Admission Control for URLLC Traffic with Computation Requirements in 5G and Beyond

Fidan Mehmeti, Valentin Thomas Haider, Wolfgang Kellerer
Chair of Communication Networks, Technical University of Munich, Germany
E-mail: {fidan.mehmeti, valentin.haider, wolfgang.kellerer}@tum.de

***Abstract*—One of the three types of services supported by 5G networks are Ultra-Reliable Low-Latency Communications, which are characterized by the stringent requirement to deliver packets within a very short time with a high reliability. Besides being successfully transmitted/received, these data need to be processed as well. To satisfy these strict requirements, one needs to determine both the required data rate and the processing rate, given the channel conditions and traffic intensity of the service. Moreover, with constraints on both the Radio Access Network and edge computing resources as well as with the competition between an ever-increasing number of users in cellular networks, a very important question which arises is that of *admission control*. This guarantees users will not suffer from deteriorating performance. In this paper, using analytical modeling, we derive admission control policies for both homogeneous and heterogeneous types of users, taking into account the delay incurred by the RAN part of the network and that caused by the finite computing capability at the edge. We validate theoretical outcomes and provide additional insights on a 5G dataset. Results show that the number of admitted users depends on the worst channel conditions, the deadline by which the data must be processed and the available resources. There is an almost linear increase in the number of admitted users with the decrease in latency.**

***Index Terms*—Admission control, 5G and beyond, URLLC.**

I. Introduction

Ultra-Reliable Low-Latency Communications (URLLC) are one of the services provided in 5G networks [1]. Services falling into this category require a very low latency (on the order of milliseconds) for delivering the vast majority of their data (almost 100%), and also support for high mobility [2].

Autonomous driving, remote surgery, remote monitoring and control [2] are some use cases that belong to services with URLLC traffic. Delivering and processing almost all URLLC packets (i.e., providing high reliability) within a very short time period is quite challenging. The difficulty becomes even more emphasized given the need for allocating two types of limited resources in the cell, and the constantly increasing number of users competing for them. These two types of resources are Radio Access Network (RAN) resources that enable the transmission/reception of information and computing resources for processing the received data. Moreover, the aforementioned services are not only sensitive to abiding by those very non-flexible requirements, but because of their nature, a failure to comply either with the low-latency or reliability requirement can give rise to a serious risk on human lives. Therefore, the paramount importance of enabling (almost) flawless operation of this type of traffic.

Enabling this impeccable functionality is particularly strenuous in cellular networks, where channel characteristics are highly variable with time because of users' mobility and processes inherent to this communication medium, like shadowing [3]. Adding the timely computation requirement complicates things even further. Therefore, to provide a given data rate and processing rate that will satisfy the delay requirements, the proper resource allocation schemes on two levels (RAN and edge cloud) must be designed. Moreover, as there are more and more users with URLLC type of traffic, the operator needs to allocate the resources to satisfy those requirements for as many of them as possible. On the other hand, there is an inter-play involved in allocating these resources, which provides an extra degree of freedom in allocating them. The operator can increase the amount of allocated RAN resources while reducing the computing resources, and vice versa, so that the delay (latency) requirement is still met.

Several interesting questions arise related to the admission of users with URLLC traffic. Firstly, given the traffic pattern of a user and its channel conditions, what is the optimal combination of RAN and processing resources so that both the latency and reliability requirements are met? Secondly, given their traffic requirements, how many URLLC users can receive satisfactory service in the cell?

To answer those questions, in this paper, we present an analytical approach which relies on realistic assumptions and captures reliably the inherent constraints of URLLC traffic. The outcomes of our model are admission policies for users with URLLC traffic, taking into account the computation requirements. We do this for both users with identical channel characteristics (homogeneous users) and heterogeneous users. The results we provide here can help cellular network operators in efficiently allocating resources in order to increase the number of admitted users. The main message of this paper is that the worst-case channel conditions jointly with users' traffic patterns and the acceptable latency are the decisive factors in determining the number of admitted users. Specifically, our main contributions are:

- We derive the maximum number of URLLC users that can be admitted, under the assumption of identical chan-

This work was supported by the Federal Ministry of Education and Research of Germany (BMBF) under the projects "6G-Life" and "6G-ANNA" with project identification numbers 16KISK002 and 16KISK107.

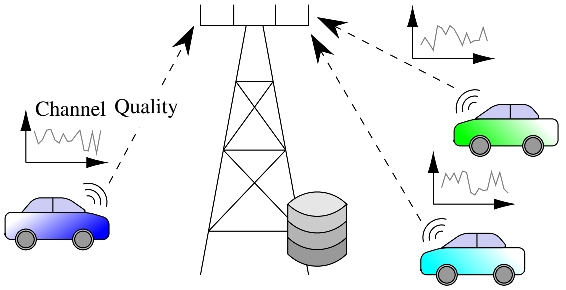

Fig. 1. Illustration of the system model.

nel and traffic distributions, and given their latency and reliability requirements, by obtaining first the Pareto frontier of the required resources.

