Stochastic Optimization of URLLC-eMBB Joint Scheduling
With Queuing Mechanism

Wenheng Zhang, Mahsa Derakhshani, Senior Member, IEEE,
and Sangarapillai Lambotharan, Senior Member, IEEE

Abstract—This letter proposes a dynamic joint scheduling for
the ultra-reliable low-latency communication (URLLC) and
enhanced mobile broadband (eMBB) traffic at a sub-frame level
with a queuing mechanism, which monitors and controls the
latency of each URLLC packet in real time to ensure its strict
requirements. We analytically derive the outage probability (i.e.,
the probability of any URLLC packet drop over all transmission
channels) and URLLC expected throughput in addition to the
expected value of served URLLC packets. Then, a stochastic
optimization problem is formulated to maximize the total
throughput for the eMBB services, constraining the URLLC outage
probability. Numerical results confirm effectiveness of our
queuing policy to significantly reduce the URLLC loss rate while
ensuring that the total eMBB throughput is not affected.

Index Terms—URLLC, eMBB, punctured scheduling, queuing.

I. INTRODUCTION

E nhanced Mobile Broadband (eMBB) and ultra-reliable
and low-latency communications (URLLC) are two main
services that will be supported by flexible network operation
in future wireless communication networks. The URLLC service
targets mission critical applications [1]–[5]. The challenge of
URLLC is an internal trade-off between its latency and
reliability requirements. Besides, the reliability of URLLC
is influenced by the coexistence with other types of traffic.
Thus, channel scheduling faces a critical challenge on simulta-
neously implementing these services with different targets.
5G services require a high peak data rate for eMBB users
up to gigabits per second, while the URLLC traffic transmits
small payload within 1ms latency and at least 99.999% success
probability [2]. In this case, the BS should provide stable and
satisfactory transmission channels for sporadic transmission
of URLLC packets but also maximize the eMBB throughput.

Dynamic Scheduling of eMBB and URLLC has received
attention recently [6]–[11]. Reference [6] proposed the idea
of scheduling eMBB and URLLC packets on different time
scales. Subsequently, the idea of punctured scheduling, i.e.,
suspension of the eMBB transmission when a URLLC packet
is decided to be served, was proposed by [7]. This scheme has
been further explored by [8] that studied the joint scheduling
optimization investigating different loss functions (i.e., linear,
convex and threshold models) to model the effects

of punctured URLLC transmission on eMBB throughput.
Moreover, [9] analyzed the eMBB throughput and operational
complexity in a dynamic mode. With a risk-aware framework,
[10] proposed a risk-sensitive optimization which
punctures more URLLC packets on the robust set of eMBB
users to protect the sensitive group. However, in these frame-
works, the incoming URLLC packets are either served imme-
diately or dropped on each mini slot. This incurs significant
URLLC packet drops, which could be avoided to some extent
with an appropriate queuing mechanism. Furthermore, further
researches are needed to focus on the time-varying channels.

Addressing these existing challenges, in this letter, we pro-
pose a stochastic scheduling scheme for URLLC and eMBB
coeexistence considering a \( M/G/\infty \) queuing mechanism for
sporadic URLLC packets, this helps the system to meet the
URLLC latency requirement by queuing each packet in the
line up to two mini slots. Comparing with the immediate drop
policy of previous works (e.g., [8]), our scheme improves the
URLLC reliability within the advised latency by 3GPP [2].

Moreover, an outage-constrained stochastic optimization
problem is formulated under assumption of time-varying
channels. The aim is to maximize the total throughput of eMBB users
on different CSI while restricting the URLLC outage proba-
bility (i.e., the probability of any URLLC drops) and URLLC
expected throughput. We use successive convex approximation
(SCA) [12] to convert the original non-convex optimization into
linear integer programming for eMBB scheduling and geometric
programming (GP) for URLLC placement scheduling. We also
built the eMBB-URLLC coexistence system of [8] (i.e.,
without applying URLLC queuing system) and compared the
optimization results in terms of mean value of total eMBB
throughput and URLLC outage probability.

