Bernd Simon, Mete Destan and Anja Klein, ”Reliable Two-Timescale Scheduling in a Multi-User Downlink Channel with Hard Deadlines,” in *Proc. of the IEEE Global Communications Conference 2021 (IEEE GLOBECOM 2021)*, December 2021.

Reliable Two-Timescale Scheduling in a Multi-User Downlink Channel with Hard Deadlines

Bernd Simon, Mete Destan, Anja Klein
Communications Engineering Lab, TU Darmstadt, Germany, {b.simon, a.klein}@nt.tu-darmstadt.de

***Abstract*—Ultra-Reliable Low-Latency Communications (URLLC) is an important part of emerging 5G and 6G networks which enables mission-critical applications like autonomous driving. These novel applications depend on the error-free delivery of short messages before an application-specific deadline, which is challenging in a fast-changing environment. In this work, we consider a wireless fading downlink channel shared for the transmission of periodically arriving messages for multiple mobile units (MUs). The message sizes, deadlines and the period of message arrival are MU-specific. The message for a MU can be split into smaller data packets, so that multiple unreliable transmissions can be combined to achieve a reliable transmission. We formulate an infinite time horizon Markov Decision Process (MDP) for the average timely throughput, and show that the MDP is periodic. We propose a novel two-timescale scheduling solution, which incorporates the uncertainty of the channel in an inter-frame problem and errors caused by short-packet coding in an intra-frame problem. Through numerical simulations, we show that the proposed approach outperforms State-of-the-Art scheduling algorithms in terms of timely throughput.**

I. Introduction

Novel mission-critical applications, e.g. autonomous driving, impose strict requirements on the error probability and latency of the transmission of short messages. Ultra-Reliable Low Latency Communication (URLLC) refers to communication under strict delay and reliability constraints in contrast to communication optimized for high average throughput or low average latency [1]. Recent research in URLLC has been focused on increasing the reliability by splitting messages into smaller data packets and adapting the modulation coding scheme (MCS) for each data packet. The introduction of short transmission time intervals (TTIs) plays an important role [2], which offers a high flexibility in the trade-off between error probability, spectral efficiency and latency.

The authors of [2] proposed a scheduling algorithm for messages with different priorities. The scheduling algorithm selects an MCS such that each transmission has a fixed low error probability and the MU is selected according to its priority. Scheduling URLLC messages with MU-specific reliability requirements is analyzed in [3]. The authors propose a knapsack-based solution to fulfill the reliability requirement of each MU, which is asymptotically optimal. Scheduling URLLC traffic for multiple MUs and multiple channels is analysed in [4], where messages can be duplicated and distributed between channels to improve the reliability. A low-complexity greedy solution is derived for the scheduling decisions. Scheduling URLLC messages in networks with multiple cells has been analysed in [5]. A centralized algorithm for MU association is proposed, and afterwards, each AP schedules messages individually to the associated MUs. The MCS is adapted such that each transmission has a fixed error probability.

Even though the mentioned works contributed significantly to the field of scheduling for URLLC, they also have their own limitations. In this work, hard deadlines of the messages are considered, meaning that a message is only useful for the MU when it completely arrives before the deadline. This is in contrast to [6], which focused on minimizing the latency of a percentile of late messages. We assume a periodic message arrival, as it is typical for wireless networked control systems [7] or age of information minimization [8]. Furthermore, we consider MU-specific deadlines, which has not been considered in [3], [4]. In contrast to [3], [4], we assume that the channel coherence time is shorter than the deadline of a message. To improve the performance of the scheduling, we incorporate a model of the probability density function of the wireless channel gains in the scheduling decision. We allow any probability distribution of the channel coefficients for each MU, not only simplified channel models as in [4].

The contributions of this work are the following. We consider an URLLC downlink scenario. Messages arrive periodically at the AP to be transmitted to the MUs within a MU-specific hard deadline. We formulate an average timely throughput maximization problem as an infinite time horizon Markov Decision Process (MDP). We prove that the MDP has a periodic structure in the state space. Based on the periodicity, we show that it is sufficient to solve a timely throughput optimization problem for one period of the infinite time horizon problem. We propose to decompose the resulting MDP into a novel two-timescale problem: An inter-frame problem where only stochastic knowledge from previous observations of the channel is available and an intra-frame problem where deterministic channel state information (CSI) from the channel estimation is available. We propose a novel URLLC scheduling algorithm based on two timescales, termed reliable two-timescale scheduling (RTTS).

The rest of this paper is organized as follows. In Section II and III, we introduce the system model and describe the problem formulation of the periodic MDP. The decomposition of the MDP into two timescales is formulated in Section IV and the proposed RTTS algorithm is presented in Section V.

This work was funded by the German Research Foundation (DFG) as a part of the projects B3, C1 and T2 within the Collaborative Research Center (CRC) 1053 - MAKI and has been supported by the BMBF project Open6GHub.

