Scheduling URLLC Users with Reliable Latency Guarantees

Apostolos Destounis, Georgios S. Paschos, Jesus Arnau, Marios Kountouris
Mathematical and Algorithmic Sciences Lab, Paris Research Center, Huawei Technologies Co. Ltd.
Arcs de Seine, 20, quai du Point du Jour, 92100, Boulogne-Billancourt, France
email: firstname.lastname@huawei.com

***Abstract*—This paper studies Ultra-Reliable Low-Latency Communications (URLLC), an important service class of emerging 5G networks. In this class, multiple unreliable transmissions must be combined to achieve reliable latency: a user experiences a frame success when the entire L bits are received correctly within a deadline, and its latency performance is reliable when the frame success rate is above a threshold. When jointly serving multiple users, a natural URLLC scheduling question arises: given the uncertainty of the wireless channel, can we find a scheduling policy that allows all users to meet a target reliable latency objective? This is called the *URLLC SLA Satisfaction* (*USS*) problem. The *USS* problem is an infinite horizon constrained Markov Decision Process, for which, after establishing a convenient property, we are able to derive an optimal policy based on dynamic programming. Our policy suffers from the curse of dimensionality, hence for large instances we propose a class of *knapsack-inspired* computationally efficient - but not necessarily optimal - policies. We prove that every policy in that class becomes optimal in a fluid regime, where both the deadline and L scale to infinity, while our simulations show that the policies perform well even in small practical instances of the *USS* problem.**

I. Introduction

Ultra-reliable low-latency communications (URLLC) refers to a use case of emerging 5G wireless networks characterized by small amount of data and strict latency and reliability constraints. Such high fidelity communications are essential to invite *vertical applications* to mobile networks, such as industry 4.0 and autonomous transportation to name a few, and to enable real-time applications like reactive virtual reality and remote surgery [1]. According to [2], 5G reliability is defined as the success probability of transmitting a layer 2/3 packet within a required user plane latency, which is the time it takes to deliver a packet from the radio protocol layer 2/3 ingress point to the radio protocol layer 2/3 egress point of the radio interface. The associated requirement for the URLLC use case is $1-10^{-5}$ probability of receiving a layer 2 protocol data unit of 32 bytes within 1 ms, i.e., a delayed delivery beyond 1 ms is allowed only once every 10000 packets.

In pursuit of such extreme requirements for latency and reliability, a key challenge is how to strategically arrange short-packet transmissions in order to offer guaranteed overall latency when the success of each individual transmission is intrinsically unreliable. The main approach is to diversify transmissions over resources, including multiple antennas, codes, frequencies, and time slots. In this paper we consider the case where the resource diversification is over time; we will design multiple retransmissions in order to achieve the required reliable performance.

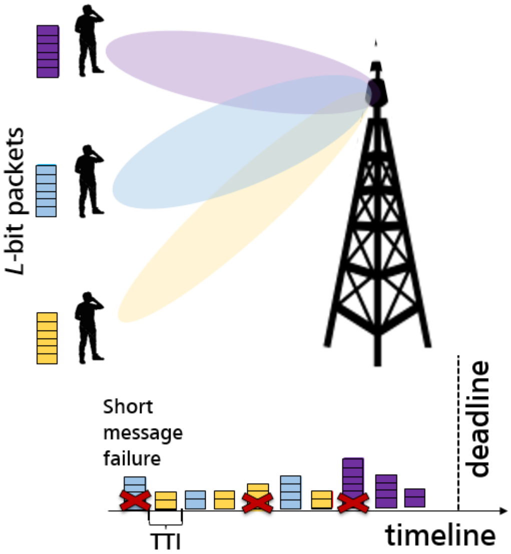

Fig. 1. A base station transmitting short packets to multiple users with the goal that L bits are received correctly before a deadline.

We study a K-user URLLC downlink as depicted in Fig.1. Time is divided in slots and a user is said to experience a *frame success* event if L bits are correctly received before the frame deadline of T slots. Accordingly, a user achieves its Service Level Agreement (SLA) if the empirical probability of frame success is higher than a given threshold. At each slot, a *scheduling policy* decides which user is served and how many bits are transmitted to that user; naturally, the higher the number of information bits in a given time span, the higher the probability of error. The objective of this paper is to design a scheduling policy that satisfies the SLAs of all users, whenever this is feasible. We call this problem the *URLLC SLA Satisfaction* problem, in short *USS*.

Our contributions can be summarized as follows:

- We define the region of reliable latencies $\mathcal{Q}$ as the set of all SLA vectors $\boldsymbol{q}^*$ which are achievable in a URLLC system by any scheduling policy, and we show that the optimal solution of *USS* can be achieved by Markovian policies.
- Exploiting the Markovian structure, we decompose the

USS problem without loss of optimality into (i) an *inter-frame policy* that dynamically adjusts user weights on a frame by frame basis and (ii) an *intra-frame policy* that solves a finite horizon weighted maximization, which is a Markov Decision Process (MDP). We thus provide a dynamic programming-based intra-frame policy, which combined with the dynamic weights optimally solves *USS*.

- Due to the well-known curse of dimensionality, the policy quickly becomes computationally intractable when the input K, L, T grows; motivated by this complexity, we propose a class of knapsack-inspired policies which are poly-time computable, and we prove that they are asymptotically ϵ–optimal in instances of *USS* where we must deliver νL bits before a νT deadline, and $\nu \to \infty$.

A. Related Work

Our work is related to the line of research in delay performance of queuing systems and scheduling policies. Prior work uses large deviations theory to examine the delay violation probability as the delay threshold becomes large [3] or by fixing the delay threshold and examining the delay violation probability as the users and the parallel channels grow large [4]. Two other systematic approaches are the theory of effective capacity/bandwidth, e.g. [5], [6], and the stochastic network calculus, e.g. [7], where approximations of the service and arrival processes are used to provide a tractable evaluation of the delay performance of resource allocation policies.

