effective bandwidth estimation method by considering a URLLC UE under three arrival types:

- Poisson traffic: Packet arrivals follow a Poisson process with a constant rate.
- Self-similar traffic: Aggregate traffic generated by multiple independent ON–OFF sources, where ON and OFF durations T follow a Pareto distribution $\mathrm{P}(T > t) = \left(\frac{t_{\min}}{t}\right)^{\alpha}, t \geq t_{\min}$, with $t_{\min} = 1, \alpha = 1.3$. Each source transmits at 1 packet/minislot during ON periods, yielding long-range dependent, heavy-tailed aggregate traffic. The aggregate arrival rate is linearly proportional to the total number of multiplexed sources.
- Bursty traffic: Generated by a Poisson process modulated by a two-state Markov chain, where the source alternates between low-rate and high-rate states (1 and 10 packet/minislot). The high-to-low transition probability is fixed at 0.1, while the low-to-high transition probability is explicitly adjusted to match the target aggregate arrival rate.

Transmission rate is set to the estimated effective bandwidth, and delay distributions are collected over 100,000 minislots. All results are averaged over 500 Monte Carlo runs.

Fig. 3 shows that the DVP of all three traffic types remains below ε_d as the arrival rate increases. With a fixed arrival rate of 5 packets/minislot, Fig. 4 indicates that relaxing the delay constraint results in a heavier tail in the probability mass function (PMF). The proposed algorithm demonstrates robust performance across different arrival rates and traffic types, achieving exponential decay in the delay distribution tail while consistently satisfying the DVP constraint.

The effects of allowable deviation r and detection threshold h on the traffic change detection algorithm are evaluated via grid search. Consider a URLLC UE with a piecewise-stationary Poisson traffic consisting of 10 stationary segments randomly distributed over 30,000 minislots, each with an arrival rate drawn from $[2, 5]$. Fig. 5 shows that the detection delay increases with r and h, where it is defined as the time gap between the detected and true change points. We adopt fixed parameters $r = 0.5$ and $h = 5$ in the following simulations.

Fig. 6 shows the delay distributions under the piecewise-stationary Poisson traffic. Transmission rates are dynamically adapted based on the effective bandwidths estimated by different methods, enabling a comparison of their impact on delay performance. For the proposed dynamic estimation method, we set $W_1 = 1000$ and $W_2 = 30$. We compare our proposal with periodic approaches that perform estimation every 25, 50, and 100 minislots, as well as with an oracle baseline that computes the effective bandwidth directly from the predefined Poisson parameters. Periodic approaches respond slowly to traffic variations, resulting in heavy-tailed delay distributions. The proposed dynamic estimation algorithm promptly tracks traffic variations and adjusts the transmission rate, achieving performance comparable to that of the oracle baseline while providing reliable delay guarantees for non-stationary traffic.

B. Intra-Slice Scheduling

Numerical results are presented to assess the performance of the proposed intra-slice scheduling algorithm under varying UE population sizes, with emphasis on compliance with URLLC DVP constraints and eMBB throughput requirements. Specifically, we consider coexisting URLLC and eMBB slices, each serving multiple UEs. The arrival traffic of each URLLC UE follows an independent piecewise-stationary Poisson process. For all UEs, each stationary segment spans 200 frames, and the arrival rate for each segment is independently and uniformly drawn from the interval $[2, 5]$. We vary the parameter ϵ in the proposed intra-slice scheduling algorithm to control the number of local search iterations, denoted by N_ϵ. Our proposal is compared with two heuristic methods: random allocation and greedy strategy. In random allocation, newly arrived packets are uniformly distributed across subchannels,

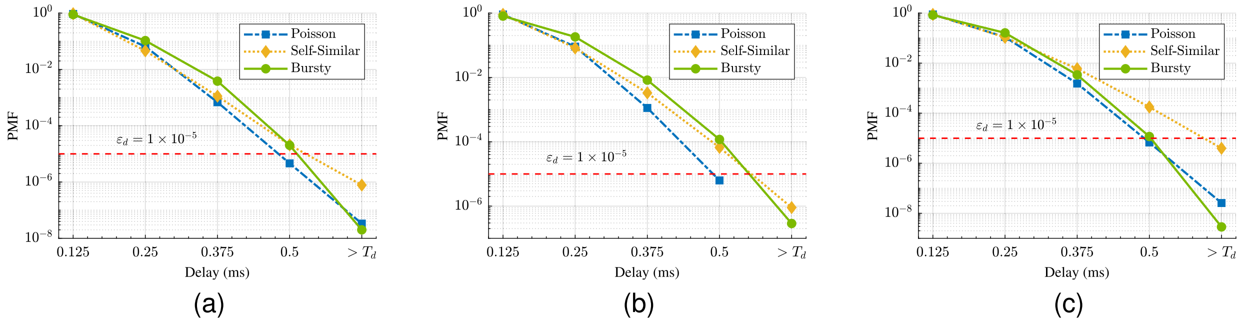

Fig. 3. Delay distributions under varying arrival rates: (a) $\bar{a} = 3$; (b) $\bar{a} = 5$; (c) $\bar{a} = 7$.

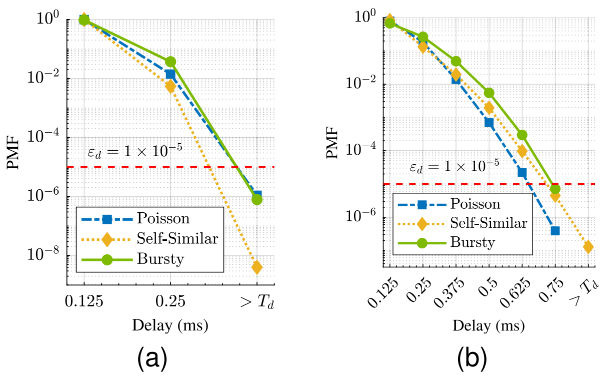

Fig. 4. Delay distributions under different delay constraints: (a) $T_d = 0.25$ ms; (b) $T_d = 0.75$ ms.