II. SYSTEM DESIGN

This letter considers a downlink dynamic scheduling of an
eMBB-URLLC coexistence system. We assume a multiple-
channel scenario and study scheduling in different frequency
channels and time intervals as shown in Fig. 1. Each eMBB
slot of one millisecond contains \( M \) URLLC mini slots and the
set is represented by \( \mathcal{M} \). The set of frequency
channels is \( \mathcal{B} = \{1, 2, \ldots, B\} \). The set of eMBB users
is \( \mathcal{U} = \{1, 2, \ldots, U\} \). The set of possible channel states on each
URLLC slot is denoted by \( \mathcal{C} = \{1, 2, \ldots, C\} \). The CSI of all
eMBB users on different channels and at different time slot is rep-
resented by a \( B \times U \) matrix, i.e., \( s_{u} = [s_{u}^{b}] \), where \( s_{u}^{b} \in \mathcal{C} \),
\( \forall \, u \in \mathcal{U}, \forall \, b \in \mathcal{B} \). We assume the CSI of URLLC user on dif-
ferent channels is represented by a \( B \times 1 \) matrix, \( s_{f} \). We use
\( s = [s_{1}, s_{u}] \) to denote the CSI of these two kinds of traffic,
the probability mass function (PMF) of \( s \) is \( P_{S}(s) \), \( \forall \, s \in \mathcal{S} \),
where the cardinality of \( \mathcal{S} \) is \( \frac{S}{C(B+1)U} \).

At each time slot, the BS decides the allocation for all
eMBB users on different channels. The eMBB allocations

Manuscript received November 13, 2020; accepted December 14, 2020.
Date of publication December 22, 2020; date of current version April 9, 2021.
The work of Mahsa Derakhshani was supported by the Royal Academy of
Engineering under the Leverhulme Trust Research Fellowship Scheme. The
associate editor coordinating the review of this article and approving it for
publication was H. Chen. (Corresponding author: Wenheng Zhang.)
The authors are with the Wolfson School of Mechanical, Electrical and
Manufacturing Engineering, Loughborough University, Loughborough LE11 3TU, U.K. (e-mail: w.zhang2@lboro.ac.uk;
m.derakhshani@lboro.ac.uk; s.lambotharan@lboro.ac.uk).
Digital Object Identifier 10.1109/LWC.2020.3046628
is presented by a $B \times S \times U$ matrix $I = [I_{u,b}^{s}]$, where
$I_{u,b}^{s} \in \{0,1\}$ is a binary variable declaring whether channel $b$ is allocated to eMBB user $u$ or not when the CSI is $s$. It is assumed that each channel is always fully occupied by one of the eMBB users on each time slot, i.e., $\sum_{u \in U} I_{u,b}^{s} = 1$, $\forall s \in S$, $\forall b \in B$. Once the BS receives an incoming URLLC packet and decides to transmit it on channel $b$, it suspends the eMBB communication and punctures URLLC on channel $b$ during next mini slot. We design and characterize a queuing mechanism to stochastically restrict the number of dropped URLLC packets which will impact the reliability. Let $A_m$ denote the number of URLLC packet arrivals between mini slot $(m-1)$ and mini slot $m$. We assume Poisson distribution for URLLC packet arrivals with a mean rate $\lambda_a$. The arriving packets are divided between $B$ different channels uniformly as shown Fig. 1. To ensure that the delay requirement of URLLC served packets is always complied with the 3GPP standard (i.e., 0.25-0.3 milliseconds [2]) while reducing the number of packet drops, it is allowed each URLLC packet to wait in the queue up to two mini slots in the proposed framework. At the beginning of each mini slot, BS decides the transmission of URLLC packets. If a URLLC packet is already queued for one mini slot and will not be served at next mini slot, it will be dropped.

We define $w_{u,b}^{s}$ as the weight of puncturing the eMBB user $u$ on channel $b$ when CSI is $s$, $w_{u,b}^{s} \in [0,1]$, $\forall s \in S$, $\forall b \in B$, $\forall u \in U$. Based on the value of $w_{u,b}^{s}$, at each mini slot, function $b(w_{u,b}^{s})$ applies Bernoulli random variable generator to obtain a URLLC transmission decision for each channel. $b(w_{u,b}^{s}) = 1$ indicates that a URLLC packet will be punctured during eMBB $u$ on channel $b$ transmission when CSI is $s$ (this occurs with probability $w_{u,b}^{s}$), and 0 indicates that the BS continues the transmission of eMBB user signal on that mini slot (this occurs with probability $1 - w_{u,b}^{s}$). If $I_{u,b}^{s} = 1$, BS will operate $b(w_{u,b}^{s})$ for each mini-time slot on channel $b$. Accordingly, $C_{m,b}^{s} \in \{0,1\}$ is a random number realizing if the mini slot $m$ will be allocated for URLLC transmission or not on channel $b$ if CSI is $s$ and can be represented as

$$C_{m,b}^{s} = \sum_{u \in U} I_{u,b}^{s} b(w_{u,b}^{s}), \forall m \in M, \forall b \in B \quad (1)$$