All the above references consider *soft* delay requirements, in the sense that a packet that exceeds the target delay still remains in the system and eventually gets delivered. Representative works on a hard deadline setting (where packets that did not meet their deadline are dropped) are [8] that analyzes the heuristic to schedule the packet closer to its deadline on a good channel state, and [9] which employs a dynamic programming problem formulation to obtain an efficient heuristic. Relatively recently, a unifying theory of resource allocation for deadline-constrained traffic has been developed, cf. [10], [11]. The main concept there is *timely throughput*, that is the fraction of packets that get delivered before their respective deadlines. This theory has been applied and extended to various settings, including scheduling under fading channels [12] and multicasting [13]. All aforementioned works consider either that a packet is transmitted in a single slot or that transmissions are without errors. Finally, a recent work [14] considers the problem of puncturing in OFDMA systems with URLLC and users with broadband traffic. The authors assume that URLLC users should be scheduled immediately on the time slot they arrive and the focus is on maximizing the utility of the broadband users subject to scheduling all arriving URLLC users.

In this paper, we expand the timely throughput framework to model URLLC systems accommodating transmissions of fragments of a packet, where the higher the number of information bits in a transmission the greater the probability of error. We show how this complicates the problem to the point of invalidating insights from previous works; for instance priority policies are no longer optimal for a large class of problems (see Example 1).

II. System Model

We consider a wireless downlink with one base station and K users, as in Fig. 1. Time is divided in slots and we impose strict latency constraints in the following manner: an L-bit packet has to be delivered to each user within T slots, else it is dropped[1]. More specifically, we split the L-bit packet into shorter messages containing a variable number of information bits, and we attempt several transmissions per user until the L bits are correctly received.

A. Communication Model

The wireless channel between the base station and each user is assumed to be i.i.d. block fading, staying constant within one frame and changing between frames.[2] We denote with $h_k[m]$ the channel state of user k at frame m, and with $\gamma_k[m] = P|h_k[m]|^2$ the corresponding receiver SNR, where P is the transmit SNR.

To model the transmission error probability, we focus on the fact that URLLC must rely on short-message transmissions which trade off blocklength (latency), efficiency, and probability of error. In classical information theory every rate below the capacity of a channel can be achieved with an arbitrarily low codeword error probability given sufficiently long messages. Here, however, we focus on transmitting *short messages*, where asymptotic information theoretic results do not apply [15]. Every transmission has an associated non-zero error probability, which in our case is approximated by [16]

$$p_e(\gamma, b) \approx Q\left(\frac{n \log_2(1+\gamma) - b + 0.5 \log_2 n}{\sqrt{V(\gamma) n}}\right) \qquad (1)$$

where n is the number of channel uses, b the number of transmitted bits, γ is the SNR, and where

$$V(x) = 1 - \frac{1}{(1+x)^2}$$

stands for the so-called *channel dispersion* and $Q(.)$ is the Gaussian Q-function. We remark that i) the above formula assumes perfect knowledge of γ at the transmitter and error-free notification of the transmission outcome, and ii) the algorithms we design are not limited to the model in (1).

B. Scheduling Reliable Latencies

At each slot t, the base station can schedule *only one* user, denoted with $k(t)$. The transmission carries $b(t)$ information bits, chosen from a finite set $\mathcal{B} = \{b^1, \ldots, b^{|\mathcal{B}|}\}$, as it is common in practical schemes with modulation and coding.

[1] Note that we implicitly assume that packets for all users arrive at the same time and all users have the same latency requirement, though the ideas presented in the paper can be extended to different requirements per user, cf. [12].

[2] Although our results are given for i.i.d. channels, we remark that they hold in the more general case where $\boldsymbol{h}[m]$ forms an ergodic Markov chain.

These two quantities, $k(t), b(t)$, constitute the scheduling variables at each slot. Formally:

Definition 1 (Scheduling Policy)**.** *A scheduling policy π is a (possibly randomized) rule to choose a pair $(k(t), b(t))$ at each slot t, where $k(t) \in \{1, \dots, K\}$ and $b(t) \in \mathcal{B}$.*

In more detail, denote with $H(t)$ the history of the system, which includes the channel SNRs, the transmission decision, and the transmission outcome of all slots up to t. A policy π specifies a history-dependent probability distribution $\boldsymbol{u}^\pi(t) = \boldsymbol{u}^\pi(H(t), t)$ where $u^\pi_{k,i}(H(t), t)$ is the probability of scheduling user k at rate b^i at slot t given the past history of the system.

We can now formalize the problem of achieving reliable latency performance. Let $s^\pi_k[m]$ be a binary variable that takes value 1 if in frame m user k successfully received the intended L bits before the latency deadline, and 0 otherwise. Clearly $s^\pi_k[m]$ is a random variable, so let us discuss how the success event $s^\pi_k[m] = 1$ is determined within a specific frame. Given a policy π, we introduce $x^\pi_k(t)$ to be *the number of bits remaining to be transmitted for the message of user k at the beginning of slot t*, and hence $x^\pi_k(mT^-)$ is the number of remaining bits at the end of frame m. Therefore, we have

$$s^\pi_k[m] = \begin{cases} 1 & \text{if } x^\pi_k(mT^-) = 0 \\ 0 & \text{if } x^\pi_k(mT^-) > 0. \end{cases} \tag{2}$$

Definition 2 (Reliable Latency)**.** *The reliable latency of user k under policy π is defined as:*

$$\overline{q}^\pi_k \triangleq \liminf_{M\to\infty} \frac{\sum_{m=0}^{M-1} \mathbb{E}\left\{s^\pi_k[m]\right\}}{M}.$$

A reliable latency $\overline{q}^\pi_k = 0.99$ guarantees that in 99 frames out of 100, user k receives the L bits with latency at most T. Reliable latency could be alternatively called (a) the empirical frequency of frame successes, or (b) *timely throughput* from the sequence of works in [10], [11].

C. Feasible Region of Reliable Latencies

Given an arbitrary system with unreliable individual transmissions, it is expected that not all possible reliable latency requirements are achievable. Let us formally define the feasible region of reliable latencies and the URLLC SLAs introduced in Sec I.