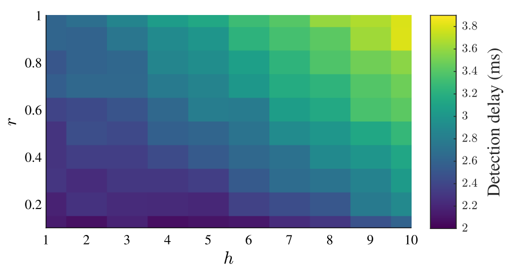

Fig. 5. Detection delay under piecewise-stationary Poisson traffic with 10 stationary segments randomly distributed over 30,000 minislots and segment-wise arrival rates uniformly drawn from $[2, 5]$.

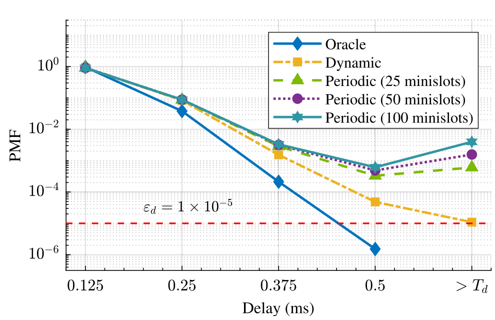

Fig. 6. Delay distributions under piecewise-stationary Poisson traffic with 10 stationary segments randomly distributed over 30,000 minislots and segment-wise arrival rates uniformly drawn from $[2, 5]$.

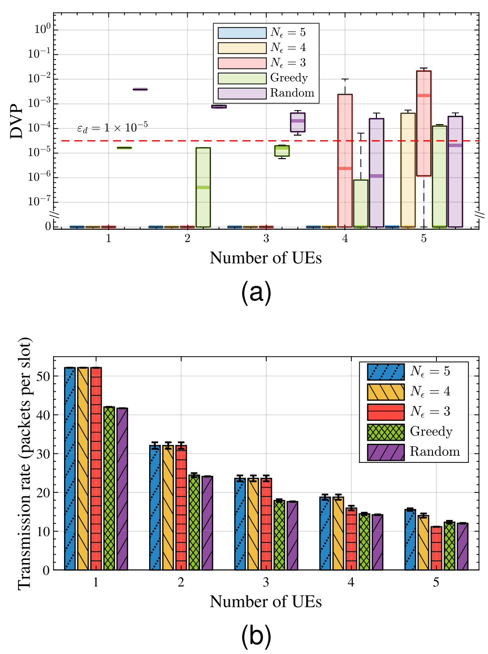

Fig. 7. Intra-slice scheduling performance versus the numbers of URLLC and eMBB UEs, under piecewise-stationary Poisson URLLC arrival processes, where each stationary segment lasts 200 frames and the segment-wise arrival rate is uniformly drawn from $[2, 5]$: (a) URLLC DVP and (b) average eMBB transmission rate.

oblivious to the channel gains [40]. The greedy strategy computes the ratio of service weight to transmission power for each UE–subchannel pair and prioritizes the pairs with the highest ratios. Inter-slice resources are fixed, with bandwidth and power equally allocated to URLLC and eMBB slices. Simulations are conducted over 2,000 frames.

As shown in Fig. 7(a), the feasible solution space of problem (7) grows exponentially with the number of UEs, resulting in higher computational complexity. Greedy and random allocation strategies cannot explore the global solution space, leading to rapid performance degradation, frequent delay violations, and failure to satisfy URLLC DVP constraints. The proposed local search algorithm performs performance-guaranteed iterative optimization, significantly reducing the DVP of all URLLC UEs as the number of iterations increases. Remarkably, only five iterations suffice to reduce the DVP of all UEs in the slice to zero, demonstrating effective resource allocation with low computational overhead.

Fig. 7(b) illustrates the variation in average transmission rate of eMBB slice with respect to the number of UEs. When

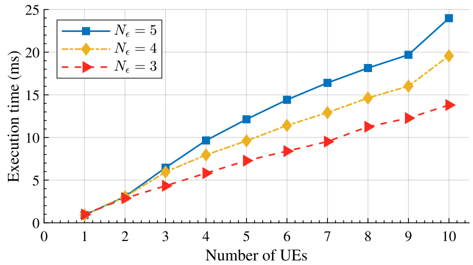

Fig. 8. Execution time of Algorithm 2 versus the number of UEs, evaluated over randomly generated scheduling instances.

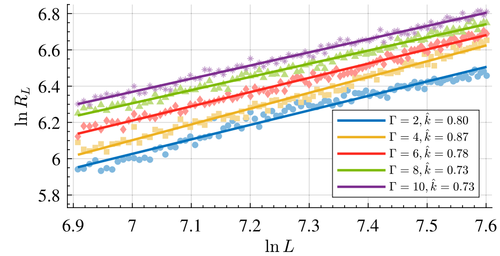

Fig. 9. Dynamic regret versus L for different Γ in an idealized piecewise-stationary utility environment.

the number of local search iterations reaches five, the proposed algorithm consistently achieves the highest average rate, outperforming the greedy and random strategies in all cases. The error bars in Fig. 7(b) represent the standard deviation of eMBB UE throughput. With five iterations of the local search, the standard deviation accounts for 2.61%, 3.19%, 3.61%, and 2.09% of the mean for UE numbers ranging from 2 to 5. The small variation in UE throughput, as indicated by the error bars, suggests that the algorithm not only enhances the average throughput but also achieves balanced resource allocation and maintains high fairness among UEs.

Fig. 8 shows how algorithm execution timescales with the number of UEs. The number of subchannels is fixed at 20, while the number of UEs is varied from 1 to 10. For each setting, 500 instances of problem (7) are randomly generated, with $w_u \in [0.1, 2]$, $\underline{s}_u \in \{0, 1, 2\}$, and $\overline{s}_u \in \{2, 3, 4\}$. The algorithm execution time is averaged over these 500 instances. The computational complexity of the proposed local search algorithm is $\mathcal{O}\left(|\mathcal{U}|^2 |\mathcal{M}_{k,u}| \ln\left(\frac{1}{\epsilon}\right)\right)$, which scales quadratically with the number of UEs. As the number of UEs increases to 10, the execution time remains low and exhibits smooth growth, indicating that the proposed local search algorithm has controllable computational overhead and is suitable for real-time decision-making. To facilitate realistic deployments, the algorithm can be optimized by transitioning to compiled languages and leveraging the architectural advantages of specialized digital signal processors, such as single-cycle multiply-accumulate operations and hardware-managed loop control, to meet the stringent timing requirements of the 5G physical layer.