Subsequently, the number of served URLLC packets at mini slot $m$ on channel $b$ when CSI is $s$ can be calculated as

$$\mu_{m,b}^{s} = \min (C_{m-1,b}^{s}, Q_{m-1,b}^{s} + A_{m,b}), \quad (2)$$

where $Q_{m-1,b}^{s}$ denotes the number of queued packet at mini time-slot $(m - 1)$. Consequently, the number of dropped packets at mini slot $m$ can be computed as

$$D_{m,b}^{s} = \max (Q_{m-1,b}^{s} - \mu_{m,b}^{s}, 0). \quad (3)$$

Based on (1)-(3), when CSI is $s$ in channel $b$, the queuing equation at mini slot $m$ can be represented as

$$Q_{m,b}^{s} = Q_{m-1,b}^{s} + A_{m,b} - (\mu_{m,b}^{s} + D_{m,b}^{s}) \quad (4)$$

Studying the queue dynamics (see Appendix A), with $\lambda = \lambda_a/B$, the expected number of served URLLC packets is a function of eMBB allocations $I$ and puncturing weights $W$ as

$$E[\mu_{m,b}^{s}] = \sum_{b \in B} \sum_{u \in U} I_{u,b}^{s} (w_{u,b}^{s})^2 \lambda (w_{u,b}^{s} \lambda - e^{-\lambda})^{-1} \quad (5)$$

**III. DYNAMIC SCHEDULING**

**A. Optimization Problem**

Here, we formulate a stochastic optimization problem aiming to optimally determine the URLLC puncturing weights $W$ and resource allocation factors $I$ for each CSI $s$ over all channels in order to schedule the incoming URLLC packets while maximizing the total throughput of eMBB users. The total expected throughput of eMBB users can be calculated as

$$T_{eMBB} = \sum_{b \in B} \sum_{u \in U} \sum_{s \in S} P_s(s) I_{u,b}^{s} R_{s,b}^{eMBB} \left(1 - w_{u,b}^{s} E[\mu_{m,b}^{s}] \right) \quad (6)$$

where $R_{s,b}^{eMBB} = B_W \log(1 + d_u^{-\kappa} s \rho_B)$ is the peak rate of eMBB user $u$ on channel $b$ where $d_u$ is the distance between the eMBB user $u$ and BS, $\kappa$ denotes the path loss exponent, $B_W$ is the channel bandwidth and $\rho_B$ is the transmitted SNR of each eMBB user. In (6), $w_{u,b}^{s} E[\mu_{m,b}^{s}]$ represents the ratio of expected number of mini-slots of eMBB user $u$ punctured for URLLC service on channel $b$ to the total number of mini-slots in one eMBB slot under CSI $s$ under all channels.

Similarly, the expected throughput of URLLC

$$T_{URLLC} = \sum_{b \in B} \sum_{u \in U} \sum_{s \in S} P_s(s) I_{u,b}^{s} w_{u,b}^{s} R_{s,b}^{URLLC} E[\mu_{m,b}^{s}] \quad (7)$$

where $R_{s,b}^{URLLC} = B_W \log(1 + d_u^{-\kappa} s \rho_r) - \frac{1}{\mu_r} \sqrt{Q^{-1}(\xi)}$ is the channel rate for URLLC on channel $b$ [13] where $b_l$ is the block-length, $\xi$ is the error probability, $d_u$ is the distance between URLLC and the BS, $\rho_r$ denotes the transmission SNR of URLLC, $V$ denotes the so-called channel dispersion, and $Q^{-1}(\xi)$ is the inverse of the Gaussian $Q$ function.