Definition 3 (Feasible Region of Reliable Latencies)**.** *Consider the set of all scheduling policies Π, and denote $\overline{\boldsymbol{q}}^\pi$ the reliable latency vector achieved by policy $\pi \in \Pi$. Then, the feasible region of reliable latencies, denoted with $\mathcal{Q}$, is the set of vectors*

$$\mathcal{Q} = \cup_{\pi\in\Pi} \left\{\boldsymbol{q} \in [0,1]^K \mid \boldsymbol{q} \leq \overline{\boldsymbol{q}}^\pi\right\}.$$

Definition 4 (URLLC SLA)**.** *The URLLC SLA for K users is a K-dimensional vector of probabilities $\boldsymbol{q}^* \in [0,1]^K$.*

- *If $\boldsymbol{q}^* \in \mathcal{Q}$ then we say that the URLLC SLA is feasible.*
- *If for a policy $\pi \in \Pi$ we have $\overline{\boldsymbol{q}}^\pi \geq \boldsymbol{q}^*$ element-wise, then we say that policy π achieves the URLLC SLA.*

A policy is called "optimal" if it achieves the URLLC SLA whenever the latter is feasible.

D. USS problem

The goal of this paper is to solve the *USS*, i.e., to find the optimal scheduling policy that achieves the URLLC SLA whenever that is feasible.

The following sections implement the following plan:

1) We establish a Markovian property which allows the decomposition of the problem into a dynamic weight adaptation over frames, and an intra-frame finite horizon MDP *without loss of optimality*. Then we provide a dynamic programming-based solution of the MDP, which combined with the weight adaptation provides an optimal policy for *USS*.
2) Due to the curse of dimensionality, the optimal policy is costly to compute in large instances of the problem. Indeed, we will prove that the per-frame problem is $\mathcal{NP}-$hard. Motivated by the problem complexity, we propose low-complexity suboptimal policies, and we prove that they become optimal in an asymptotic regime.

III. *USS* as a Markov Decision Process

Recall that $\boldsymbol{x}^\pi(t)$ expresses the remaining bits at slot t. Due to the inherent randomness of the channel, the system state $\boldsymbol{x}^\pi(t)$ evolves within the $m-$th frame as a controlled Markov Chain, described next. At each frame, the state is initialized as $\boldsymbol{x}(mT) = L\mathbf{1}$, and hence all users have L bits to transmit. Denote with $\boldsymbol{u}^\pi(t)$ the decision of a policy π in slot t: for example if $u^\pi_{k,i}(t) = 1$, then the policy decided in slot t is to activate user k by transmitting b^i bits. The transitions of the Markov Chain $\boldsymbol{x}^\pi(t)$ for $t \in \{(m-1)T, (m-1)T + 1, ..., mT-2\}$ are as follows:

$$x^\pi_k(t+1) = \begin{cases} \left(x^\pi_k(t) - b^i\right)^+, & \text{w.p. } u^\pi_{k,i}(t)(1-\hat{p}_e) \\ x^\pi_k(t), & \text{w.p. } \sum_{k,i} u^\pi_{k,i}(t)\hat{p}_e \end{cases}$$

where $\hat{p}_e = p_e(\gamma_k(m), b^i)$. Recall that $\boldsymbol{s}^\pi[m]$ takes value 1 for users with success frame m and 0 otherwise, as given by (2). The SLA satisfaction problem amounts therefore to finding a control policy π such that

$$\liminf_{M\to\infty} \frac{1}{M} \sum_{m=0}^{M-1} \mathbb{E}\left\{s^\pi_k[m]\right\} \geq q^*_k.$$

The above belongs to the general class of constrained Markov Decision Processes (MDPs), which are usually very complicated problems to solve.

A. Markovian Characterization of $\mathcal{Q}$

The MDP optimization space is the set Π that contains policies that act on all the past information of the system. It is, however, possible to show that Markovian policies, acting only on the current system state, suffice to achieve all reliable latencies.

Definition 5. *A policy π is "Markovian" if the following holds*

$$\boldsymbol{u}^\pi(H(mT + \hat{t}), mT + \hat{t}) = \boldsymbol{u}^\pi(\boldsymbol{x}(\hat{t}), \boldsymbol{h}[m], \hat{t}),$$

for all (i) $\hat{t} \in \{0,1,.,.,,T-1\}$, (ii) $m \geq 0$, and (iii) any $H(mT+\hat{t})$. We denote with Π^{MP} the set of Markovian policies.

Then let $\phi_{\boldsymbol{h}}$ denote the probability distribution over channel states, and $\rho_{\pi'}(\boldsymbol{h})$ denote the probability of using (Markovian) policy π' at a frame where the channel state is $\boldsymbol{h}$. We can show the following result.

Theorem 1 (Markovian characterization of reliable latencies)**.** *A reliable latency demand $\boldsymbol{q}^*$ is feasible if and only if for each channel state $\boldsymbol{h}$ there exists a set of Markovian policies $\Pi(\boldsymbol{h}) \subset \Pi^{MP}$ with cardinality at most $K+1$, and a probability distribution $\rho_{\pi'}(\boldsymbol{h})$ over $\Pi(\boldsymbol{h})$ such that*

$$\boldsymbol{q}^* = \sum_{\boldsymbol{h}} \phi_{\boldsymbol{h}} \sum_{\pi' \in \Pi(\boldsymbol{h})} \rho_{\pi'}(\boldsymbol{h}) \overline{\boldsymbol{q}}^{\pi'}(\boldsymbol{h}). \tag{3}$$

To provide some intuition into the above, note that since the channel is fixed within frames, and ergodic across them, we can consider each channel state separately and then average over the channel states. For a given channel state, we can study the feasibility of the probability $\Pr\{s_k^{\pi}=1\} > \overline{q}_k^*$. Also, by Caratheodory's theorem, we can describe every point in the region $\mathcal{Q}$ by as a linear combination of at most $K+1$ points. Hence, the idea is to use Markovian policies that achieve these $K+1$ extreme points and then satisfy the SLA by time sharing between them. This result is along the lines of conventional scheduling problems [17], [18], but extended to the case where we have a function that specifies what action should be taken according to the queue state and closeness to the deadline instead of a single action.