C. Inter-Slice Scheduling

The common scheduler performs online inter-slice resource allocation, with each decision affecting the transmission of the subsequent frame. The mapping from allocation decisions to the resulting utility is complex and analytically intractable, rendering inter-slice resource allocation a sequential black-box optimization in a non-stationary environment. Algorithm 3 is compared against the following sequential model-based optimization methods:

- Tree-structured Parzen estimator (TPE): TPE partitions historical observations into high- and low-performing groups based on specified utility quantiles, models the conditional probability density of each group via kernel density estimation, and selects the next decision by maximizing their density ratio [41]. In partitioning historical samples, the top 25% of utility values are designated high-performing and the remainder low-performing, with Gaussian kernels employed for density estimation.
- Gaussian process-based Bayesian optimization (GPBO): GPBO uses Gaussian process regression [42] as a surrogate model to predict the expected utility of resource allocation decisions, along with the associated uncertainty. It has been applied to joint resource allocation for eMBB and URLLC in [8]. Following the setup in [8], we adopt four sampling criteria to guide the next decision: expected improvement (EI), probability of improvement (PI), upper confidence bound (UCB), and a greedy strategy with an exploration probability of 0.1.

All alternative methods use the last 100 frames of decision–utility history to guide resource allocation in the subsequent frame.

We evaluate the sublinear growth of regret for Algorithm 3 when the number of stationary segments is precisely known. Specifically, consider the expected utility function $\mu(\boldsymbol{c}) = \max\left[0, 1 - \left(\frac{\|\boldsymbol{c}-\boldsymbol{c}_0\|_2}{r_0}\right)^p\right]$, $p > 1$, which attains its maximum at $\boldsymbol{c}_0$ and is Lipschitz continuous. We set $r_0 = 1$ and $p = 1.5$, with L ranging from 1,000 to 2,000 in increments of 10. The L frames are divided into Γ segments, with the optimum fixed within each segment and randomly varying across segments to emulate a piecewise-stationary environment. All results are averaged over 500 Monte Carlo runs.

Fig. 9 depicts the regret R_L of Algorithm 3 for varying total frames and different numbers of stationary segments, with hyperparameters set according to Theorem 4. Both L and R_L are shown on a logarithmic scale as scatter points, with a least-squares line fitted to the data. The fitted slopes $\hat{k}$ are all below 1, confirming the sublinear regret of the proposed algorithm in piecewise-stationary environments. We further evaluate the performance of the proposed algorithm when the number of stationary segments is unknown. For all subsequent simulations, the hyperparameters of Algorithm 3 are set to $W' = 100$, $\boldsymbol{\rho} = (0.15, 0.15)$, and $\varsigma = 0.2$. Fig. 10 depicts the dynamic regret of all methods for $\Gamma = 4$ and $L = 2{,}000$, showing that Algorithm 3 attains the lowest dynamic regret compared with the alternative methods, further demonstrating its superior performance in non-stationary environments.

We examine how system utility and service performance vary with the number of URLLC UEs when the number of

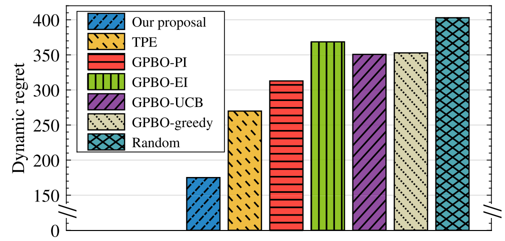

Fig. 10. Dynamic regret comparison in an idealized piecewise-stationary utility environment with $L = 2{,}000$ and $\Gamma = 4$.

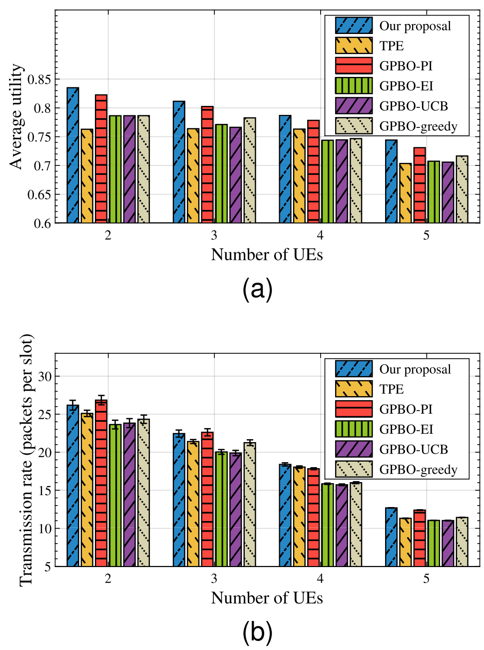

Fig. 11. Inter-slice scheduling performance versus the number of URLLC UEs with the number of eMBB UEs fixed at 3, under piecewise-stationary Poisson URLLC arrival processes, where each stationary segment lasts 200 frames and the segment-wise arrival rate is uniformly drawn from $[2, 5]$: (a) system utility; (b) average eMBB transmission rate.

eMBB UEs is fixed at 3. Both eMBB and URLLC intra-slice scheduling employ Algorithm 2 with $\epsilon = 0.01$. URLLC traffic arrivals follow a piecewise-stationary Poisson process, where each segment lasts 200 frames and has a constant arrival rate uniformly drawn from the interval $[2, 5]$. Fig. 11(a) shows that the average system utility decreases as the number of URLLC UEs increases, reflecting performance loss due to intensified resource competition. Nevertheless, the proposed algorithm consistently outperforms the alternative methods across the entire URLLC traffic range, maintaining high utility.

Fig. 11(b) shows that the average eMBB transmission rate decreases with the number of URLLC UEs but remains above the threshold $\underline{s}_e$. Meanwhile, across all experiments, the URLLC DVP under the proposed algorithm remains below 10^{-5}, owing to effective bandwidth estimation and dynamic scheduling. Numerical results indicate that the proposed algorithm effectively preserves slice isolation and satisfies eMBB transmission rate requirement while ensuring URLLC latency guarantee, achieving efficient scheduling and resource sharing for both service types.