- We also provide the admission policy for users with heterogeneous channels.
- Using extensive simulations run with input parameters from a publicly-available 5G trace, we validate our theoretical results and provide some other interesting insights.

The remainder of this paper is organized as follows. In Section II, we introduce the system model and the type of traffic of interest. Section III and Section IV present the analysis which determines the maximum number of homogeneous URLLC users that can be admitted. An admission policy for admitting heterogeneous users is presented in Section V. Some performance evaluation results are presented in Section VI. In Section VII, we discuss some related work. Finally, Section VIII concludes the paper.

II. Performance Modeling

A. System model

The possibility of network slicing in 5G [4], which introduced a paradigm shift in the operation of cellular networks, enables assigning *dedicated* resources to the same use case, e.g., users with URLLC traffic having the same latency and reliability requirements, and which are located in the area covered by the same Base Station (BS). In this paper, we assume that users with URLLC traffic in the cell belong to the same use case, and hence require the same level of service.

We consider users within the coverage area of a 5G macro BS, operating in Frequency Range 1, i.e., in the sub-6 GHz bands. The focus is on the uplink (see Fig. 1). After packets are received at the BS, there is also processing involved, which is performed on the edge cloud, where the latter is collocated with the BS. So, there are two types of resources needed to realize the communication successfully. The first are Physical Resource Blocks (PRBs) [5] for the users to send the information to the BS, which represent the unit of RAN resource allocation per slot. The second type are the computing resources on the edge cloud to process the information, which can be Virtual Machines for instance. The number of available PRBs is K, whereas the number of computing resources is M.

Given the nature of mobile communications, we assume that channel conditions vary from one slot to another for all users. To quantify the channel quality for a user in a slot, the metric known as Channel Quality Indicator (CQI) is used. There are 15 possible values of the CQI [6], with 1 denoting the worst channel conditions, whereas 15 representing excellent channel conditions. In general, users experience different channel conditions, i.e., different values of CQI, across different PRBs even within the same slot. Because of the user's mobility and time-varying nature of the channels, per-PRB CQI (which is a function of Signal-to-Interference-Plus-Noise-Ratio (SINR)) varies from one slot to another, whose value depending on the Modulation and Coding Scheme (MCS) used sets the per-PRB rate [6]. Therefore, the allocation of RAN resources has to be performed across two dimensions, *time* and *frequency*. For each user, we assume flat blocks in a slot, i.e., the per-PRB rate does not change during the slot, but it changes from one slot to another randomly.

For analytical tractability, we make a simplifying assumption. Namely, we assume that the BS splits the transmission power equally among all PRBs it transmits on, and that the channel characteristics for a user remain static across all PRBs (identical CQI over all PRBs for a given user), but change randomly (according to some distribution) from one slot to another, and are mutually independent among users. These assumptions reduce the RAN allocation to the number of allocated PRBs and not to which PRBs are assigned to a user.

From the previous assumptions, it follows that in every slot user's i per-PRB rate can be modeled as a discrete random variable, R_i, with values in $\{r_1, r_2, \ldots, r_{15}\}$, where $r_1 < r_2 < \ldots < r_{15}$, with a Probability Mass Function (PMF) $p_{R_i}(x)$.

The situation is less complicated with computing resources because any assigned unit performs the same across all users, i.e., it is insensitive to channel conditions. We denote the processing capacity of a single unit by q (expressed in Mbps).

B. URLLC traffic

The main feature of URLLC traffic is the requirement to have an extremely low delay and on top of that to be reliable, i.e., the vast majority of its packets to be transmitted (received) and processed within that maximum allowed latency.[1] To capture this, we use T_{max} to denote the maximum allowed latency (transmission and processing) of the packets. If T denotes the total delay, we describe the reliability by

$$\mathbb{P}\left(T \leq T_{max}\right) \geq 1 - \epsilon, \tag{1}$$

where ϵ has a very small value. It denotes the outage probability. E.g., if the requirement is for 99% reliability, $\epsilon = 0.01$.

Note that there is also the propagation delay contributing to latency. Nevertheless, there are two reasons we do not consider it here. The first is that we cannot affect it, and the second is that the propagation delays are much lower than transmission and computing delays. Therefore, in this work, we assume that the transmission and computation times comprise the latency.