Theorem 1 may characterize $\mathcal{Q}$ not only in an implicit manner, but can also give us an important tool: It implies that in order to achieve any given feasible SLA, it suffices to restrict to policies that observe the channel state at the beginning of each frame and select at random a Markovian policy to follow for the current frame. In the next section we capitalize on this insight to decompose *USS* to simpler problems without loss of optimality. The idea is that we use weights to keep track of how much the SLA constraint has been satisfied and we find the policy that maximizes the weighted sum deliver rate periodically per frame. This procedure, as the system evolves, essentially finds the correct time sharing over Markovian policies that achieve the appropriate extreme points. We formalize these next.

B. Periodic Decomposition

We define the process $w_k[m]$, which is updated at the end of each frame as

$$w_k[m+1] = \left[w_k[m] - s_k^{\pi}[m] + A_k[m]\right]^+, \tag{4}$$

where $A_k[m]$ is a Bernoulli random variable with parameter q_k^*, i.e., the requested reliability of user k. Note that $w_k[m]$ evolves over frames as a queue with arrival rate q_k^* and service rate $\overline{q}_k^{\pi}$, and therefore its stability is equivalent to satisfying $q_k^* < \overline{q}_k^{\pi}$. Indeed, in our scheme $w_k[m]$ acts as a counter that tracks how well the user-k SLA constraint is satisfied in the past frames. A large value indicates that the user is lagging behind in satisfying its target empirical probability, and therefore our scheme will prioritize this user within the next frames. Specifically, within frame m we use $\boldsymbol{w}[m]$ as weights to solve the following intra-frame control problem:

$$\max_{\pi} \mathbb{E}\left\{\sum_{k=1}^{K} w_k[m] s_k^{\pi}[m]\right\}. \tag{5}$$

Theorem 2 (Optimal Decomposition)**.** *The policy that, at each frame, solves the finite horizon MDP* (5) *and adjusts the weights as per* (4) *satisfies any feasible SLA requirement $\boldsymbol{q}^* \in \mathcal{Q}$.*

Proof: The proof is based on the behavior of the process $\boldsymbol{w}[m]$, specifically on the fact that under the above policy (let us denote it by π^*) this process is mean rate stable. Let us take any $\boldsymbol{q}^* \in \mathcal{Q}$ and define $\overline{\pi}$ the policy that achieves this reliable latency demand by randomizing over Markovian policies, as per Theorem 1. In addition, consider the quadratic Lyapunov function $L(\boldsymbol{x}) = \sum_i x_i^2$. We have

$$\mathbb{E}\left\{L(\boldsymbol{w}^{\pi^*}[m+1]) - L(\boldsymbol{w}^{\pi^*}[m])|\boldsymbol{h}[m], \boldsymbol{w}^{\pi^*}[m] = \boldsymbol{w}\right\}$$
$$\leq C - 2\left(\mathbb{E}\left\{\sum_{k=1}^{K} w_k s_k^{\pi^*}[m]|\boldsymbol{h}\right\} - \mathbb{E}\left\{\sum_{k=1}^{K} w_k A_k[m]\right\}\right).$$

In the above, $C = K + \left(\sum_k q_k^*\right)^2$ is a constant. Observe that policy π^* maximizes the first expectation for each channel state. Taking expectations over the channel states and comparing with policy $\overline{\pi}$ thus gives

$$\mathbb{E}\left\{L(\boldsymbol{w}^{\pi^*}[m+1]) - L(\boldsymbol{w}^{\pi^*}[m])|\boldsymbol{w}^{\pi^*}[m] = \boldsymbol{w}\right\}$$
$$\leq C - 2\left(\sum_{k=1}^{K} w_k \mathbb{E}\left\{s_k^{\overline{\pi}}[m]\right\} - \sum_{k=1}^{K} w_k q_k\right) = C.$$

Taking expectations over $\boldsymbol{w}^{\pi^*}[m]$ we have

$$\mathbb{E}\left\{L(w^{\pi^*}[m])\right\} \leq mC + \mathbb{E}\{L(\boldsymbol{w}^{\pi^*}[0])\},$$

and since $w_k^2[m] \leq \sqrt{\sum_k w_k^2[m]}$ it follows that

$$\frac{\mathbb{E}\left\{w_k^{\pi^*}[M]\right\}}{M} \leq \frac{C}{\sqrt{M}} + \frac{\mathbb{E}\{L(\boldsymbol{w}^{\pi^*}[0])\}}{M}. \tag{6}$$

Finally, note that

$$w_k^{\pi^*}[M] \geq \sum_{m=0}^{M-1} A_k[m] - \sum_{m=0}^{M-1} s_k^{\pi^*}[m],$$

which, combined with (6) gives

$$\frac{\sum_{m=0}^{M-1} \mathbb{E}\left\{s_k^{\pi^*}[m]\right\}}{M} \geq \frac{\sum_{m=0}^{M-1} \mathbb{E}\left\{A_k[m]\right\}}{M}$$
$$- \frac{C}{\sqrt{M}} - \frac{\mathbb{E}\{L(\boldsymbol{w}^{\pi^*}[0])\}}{M}.$$

As M grows, the last term can be tuned to tend to zero (e.g. choose the counter to be zero at the beginning of system operation). Hence taking limits to infinity we obtain:

$$\liminf_{M\to\infty} \frac{1}{M} \sum_{m=0}^{M-1} \mathbb{E}\left\{ s_k^{\pi^*}[m] \right\} \geq q_k^*,$$

proving the theorem. ■

C. Optimal URLLC Algorithm

In this subsection, we focus on one frame and provide a dynamic programming algorithm that solves (5) optimally. Combining this algorithm with the weight update mechanism (4) we have an optimal policy for *USS*.