VIII. Conclusion

We proposed a hierarchical scheduling approach for eMBB and URLLC slices in non-stationary mobile networks. To address non-stationary URLLC traffic, we develop an effective bandwidth estimation algorithm that is activated by traffic change detection and accordingly updates the transmission rate to ensure ultra-reliable delay guarantees. We design a local search algorithm for intra-slice resource allocation and traffic scheduling and establish its provable approximation ratio. For inter-slice resource allocation, we propose an online strategy and establish a sublinear dynamic regret bound in piecewise stationary environments. Numerical results show that the proposed approach outperforms existing methods, highlighting its potential for efficient multi-slice management. Specifically, the proposed method maintains URLLC DVP below 10^{-5} while satisfying the eMBB slice throughput requirements with low variability, providing concrete evidence of its stable and reliable performance. In the future, we intend to explore joint multi-slice management under continuously evolving non-stationary conditions and the coordinated scheduling of radio access and core networks, aiming to provide end-to-end guarantees for URLLC latency and eMBB throughput.

Appendix A
Proof of Theorem 2

Let $\widetilde{\mathcal{S}}^*$ denote the optimal solution to problem (8). Define the sets

$$\mathcal{M}^{(1)} := \left\{ m \in \mathcal{M}_{k,u} \mid v_m(\widetilde{\mathcal{S}}^*) > v_m(\widetilde{\mathcal{S}}) \right\},$$
$$\mathcal{M}^{(2)} := \mathcal{M}_{k,u} \setminus \mathcal{M}^{(1)}.$$

Given a subset $\mathcal{M}' \subseteq \mathcal{M}_{k,u}$, define $f(\mathcal{M}')$ and $g(\mathcal{M}')$ as the total utility of $\mathcal{M}'$ under $\widetilde{\mathcal{S}}^*$ and $\widetilde{\mathcal{S}}$, respectively. It follows that $v(\widetilde{\mathcal{S}}^*) = f(\mathcal{M}^{(1)}) + f(\mathcal{M}^{(2)})$ and $v(\widetilde{\mathcal{S}}) = g(\mathcal{M}^{(1)}) + g(\mathcal{M}^{(2)})$.

Let $\mathcal{M}_u$ denote the set of subchannels in $\mathcal{M}^{(1)}$ that are allocated to UE u under $\widetilde{\mathcal{S}}^*$. For each $m \in \mathcal{M}_u$, since $w_u - w_m(\widetilde{\mathcal{S}}) \geq w_u - v_m(\widetilde{\mathcal{S}})$, the total marginal utility of $\mathcal{M}_u$ with respect to UE u is at least $\sum_{m\in\mathcal{M}_u}(w_u - v_m(\widetilde{\mathcal{S}})) = f(\mathcal{M}_u) - g(\mathcal{M}_u)$. By applying a ξ-approximation algorithm to problem (9), we obtain $\sum_{u\in\mathcal{U}}\sum_{m\in\mathcal{S}'_u}\delta_m \geq \xi\sum_{u\in\mathcal{U}}(f(\mathcal{M}_u) - g(\mathcal{M}_u)) = \xi(f(\mathcal{M}^{(1)}) - g(\mathcal{M}^{(1)}))$. In light of $f(\mathcal{M}^{(2)}) \leq g(\mathcal{M}^{(2)})$, it follows that $\sum_{u\in\mathcal{U}}\sum_{m\in\mathcal{S}'_u}\delta_m \geq \xi(f(\mathcal{M}^{(1)}) - g(\mathcal{M}^{(1)}) + f(\mathcal{M}^{(2)}) - g(\mathcal{M}^{(2)})) = \xi(v(\widetilde{\mathcal{S}}^*) - v(\widetilde{\mathcal{S}}))$. Accordingly,

$$\begin{aligned}\sum_{u\in\mathcal{U}}\Delta_u &= \sum_{u\in\mathcal{U}}\left(\sum_{m\in\mathcal{S}'_u}\delta_m - V_u(\widetilde{\mathcal{S}})\right)\\ &\geq \xi\left(v(\widetilde{\mathcal{S}}^*) - v(\widetilde{\mathcal{S}})\right) - v(\widetilde{\mathcal{S}})\\ &= \xi v(\widetilde{\mathcal{S}}^*) - (1+\xi)v(\widetilde{\mathcal{S}}).\end{aligned}$$

It then follows that $\Delta_{u^*} \geq \frac{\xi}{|\mathcal{U}|}v(\widetilde{\mathcal{S}}^*) - \frac{1+\xi}{|\mathcal{U}|}v(\widetilde{\mathcal{S}})$. Let $\widetilde{\mathcal{S}}'$ denote the allocation obtained by replacing $\mathcal{S}_{u^*}$ with $\mathcal{S}'_{u^*}$ in $\widetilde{\mathcal{S}}$, and we have

$$v(\widetilde{\mathcal{S}}') = v(\widetilde{\mathcal{S}}) + \Delta_{u^*}$$

$$\geq v(\widetilde{\mathcal{S}}) + \frac{\xi}{|\mathcal{U}|} v(\widetilde{\mathcal{S}}^*) - \frac{1+\xi}{|\mathcal{U}|} v(\widetilde{\mathcal{S}})$$
$$= \left(1 - \frac{1+\xi}{|\mathcal{U}|}\right) v(\widetilde{\mathcal{S}}) + \frac{\xi}{|\mathcal{U}|} v(\widetilde{\mathcal{S}}^*).$$

Denote by $v^{(n)}$ the total utility after the n-th iteration, and note that $v^{(n+1)} \geq \left(1 - \frac{1+\xi}{|\mathcal{U}|}\right) v^{(n)} + \frac{\xi}{|\mathcal{U}|} v(\widetilde{\mathcal{S}}^*)$, where $v^{(0)} = 0$. One can show by induction that $v^{(n)} \geq \frac{\xi}{1+\xi}\left(1 - \left(1 - \frac{1+\xi}{|\mathcal{U}|}\right)^n\right) v(\widetilde{\mathcal{S}}^*)$. With $n \geq \frac{|\mathcal{U}|}{1+\xi} \ln\left(\frac{1}{\epsilon'}\right)$, we have $v^{(n)} \geq \frac{\xi}{1+\xi}(1 - e^{\ln \epsilon'}) v(\widetilde{\mathcal{S}}^*) = \frac{\xi}{1+\xi}(1-\epsilon') v(\widetilde{\mathcal{S}}^*)$. Setting $\epsilon' = \frac{\xi+1}{\xi}\epsilon$, we obtain

$$v^{(n)} \geq \left(\frac{\xi}{\xi+1} - \epsilon\right) v(\widetilde{\mathcal{S}}^*),$$

which completes the proof.