II. System model

This work considers a shared wireless downlink scenario with one AP and K MUs, as depicted in Figure 1. We assume that the transmission from the AP to the MUs is performed in frames [9]. The current frame is identified by its index m. Messages arrive periodically with an MU-specific period duration of λ_k frames at the AP to be transmitted to the corresponding MU k. The message has an MU-specific size of L_k bits and an MU-specific deadline of d_k frames. A new message is available at the AP for MU k directly after the deadline d_k of the previous message is exceeded, so we assume that the period duration λ_k is the same as the deadline d_k. The AP has K data buffers, one for each MU k, and the number of bits in this buffer is denoted as $x_{k,m}$. When a message arrives for MU k, the number $x_{k,m}$ of bits in the data buffer is equal to the message size L_k. After the deadline of a message exceeded, the remaining bits of the message are removed from the data buffer at the AP. Therefore, at maximum one message is stored in the data buffer for each MU at the AP. Furthermore, the time remaining until the deadline of the message for MU k in frame m will be denoted as $\tau_{k,m}$.

At the beginning of each frame m, the channel gain $h_{k,m} \in \mathbb{C}$ between the AP and MU k is available at the AP. The wireless channel is assumed to follow an i.i.d. block fading model, where the channel gain $h_{k,m}$ remains constant during one frame. The AP transmits with a constant transmission power P_{AP} and the power of the additive white Gaussian noise (AWGN) at the receiver is σ^2. Consequently, the SNR at MU k in frame m is $\gamma_{k,m} = (P_{\text{AP}} \cdot |h_{k,m}|^2)/\sigma^2$. After each frame, a new value for the SNR $\gamma_{k,m}$ at MU k is drawn from a random distribution $p_\gamma^k(\gamma)$.

Each frame m consists of T transmission time intervals (TTIs) with the same duration, and each TTI contains n modulation symbols. The transmission of a message is done in data packets, where only one data packet can be transmitted in one TTI. The scheduling algorithm has to decide in each frame m how many data packets are transmitted to MU k, which is denoted as $c_{k,m}$. Further, the number $b_{k,m}$ of bits in each data packet for MU k in frame m can be selected by adapting the MCS. After the transmission of a frame, an acknowledgement (ACK) is received from each MU, which is assumed to be error-free. Further, the time $\tau_{k,m}$ remaining until the deadline is decreased by one frame duration, i.e.

$$\tau_{k,m+1} = \begin{cases} \tau_{k,m} - 1, & \text{if } \tau_{k,m} > 0 \\ d_k, & \text{if } \tau_{k,m} = 0 \end{cases} \tag{1}$$

for each message which is available at the AP. If the deadline of the message is exceeded, i.e. $\tau_{k,m} = 0$, a new message with deadline d_k and size L_k arrives in the next frame.

The number of correctly received data packets of MU k is denoted as $r_{k,m}$ which is a random variable between 0 and the number $c_{k,m}$ of transmitted data packets. The successfully transmitted data packets are removed from the data buffer of the respective MU at the AP. The remaining bits are available for retransmission in the next frame. The number of bits $x_{k,m}$ in the data buffer of MU k is updated after each frame m according to

$$x_{k,m+1} = \begin{cases} x_{k,m} - r_{k,m} \cdot b_{k,m}, & \text{if } \tau_{k,m} > 0 \\ L_k, & \text{if } \tau_{k,m} = 0. \end{cases} \tag{2}$$

When a message of L_k bits arrives at the AP for MU k, the number of bits in the data buffer $x_{k,m}$ is set to L_k bits. A message is transmitted successfully only if the MU k receives the whole message error-free before the deadline d_k.

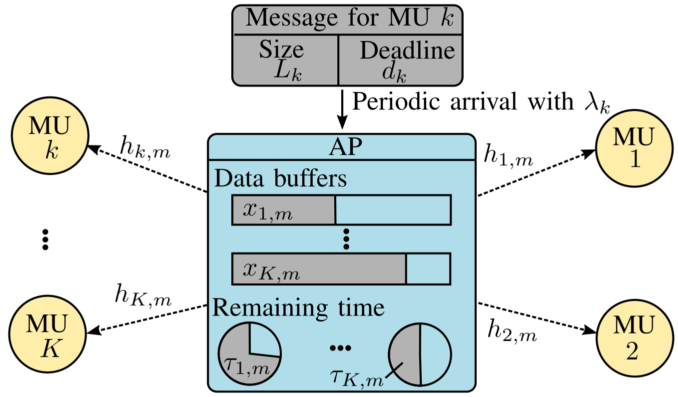

Fig. 1. URLLC scenario with periodic message arrival.