Initially, we drop the frame indices and take $t \in \{1, ..., T\}$. We define $F(\boldsymbol{x}, t)$ as the value function of the problem, i.e. the optimal value of (5) if we start at slot t with $\boldsymbol{x}(t) = \boldsymbol{x}$. We also denote with $\mathcal{U}(\boldsymbol{x})$ the set of feasible decisions when the remaining bits of user k are x_k, we have

$$\mathcal{U}(\boldsymbol{x}) = \{(k, b) : b \in \{0\} \cup \mathcal{B}, b = 0 \text{ if } x_k = 0\},$$

which essentially says that we exclude transmissions to users whose packet has already been delivered.[3] Then although in problem (5) the reward is collected at the end, using the constrained decision set $\mathcal{U}(\boldsymbol{x})$ we may collect reward w_k at the slot when the whole packet for user k is successfully transmitted. Then, the value function of the problem should satisfy the following recursive relations [19]:

$$F(\boldsymbol{x}, T) = \max_{(k,b)\in\mathcal{U}(\boldsymbol{x})} \left[w_k(1 - p_e(\gamma_k, b))\mathbb{1}_{\{x_k(t) - b \leq 0\}} \right] \quad (7)$$

$$F(\boldsymbol{x}, t) = \max_{(k,b)\in\mathcal{U}(\boldsymbol{x})} \Big[F(\boldsymbol{x}, t+1)p_e(\gamma_k, b) \quad (8)$$

$$+ (w_k \mathbb{1}_{\{x_k(t) - b \leq 0\}} + F(\boldsymbol{x} - \boldsymbol{e}_k, t+1))(1 - p_e(\gamma_k, b)) \Big].$$

The optimal algorithm works as follows. At the beginning of each frame, we read the current weights and channel qualities and then determine the function $F(\boldsymbol{x}, t)$ by solving the recursion (7)-(8), e.g. using *policy* or *value iteration* methods [19]. The optimal decisions within the frame are then given by the minimizer of the right hand side of (8), with ties broken arbitrarily; formally $u_{k,i}(\mathbf{x}, \mathbf{h}, t) = 1$ if the pair (k, b^i) is the minimizer in (8) and zero otherwise.

IV. Complexity of the Intra-frame Problem

The dynamic programming-based algorithm explained in the previous section solves optimally the finite horizon MDP (5). The algorithm can be used for small instances (e.g. few users, few possible modulation and coding schemes and short frame duration); however, as the instance size grows, it becomes impractical due to the so-called *curse of dimensionality*, i.e., the number of possible system states $\boldsymbol{x}$ grows exponentially large with the problem size (i.e. K, T and $|\mathcal{B}|$).

A natural question then arises: are there low-complexity optimal policies? Prior works on timely throughput [11], [10], [13] solve problems similar to *USS* using low-complexity *strict priority policies*. These works can be seen as special cases of our problem where: (i) only one successful transmission is needed for the whole packet to go through (i.e. the only transmission rate available is L/n) or (ii) the blocklength n is high enough so transmissions at any rate below the Shannon capacity of the each user has negligible transmission error. Therefore, since the *USS* problem is more general, it remains unclear whether it is a complex problem to solve or not.

First we provide an illustrative example of the suboptimality of strict priority policies in our problem.

Example 1. *Consider 3-slot frames, two users and a single transmission rate $\mathcal{B} = \{L/2\}$, such that each packet requires two successful transmissions. Let the success probabilities be $(p_1, p_2) = (1, 1/2)$ and the rewards $(w_1, w_2) = (1, 3)$. In this specific example it is better to first "gamble" with the high reward user and then if the first transmission fails switch to the user with the reliable channel. Here, as the deadline approaches the policy becomes more conservative.*

It follows that generalizing [11], [13], to include multiple successful transmissions renders strict priority policies suboptimal. Therefore we cannot use [11], [13] to conclude the complexity of our problem.

The main result of this Section is the complexity of the intra-frame problem (5):

Proposition 3. *Problem* (5) *is $\mathcal{NP}$-hard.*

Proof: We will prove $\mathcal{NP}$-hardness by reducing the intra-frame scheduling problem to a knapsack problem. Indeed, a special case of the intra-frame problem is the one where the blocklength for each transmission is asymptotically large, i.e. $n \to \infty$. In that case, the probabilities of error take the following form

$$p_e(\gamma, b) = \begin{cases} 1, b > n\log_2(1+\gamma) \\ 0, b \leq n\log_2(1+\gamma) \end{cases}$$

and the best rate for user k takes the form $b_k^* = \max\{b \in \mathcal{B} | b \leq n\log_2(1+\gamma_k)\}$. Since there is no probability of error, user k then needs $s_k = \lceil L/b^*k \rceil$ transmissions (i.e. needs to use s_k slots) to send its packet.

Since SNRs can be arbitrary, this is a general instance of a knapsack problem, where the knapsack has capacity T and the packet for user k is considered item k with reward w_k and size s_k. ■

With the above result we showed that our systematic way to solve the *USS* through a problem decomposition turned out to lead to an $\mathcal{NP}$–hard problem. The hardness result of Proposition 3 relates to the decomposition approach, and does not characterize the complexity of the *USS* problem, hence to this point, it remains unclear whether the *USS* problem is indeed a complex problem.

It is then important to note that we believe that the *USS* is *much harder* than the deterministic knapsack problem, which was used to prove hardness of the intra-frame problem of

[3] We use this convention throughout the paper.

our decomposition. First, it might not even be in $\mathcal{NP}$: it is not straightforward to test a given optimal solution without checking the policy over all possible system evolutions, which grow exponentially. In addition, our setting is conceptually a more general version of the stochastic knapsack problem, where the size of an item is random and revealed only after the item is put in the knapsack [20]. The best known approximation ratio for this problem is $1/2 - \epsilon$, however the corresponding algorithms are exponential in (polynomials of) $1/\epsilon$ [21], [22]. Whether better approximations exist or whether fully polynomial time approximation algorithms exist for approximation ratio better than $1/4$ [20] remain open questions.

On the other hand, when the deadline T grows we can exploit the fact that the effect of the random variables describing the success probabilities will tend to concentrate around their mean values in order to obtain efficient approximation algorithms for such asymptotic regimes. Indeed, based on this approach, we show in the next Section that the *USS* has a Fully Polynomial Time Approximation Scheme (FPTAS) at the asymptotic regime where the deadline T scales with the number of bits L per packet.