Appendix B
Proof of Theorem 3

Let $s^*_{e,t} = \sum_{m \in \mathcal{M}_{k,e}} x^*_{e,m,t}$. It follows from (11) that

$$\Omega\nu_t - \Delta F_t \geq \Omega\nu_t - C - \sum_{e\in\mathcal{E}} z_{e,t}(\underline{s}_e - s_{e,t})$$
$$\geq \sum_{e\in\mathcal{E}} \left(\xi'\Omega w_e s^*_{e,t} - z_{e,t}(\underline{s}_e - \xi' s^*_{e,t})\right) - C. \quad (20)$$

For each $t \in \{t_1, \cdots, t_1+K-1\}$, let $z_{e,t}\left(\underline{s}_e - \xi' s^*_{e,t}\right) = z_{e,t_1}(\underline{s}_e - \xi' s^*_{e,t}) + (z_{e,t} - z_{e,t_1})(\underline{s}_e - \xi' s^*_{e,t})$. Noting that

$$|z_{e,t} - z_{e,t_1}| \leq (t - t_1)\max(|\underline{s}_e - \overline{s}_e|, \underline{s}_e),$$
$$|\underline{s}_e - \xi' s^*_{e,t}| \leq \max(|\underline{s}_e - \overline{s}_e|, \underline{s}_e),$$

we have $(z_{e,t} - z_{e,t_1})(\underline{s}_e - \xi' s^*_{e,t}) \leq (t-t_1)\max(|\underline{s}_e - \overline{s}_e|, \underline{s}_e)^2$. Since $\sum_{e\in\mathcal{E}} \max(|\underline{s}_e - \overline{s}_e|, \underline{s}_e)^2 \leq 2C$, combining with (20), we obtain

$$\Omega\nu_t - \Delta F_t \geq \sum_{e\in\mathcal{E}} \left(\xi'\Omega w_e s^*_{e,t} - z_{e,t_1}(\underline{s}_e - \xi' s^*_{e,t})\right)$$
$$- 2C(t - t_1) - C. \quad (21)$$

Summing (21) over $t \in \{t_1, \cdots, t_1+K-1\}$ yields

$$\Omega \sum_{t=t_1}^{t_1+K-1} \nu_t - (F_{t_1+K} - F_{t_1})$$
$$\geq \xi'\Omega K \nu^*_k - CK^2 - \sum_{e\in\mathcal{E}} z_{e,t_1} \sum_{t=t_1}^{t_1+K-1} (\underline{s}_e - \xi' s^*_{e,t})$$
$$\geq \xi'\Omega K \nu^*_k - CK^2, \quad (22)$$

where the final inequality follows from C'_6 in Eq. (13). Summing (22) over $k \in \{1, 2, \cdots, L\}$, it follows that

$$\Omega \sum_{t=1}^{LK} \nu_t - F_{LK+1} \geq \xi'\Omega K \sum_{k=1}^{L} \nu^*_k - CK^2 L. \quad (23)$$

Dividing both sides of (23) by ΩKL and noting that $F_{LK+1} \geq 0$, we obtain

$$\frac{1}{LK}\sum_{t=1}^{LK} \nu_t \geq \frac{1}{L}\sum_{k=1}^{L} \xi'\nu^*_k - \frac{CK}{\Omega}.$$

Let $T = KL$ and take the limit as $L \to \infty$, yielding

$$\lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \nu_t \geq \lim_{L\to\infty} \frac{1}{L}\sum_{k=1}^{L} \xi'\nu^*_k - \frac{CK}{\Omega}.$$

It can be shown from (23) that there exists a constant $C' > 0$ such that $F_{T+1} = \frac{1}{2}\sum_{e\in\mathcal{E}} z^2_{e,T+1} \leq C'T$. For any $e \in \mathcal{E}$, we have $z_{e,T+1} \leq \sqrt{2C'T}$, which implies

$$\lim_{T\to\infty} \frac{1}{T} z_{e,T+1} \leq \lim_{T\to\infty} \sqrt{\frac{2C'}{T}} = 0.$$

Using the fact that $z_{e,T+1} \geq \sum_{t=1}^{T}(\underline{s}_e - s_{e,t})$, we have

$$\lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T}(\underline{s}_e - s_{e,t}) \leq 0,$$

which implies that the transmission rate constraint is satisfied.

Appendix C
Proof of Theorem 4

We divide the L frames into Γ epochs such that in each epoch the utility distribution is stationary

$$[1, L] = [\iota_1, \iota_2) \cup [\iota_2, \iota_3) \cup \cdots \cup [\iota_\Gamma, L+1).$$

where ι_i denotes the i-th change point of the utility distributions, i.e., $\exists \boldsymbol{c} \in \mathcal{C}, \mathcal{D}_{\iota_i}(\boldsymbol{c}) \neq \mathcal{D}_{\iota_{i-1}}(\boldsymbol{c})$. We define $\iota_{\Gamma+1} = L+1$ for notational convenience. For epoch $[\iota_i, \iota_{i+1})$, let $\widehat{\mathcal{C}}_i = \{\boldsymbol{c} \in \widehat{\mathcal{C}} | \mu_{\iota_i}(\boldsymbol{c}) = \max_{\boldsymbol{c}'\in\widehat{\mathcal{C}}} \mu_{\iota_i}(\boldsymbol{c}')\}$. For each $\boldsymbol{c} \in \widehat{\mathcal{C}}$, define $\Delta_i(\boldsymbol{c}) = \mu_{\iota_i}(\tilde{\boldsymbol{c}}_i) - \mu_{\iota_i}(\boldsymbol{c})$, where $\tilde{\boldsymbol{c}}_i \in \widehat{\mathcal{C}}_i$. Define $\widehat{\mathcal{C}}_{i,\rho} = \{\boldsymbol{c} \in \widehat{\mathcal{C}} | \Delta_i(\boldsymbol{c}) > \rho\}$, where $\rho = \sqrt{\rho_1^2 + \rho_2^2}$. We have