Based on (6) and (7), the optimization problem can be represented as

$\mathbb{P}$: $\max \mathbf{I}, \mathbf{W} \sum_{b \in B} \sum_{u \in U} \sum_{s \in S} \varphi_u P_s(s) I_{u,b}^{s} R_{s,b}^{eMBB} \left(1 - w_{u,b}^{s} E[\mu_{m,b}^{s}] \right)$, s.t.,

- $C_1$: $\sum_{s \in S} P_s(s) \prod_{b \in B} \Pr (D_{m,b}^{s} = 0) \geq \zeta, \forall m \in M$
- $C_2$: $\sum_{u \in U} I_{u,b}^{s} = 1$, $\forall b \in B$, $\forall s \in S$
- $C_3$: $I_{u,b}^{s} \in \{0,1\}$, $\forall b \in B$, $\forall s \in S$, $\forall u \in U$
- $C_4$: $w_{u,b}^{s} \in [0,1]$, $\forall b \in B$, $\forall s \in S$,
- $\forall u \in U$
- $C_5$: $\sum_{b \in B} \sum_{u \in U} \sum_{s \in S} P_s(s) I_{u,b}^{s} w_{u,b}^{s} R_{s,b}^{URLLC} E[\mu_{m,b}^{s}] \geq \eta$, where the stochastic constraint in $C_1$ guarantees that the probability of no URLLC drop is always higher than a target threshold $\zeta$ (equivalently, probability of any drops is kept limited). $C_2$ ensures that each channel is always occupied exclusively by one of the eMBB users on each time slot. $C_3$ and $C_4$ ensure the values of the eMBB allocation variable $I_{u,b}^{s}$ should be binary and puncturing weight $w_{u,b}^{s}$ should be a decimal for each eMBB user $u$ on channel $b$ and CSI $s$. Also, $C_5$ guarantees that the expected throughput of URLLC service is...
Algorithm 1: Dynamic URLLC-eMBB Scheduling

Variables: I, W, Y, P and X
Results: Maximize the total throughput of eMBB users

1. Initialize the puncturing matrix \( W_0 \);
2. while Not reaching the maximum iteration number or variables do not converge do
   3. eMBB allocation scheduling as \( P_1 \);
   4. Initialize the iteration variables;
   5. while Not reaching the maximum iteration number or variables do not converge do
      6. URLLC placement scheduling as \( P_2 \);
      7. Update the iteration variables as (8) to (13);

no less than a target threshold \( \eta \). We use \( \varphi_u \) to denote the priority weights of eMBB user \( u \).

B. Proposed Solution

The dynamic scheduling problem is decomposed into two stages. The BS schedules eMBB allocations by fixing the puncturing weights \( W \). After updating the eMBB allocation matrix \( I \), URLLC packets placement scheduling is optimized.

1) eMBB Scheduling: In this stage, we assume that the URLLC puncturing weight matrix \( W \) is fixed when optimizing eMBB allocation matrix \( I \). For the eMBB scheduling, we solve the following linear integer programming:

\[
P_1: \max \sum_{b \in B} \sum_{u \in B} \sum_{s \in S} \varphi_u P_a(s) I_{u,b}^a R_{u,b}^{s_a} \left( 1 - \frac{\alpha(w_{u,b}^a)^3}{w_{u,b}^a \lambda - e^{-\lambda}} \right)
\]

s.t. C1, C2, C3

2) URLLC Placement Scheduling: For optimizing the URLLC puncturing weights, we fix \( I \) and then solve \( P_2 \). A new variable matrix is defined as \( Y = [y_{u,b}^a] \) where \( y_{u,b}^a = e^{-\lambda} - w_{u,b}^a \lambda, \forall b \in B, \forall s \in S, \forall u \in U \). Thus, the optimization problem can be transformed into the complementary geometric programming form (CGP) as:

\[
P_2: \max \sum_{b \in B} \sum_{u \in B} \sum_{s \in S} \varphi_u P_a(s) I_{u,b}^a R_{u,b}^{s_a} \left( 1 - (w_{u,b}^a)^2 y_{u,b}^a \lambda - e^{-\lambda} \right)
\]

s.t. C1, C4, C5

C6: \( e^{-\lambda} (y_{u,b}^a + w_{u,b}^a \lambda)^{-1} \leq 1, \forall b \in B, \forall s \in S, \forall u \in U \)

We use \( p_b^a \) to denote the probability of no URLLC packet drop over one time-slot on channel \( b \) when CSI is \( s \). As shown in Appendix B, \( p_b^a \) is calculated as:

\[
p_b^a = \sum_{u \in U} I_{u,b}^a \left( \frac{\lambda w_{u,b}^a - (\lambda w_{u,b}^a)^2 (2 + \lambda) e^{-\lambda}}{e^{-\lambda} - w_{u,b}^a \lambda} \right)^{\gamma_{u,b}^a}, \forall s \in S, \forall b \in B
\]