V. Asymptotically Optimal Policies

We begin by noting that our intra-frame scheduling problem bears a striking similarity into the *knapsack* problem: we must "pack" items of different weights (the number of transmissions to a user) in a knapsack of T slots, such that the total value (the sum of w_k of successful users) is maximized. However, there are certain complications: (i) the value of an item is obtained only if *all* reliable bits are successfully transmitted, hence the value of each item is random, (ii) the weight of each item increases with transmission failures and hence it is also random. As mentioned in the previous Section, such stochastic knapsack problems have been studied in the literature, but without success in obtaining efficient, close to optimal approximation algorithms. Therefore, our strategy in this Section is as follows. We propose a class of *knapsack-inspired* policies, which are essentially simple policies that solve a random knapsack problem where the weights of each item are concentrated to their mean values, and then we prove that these policies attain asymptotic near-optimality in our problem for the regime where T and the number of reliable bits are both large. First, we formalize the notion of a near-optimal policy:

Definition 6 ($\epsilon-$optimal policies)**.** *A policy π will be called* ϵ-optimal *for the USS problem if it achieves any SLA demand vector $\boldsymbol{q} \in (1-\epsilon)\mathcal{Q}$.*

We then note that the most efficient way of transmitting information to user k is to select the number of transmitting bits b that maximizes the average rate of successful bits/slot, which is given by b times the probability of success. Denoting this choice with b_k^* for user k,[4] we have

$$b_k^* = \underset{b\in\mathcal{B}}{\operatorname{argmax}}\,[b(1-p_e(b,\gamma_k))]\,,$$

Then we define the following class of policies:

Definition 7 (Knapsack-inspired class of policies)**.** *A policy π is said to be knapsack-inspired if*

- *Transmission to user k is attempted at rate b_k^*.*
- *User selection is given by the solution of a (deterministic) knapsack problem where user k brings reward w_k, has size L/b_k^* and the knapsack has total capacity of T.*

The complexity of the knapsack-inspired policies is dominated by the user selection phase, for which there exists a Fully Polynomial Time Approximation Scheme within $1-\epsilon$ of optimality, for any $\epsilon > 0$, see for example [23, Chapter 8].

The main result of this Section is that any knapsack-inspired policy has close to optimal performance in a scaling system where the deadline T and the reliable bits L grow large.

Theorem 4 (Asymptotic optimality of the knapsack-inspired class)**.** *Consider an instance of our problem where each frame lasts for νT slots, and each user has to transmit νL reliable bits. Then any knapsack-inspired policy is asymptotically $\epsilon-$optimal as $\nu \to \infty$.*

Before showing the proof of the theorem, we present a lemma regarding $\epsilon-$optimal scheduling.

Lemma 5. *Let π^ϵ be a policy such that*

$$\mathbb{E}\left\{\sum_{k=1}^{K} w_k[m]s_k[m]\middle|\pi^\epsilon, \boldsymbol{h}[m]\right\}$$
$$\geq (1-\epsilon)\max_{\pi}\mathbb{E}\left\{\sum_{k=1}^{K} w_k[m]s_k[m]\middle|\pi, \boldsymbol{h}[m]\right\}$$

for all possible values of the channel states $\boldsymbol{h}[m]$ and $\boldsymbol{w}[m] \in \mathbb{R}_+^{\mathbb{K}}$. Then this policy is $\epsilon-$optimal.

The proof is similar to that of the results on approximate scheduling in stochastic optimization problems not involving deadline constraints, cf. [18, Chapter 6.2]. We now turn to the proof of the Theorem.

Proof of Theorem 4: Recall that $\boldsymbol{x}(i)$ denotes the vector of remaining reliable bits at slot $i \in \{0,1,\ldots,T\}$. Under any policy, the scaled system is described by the vector:

$$\boldsymbol{x}^\nu(t) = \frac{1}{\nu}\boldsymbol{x}(\lfloor \nu t\rfloor), t \in [0,T], \tag{9}$$

such that for any scaling parameter $\nu > 0$ we have $x^\nu(0) = L$. The evolution of the scaled system can be written concisely as

$$\boldsymbol{x}^\nu(t) = L\mathbf{1} - \frac{1}{\nu}\sum_{\tau=1}^{\lfloor \nu t\rfloor}\boldsymbol{R}(\tau-1)\boldsymbol{A}(\boldsymbol{x}^\nu(\tau-1))\boldsymbol{v}(\tau-1), \tag{10}$$

[4] Since we focus on the intra-frame problem, hereinafter we drop the frame index from the notation.

where [5]

(i) $\boldsymbol{v}(t)$ is here a $K|\mathcal{B}|$–dimensional vector with zeros, except one element which has value 1. The vector indicates the choice of the policy in the slot where time instance t belongs, regarding which user to serve and how many bits to transmit.

(ii) $\boldsymbol{R}(t)$ is a $K \times K|\mathcal{B}|$ matrix with elements given by

$$\boldsymbol{R}_{k,i}(t) = \begin{cases} \text{Ber}\left(1 - p_e(b^{\text{rem}(i,k)}, \gamma_k)\right), \\ \qquad \text{if } (k-1)|\mathcal{B}| + 1 \leq i \leq k|\mathcal{B}| \\ 0, \quad \text{otherwise} \end{cases}$$

where $\text{rem}(i,k)$ is the remainder of the division i/k and $\text{Ber}(p)$ denotes a Bernoulli random variable with success probability p.