Traffic generation: Data from every user to the BS are transmitted in packets in regular time intervals (periodically).

[1] In practice, this latency is on the order of *ms*, with the reliability requirement usually going above 99%.

TABLE I
NOTATION

$R_i(t)$	Per-PRB rate of user i in slot t
q	Per-unit processing capacity
$p_{R_i}(x)$	PMF of user's i per-PRB rate
K	Total number of PRBs
M	Total number of computing resources
ϵ	Outage probability
Δ	Size of transmitted data
T_{max}	Maximum allowed delay (latency)
τ	Inter-transmission period
$\lfloor x \rfloor$	Largest integer that is $\leq x$

These periods are assumed to be longer than the slot duration. We denote these periods by τ.

Amount of data: At the moment of generation, we assume that there are several packets that are being transmitted. The amount of data transmitted at once by a user is assumed to be constant [7], and is equal to Δ.

As for this traffic type packets are small [7], and not too many of them are transmitted simultaneously, we assume that the data generated at once is transmitted with the same rate.

If K_i PRBs and M_i processing units are assigned to user i, the total delay of the data transmitted at once by that user is

$$T_i = \frac{\Delta}{K_i R_i} + \frac{\Delta}{M_i q}, \tag{2}$$

where the first term denotes the transmission delay, while the second term denotes the processing delay.

Table I summarizes the notation used throughout this paper.

III. Admission Control for Homogeneous Users

We present an approach for homogeneous users, by deriving the maximum number of users that can be admitted. Then, we show why that approach leads to a contradictory result.

A. Equal-share of resources

As users are homogeneous, their per-PRB rates R_i in (2) undergo the same distribution, which we denote by R. In this case, the number of PRBs an admitted user receives is $\frac{K}{n}$, given that there are n users in total. Similarly, every user will receive $\frac{M}{n}$ computing resources. Let us see the maximum number of users that can be admitted in the cell this way. Substituting the aforementioned facts into (2) and (1), and rearranging we obtain

$$\mathbb{P}\left(\frac{1}{R} \leq \frac{KT_{max}}{n\Delta} - \frac{K}{Mq}\right) \geq 1 - \epsilon. \tag{3}$$

The left-hand side of (3) is the Cumulative Distribution Function (CDF) of the inverse of the per-PRB rate at point $\frac{KT_{max}}{n\Delta} - \frac{K}{Mq}$. So, we have

$$F_{\frac{1}{R}}\left(\frac{KT_{max}}{n\Delta} - \frac{K}{Mq}\right) \geq 1 - \epsilon. \tag{4}$$

As CDF is a monotonous increasing function, (4) yields

$$\frac{KT_{max}}{n\Delta} - \frac{K}{Mq} \geq F^{-1}_{\frac{1}{R}}(1 - \epsilon), \tag{5}$$

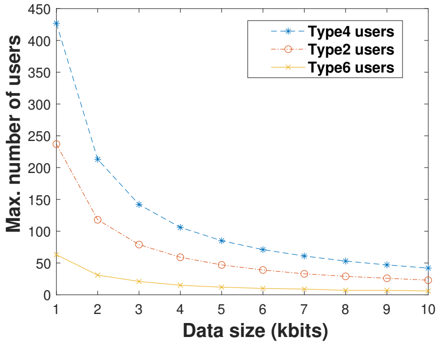

Fig. 2. The number of users with URLLC traffic with a maximum latency of $T_{max} = 5$ ms, which can be admitted in the cell. There are $K = 273$ PRBs that can be used, as well as $M = 500$ processing units on the edge cloud. We consider three types of users, characterized by different channel statistics (see Table II).

where $F^{-1}_{\frac{1}{R}}(1-\epsilon)$ is the inverse of the CDF at point $1 - \epsilon$. Using simple algebraic operations, we obtain

$$n \leq \frac{KT_{max}}{\Delta} \cdot \frac{1}{F^{-1}_{\frac{1}{R}}(1-\epsilon) + \frac{K}{Mq}}, \tag{6}$$

or equivalently,

$$n_{max} = \frac{KT_{max}}{\Delta} \cdot \frac{1}{F^{-1}_{\frac{1}{R}}(1-\epsilon) + \frac{K}{Mq}}. \tag{7}$$

We describe next why this result is not correct.

B. Contradictory result

Let us look at the number of users that can be admitted in the cell for a given system configuration. For a subcarrier spacing of 30 KHz, the number of available PRBs is $K = 273$ PRBs [6]. The number of processing units on the edge cloud is $M = 500$, with each processing unit of $q = 1$ Mbps. The maximum allowed latency is $T_{max} = 5$ ms. We consider different data sizes transmitted at once. Fig. 2 illustrates the maximum number of users that can be admitted for this type of URLLC traffic vs. the data size (Δ) for three types of channel characteristics of users. More details on this can be found in Section VI, and Table II more specifically.