As the number of modulation symbols n is small in each TTI, the error probability for short data packets has to be considered [10]. The error probability of transmitting a data packet with $b_{k,m}$ bits with the SNR $\gamma_{k,m}$ can be approximated by

$$p_e(\gamma_{k,m}, b_{k,m}) \approx Q\left(\frac{n \log_2(1+\gamma_{k,m}) - b_{k,m} + \frac{1}{2}\log_2 n}{\sqrt{(1 - \frac{1}{(1+\gamma_{k,m})^2})n}}\right), \tag{3}$$

where $Q(.)$ is the Gaussian Q-function, n is the number of modulation symbols [10].

The scheduling algorithm selects one of the following transmission modes for each message. The transmission mode is determined based the number of bits $b_{k,m}$ per data packet and the number $c_{k,m}$ of data packets:

1) *Message transmission with potential retransmission:* The whole message of L_k bits is transmitted in a single data packet ($c_{k,m} = 1$). The data packet contains $b_{k,m} = L_k$ bits. In case of a transmission error, the message remains in the data buffer for potential retransmission.
2) *Message duplication:* The whole message of L_k bits is transmitted in multiple data packets ($c_{k,m} > 1$) in frame m. This increases the probability that the transmission is successfull, as it is sufficient that one of the data packets is received.
3) *Message splitting:* The message of L_k bits is transmitted in multiple smaller data packets ($c_{k,m} > 1$) with a number of bits $b_{k,m} < L_k$. These data packets can be distributed between several frames or sent in the same frame.

4) *Message dropping:* The message is proactively removed at the AP. It can be a good decision to remove single messages which incur a high resource demand for transmission [11].

III. Problem Formulation

In this section, the scheduling problem for timely throughput maximization is formulated. First, we identify the objective function of our problem and then, we present an infinite-time horizon MDP formulation. Afterwards, we show the periodicity of the MDP and formulate a finite-horizon MDP for one period.

A. Objective function

Let $y_{k,m}$ denote a binary random variable that equals 1 if MU k successfully received the intended message of L_k bits in frame m before the deadline d_k, and 0 otherwise. Formally,

$$y_{k,m} = \begin{cases} 1, & \text{if } x_{k,m} - r_{k,m} \cdot b_{k,m} = 0 \text{ and } \tau_{k,m} = 0 \\ 0, & \text{else} \end{cases} \tag{4}$$

As proposed in [12], the timely throughput R is defined as the number of messages arriving before the respective deadline. As we consider an infinite time horizon, the timely throughput is divided by the number of messages that arrived at the AP to obtain an average timely throughput $\bar{R}$. The number $q_{M,k}$ of messages which arrives for MU k in a duration of M frames is given by the quotient of the duration M and the periodicity λ_k of the message arrival, which is denoted as $q_{M,k} = \lceil \frac{M}{\lambda_k} \rceil$. Consequently, the average timely throughput $\bar{R}$ at the AP for an infinite time horizon is given as

$$\bar{R} = \lim_{M\to\infty} \sum_{k=1}^{K} \frac{\sum_{m=1}^{M} y_{k,m}}{q_{M,k}} \tag{5}$$

which is the proportion of messages that arrived error-free before their deadline. We assume that every message has the same importance for the communication system, no weight is assigned to the individual messages.

B. Markov Decision Process for Timely Throughput

An MDP is defined by a state set $\mathcal{S}$, a set $\mathcal{A}$ of actions, a transition model $\mathcal{P}$ and a set $\mathcal{R}$ of rewards [13]. Let $\boldsymbol{x}_m = [x_{0,m}, \dots, x_{K,m}]$ denote the vector with the number of bits in each data buffer. Let $\boldsymbol{\tau}_m = [\tau_{0,m}, \dots, \tau_{K,m}]$ denote the vector with the remaining times until the deadlines of all MUs. We denote the vector with all channel coefficients in frame m by $\boldsymbol{h}_m = [h_{0,m}, \dots, h_{K,m}]$. In frame m, the state $\boldsymbol{s}_m \in \mathcal{S}$ is composed of $\boldsymbol{x}_m$, $\boldsymbol{\tau}_m$ and $\boldsymbol{h}_m$.

In each frame m, the AP transmits $c_{k,m}$ data packets to MU k and selects a number $b_{k,m}$ of bits in each data packet for MU k. Actions $\boldsymbol{a}_m \in \mathcal{A}$ in each frame correspond to scheduling actions $\boldsymbol{a}_m = [c_{k,m}, b_{k,m}]$. After the frame was transmitted, the new state $\boldsymbol{s}_{m+1}$ is determined by considering the number of data packets $r_{k,m}$ that were successfully received (2).