(iii) $\boldsymbol{A}(\boldsymbol{x}^{\nu}(\tau-1))$ is a $K|\mathcal{B}| \times K|\mathcal{B}|$ matrix with off-diagonal elements equal to zero and diagonal elements given as

$$A_{jj}(\boldsymbol{x}^{\nu}(t)) = \min\left[b^{i_j}, x_k^{\nu}(t)\right]$$

The role of this matrix is to restrict the number of information bits delivered by a transmission to the number of bits still in the queue (hence keeping the system state nonnegative)

We may rewrite eq. (10) as

$$\boldsymbol{x}^{\nu}(t) = L\boldsymbol{1} - \bar{\boldsymbol{R}}\frac{1}{\nu}\sum_{\tau=1}^{\lfloor \nu t \rfloor} \boldsymbol{A}(\boldsymbol{x}^{\nu}(\tau-1))\boldsymbol{v}(\tau-1) + \boldsymbol{\mu}^{\nu}(t), \quad (11)$$

where $\bar{\boldsymbol{R}} \triangleq \mathbb{E}\{\boldsymbol{R}(\tau)\}$ and

$$\boldsymbol{\mu}^{\nu}(t) = \frac{1}{\nu}\sum_{\tau=1}^{\lfloor \nu t \rfloor}\left(\bar{\boldsymbol{R}} - \boldsymbol{R}(\tau-1)\right)\boldsymbol{A}(\boldsymbol{x}^{\nu}(\tau-1))\boldsymbol{v}(\tau-1). \quad (12)$$

We have that $||\boldsymbol{\mu}^{\nu}(t) - \boldsymbol{\mu}^{\nu}(t-1)|| \leq K\max_{i\in\{1,..,|\mathcal{B}|\}}[b^i]$, almost surely $\forall \nu > 0$. In addition, since $\bar{\boldsymbol{R}}$ and $\boldsymbol{R}(\tau)$ are i.i.d and independent of $\boldsymbol{v}(\tau)$, the process $\boldsymbol{\mu}^{\nu}(t)$ is a Martingale for any ν. From [24, Th. 10.2.4] we have for any T

$$\lim_{\nu\to\infty}\sup_{0\leq t\leq T}||\boldsymbol{\mu}^{\nu}(t)|| = 0, \quad \text{a.s.}.$$

This implies that the scaled controlled stochastic process $\boldsymbol{x}^{\nu}(t)$ converges to $\bar{\boldsymbol{x}}(t)$ almost surely and uniformly on compact sets (see [24, Propositions 10.3.2, 10.3.3]), where $\bar{\boldsymbol{x}}(t)$ is defined as:

$$\bar{\boldsymbol{x}}(t) = L - \bar{\boldsymbol{R}}\int_0^t \boldsymbol{A}(\bar{\boldsymbol{x}}(s))\bar{\boldsymbol{v}}(s)ds, \quad t\in[0,T]. \quad (13)$$

Considering hereinafter the deterministic fluid system (13), our problem becomes to maximize

$$J(\bar{\boldsymbol{x}}(T)) = \sum_{k=1}^{K} w_k \mathbb{1}_{\{\bar{x}_k(T)=0\}}. \quad (14)$$

As $\nu \to \infty$, a policy that solves the above problem also solves our original stochastic problem. However, problem (14) is now simplified; since everything is deterministic, the evolution depends only on the control and not the queue state. We can thus obtain the optimal solution by planning the transmissions beforehand. Let us denote $\tau_{i,k}$ the amount of time where user k was scheduled with rate b^i. Then, (14) is equivalent to the following optimization problem:

$$\begin{aligned} \max \quad & \sum_{k=1}^{K} w_k \mathbb{1}_{\left\{\sum_{i=1}^{|\mathcal{B}|}\tau_{i,k}b^i(1-p_e(\gamma_k,b^i))\geq L\right\}} \\ \text{s.t.} \quad & \sum_{i,k}\tau_{i,k} = T. \end{aligned}$$

Notice that the most efficient allocation for user k is the one which assigns time only to the modulation scheme b_k^* that maximizes $b(1-p_e(\gamma_k,b))$. The problem then is equivalent to a knapsack problem where the knapsack capacity is T, the items are the user reliable bits, and item k (user k) yields reward w_k and takes space of $s_k = \lceil L/b_k^* \rceil$. Then our problem can be solved within accuracy ϵ by the poly-time algorithm mentioned in [23, Chapter 8]. Coupled with Lemma 5, this implies that we can achieve $(1-\epsilon)$ fraction of the Feasible Region of Reliable Latencies in the asymptotic regime. ■

In practice, as the deadline is approaching, transmissions of b_k^* bits may not be enough to transmit a packet by the deadline, even if they are all successful. The policies we are using modify the knapsack-inspired policies as follows: at slot t, the rate $min\,[b_k^*, \min_{b\in\mathcal{B}}\{b : (T-t)b \geq x_k(t)\}]$ is chosen for transmission. These modified knapsack-inspired policies are without loss of optimality in the asymptotic regime, and perform better in the non asymptotic cases.

VI. Numerical Results

In this section we report numerical results that illustrate the performance of the policy proposed in the previous section. We simulate a Rayleigh fading wireless system where the average SNR is equal to 4 dB and each codeword spans 168 symbols. Each user has to transmit 32 bytes of information in each frame and the number of information bits per codeword is constrained to the set $\{64, 96, 128, 160, 192\}$.

Figure 2 shows the reliable latency region achieved by different policies for two users. There are 4 slots per frame and 10^5 frames have been simulated for each point. The proposed policy can be seen to perform very close to the optimal one despite its much lower computational complexity.

Figure 3 plots the average reliability as a function of the number of users served. Each frame consists of 14 slots and 10^4 frames have been simulated per point. We can see to which extent the proposed policy outperforms round robin, and how much this gap increases with the number of users.

VII. Conclusions

We have studied the URLLC SLA satisfaction problem and we design a scheduling policy to activate users and uncertain short-packet transmissions with the goal to establish reliable latency performance A dynamic programming optimal policy is described, which can be used to solve small instances. The proposed policy leads to periodically solving an $\mathcal{NP}$–hard problem, however whether the original problem of SLA

[5] In the following, the dimension $K|\mathcal{B}|$ is interpreted as follows: The first $|\mathcal{B}|$ elements/columns/rows correspond to the rate selections schemes for user 1, the next for user 2 etc.