Thus, C1 can be rewritten as C1:

\[
C1: \sum_{s \in S} P_a(s) \prod_{b \in B} p_b^a \geq \zeta
\]

Applying arithmetic-geometric mean approximation (AGMA) (by adding a new constraint C7) to change the non-posynomial constraint C1 to posynomial form. Variable \( X = [x_{u,b}^a] \) is introduced to change the non-posynomial objective function \( P_2 \) into a posynomial form. \( x_{u,b}^a = (w_{u,b}^a)^2 (y_{u,b}^a - e^{-\lambda}) (y_{u,b}^a)^{-1}, \forall b \in B, \forall s \in S, \forall u \in U \). C8 converts \( X \) to the posynomial form. The URLLC scheduling optimization becomes,

\[
P_2: \min \sum_{b \in B} \sum_{u \in B} \sum_{s \in S} \varphi_u P_a(s) I_{u,b}^a R_{u,b}^{s_a} x_{u,b}^a \quad s.t. C1, C4, C5, C7
\]

\[
C7: \frac{p_b^a y_{u,b}^a e^\lambda + y_{u,b}^a e^{\lambda + e^{-\lambda} (2 + \lambda) (2 + \lambda) e^{-\lambda}}}{\forall b \in B, \forall s \in S, \forall u \in U}
\]

\[
P_2: \min \sum_{b \in B} \sum_{u \in B} \sum_{s \in S} \varphi_u P_a(s) I_{u,b}^a R_{u,b}^{s_a} x_{u,b}^a \quad s.t. C1, C4, C5, C7
\]

\[
C7: \frac{p_b^a y_{u,b}^a e^\lambda + y_{u,b}^a e^{\lambda + e^{-\lambda} (2 + \lambda) (2 + \lambda) e^{-\lambda}}}{\forall b \in B, \forall s \in S, \forall u \in U}
\]

With AGMA, the posynomial constraints C5-C8 become monomial and \( P_2 \) is converted into a standard GP (i.e., \( \overline{P}_2 \)), the constraints C4, C6 – C8 satisfy \( \forall b \in B, \forall s \in S, \forall u \in U \). Finally, SCA algorithm updates the variables in each iteration \( l \). The optimization problem under different time iterations is:

\[
P_2: \min \sum_{b \in B} \sum_{u \in B} \sum_{s \in S} \varphi_u P_a(s) I_{u,b}^a R_{u,b}^{s_a} x_{u,b}^a \quad s.t. C1, C4, C5, C7
\]

\[
C7: \frac{p_b^a y_{u,b}^a e^\lambda + y_{u,b}^a e^{\lambda + e^{-\lambda} (2 + \lambda) (2 + \lambda) e^{-\lambda}}}{\forall b \in B, \forall s \in S, \forall u \in U}
\]

and the iteration variables are updated as (8)-(13), and all of them satisfy \( \forall b \in B, \forall s \in S, \forall u \in U \).

\[
\alpha_{u,b}^a(l) = \frac{y_{u,b}^a(l)}{y_{u,b}^a(l) + w_{u,b}^a(l) \lambda}, \beta_{u,b}^a(l) = \frac{w_{u,b}^a(l) \lambda}{y_{u,b}^a(l) + w_{u,b}^a(l) \lambda}
\]

\[
\delta_{u,b}^a(l) = p_b^a y_{u,b}^a e^{\lambda (\gamma_{u,b}^a(l)^{-1} - \lambda)}, \delta_{u,b}^a(l) = e^{\lambda y_{u,b}^a(l)(\gamma_{u,b}^a(l) - 1)}
\]

\[
\epsilon_{u,b}^a(l) = \frac{(y_{u,b}^a(l))^2 (2 + \lambda)}{2 \lambda y_{u,b}^a(l)}, \epsilon_{u,b}^a(l) = \frac{(2 + \lambda) e^{-2\lambda}}{2 \lambda y_{u,b}^a(l)}
\]

where, \( y_{u,b}^a(l) = (p_b^a(l) y_{u,b}^a(l) + w_{u,b}^a(l)^2) e^{\lambda l} \cdot (2 + \lambda) e^{-2\lambda} \)

\[
\nu_{u,b}^a(l) = y_{u,b}^a(l) x_{u,b}^a(l) \left( y_{u,b}^a(l) x_{u,b}^a(l) + w_{u,b}^a(l)^2 e^{-\lambda w_{u,b}^a(l)^2} \right)^{-1}
\]

\[
\tau_{u,b}^a(l) = e^{-\lambda w_{u,b}^a(l)^2} \left( y_{u,b}^a(l) x_{u,b}^a(l) + e^{-\lambda w_{u,b}^a(l)^2} \right)^{-1}
\]