As we consider AWGN, the success of decoding of one of the data packets is independent of the other data packets. The transition probability $P(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a}) \in \mathcal{P}$ is the probability of decoding $r_{k,m}$ data packets successfully in case that $c_{k,m}$ data packets with $b_{k,m}$ bits per data packet were transmitted. This probability is given by a Bernoulli distribution

$$P^{\text{MDP}}(r_{k,m}|c_{k,m}) = \binom{c_{k,m}}{r_{k,m}} p_e^{c_{k,m}-r_{k,m}} \cdot (1-p_e)^{r_{k,m}}, \tag{6}$$

where p_e denotes the error-probability in (3) when transmitting $b_{k,m}$ bits in a data packet. The reward $Y_m \in \mathcal{R}$ in frame m is the number

$$Y_m = \sum_{k=0}^{K} y_{k,m} \tag{7}$$

of messages that arrived before their deadline.

We denote a sequence of actions as $\{\boldsymbol{a}_m\}$ and the optimal sequence of actions as $\{\boldsymbol{a}_m\}^*$. The objective is to maximize the average timely throughput $\bar{R}$ of all K MUs. The corresponding optimization problem is formulated as

$$\{\boldsymbol{a}_m\}^* = \underset{\{\boldsymbol{a}_m\}}{\arg\max} \quad \lim_{M\to\infty} \mathbb{E}\left\{ \sum_{k=1}^{K} \frac{\sum_{m=1}^{M} y_{k,m}}{q_{M,k}} \right\} \tag{8}$$

$$\text{s.t.} \quad \sum_{k=0}^{K} c_{k,m} \leq T, \quad \forall m \tag{8a}$$

$$b_{k,m} \cdot c_{k,m} \leq x_{k,m}, , \quad \forall k, m \tag{8b}$$

The problem (8) is identified as an infinite time horizon MDP with average reward criterion [13]. Constraint (8a) states that only T TTIs are available in each frame, and therefore only T data packets can be transmitted in a frame. Constraint (8b) states that the number of transmitted bits is not larger than the number $x_{k,m}$ of bits in the data buffer of each MU.

C. Periodicity of the MDP

We observe that frames exist where a new message arrives for each MU k. This is the case every

$$Q = \text{lcm}(\lambda_1, \dots, \lambda_K) \tag{9}$$

frames, where Q is the least common multiple (lcm) of all periods λ_k of packet arrival.

Theorem 1. *The timely throughput* (5) *in a period of Q frames is independent from the scheduling decisions $\{\boldsymbol{a}_m\}$ from the previous periods and does not affect the scheduling decision in the next periods of Q frames.*

Proof: The state $\boldsymbol{s}_Q$ in frame Q is independent of the state $\boldsymbol{s}_{Q-1}$ in the previous frame. We will show this for each part of the state space. We can show that $\boldsymbol{x}_Q = [L_0, \dots, L_k]$ and $\boldsymbol{\tau}_Q = [d_0, \dots, d_k]$ as each MU's deadline exceeded before the end of frame $Q-1$. All periods λ_k of packet arrival are divisors of Q by definition from (9), so a new message is available at the AP for each MU. The channel coefficients are independently drawn at each frame m, so the channel coefficient state $\boldsymbol{h}_Q$ is independent of $\boldsymbol{h}_{Q-1}$.

By Theorem 1, it is sufficient to maximize the timely throughput individually for each period of Q frames of the

MDP. We formulate the optimization problem for one period of the MDP as:

$$\{\boldsymbol{a}_m\}^* = \underset{\{\boldsymbol{a}_m\}}{\arg\max} \quad \mathbb{E}\left\{\sum_{m=1}^{Q} Y_m\right\} \tag{10}$$

$$\text{s.t.} \quad \text{(8a), (8b).} \tag{10a}$$

IV. Reformulation of the Timely Throughput problem

In this section, a decomposition of (10) into the following two sub-problems is proposed: 1) The inter-frame problem and 2) the intra-frame problem. The inter-frame problem is to estimate the long-term reward of scheduling actions which is given by a state-value function in the timescale of frames. The time horizon of this problem includes future frames for which only the distributions of the channel gains are known. As the random process of channel gains is stationary, the solution to this problem can be calculated offline, meaning before the scheduling starts. In Section IV-B we formulate the intra-frame problem, which is in the timescale of TTIs. This problem is to select the scheduling action $\boldsymbol{a}_m$ for the current frame when full knowledge of the current channel state $\boldsymbol{h}_m$ is available. To estimate the influence of $\boldsymbol{a}_m$ on the timely throughput in future frames, the state-value function from the inter-frame problem is required.