What can be observed from Fig. 2 is the fact that when all the users of Type 4 transmit regularly (periodically) data of size $\Delta = 1$ kbits, a total of 430 users can be admitted in the cell. However, this is in contradiction to the number of available PRBs assumed in this scenario, which is (only) 273! As is well known, the granularity level in resource allocation in 5G is the PRB per slot. So, the maximum number of simultaneously transmitting users in this scenario would be 273, not 430 as (7) implies. The reason for this contradiction stems from the fact that with the above approach a user can receive a non-integer number of PRBs or processing units, i.e., $\frac{K}{n}$ and $\frac{M}{n}$, respectively. This would lead to a user receiving an amount of PRBs lower than 1, e.g., 0.8, especially with users with good channel conditions and a low amount of submitted data. This, for apparent reasons, is infeasible.

The correct way of writing the amount of PRBs a user receives, if there are in total n users, would be $\left\lfloor \frac{K}{n} \right\rfloor$, whereas the number of processing units in the edge cloud would be $\left\lfloor \frac{M}{n} \right\rfloor$, where $\lfloor x \rfloor$ denotes the value of x rounded down.

Combining (1) and (2) would then result in

$$\mathbb{P}\left(\frac{\Delta}{\left\lfloor \frac{K}{n} \right\rfloor R} + \frac{\Delta}{\left\lfloor \frac{M}{n} \right\rfloor q} \leq T_{max}\right) \geq 1 - \epsilon. \tag{8}$$

Note that $\left\lfloor \frac{K}{n} \right\rfloor \neq K \left\lfloor \frac{1}{n} \right\rfloor$. Hence, solving inequality (8) is not analytically tractable.

Given the previous reasoning, we need to follow a different approach in determining the maximum number of URLLC users that can be admitted in the cell, while taking into account two types of resources (RAN and processing units). This is shown in the next section.

IV. RAN RESOURCES VS. PROCESSING UNITS TRADEOFF

In this section, we determine the optimal tradeoff between the number of PRBs and processing units that need to be allocated to a user, such that the number of admitted homogeneous users is maximized. In order to derive the admission policies for a reliability of $1 - \epsilon$, we need first the results for the strictest reliability possible (100%). Therefore, we obtain the maximum number of admitted users for $\epsilon = 0$ first.

A. *Homogeneous users:* 100% *reliability*

In this scenario, user i will receive K_i PRBs and M_i computing resources. It holds that $\sum_i K_i \leq K$ and $\sum_i M_i \leq M$. As the reliability requirement is 100%, then we have

$$\frac{\Delta}{K_i R_i} + \frac{\Delta}{M_i q} \leq T_{max}. \tag{9}$$

Further, since (9) has to be always satisfied, we need to consider it for the worst-case scenario in terms of the channel conditions (when the user has the lowest per-PRB rate), because in that case the user needs the largest amount of resources to meet the latency requirement.

Let $\rho_i = \min\{r_j | p_{R_i}(r_j) > 0, j \in \{1, \ldots, 15\}\}$ denote the lowest possible per-PRB rate for user i; $\rho_i \in \{r_1, \ldots, r_{15}\}$. In order to admit as many users as possible, we can be flexible in terms of the latency and allow that for every packet it is strictly equal to the allowed maximum, i.e., $T = T_{max}$, which transforms (9) into

$$\frac{1}{K_i \rho_i} + \frac{1}{M_i q} = \frac{T_{max}}{\Delta}. \tag{10}$$

Next, from (10) we express the amount of needed computation resources as a function of the number of assigned PRBs:

$$M_i = \frac{1}{q} \cdot \frac{1}{\frac{T_{max}}{\Delta} - \frac{1}{K_i \rho_i}}. \tag{11}$$

Since $M_i > 0$, from (11), it must hold that $\frac{T_{max}}{\Delta} > \frac{1}{K_i \rho_i}$, resulting in

$$K_i > \frac{\Delta}{\rho_i T_{max}}. \tag{12}$$

Following a similar reasoning for M_i from (9), it should hold that $\frac{T_{max}}{\Delta} > \frac{1}{M_i q}$, leading to