C. Computational Complexity Analysis

The computational complexity of Algorithm 1 is dominated by the URLLC placement scheduling, in which we solve a series of GP sub-problems with interior point method. From [12], the order of computational complexity for \( \overline{P}_2 \) is \( i_{urllc} \times \frac{\log(C_{urllc}/(I^0 \delta))}{\log(I^0)} \), where \( i_{urllc} \) is the number of required computations to locally approximate \( \overline{P}_2 \) with a GP using AGMA and the later represents the number of required iterations, in which \( I^0 \) is the initial point to approximate the
accuracy, $0 < \rho < 1$ is the stopping criterion and $\partial$ is used for updating the accuracy of interior point method. Also, $C_{urllc}$ denotes the number of URLLC scheduling constraints in $\mathbb{P}_2$ which is $C_{urllc} = 2 + 4BSU$. The number of computations for URLLC placement scheduling is $i_{urllc} = B^2SU + 5BSU$. It is worth noting that the studied scheduling problem is a stochastic problem and thus needs to be run once. The obtained optimal values (i.e., $w_{u,\varphi}$ and $I_{s,\varphi}$) can be applied in each time-slot and each channel to have the allocations depending on what the instantaneous channel is.

### IV. Simulation Results

We consider a system with three eMBB users with the setting in Table I. We assumed Rayleigh fading model considering three different ranges of fading power gains $[0, 0.5]$, $[0.5, 1.5]$ and $[1.5, \infty]$. The PMF of possible channel states is calculated as $\sum_{\chi \in \chi_1} g(\chi) d\chi / \int_{\chi_1}^{\chi_2} g(\chi) d\chi$, where $\chi$, $g(\chi)$, $\chi_1$ and $\chi_2$ represent the value of the fading power gains, its probability density function, and its lower and upper bounds respectively. Therefore, the set of channel states $\mathcal{C} = \{0.2287, 0.9129, 2.5097\}$ and the PMF of corresponding different possible channel states is $\{0.39, 0.38, 0.23\}$.

First, we consider a case in which for all eMBB users $d = 0.3$, which is the normalized distance to the maximum radius of the cell and consider the same priority weights for all eMBB users ($\varphi_u = 1/3$).

Fig. 2 numerically explores the average queuing time for served URLLC packets against URLLC arrival rate $\lambda_u$. From the results, the value of mean latency is below 0.65 mini-slots when $\lambda_u$ changing within $[1, 5]$ packets/mini-slot. However, when the load rate is greater than 4 packets/mini-slot, there is a rapid rise on such delay, since the channel limitation leads to longer waiting time.

Fig. 3 depicts the trade-off between URLLC average served rate and the expected total eMBB throughput with the proposed queuing policy and resource proportional (RP) placement scheduling policy [9]. The latter policy assumes that puncturing weights are in proportion to the percentage of resource blocks that allocated to eMBB user $u$ in each time-slot. The curves have been obtained by changing $\lambda_u$ and measuring eMBB throughput and URLLC served packet rate. It is shown that with the same URLLC served rate, the system with queuing can support a higher eMBB total throughput for higher range of URLLC served rate. As $\lambda_u$ being increased, the BS reduces eMBB throughput to allocate more mini slots for enhancing URLLC reliability in both systems, but the RP placement scheduling policy drops more URLLC packets. This has been also confirmed in Fig. 4 which demonstrates achievable URLLC outage probability versus URLLC served rate. Compared with the RP scheduling policy, it is clear that the queuing mechanism is effective in keeping URLLC outage probability limited even with a larger URLLC rate.