$$\sum_{k=\iota_i}^{\iota_{i+1}-1} \mu_k(\tilde{\boldsymbol{c}}_i) - \mu_k(\boldsymbol{c}_k) = \sum_{k=\iota_i}^{\iota_{i+1}-1} \Delta_i(\boldsymbol{c}_k)$$
$$\leq \sqrt{2}\ell W' + \sum_{k=\iota_i+W'}^{\iota_{i+1}-1} \Delta_i(\boldsymbol{c}_k)$$
$$\leq \sqrt{2}\ell W' + (\iota_{i+1} - \iota_i)\rho + \sum_{\boldsymbol{c}\in\widehat{\mathcal{C}}_{i,\rho}} \Delta_i(\boldsymbol{c}) \sum_{k=\iota_i+W'}^{\iota_{i+1}-1} \mathbb{I}[\boldsymbol{c}_k = \boldsymbol{c}]$$
$$\leq \sqrt{2}\ell W' + (\iota_{i+1} - \iota_i)\rho + \sqrt{2}\ell \sum_{\boldsymbol{c}\in\widehat{\mathcal{C}}_{i,\rho}} \sum_{k=\iota_i+W'}^{\iota_{i+1}-1} \mathbb{I}[\boldsymbol{c}_k = \boldsymbol{c}]. \quad (24)$$

For each $k \in [\iota_i, \iota_{i+1})$, we assume that the inequality $\Delta_i(\boldsymbol{c}) \leq 2\zeta_k(\boldsymbol{c})$ holds. From (18), we obtain $\Delta_i(\boldsymbol{c})^2 \leq \frac{2\ln(2/\varsigma)}{n_k(\boldsymbol{c})}$. Define $f_i(\boldsymbol{c}) = 2\ln\left(\frac{2}{\varsigma}\right)\Delta_i^{-2}(\boldsymbol{c})$, and we have $n_k(\boldsymbol{c}) \leq f_i(\boldsymbol{c})$.

Note that $\mathbb{I}[\boldsymbol{c}_k = \boldsymbol{c}] = \mathbb{I}[\boldsymbol{c}_k = \boldsymbol{c}, n_k(\boldsymbol{c}) \leq f_i(\boldsymbol{c})] + \mathbb{I}[\boldsymbol{c}_k = \boldsymbol{c}, \Delta_i(\boldsymbol{c}) > 2\zeta_k(\boldsymbol{c})]$. For $\mathbb{I}[\boldsymbol{c}_k = \boldsymbol{c}, n_k(\boldsymbol{c}) \leq f_i(\boldsymbol{c})]$, we have

$$\sum_{k=\iota_i+W'}^{\iota_{i+1}-1} \mathbb{I}[\boldsymbol{c}_k = \boldsymbol{c}, n_k(\boldsymbol{c}) \leq f_i(\boldsymbol{c})]$$
$$\leq \left(\sum_{k=\iota_i+W'}^{\iota_{i+1}-1} \mathbb{I}[\boldsymbol{c}_k = \boldsymbol{c}]\right)$$

$$\times \mathbb{I}[\,\forall k \in [\iota_i + W', \iota_{i+1}), n_k(\boldsymbol{c}) \le f_i(\boldsymbol{c})]$$
$$\le \left\lfloor \frac{\iota_{i+1} - \iota_i}{W'} \right\rfloor f_i(\boldsymbol{c}). \tag{25}$$

For $\mathbb{I}[\boldsymbol{c}_k = \boldsymbol{c}, \Delta_i(\boldsymbol{c}) > 2\zeta_k(\boldsymbol{c})]$, we have

$$\begin{aligned}
&\mathbb{E}\left[\mathbb{I}[\boldsymbol{c}_k = \boldsymbol{c}, \Delta_i(\boldsymbol{c}) > 2\zeta_k(\boldsymbol{c})]\right] \\
&= \mathrm{P}\left(\boldsymbol{c}_k = \boldsymbol{c}, \Delta_i(\boldsymbol{c}) > 2\zeta_k(\boldsymbol{c})\right) \\
&= \mathrm{P}\begin{pmatrix} \hat{\mu}_k(\boldsymbol{c}) + \zeta_k(\boldsymbol{c}) \ge \hat{\mu}_k(\tilde{\boldsymbol{c}}_i) + \zeta_k(\tilde{\boldsymbol{c}}_i), \\ \mu_k(\tilde{\boldsymbol{c}}_i) - \mu_k(\boldsymbol{c}) > 2\zeta_k(\boldsymbol{c}) \end{pmatrix} \\
&\le \mathrm{P}\left(\hat{\mu}_k(\boldsymbol{c}) > \mu_k(\boldsymbol{c}) + \zeta_k(\boldsymbol{c})\right) \\
&+ \mathrm{P}\left(\hat{\mu}_k(\tilde{\boldsymbol{c}}_i) < \mu_k(\tilde{\boldsymbol{c}}_i) - \zeta_k(\tilde{\boldsymbol{c}}_i)\right) \\
&\le \frac{\varsigma}{2} + \frac{\varsigma}{2} = \varsigma.
\end{aligned} \tag{26}$$

Based on (24), (25), and (26), we obtain

$$\begin{aligned}
&\mathbb{E}\left[\sum_{k=\iota_i}^{\iota_{i+1}-1} \mu_k(\tilde{\boldsymbol{c}}_i) - \mu_k(\boldsymbol{c}_k)\right] \\
&\le \sqrt{2}\ell W' + (\iota_{i+1} - \iota_i)\rho \\
&\quad + \sqrt{2}\ell(\iota_{i+1} - \iota_i) \sum_{\boldsymbol{c} \in \widehat{\mathcal{C}}_{i,\rho}} \left(\frac{2\ln(2/\varsigma)}{W'\rho^2} + \varsigma\right).
\end{aligned} \tag{27}$$

For each $k \in [\iota_i + W', \iota_{i+1})$, let $\boldsymbol{c}_k^*$ be the optimal decision in frame k. According to (14), there exists $\boldsymbol{c}_k' \in \widehat{\mathcal{C}}$ such that $\|\boldsymbol{c}_k^* - \boldsymbol{c}_k'\| \le \rho$, it follows that

$$\begin{aligned}
&\mathbb{E}\left[\sum_{k=\iota_i}^{\iota_{i+1}-1} \mu_k(\boldsymbol{c}_k^*) - \mu_k(\tilde{\boldsymbol{c}}_i)\right] \\
&\le \mathbb{E}\left[\sum_{k=\iota_i}^{\iota_{i+1}-1} \mu_k(\boldsymbol{c}_k^*) - \mu_k(\boldsymbol{c}_k')\right] \le (\iota_{i+1} - \iota_i)\ell\rho.
\end{aligned} \tag{28}$$