$$M_i > \frac{\Delta}{q T_{max}}. \tag{13}$$

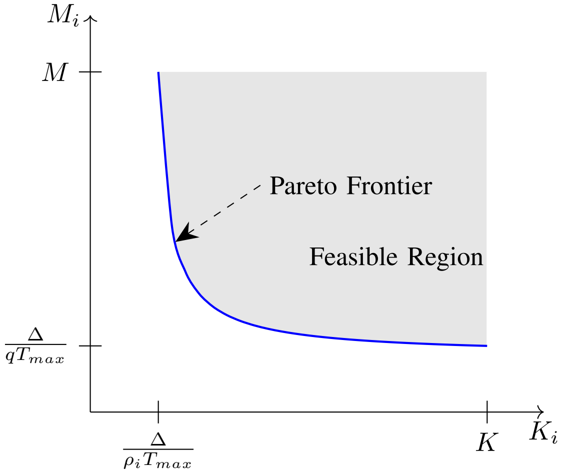

Fig. 3. The general shape of the Pareto frontier and the feasible region for the amount of needed RAN resources (PRBs) and processing units in the edge cloud so that latency is met with a reliability of 100%. Note the dependence of the Pareto frontier on the worst channel conditions (ρ_i).

So far, we know what are the infimums of the amount of RAN ($\frac{\Delta}{\rho_i T_{max}}$) and edge cloud ($\frac{\Delta}{q T_{max}}$) resources needed. Apparently, to satisfy the latency constraint, there are multiple combinations of (K_i, M_i). Providing more RAN resources (reducing the transmission delay) will compensate fewer computing resources allocated (higher processing delay). So, *what is the optimal combination of* (K_i, M_i) *that will enable admitting the highest number of URLLC users*? To answer the question, we must understand first the dependency of M_i on K_i. To that end, we look at the first derivative of $M_i(K_i)$ from (11). We have

$$M_i^{'}(K_i) = \frac{-1}{q \rho_i K_i^2} \cdot \frac{1}{\left(\frac{T_{max}}{\Delta} - \frac{1}{K_i \rho_i}\right)^2} < 0, \tag{14}$$

implying that M_i is a monotonous decreasing function in K_i, as expected. For the second derivative, after some calculus, we obtain

$$M_i^{''}(K_i) = \frac{2 T_{max}}{q \Delta \rho_i} \cdot \frac{1}{\left(\frac{T_{max} K_i}{\Delta} - \frac{1}{\rho_i}\right)^3} > 0, \tag{15}$$

which implies the fact that M_i is a convex function in K_i.

Taking into account (12), (13), (14), and (15), we can obtain the general shape of the dependency between M_i and K_i for a given ρ_i. The feasible region of the values for the ordered pair (K_i, M_i) is shown in Fig. 3. Having this in mind, we need to look for our solution along the curve shown in Fig. 3. This is the well-known *Pareto frontier* [8].

Note: In Fig. 3, we show the general "continuous" Pareto frontier to illustrate the dependency of M_i on K_i. In practice, it would be a discrete function where the values of M_i would have to be rounded up accordingly. But, the shape of the frontier would not change.

The convexity of $M_i(K_i)$ provides an interesting observation. Namely, reducing the value of K_i implies a higher incline in M_i, and vice versa (the *increasing return* property of convex functions, the opposite of the *diminishing return* property encountered in concave functions). The previous observation implies the need to choose as a solution of allocated resources the point (K_0, M_0) from the Pareto frontier such that both resource components K_0 and M_0 are sufficient to satisfy the traffic requirements of the highest possible number of users.

So, what is the optimal choice on the Pareto frontier? Let us denote by (K_0, M_0) any ordered set of points on the Pareto frontier. For any such point, the maximum number of users that can be admitted if considering only the amount of needed RAN resources K_0 would be $\lfloor \frac{K}{K_0} \rfloor$. In case the number of users is determined based solely on the number of processing resources, its maximum number would be $\lfloor \frac{M}{M_0} \rfloor$. Therefore, for a given point (K_0, M_0) on the Pareto frontier, if we consider both resources, the maximum number of users that can be admitted would be

$$n_{max}(K_0, M_0) = \left\lfloor \min\left(\frac{K}{K_0}, \frac{M}{M_0}\right) \right\rfloor. \tag{16}$$

When looking over the entire possible set of ordered pairs (K_0, M_0), we have the following result for the maximum number of admitted users in the cell:

Result 1. *Assume a BS with K PRBs and M edge computing resources (with a processing rate of q per resource). The maximum number of users with URLLC traffic, whose worst-case per-PRB rate is ρ_i and the amount of transmitted data at once Δ, which should never experience (i.e., the reliability is 100%) a latency higher than T_{max} that can be admitted by that BS is*

$$n_{max} = \max_{K_0, M_0} \left\{ \left\lfloor \min\left(\frac{K}{K_0}, \frac{M}{M_0}\right) \right\rfloor \right\}, \tag{17}$$

where the ordered set (K_0, M_0) satisfies the inequality

$$\frac{1}{K_0 \rho_i} + \frac{1}{M_0 q} \leq \frac{T_{max}}{\Delta}. \tag{18}$$

The interesting thing to observe from Result 1 is that with this approach for all the users with the same lowest possible CQI, the amount of resources needed is the same, i.e., this approach is valid not only for users with identical per-PRB rate distributions, *but for all users with the same lowest CQI*. Said differently, the approach is oblivious to the entire channel condition statistics. Apparently, the higher the ρ, the higher the number of admitted users.