Here, we consider a setting in which eMBB users are at different distances from the BS (i.e., $d_1 = 0.1$, $d_2 = 0.5$, $d_3 = 0.9$). We change the priority weights to observe the influence on expected throughput of each eMBB user. In Fig. 5, we apply the same priority weights on eMBB users, i.e., $\varphi_u = 1/3$, $\forall u \in \mathcal{U}$. The BS allocates more time slots to the eMBB user which has the highest average channel rate (i.e., $U_1$) and also punctures abundant URLLC packets on it. As $\lambda_u$ increases, larger puncturing weights are required to support URLLC reliability requirement and this reduces the average throughput of $U_1$. Thus, the BS enhances the total eMBB throughput by allocating time slots to both $U_2$ and

### Table I

**System Configuration**

| Parameters                        | Values       |
|-----------------------------------|--------------|
| Transmission error rate           | $\xi = 10^{-6}$ |
| Channel block-length              | $b_1 = 200$  |
| The number of channels            | $B = 4$      |
| The threshold of URLLC outage probability | $\zeta = 0.9$ |
| The threshold of URLLC mean throughput | $\eta = 1$Mbps |
| Transmit SNR of eMBB users        | $\rho = 10^4$ (i.e., 40dB) |
| Transmit SNR of URLLC users       | $\rho_r = 10^4$ (i.e., 40dB) |
| Channel bandwidth                 | $B_{ch} = 180$KHz |

---

**Fig. 2.** Average delay of served URLLC packets vs. URLLC arrival rate.

**Fig. 3.** Mean eMBB total throughput versus URLLC mean served rate.

**Fig. 4.** URLLC outage probability vs. URLLC mean served rate.

**Fig. 5.** eMBB throughput vs URLLC arrival rate ($\varphi_u = 1/3$, $\forall u \in \mathcal{U}$).
U1 under different instantaneous CSI and never allocates any time slot to the user at the edge. To improve such unfairness among eMBBs, in Fig. 6, we adjust the priority weights to achieve fair allocations. The priority weight of a user is set inversely proportional to the distance of that user from the BS, i.e., ϕ1 = 0.25, ϕ2 = 0.35, ϕ3 = 0.45. Thus, the priority weights can adjust the allocations even if the eMBB users are at different distances from the BS.

V. CONCLUSION

In this letter, we developed an optimal joint scheduling strategy for the eMBB and URLLC coexistence system with a queuing mechanism. The results showed that the system mechanism satisfied URLLC reliability and latency requirements when comparing with the no queue system under RP placement policy. With suitable priority factors, the system also maintains the throughput of each eMBB user at different transmission distances.

APPENDIX A

PROOF OF EXPECTED VALUE FOR SERVED URLLC PACKETS

We assumed all channels are the same in different channels and scheduled each channel individually. At the beginning of each mini slot, the number of URLLC served packet in each channel is either zero or one, thus, the expected number of URLLC served packets for all transmission channels at each mini slot m when CSI is s can be expressed as

\[ E[\mu^s_m] = \sum_{b \in B} \Pr(\mu^s_{m,b} = 1) \]

\[ = \sum_{b \in B} \Pr(C^s_{m,b} = 1)(1 - \Pr(Q^s_{m-1,b} = 0))\Pr(A^s_{m,b} = 0) \]  \hspace{1cm} (14)

We assume the URLLC packets as Possion distribution, therefore, \( \Pr(A^s_{m,b} = 0) = e^{-\lambda} \). The problem becomes to find the distribution of the provided mini slots (i.e., \( \Pr(C^s_{m,b} = c) \), \( \forall c \in \{0,1\} \)) and the distribution of queue (i.e., \( \Pr(Q^s_{m,b} = q) \), \( \forall q \in \{0,1,\ldots,\infty\} \)) for each channel b.

The PMF of provided mini slots for channel b, \( \forall b \in B \) is

\[ \Pr(C^s_{m,b} = c) = (w^s_b)^c (1 - (w^s_b))^{1-c}, \forall c \in \{0,1\} \]  \hspace{1cm} (15)

where \( w^s_b \) is the total puncturing weights for channel b when CSI is s and can be calculated with

\[ w^s_b = \text{diag}(W^s_b) T I_b = \sum_{u \in \mathcal{U}} w^s_{u,b} T I_{u,b}, \forall s \in \mathcal{S}, \forall b \in B \]  \hspace{1cm} (16)

The state transition matrix is used to find the steady state probability of the queue system. The equation for the state transition matrix for each sub queue is in Fig. 7.