A. Inter-frame problem

Let $V(\boldsymbol{x}_m, \boldsymbol{\tau}_m)$ denote a state-value function which estimates the expected reward in the current period of the MDP when starting with $\boldsymbol{x}_m$ bits in the data buffers and the times $\boldsymbol{\tau}_m$ remaining until the deadlines. For each scheduling action $\boldsymbol{a}_m$, there are two possibilities: 1) With probability P_a the channels are good enough to transmit $b_{k,m} \cdot c_{k,m}$ bits to MU k, or 2) with probability $1 - P_a$ the channels do not support the transmission in T data packets. In case 1) the number of bits $\boldsymbol{x}'_{m+1}$ in the data buffer is reduced by the number of transmitted bits. The reward (7) in case of a successful transmission is denoted as Y'_m. In case 2) the number of bits $\boldsymbol{x}_{m+1} = \boldsymbol{x}_m$ in the data buffer remains the same. In both cases, the times $\tau_{k,m}$ remaining until the deadlines are reduced according to (1). The recursive relation of the expected timely throughput can be formulated as

$$\begin{aligned} V(\boldsymbol{x}_m, \boldsymbol{\tau}_m) = \max_{\boldsymbol{a}_m} \; & P_a \cdot (Y'_m + V(\boldsymbol{x}'_{m+1}, \boldsymbol{\tau}_{m+1})) \\ & + (1 - P_a) \cdot (Y_m + V(\boldsymbol{x}_{m+1}, \boldsymbol{\tau}_{m+1})) \end{aligned} \tag{11}$$

In the following steps the probability P_a is approximated using the knowledge of $p^k_\gamma(\gamma)$. For a given action $\boldsymbol{a}_m$, each MU transmits $c_{k,m}$ data packets with $b_{k,m}$ bits per data packet. The required SNR $\gamma_{k,m}$ to transmit with b bits in a data packet with n modulation symbols is approximated [14] as

$$\gamma_{\min}(b) = 2^{\frac{b}{n}} - 1. \tag{12}$$

The probability $P^k_b(b)$ that MU k can transmit with rate b in frame m can be estimated as:

$$P^k_b(b) = \int_{\gamma_{\min}(b)}^{\infty} p^k_\gamma(\gamma)\, \mathrm{d}\gamma. \tag{13}$$

Let n_k denote the random variable that indicates how many TTIs are required to transmit a data packet with the size $b_{k,m} c_{k,m}$ successfully to MU k. The probability of n_k being equal to n can be calculated as

$$P^k_n(n) = P^k_b\left(\frac{b_{k,m} c_{k,m}}{n}\right). \tag{14}$$

The probability distribution of the sum of independent random variables $N = \sum_{k=1}^{K} n_k$ is given as the convolution of the individual probability distributions

$$P_N(N) = P^k_n(n) * \cdots * P^K_n(n). \tag{15}$$

From the distribution of the required TTIs n_k for a successful transmission, the approximation for the probability P_a is derived. The probability that action $\boldsymbol{a}_m$ is successful is approximated by the sum of all probabilities

$$P_a \approx \sum_{n=0}^{T} P_N(N = n) \tag{16}$$

that not more than T TTIs are required for successfull transmission. The solution of the inter-frame problem is $V(\boldsymbol{x}_m, \boldsymbol{\tau}_m)$ from (11), which gives the expected reward Y_m when starting a with $\boldsymbol{x}_m$ bits in the data buffers and the remaining times $\boldsymbol{\tau}_m$ until the deadline.

B. Intra-frame problem

In the intra-frame problem, the solution from the inter-frame problem $V(\boldsymbol{x}_m, \boldsymbol{\tau}_m)$ is used. We optimize the scheduling decision such that the expected value of the reward Y_m of the current frame m and the state-value function $V(\boldsymbol{x}_{m+1}, \boldsymbol{\tau}_{m+1})$ of the next frame $m + 1$ is maximized.

After obtaining the channel estimate $h_{k,m}$ for the current frame m, the number $b_{k,m}$ of bits per data packet is selected such that the expected throughput is maximized [3]

$$b^*_{k,m} = \underset{b}{\arg\max} \quad (1 - p_e(\gamma_{k,m}, b)) \cdot b. \tag{17}$$

To find the optimal allocation of TTIs $\boldsymbol{c}_m = [c_{1,m}, \ldots, c_{K,m}]$, the intra-frame problem

$$\boldsymbol{c}^*_m = \underset{\boldsymbol{c}_m}{\arg\max} \sum_{\boldsymbol{r}_m \in \mathcal{R}} P^{\text{MDP}}(\boldsymbol{r}_m | \boldsymbol{c}_m) \cdot (Y'_m + V(\boldsymbol{x}'_{m+1}, \boldsymbol{\tau}_{m+1})) \tag{18}$$

$$\text{s.t.} \quad \text{(8a), (8b),}$$

has to be solved. The values Y'_m and $\boldsymbol{x}'_{m+1}$ denote the reward and number of bits in the data buffer for the case that $\boldsymbol{r}_m$ data packets are received, respectively. The transition probability is denoted by $P^{\text{MDP}}(\boldsymbol{r}_m | \boldsymbol{c}_m)$ from (6).

V. Two-Timescale Scheduling Algorithm

Our algorithm, termed reliable two-timescale scheduling (RTTS) consists of two parts: In Section V-A, the inter-frame value iteration is presented, which calculates the state-value function (11) under the uncertainty of the channel state $\boldsymbol{h}_m$ in future frames. The selection how many data packets are transmitted to MU k is done on the timescale of TTIs in the intra-frame scheduling algorithm, which is described in Section V-B.