B. Homogeneous users: General reliability

With a reliability lower than 100%, the constraint (1) that needs to be fulfilled for every user i reads as

$$\mathbb{P}\left(\frac{1}{K_i \rho_i} + \frac{1}{M_i q} \leq \frac{T_{max}}{\Delta}\right) \geq 1 - \epsilon, \quad \forall i. \tag{19}$$

Obviously, relaxing the reliability requirement should lead to an increased number of admitted users. However, determining that number is not feasible via a closed-form expression. Instead, we use a rather different approach.

We assume w.l.o.g. that all per-PRB rates are possible, i.e., $p_R(r_j) > 0, \forall j \in \{1, \ldots, 15\}$. Then, in a given setup, for the worst-possible per-PRB rate, i.e., ρ_i, we find the maximum number of admitted users when $\epsilon = 0$ (using (17)). We denote this as $n(\rho_i)$. This is a lower bound, as relaxing the reliability to $\epsilon > 0$ enables more users to be admitted in the cell.

Next, we increase the "minimum" possible per-PRB rate to the next higher rate and denote this by ρ_i^+, e.g., if $\rho_i = r_6$ then $\rho_i^+ = r_7$. For the latter value, using (17), we obtain the corresponding maximum number of users that can be admitted for $\epsilon = 0$. That is the new reference value, corresponding to K_0^+ and M_0^+. Then, as this new value was planned for better channel conditions, but with strict reliability (of 100%), we check whether this new number of users, which we denote as $n(\rho_i^+)$, can be admitted in the cell, such that their latency is satisfied for $1 - \epsilon$ of the time, for which it holds

$$\mathbb{P}\left(\frac{1}{K_0^+ \rho_i} + \frac{1}{M_0^+ q} \leq \frac{T_{max}}{\Delta}\right) \geq 1 - \epsilon. \tag{20}$$

If the previous condition is satisfied, then we increase ρ_i^+ to the next per-PRB rate, and find the new K_0^+ and M_0^+, as well as the new $n(\rho_i^+)$ using (17). Afterwards, we again check if the updated (20) is satisfied. This procedure continues until the corresponding (20) is not fulfilled for the first time.

Once (20) does not hold, we know that we cannot admit the checked number of users. Nevertheless, we know that we were able to admit the number of users corresponding to the previous ρ_i^+ for ϵ. Hence, an upper and a lower bound on the maximum number of users that can be admitted were determined. Therefore, the binary search algorithm [9] can be employed to find the largest possible number $n_{max}(\rho_i, \epsilon)$ between $n(\rho_i)$ and $n(\rho_i^+)$. For every iteration of the binary search, using (20), it is checked whether the "new" number of users can be admitted to the network. If yes, the upper interval is taken as the new range, while the lower interval is analogously taken over as the new range if the condition did not hold. This procedure is repeated until the largest number of users which satisfies (20) is found.

The procedure is summarized in Algorithm 1. The complexity of the algorithm is $O(\log_2 n)$.

V. Admission policy for heterogeneous users

When it comes to users with heterogeneous conditions, we provide a simple admission policy for a newly arriving user, for any ϵ. The first step is to check whether the newly arriving user and the current $n - 1$ users receiving service satisfy the inequality (heterogeneous users have different ρ_i)

$$\sum_{i=1}^{n} \frac{1}{\rho_i} \leq K \left(\frac{T_{max}}{\Delta} - \frac{1}{\lfloor \frac{M}{n} \rfloor q}\right). \tag{21}$$

If that is the case, then the *new user can be admitted*. Condition (21) is obtained by combining the latency requirement and $\sum_{i=1}^{n} K_i \leq K$, after some algebra. Due to space limitations, we do not show the rest of the procedure here. Condition (21) pertains to the case of $\epsilon = 0$. Essentially, if there are enough resources for the newly arriving user to be admitted for the most restrictive case (that of $\epsilon = 0$), the user can be admitted for any other lower reliability, i.e., higher ϵ. If user n has a high ρ_n, it would lead to a lower LHS of (21), and thus to a higher chance for the user to be admitted.