By summing (27) and (28) over Γ epochs, we have

$$\begin{aligned}
&\mathbb{E}[R_L] \le \sqrt{2}\ell W'\Gamma + (\ell + 1)L\rho \\
&+ \sqrt{2}\ell \sum_{i=1}^{\Gamma} (\iota_{i+1} - \iota_i) \sum_{\boldsymbol{c} \in \widehat{\mathcal{C}}_{i,\rho}} \left(\frac{2\ln(2/\varsigma)}{W'\rho^2} + \varsigma\right).
\end{aligned}$$

Considering that the number of elements in $\widehat{\mathcal{C}}_{i,\rho}$ is less than $\frac{1}{\rho_1\rho_2}$, and the inequality $\rho^2 = \rho_1^2 + \rho_2^2 \ge 2\rho_1\rho_2$, we obtain

$$\begin{aligned}
&\mathbb{E}[R_L] \le \sqrt{2}\ell W'\Gamma + (\ell + 1)L\rho \\
&+ \sqrt{2}\ell L \left(\frac{\ln(2/\varsigma)}{W'\rho_1^2\rho_2^2} + \frac{\varsigma}{\rho_1\rho_2}\right).
\end{aligned}$$

Set $W' = \lceil \left(\frac{L}{\Gamma}\right)^{\frac{5}{6}} \rceil$, $\varsigma = L^{-1}$, and $\rho_1 = \rho_2 = \left(\frac{\Gamma}{L}\right)^{\frac{1}{6}}$. The dynamic regret of Algorithm 3 satisfies $\mathbb{E}[R_L] \le \mathcal{O}\left(L^{\frac{5}{6}} \ln L \cdot \Gamma^{\frac{1}{6}}\right)$.

References

[1] N. Varsier, L.-A. Dufrène, M. Dumay, Q. Lampin, and J. Schwoerer, "A 5G new radio for balanced and mixed IoT use cases: Challenges and key enablers in FR1 band," *IEEE Commun. Mag.*, vol. 59, no. 4, pp. 82–87, Apr. 2021.

[2] C. She, C. Yang, and T. Q. S. Quek, "Radio resource management for ultra-reliable and low-latency communications," *IEEE Commun. Mag.*, vol. 55, no. 6, pp. 72–78, Jun. 2017.

[3] H. Yang, K. Zheng, K. Zhang, J. Mei, and Y. Qian, "Ultra-reliable and low-latency communications for connected vehicles: Challenges and solutions," *IEEE Netw.*, vol. 34, no. 3, pp. 92–100, May 2020.

[4] H. Ji, S. Park, J. Yeo, Y. Kim, J. Lee, and B. Shim, "Ultra-reliable and low-latency communications in 5G downlink: Physical layer aspects," *IEEE Wireless Commun.*, vol. 25, no. 3, pp. 124–130, Jun. 2018.

[5] 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.

[6] Y. Huang, Y. T. Hou, and W. Lou, "A deep-learning-based link adaptation design for eMBB/URLLC multiplexing in 5G NR," in *Proc. IEEE Conf. Comput. Commun.*, May 2021, pp. 1–10.

[7] M. Almekhlafi, M. A. Arfaoui, C. Assi, and A. Ghrayeb, "Superposition-based URLLC traffic scheduling in 5G and beyond wireless networks," *IEEE Trans. Commun.*, vol. 70, no. 9, pp. 6295–6309, Sep. 2022.

[8] W. Zhang, M. Derakhshani, G. Zheng, C. S. Chen, and S. Lambotharan, "Bayesian optimization of queuing-based multichannel URLLC scheduling," *IEEE Trans. Wireless Commun.*, vol. 22, no. 3, pp. 1763–1778, Mar. 2023.

[9] R. Chai and S. Wang, "Energy-efficient resource allocation for latency-constrained wireless transmissions," in *Proc. IEEE/CIC ICCC*, Aug. 2024, pp. 1051–1056.

[10] W. Zhang, M. Derakhshani, G. Zheng, and S. Lambotharan, "Constrained risk-sensitive deep reinforcement learning for eMBB-URLLC joint scheduling," *IEEE Trans. Wireless Commun.*, vol. 23, no. 9, pp. 10608–10624, Sep. 2024.

[11] H. Babbar, S. Rani, A. A. AlZubi, A. Singh, N. Nasser, and A. Ali, "Role of network slicing in software defined networking for 5G: Use cases and future directions," *IEEE Wireless Commun.*, vol. 29, no. 1, pp. 112–118, Feb. 2022.

[12] S. Vittal and A. A. Franklin, "HARNESS: High availability supportive self reliant network slicing in 5G networks," *IEEE Trans. Netw. Service Manage.*, vol. 19, no. 3, pp. 1951–1964, Sep. 2022.

[13] J. F. Santos et al., "Breaking down network slicing: Hierarchical orchestration of end-to-end networks," *IEEE Commun. Mag.*, vol. 58, no. 10, pp. 16–22, Oct. 2020.

[14] R. Chai and S. Wang, "Resource allocation for radio access network slicing with delay guarantees: A Lyapunov optimization method," *IEEE Trans. Veh. Technol.*, vol. 74, no. 11, pp. 18313–18317, Nov. 2025.

[15] H. Yin, L. Zhang, and S. Roy, "Multiplexing URLLC traffic within eMBB services in 5G NR: Fair scheduling," *IEEE Trans. Commun.*, vol. 69, no. 2, pp. 1080–1093, Feb. 2021.

[16] T. Wang and S. Wang, "Online convex optimization for efficient and robust inter-slice radio resource management," *IEEE Trans. Commun.*, vol. 69, no. 9, pp. 6050–6062, Sep. 2021.

[17] J. Zhu and S. Wang, "QoS-guaranteed resource allocation in mobile communications: A stochastic network calculus approach," *IEEE/ACM Trans. Netw.*, vol. 32, no. 6, pp. 5159–5171, Dec. 2024.