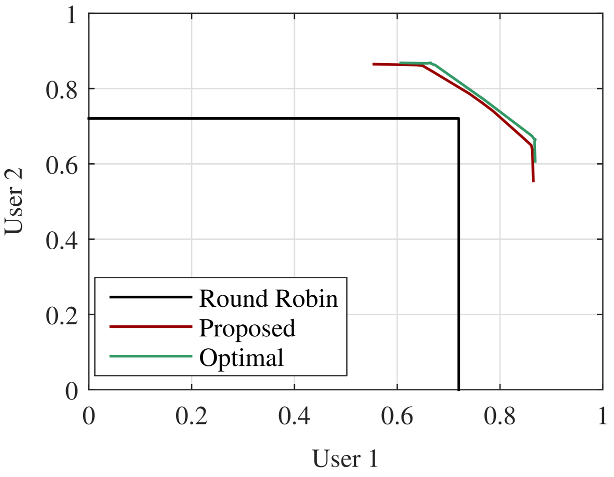

Fig. 2. Reliable latency region.

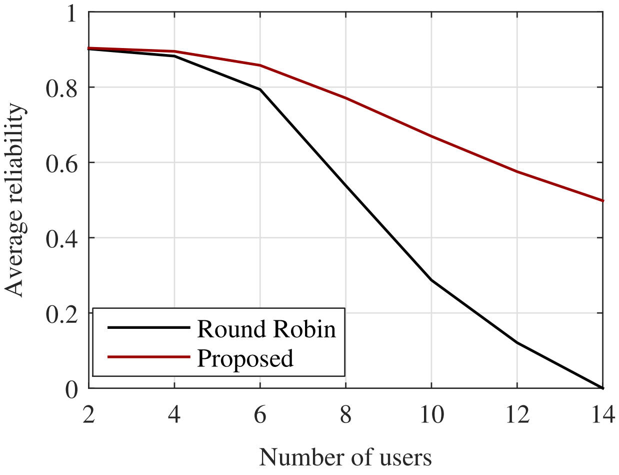

Fig. 3. Reliability as a function of the number of users.

satisfaction is $\mathcal{NP}$–hard remains open. Nevertheless, For specific large problem instances, we show that low complexity knapsack-inspired heuristics become asymptotically near-optimal in a fluid regime. Our work sheds light to the *USS* problem, and hence it is an important first step towards designing more sophisticated URLLC schemes with multiple different SLAs. In such settings, the region of reliable latencies $\mathcal{Q}$ varies over time following the changes in the environment, and the system can possibly negotiate the SLA level with each user.

References

[1] "GSA white paper: 5G network slicing for vertical industries," Sep. 2017. [Online]. Available: http://www.huawei.com/minisite/5g/img/5g-network-slicing-for-vertical-industries-en.pdf

[2] "3GPP TR 38.913: Study on scenarios and requirements for next generation access technologies," Mar. 2017.

[3] A. L. Stolyar, "Large deviations of queues sharing a randomly time-varying server," *Queueing Systems*, vol. 59, no. 1, Jun 2008.

[4] S. Bodas, S. Shakkottai, L. Ying, and R. Srikant, "Low-complexity scheduling algorithms for multichannel downlink wireless networks," *IEEE/ACM Trans. Netw.*, vol. 20, no. 5, pp. 1608–1621, 2012.

[5] 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, Jul. 2003.

[6] S. Vassilaras, "A cross-layer optimized adaptive modulation and coding scheme for transmission of streaming media over wireless links," *Wireless Networks*, vol. 16, no. 4, pp. 903–914, May 2010.

[7] Y. Jiang, "A basic stochastic network calculus," *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 4, pp. 123–134, Aug. 2006.

[8] S. Shakkottai and R. Srikant, "Scheduling real-time traffic with deadlines over a wireless channel," *Wirel. Netw.*, vol. 8, no. 1, pp. 13–26, 2002.

[9] A. Dua and N. Bambos, "Downlink wireless packet scheduling with deadlines," *IEEE Transactions on Mobile Computing*, vol. 6, no. 12, pp. 1410–1425, Dec 2007.

[10] I. H. Hou, V. Borkar, and P. R. Kumar, "A Theory of QoS for Wireless," in *IEEE INFOCOM*, Apr. 2009.

[11] I. H. 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, Feb. 2012.

[12] I.-H. Hou, "Scheduling heterogeneous real-time traffic over fading wireless channels," *IEEE/ACM Trans. Netw.*, vol. 22, no. 5, pp. 1631–1644, Oct. 2014.

[13] K. S. Kim, C. P. Li, and E. Modiano, "Scheduling multicast traffic with deadlines in wireless networks," in *IEEE INFOCOM*, 2014.

[14] A. Anand, G. de Veciana, and S. Shakkottai, "Joint Scheduling of URLLC and eMBB Traffic in 5G Wireless Networks," in *IEEE INFOCOM*, 2018.

[15] G. Durisi, T. Koch, and P. Popovski, "Toward massive, ultrareliable, and low-latency wireless communication with short packets," *Proc. IEEE*, vol. 104, no. 9, pp. 1711–1726, Sep. 2016.

[16] 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.

[17] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, R. Vijayakumar, and P. Whiting, "Scheduling in a queueing system with asynchronously varying service rates," *Probability in the Engineering and Informational Sciences*, vol. 18, no. 2, p. 191217, 2004.

[18] M. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan & Claypool, 2010.

[19] D. P. Bertsekas, *Dynamic Programming and Optimal Control Vol. I*. Athena scientific Belmont, MA, 1995.

[20] B. C. Dean, M. X. Goemans, and J. Vondrk, "Approximating the stochastic knapsack problem: The benefit of adaptivity," *Mathematics of Operations Research*, vol. 33, no. 4, pp. 945–964, 2008.

[21] A. Bhalgat, "A $(2+\epsilon)$-approximation algorithm for the stochastic knapsack problem," *ArXiv e-prints*, 2011.

[22] J. Li and W. Yuan, "Stochastic combinatorial optimization via poisson approximation," in *ACM/SIAM STOC*, 2013.

[23] V. Vazirani, *Approximation Algorithms*. Springer, 2001.

[24] S. Meyn, *Control Techniques for Complex Networks*, 1st ed. New York, NY, USA: Cambridge University Press, 2007.