Algorithm 1 Inter-Frame Value Iteration

1: Input K, T, L_k, n, d_k
2: Determine Q ▷ Eq. (9)
3: Initialize $V(\boldsymbol{x}_m, \boldsymbol{\tau}_m) = 0$
4: **for** $a \in \mathcal{A}$ **do**
5: Calculate transition probability P_a ▷ Eq. (16)
6: **end for**
7: **for** $i \in \{0, \ldots, Q\}$ **do** ▷ Timescale 1: Frames
8: Calculate $\boldsymbol{\tau}_{Q-i}$
9: **for all** $\boldsymbol{x}_{Q-i} \in \mathcal{X}$ **do**
10: Update $V(\boldsymbol{x}_{Q-i}, \boldsymbol{\tau}_{Q-i})$ ▷ Eq. (11)
11: **end for**
12: **end for**
13: return $V(\boldsymbol{x}_m, \boldsymbol{\tau}_m)$

Algorithm 2 Intra-Frame Scheduling Algorithm

1: Input K, T, L, n, d_k
2: Input $V(\boldsymbol{x}_m, \boldsymbol{\tau}_m)$ from algorithm 1
3: **for** each frame m **do**
4: Obtain channel estimate $h_{k,m}$ from each MU k
5: Select number $b^*_{k,m}$ of bits per data packet for each MU ▷ Eq. (17)
6: Initialize $c_k = 0$
7: **for** $t \in \{1, \ldots, T\}$ **do** ▷ Timescale 2: TTIs
8: Select MU k_0 ▷ Eq. (19)
9: $c_{k_0} \leftarrow c_{k_0} + 1$
10: **end for**
11: Transmit frame with $b^*_{k,m}$ bits per data packet and c_k data packets for MU k
12: Receive ACK with $r_{k,m}$ from each MU
13: Update remaing time $\boldsymbol{\tau}_{m+1}$ until the deadline of MU k ▷ Eq. (1)
14: Update the number of bits in the data buffer $\boldsymbol{x}_{m+1}$ ▷ Eq. (2)
15: **end for**

A. Timescale 1: Inter-Frame Value Iteration

The proposed inter-frame value iteration is summarized in Algorithm 1. After the input of the system parameters, the periodicity Q of the MDP is determined (line 3). Then for each action $\boldsymbol{a}_m \in \mathcal{A}$, the probability P_a is calculated and stored in a table (line 4-6). An iteration over all frames in the period Q of the MDP starts. Firstly, the deadline of each message in frame $Q-i$ is calculated (line 8). Afterwards, the state-value function of all states $\boldsymbol{x}_{Q-i}$ is updated and stored in a table (line 10). To overcome limitations by large state or action spaces, approximation techniques, e.g. linear function approximation or quantization, can be used in lines 5 and 10. The result of the Inter-Frame Value Iteration is available for the inter-frame problem.

B. Timescale 2: Intra-Frame Scheduling

The proposed intra-frame scheduling is summarized in Algorithm 2. The algorithm starts with the input of the system parameters and the results from the inter-frame value iteration, which is stored in a lookup table. First, the channel estimate $h_{k,m}$ of the current frame is obtained from each MU and the number $b^*_{k,m}$ of bits per data packet is selected accordingly (line 4 and 5). Then a greedy approximate solution to problem (6) is used to determine for each TTI the MU k which maximizes the expected timely throughput (line 8). We start with the first TTI $t = 0$ and select the MU k_t which maximizes

$$k_t = \arg\max_k \sum_{r_m \in \mathcal{R}} P^{\text{intra}}(\boldsymbol{r}_m | c_k) \cdot (Y'_m + V(\boldsymbol{x}'_{m+1}, \boldsymbol{\tau}_{m+1}))$$
$$\text{subject to} \quad b^*_{k,m} \cdot c_k \leq x_{k,m} \tag{19}$$

One TTI is then allocated to the MU k_t therefore c_{k_t} is increased by one (line 9). These steps (line 7-9) are repeated T times until all TTIs are allocated. After all the TTIs are allocated, the frame is transmitted by the AP, and then the ACK from each MU is received (line 11 and 12). The deadlines and data buffers are updated at the AP (line 13 and 14). This procedure is repeated for each frame m.

VI. Numerical evaluation

A. Simulation Setup

In this section, the numerical results for the evaluation of the proposed RTTS algorithm are presented. We consider a single-cell scenario with one AP, which serves $K = 4$ MUs. Orthogonal Frequency Division Multiplexing (OFDM) is used in the downlink with a frame duration of $T_f = 0.5\,\text{ms}$. Each OFDM frame is separated into $T = 3$ TTIs with $n = 168$ modulation symbols each. The channel gains $h_{k,m}$ are taken from an i.i.d. Rayleigh fading process with a variance of the average SNR of the MU. We introduce two types of MUs: MU type 1 [9] has to receive a message of $L_k = 32$ bytes before the deadline $d_k = 1\,\text{ms}$ and MU type 2 has to receive $L_k = 48$ bytes before the deadline $d_k = 1.5\,\text{ms}$. The following results are obtained by simulating $M = 10^5$ frames.