[18] O. Adamuz-Hinojosa, L. Zanzi, V. Sciancalepore, A. Garcia-Saavedra, and X. Costa-Pérez, "ORANUS: Latency-tailored orchestration via stochastic network calculus in 6G O-RAN," in *Proc. IEEE Conf. Comput. Commun.*, May 2024, pp. 61–70.

[19] M. Katwe et al., "Enhanced user fairness and performance for eMBB-URLLC uplink traffic with rate-splitting based super-positioning," *IEEE Trans. Wireless Commun.*, vol. 23, no. 9, pp. 12513–12530, Sep. 2024.

[20] M. Tian et al., "On-demand multiplexing of eMBB/URLLC traffic in a multi-UAV relay network," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 6, pp. 6035–6048, Jun. 2024.

[21] F. Wei, G. Feng, S. Qin, Y. Peng, and Y. Liu, "Hierarchical network slicing for UAV-assisted wireless networks with deployment optimization," *IEEE J. Sel. Areas Commun.*, vol. 42, no. 12, pp. 3705–3718, Dec. 2024.

[22] M. Setayesh, S. Bahrami, and V. W. S. Wong, "Resource slicing for eMBB and URLLC services in radio access network using hierarchical deep learning," *IEEE Trans. Wireless Commun.*, vol. 21, no. 11, pp. 8950–8966, Nov. 2022.

[23] J. Zhang and J. G. Andrews, "Adaptive spatial intercell interference cancellation in multicell wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 9, pp. 1455–1468, Dec. 2010.

[24] Y. Polyanskiy, H. V. Poor, and S. Verdu, "Channel coding rate in the finite blocklength regime," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2307–2359, May 2010.

[25] J. Gärtner, "On large deviations from the invariant measure," *Theory Probab. Appl.*, vol. 22, no. 1, pp. 24–39, Dec. 1977.

[26] R. S. Ellis, "Large deviations for a general class of random vectors," *Ann. Probab.*, vol. 12, no. 1, pp. 1–12, Feb. 1984.

[27] C.-S. Chang and J. A. Thomas, "Effective bandwidth in high-speed digital networks," *IEEE J. Sel. Areas Commun.*, vol. 13, no. 6, pp. 1091–1100, Aug. 1995.
[28] G. L. Choudhury, D. M. Lucantoni, and W. Whitt, "Squeezing the most out of ATM," *IEEE Trans. Commun.*, vol. 44, no. 2, pp. 203–217, Feb. 1996.
[29] D. Wu and R. Negi, "Effective capacity: A wireless link model for support of quality of service," *IEEE Trans. Wireless Commun.*, vol. 2, no. 4, pp. 630–643, May 2003.
[30] L. Fleischer, M. X. Goemans, V. S. Mirrokni, and M. Sviridenko, "Tight approximation algorithms for maximum general assignment problems," in *Proc. ACM/SIAM SODA*, Jan. 2006.
[31] W. Li, J. Lee, and N. Shroff, "A faster FPTAS for knapsack problem with cardinality constraint," *Discrete Appl. Math.*, vol. 315, pp. 71–85, Jul. 2022.
[32] A. Caprara, H. Kellerer, U. Pferschy, and D. Pisinger, "Approximation algorithms for knapsack problems with cardinality constraints," *Eur. J. Oper. Res.*, vol. 123, no. 2, pp. 333–345, Jun. 2000.
[33] N. Megiddo and A. Tamir, "Linear time algorithms for some separable quadratic programming problems," *Operations Res. Lett.*, vol. 13, no. 4, pp. 203–211, May 1993.
[34] M. Neely, *Stochastic Network Optimization With Application to Communication and Queueing Systems*. San Rafael, CA, USA: Morgan & Claypool, 2010.
[35] H. Deng, T. Zhao, and I.-H. Hou, "Online routing and scheduling with capacity redundancy for timely delivery guarantees in multihop networks," *IEEE/ACM Trans. Netw.*, vol. 27, no. 3, pp. 1258–1271, Jun. 2019.
[36] Y. Yang and S. Wang, "EdgeOPT: A competitive algorithm for online parallel task scheduling with latency guarantee in mobile edge computing," *IEEE Trans. Commun.*, vol. 72, no. 11, pp. 7077–7092, Nov. 2024.
[37] C. Tsanikidis and J. Ghaderi, "Online scheduling and routing with end-to-end deadline constraints in multihop wireless networks," *IEEE Trans. Netw.*, vol. 33, no. 5, pp. 2396–2409, Oct. 2025.
[38] S. Lu et al., "Non-stationary continuum-armed bandits for online hyper-parameter optimization," in *Proc. ACM WSDM*, Feb. 2022, pp. 618–627.
[39] K. Haneda et al., "5G 3GPP-like channel models for outdoor urban microcellular and macrocellular environments," in *Proc. IEEE VTC Spring*, May 2016, pp. 1–7.
[40] W. Zhang, M. Derakhshani, and S. Lambotharan, "Stochastic optimization of URLLC-eMBB joint scheduling with queuing mechanism," *IEEE Wireless Commun. Lett.*, vol. 10, no. 4, pp. 844–848, Apr. 2021.
[41] J. Bergstra and Y. Bengio, "Algorithms for hyper-parameter optimization," in *Proc. NeurIPS*, Dec. 2011, pp. 2546–2554.
[42] M. P. Deisenroth and J. W. Ng, "Distributed Gaussian processes," in *Proc. ICML*, Jun. 2015, pp. 1481–1490.

Rong Chai (Graduate Student Member, IEEE) received the B.S. degree in electronic information engineering from Northwestern Polytechnical University, Xi'an, China, in 2023. He is currently pursuing the Ph.D. degree with the School of Electronic Science and Engineering, Nanjing University, Nanjing, China. His research interests include machine learning and deterministic networking.

Shaowei Wang (Senior Member, IEEE) received the Ph.D. degree from Wuhan University, Wuhan, China, in 2006. In 2006, he joined the School of Electronic Science and Engineering, Nanjing University, Nanjing, China, as a Faculty Member, where he is currently a Full Professor. From 2012 to 2013, he was a Visiting Scholar/a Professor with Stanford University, Stanford, CA, USA, and The University of British Columbia, Vancouver, BC, Canada. His research interests include communications and networking, operations research, and machine learning.