The matrix obeys the steady state equation as

\[ (1 - P^T)[\Pr(Q^s_{m,b} = 0), \ldots, \Pr(Q^s_{m,b} = \infty)] = 0 \]  \hspace{1cm} (17)

The sum of all steady state probabilities equals to one, i.e.,

\[ \sum_{q \in \{0,\ldots,\infty\}} \Pr(Q^s_{m,b} = q) = 1 \]  \hspace{1cm} (18)

Therefore, the steady state probabilities for \( \Pr(Q^s_{m,b} = 0) \) at channel b is

\[ \Pr(Q^s_{m,b} = 0) = \sum_{u \in \mathcal{S}} I^s_{u,b} e^{-\lambda - \lambda w^s_{u,b} - 1}, \forall m \in \mathcal{M} \]  \hspace{1cm} (19)

With (14)-(19), \( E[\mu^s_m] \) can be derived as (5).

APPENDIX B

PROOF OF PROBABILITY FOR NO DROPPED URLLC CASE

The formula of dropped packets on channel b in (3) can be written as

\[ D^s_{m,b} = \begin{cases} 0, & \text{if} \sum_{u \in \mathcal{S}} I^s_{u,b} e^{-\lambda - \lambda w^s_{u,b} - 1} - \sum_{u \in \mathcal{S}} I^s_{u,b} e^{-\lambda} - \lambda w^s_{u,b} - 1 \geq 0 \ I^s_{u,b} e^{-\lambda} - \lambda w^s_{u,b}, & \text{otherwize} \end{cases} \]  \hspace{1cm} (20)

Thus, based on (15)-(19), \( \Pr(D^s_{m,b} = 0) \) can be derived as

\[ \Pr(D^s_{m,b} = 0) = \Pr(Q^s_{m-1,b} = 0) + \Pr(Q^s_{m-1,b} = 1)\Pr(C^s_{m,b} = 1) \]

\[ = \sum_{u \in \mathcal{S}} I^s_{u,b} e^{-\lambda} \left[ \lambda w^s_{u,b} + \left( w^s_{u,b} - 1 \right) e^{-\lambda} \right] e^{-\lambda - \lambda w^s_{u,b}}, \forall s \in \mathcal{S}, \forall b \in B. \]  \hspace{1cm} (21)

REFERENCES

[1] Y. Zikria et al., “5G mobile services and scenarios: Challenges and solutions,” Sustainability, vol. 10, no. 10, p. 3626, 2018.
[2] 3GPP, “New services and applications with 5G ultra-reliable low latency communication,” Bellevue, WA, USA, 5G Americas, White Paper, Nov. 2018.
[3] M. Lusisotto et al., “Ultra high performance wireless control for critical applications: Challenges and directions,” IEEE Trans. Ind. Informat., vol. 13, no. 3, pp. 1448–1459, Jun. 2017.
[4] H. Ji et al., “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] M. Bennis et al., “Ultra-reliable and low-latency wireless communication: Tail risk and scale,” Proc. IEEE, vol. 106, no. 10, pp. 1834–1853, Oct. 2018.
[6] C. P. Li et al., “5G ultra-reliable and low-latency systems design,” in Proc. EuCNC, 2017, pp. 1–5.
[7] K. I. Pedersen et al., “Punctured scheduling for critical low latency data on a shared channel with mobile broadband,” in Proc. IEEE Veh. Technol. Conf. (VTC-Fall), 2017, pp. 1–6.
[8] A. Anand et al., “Joint scheduling of URLLC and eMBB traffic in 5G wireless networks,” IEEE/ACM Trans. Netw., vol. 28, no. 2, pp. 477–490, Apr. 2020.
[9] T. R. Pijnappel et al., “Joint scheduling of low-latency and best-effort flows in 5G wireless networks,” in Proc. 18th WiOpt, 2020, pp. 1–8.
[10] M. Alsenwi et al., “eMBB-URLLC resource slicing: A risk-sensitive approach,” IEEE Commun. Lett., vol. 23, no. 4, pp. 740–743, Apr. 2019.
[11] Y. Huang et al., “A deep-reinforcement-learning-based approach to dynamic eMBB/URLLC multiplexing in 5G NR,” IEEE Internet Things J., vol. 7, no. 7, pp. 6439–6456, Jul. 2020.
[12] S. Parsaeefard et al., “Joint user-association and resource-allocation in virtualized wireless networks,” IEEE Access, vol. 4, pp. 2738–2750, 2016.
[13] G. Durisi et al., “Toward massive, ultra reliable, and low-latency wireless communication with short packets,” Proc. IEEE, vol. 104, no. 9, pp. 1711–1726, Sep. 2016.