For comparison purposes, besides the proposed RTTS algorithm, two State-of-the-Art algorithms are implemented. The *Modified Largest Weighted Delay First (M-LWDF)* is a popular channel and QoS aware scheduling algorithm [15]. M-LWDF schedules MUs with the highest M-LWDF metric greedily. The metric $\delta_{k,m}$ for a MU k in frame m is calculated by

$$\delta_{k,m} = \frac{b^*_{k,m}}{\bar{b}_k} \cdot \frac{d_k - \tau_{k,m}}{d_k} \tag{20}$$

using the number of frames until the deadline d_k, average throughput $\bar{b}_k$ and the current optimum number $b^*_{k,m}$ of bits per data packet. The average throughput $\bar{b}_k$ is estimated in each frame as $\bar{b}_k \leftarrow (1-\alpha) \cdot \bar{b}_k + \alpha \cdot b^*_{k,m}$, with $\alpha = 0.05$.

A *Framewise-Knapsack* solution was proposed in [3]. A value is associated to each MU which is proportional to the number of messages this MU hasn't successfully received in past frames. The weight for each MU is given by $\lceil x_{k,m}/b^*_{k,m} \rceil$. Each frame, a set of MUs has to be fitted in a knapsack with a capacity of T such that the sum of the values is maximized.

B. Numerical results

Figure 2 shows the loss rate of messages for increasing average SNRs with the same average SNR for each MU. The proposed RTTS algorithm requires $3\,\text{dB}$ $(2.5\,\text{dB})$ less SNR than the M-LWDF algorithm (Framewise-Knapsack) for a loss rate of messages of 10^{-3}. With an SNR of $0\,\text{dB}$, the proposed RTTS algorithm has a timely throughput which is 44% (9.5%) higher than the M-LWDF algorithm (Framewise-Knapsack). At low SNR values, the proposed RTTS algorithm proactively drops messages according to the state-value function and therefore avoids partial transmission. M-LWDF allocates the TTIs according to the metric $\delta_{k,m}$, which can lead to partial transmission. To avoid this problem, the Framewise-Knapsack

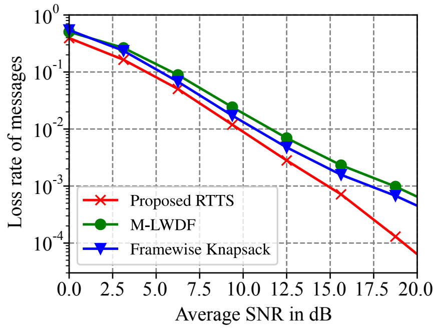

Fig. 2. Loss rate of messages vs. average SNR for 2 MUs of type 1 and 2 MUs of type 2

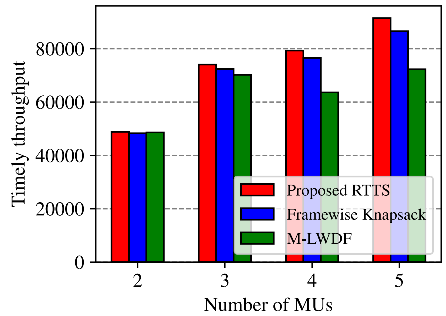

Fig. 3. Timely throughput for K MUs with average SNR of $5\,\mathrm{dB}$. Half of the MUs are of type 1.

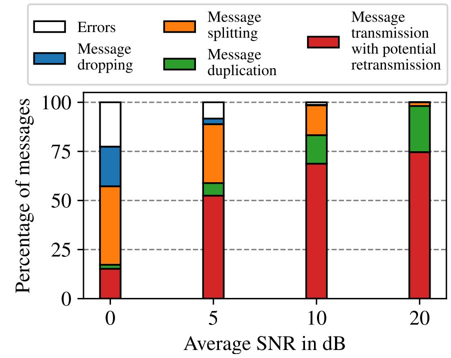

Fig. 4. Transmission modes of the proposed RTTS vs. average SNR

approach always tries to transmit a whole message, therefore in the case of hard deadlines it performs better than the M-LWDF. Nevertheless, the Framewise-Knapsack algorithm does not take the average channel quality into account, which causes loss of messages if the channel changes during the deadline.

Figure 3 shows the timely throughput for an increasing number of MUs with a fixed average SNR for all MUs. Half of the MUs are of type 1. At $K = 5$ MUs, the proposed RTTS algorithm has 6% and 26.5% more timely throughput than the M-LWDF and Framewise-Knapsack respectively. The performance of the M-LWDF can decrease with more MUs, e.g. seen between $K = 3$ and 4 MUs, because the number of partially transmitted messages increases.

Figure 4 shows the distribution of transmission modes from Section II for different average SNRs for RTTS. At low SNR values, the proposed RTTS algorithm can proactively drop messages to avoid partial transmission of messages. Furthermore, over 30% of the messages are split into smaller data packets for an increased reliability. Only 13% of the messages are transmitted without splitting or duplication. With increasing SNR values, the message transmission and message duplication increases. With higher possible transmission rates on the channel, RTTS selects message duplication to increase the reliability.

VII. Conclusion

In this work, we studied an URLLC scenario with MU-specific deadlines and MU-specific message sizes for MUs with periodic traffic. In addition, we considered the error probability of the transmission of short data packets. Our goal was to find a scheduling algorithm to maximize the timely throughput of messages. To this aim, we formulated the scheduling problem as an infinite-time horizon MDP and showed that the MDP is periodic. We propose to decompose the problem into two sub-problems with different timescales: The inter-frame problem and the intra-frame problem. The inter-frame problem is the estimation of the long-term reward of scheduling actions, which is stored in a state-value function in the timescale of frames. The intra-frame problem is to select which data packet is transmitted on the timescale of TTIs. To solve the two sub-problems, we proposed a novel two-timescale scheduling algorithm, which maximizes the expected average timely throughput of periodic messages with hard deadlines. Through numerical simulations we showed that the proposed two-timescales algorithm outperforms the reference algorithms.

References

[1] M. Bennis, M. Debbah, and H. V. Poor, "Ultrareliable and Low-Latency Wireless Communication: Tail, Risk, and Scale," *Proceedings of the IEEE*, vol. 106, no. 10, pp. 1834–1853, 2018.

[2] G. Pocovi, K. I. Pedersen, and P. Mogensen, "Joint Link Adaptation and Scheduling for 5G Ultra-Reliable Low-Latency Communications," *IEEE Access*, vol. 6, pp. 28 912–28 922, 2018.

[3] A. Destounis, G. S. Paschos, J. Arnau, and M. Kountouris, "Scheduling URLLC users with reliable latency guarantees ," in *2018 16th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, 2018, pp. 1–8.

[4] N. Ben Khalifa, V. Angilella, M. Assaad, and M. Debbah, "Low-Complexity Channel Allocation Scheme for URLLC Traffic," *IEEE Transactions on Communications*, vol. 69, no. 1, pp. 194–206, 2021.

[5] A. Karimi, K. I. Pedersen, N. H. Mahmood, J. Steiner, and P. Mogensen, "5G Centralized Multi-Cell Scheduling for URLLC: Algorithms and System-Level Performance," *IEEE Access*, vol. 6, pp. 72 253–72 262, 2018.

[6] J. Choi, "On Delay-Constrained Transmissions of a Finite Number of Short-Length Packets," *IEEE Access*, vol. 9, pp. 1005–1015, 2021.

[7] R. Jurdi, S. R. Khosravirad, H. Viswanathan, J. G. Andrews, and R. W. Heath, "Outage of Periodic Downlink Wireless Networks With Hard Deadlines," *IEEE Transactions on Communications*, vol. 67, no. 2, pp. 1238–1253, 2019.

[8] J. P. Champati, H. Al-Zubaidy, and J. Gross, "Statistical Guarantee Optimization for AoI in Single-Hop and Two-Hop FCFS Systems With Periodic Arrivals," *IEEE Transactions on Communications*, vol. 69, no. 1, pp. 365–381, 2021.

[9] 3GPP, "Summary of Rel-15 Work Items (Release 15)," Tech. Rep., 2019, 3GPP TR 21.915 V15.0.0.

[10] Y. Polyanskiy, H. V. Poor, and S. Verdu, "Channel Coding Rate in the Finite Blocklength Regime," *IEEE Transactions on Information Theory*, vol. 56, no. 5, pp. 2307–2359, 2010.

[11] B. Chang, G. Zhao, Z. Chen, L. Li, and M. A. Imran, "Packet-Drop Design in URLLC for Real-Time Wireless Control Systems," *IEEE Access*, vol. 7, pp. 183 081–183 090, 2019.

[12] I. Hou and P. R. Kumar, "Real-time communication over unreliable wireless links: a theory and its applications," *IEEE Wireless Communications*, vol. 19, no. 1, pp. 48–59, 2012.

[13] M. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, ser. in Probability and Statistics. Wiley, 2014.

[14] P. Mary, J.-M. Gorce, A. Unsal, and H. V. Poor, "Finite Blocklength Information Theory: What Is the Practical Impact on Wireless Communications?" in *2016 IEEE Globecom Workshops*, 2016, pp. 1–6.

[15] D. Nguyen, H. Nguyen, and E. Renault, "E-MQS - A New Downlink Scheduler for Real-Time Flows in LTE Network," in *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*, 2016, pp. 1–5.