Algorithm 1 Admission control with general reliability for homogeneous users

```
Input: $\rho_i$, $K$, $M$, $q$, $T_{max}$, $\Delta$, $\epsilon$
Output: $n_{max}(\rho_i, \epsilon)$, $K_0 \in \mathbb{N}$, $M_0 \in \mathbb{N}$
function GenRelAdmission($\rho_i$, $K$, $M$, $q$, $T_{max}$, $\Delta$, $\epsilon$)
    Calculate $n(\rho_i) = \max_{K_0, M_0} \left\{ \left\lfloor \min\left( \frac{K}{K_0}, \frac{M}{M_0} \right) \right\rfloor \right\}$
    s.t. $\frac{1}{K_0 \rho_i} + \frac{1}{M_0 q} \le \frac{T_{max}}{\Delta}$.
    Note $K_0$ and $M_0$.
    Set $\rho_i^+$ to $r_k$ where $k = j+1$ if $\rho_i = r_j$.
    Calculate $n(\rho_i^+) = \max_{K_0^+, M_0^+} \left\{ \left\lfloor \min\left( \frac{K}{K_0^+}, \frac{M}{M_0^+} \right) \right\rfloor \right\}$
    s.t. $\frac{1}{K_0^+ \rho_i^+} + \frac{1}{M_0^+ q} \le \frac{T_{max}}{\Delta}$.
    Note $K_0^+$ and $M_0^+$.
    while $\mathbb{P}\left( \frac{1}{K_0^+ \rho_i} + \frac{1}{M_0^+ q} \le \frac{T_{max}}{\Delta} \right) \ge 1 - \epsilon$ & $\rho_i^+ < r_{15}$ do
        Set $n(\rho_i) = n(\rho_i^+)$.
        Set $K_0 = K_0^+$ and $M_0 = M_0^+$.
        Increase $\rho_i^+$ to the next higher $r_k$, i.e., set $k = k+1$.
        Calculate $n(\rho_i^+) = \max_{K_0^+, M_0^+} \left\{ \left\lfloor \min\left( \frac{K}{K_0^+}, \frac{M}{M_0^+} \right) \right\rfloor \right\}$
        s.t. $\frac{1}{K_0^+ \rho_i^+} + \frac{1}{M_0^+ q} \le \frac{T_{max}}{\Delta}$.
        Note $K_0^+$ and $M_0^+$.
    end while
    while $n(\rho_i^+) - n(\rho_i) > 1$ do
        Set $K_0^t = \left\lfloor \frac{K}{n(\rho_i) + \left\lfloor \frac{n(\rho_i^+) - n(\rho_i)}{2} \right\rfloor} \right\rfloor$.
        Set $M_0^t = \left\lfloor \frac{M}{n(\rho_i) + \left\lfloor \frac{n(\rho_i^+) - n(\rho_i)}{2} \right\rfloor} \right\rfloor$.
        if $\mathbb{P}\left( \frac{1}{K_0^t \rho_i} + \frac{1}{M_0^t q} \le \frac{T_{max}}{\Delta} \right) \ge 1 - \epsilon$ then
            Set $n(\rho_i) = n(\rho_i) + \left\lfloor \frac{n(\rho_i^+) - n(\rho_i)}{2} \right\rfloor$.
            Set $K_0 = K_0^t$ and $M_0 = M_0^t$.
        else
            Set $n(\rho_i^+) = n(\rho_i) + \left\lfloor \frac{n(\rho_i^+) - n(\rho_i)}{2} \right\rfloor$.
        end if
    end while
    return $n_{max}(\rho_i, \epsilon) = n(\rho_i)$, $K_0$, $M_0$
end function
```

If (21) does not hold, we need to look with what probability the worst-case scenario occurs, i.e., what is the probability that all the users will have their lowest corresponding per-PRB rates ρ_i simultaneously. That probability is $\prod_{i=1}^{n} p_{R_i}(\rho_i)$, and if it is lower than the outage, i.e., if $\prod_{i=1}^{n} p_{R_i}(\rho_i) \le \epsilon$, it means that the planning can be done not for the worst-case per-PRB rate, but for higher ones, which in turn implies that fewer resources are needed for a user. This means that there are enough resources for user n to be admitted. Summarizing, we have the following admission policy for heterogeneous users:

Result 2. *Given a set of $n-1$ users with URLLC traffic in the cell, whose worst-case per-PRB rates are ρ_i, $i = 1, \dots, n-1$, with reliability requirement of $1-\epsilon$, a sufficient condition for a new user with worst-case per-PRB rate ρ_n to be admitted is if one of the following holds:*

$$\sum_{i=1}^{n} \frac{1}{\rho_i} \le K \left(\frac{T_{max}}{\Delta} - \frac{1}{\lfloor \frac{M}{n} \rfloor q} \right), \qquad or \tag{22}$$

$$\prod_{i=1}^{n} p_{R_i}(\rho_i) \le \epsilon. \tag{23}$$

VI. Performance Evaluation

A. Simulation setup

We have used a 5G trace with data measured in the Republic of Ireland as input parameters. These traces can be found in [10], with a detailed description in [11], and statistical analysis in [12]. The parameter of interest from the trace is the CQI with 15 levels, which serves to determine the per-PRB rate of a user in a slot. We have picked 6 users that were moving around. Based on the frequency of occurrence of a per-PRB rate for every user, we obtained the corresponding per-PRB rate probabilities (Table II).

The slot duration is 0.5 ms. The subcarrier spacing is 30 KHz, with 12 subcarriers per block, making the PRB width 360 KHz. The total number of PRBs is $K = 273$ [6], whereas the total number of computing resources is $M = 500$, where the processing rate per resource is $q = 1$ Mbps [13]. The inter-transmission period is $\tau = 1$ s.

B. Validations

We start by validating our theoretical result for the maximum number of homogeneous users that can be admitted (17). The reliability requirement is 100%. Three types of users from Table II are considered: 1, 3, and 5. The deadline is $T_{max} = 5$ ms. To obtain simulation results (in MATLAB), we start increasing the number of users, run the simulation and check whether there is a case when the packet is not processed within the deadline. If that is not the case, we increase the number of users by 1. Otherwise, we stop and the previous number is the maximum number of users that can be admitted. Fig. 4 shows the results vs. the size of the data transmitted at once. The first thing to observe is the perfect match between simulation and theory, which corroborates the validity of our analytical approach. The second observation we can make is the decline in the admissions as the data size increases. This is to be expected as it would take more resources to deliver and process more data during the same time. The third outcome is the higher number of type 3 users that can be admitted. The reason lies in the best worst-case channel conditions of user type 3. Namely, for user type 3, the worst per-PRB rate is r_5, as opposed to r_2 and r_1 for user types 1 and 5, respectively.

Next, we validate our result as a function of the maximum allowed latency (T_{max}). The reliability is again 100% ($\epsilon = 0$). The simulation is realized similarly to the previous scenario. To introduce diversity, now we show results for user types 2, 4, and 6 from Table II. The data size is 5 kbits. The other parameters remain unchanged from the previous scenario. Fig. 5

TABLE II
PER-PRB RATES AND THE CORRESPONDING PROBABILITIES FOR EVERY USER TYPE (6 IN TOTAL) FROM THE REPUBLIC OF IRELAND TRACE [11]

R (kbps)	48	73.6	121.8	192.2	282	378	474.2	712	772.2	874.8	1063.8	1249.6	1448.4	1640.6	1778.4
$p_{1,k}$	0	0.1	0.72	0.04	0.05	0.09	0	0	0	0	0	0	0	0	0
$p_{2,k}$	0	0	0.2	0.7	0.1	0	0	0	0	0	0	0	0	0	0
$p_{3,k}$	0	0	0	0	0.01	0.12	0.51	0.32	0.01	0.01	0.02	0	0	0	0
$p_{4,k}$	0	0	0	0	0	0.01	0.98	0.01	0	0	0	0	0	0	0
$p_{5,k}$	0.24	0.04	0.07	0.04	0.04	0.06	0.16	0.15	0.01	0.01	0.06	0.06	0	0.03	0.03
$p_{6,k}$	0.18	0.11	0.1	0.06	0.05	0.1	0.17	0.11	0.02	0.04	0	0.03	0	0.02	0.01

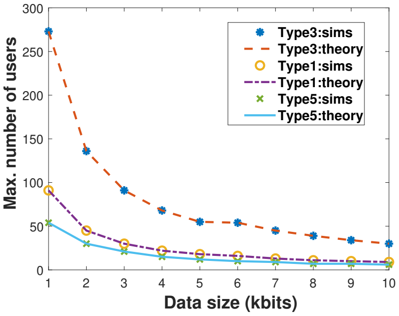

Fig. 4. The maximum number of users to be admitted with $T_{max} = 5$ ms and 100% reliability.

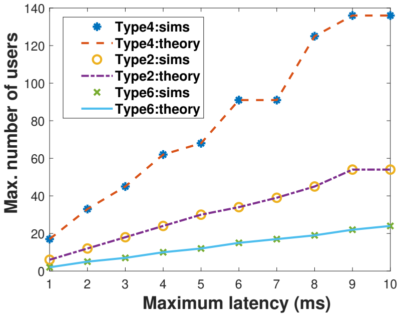

Fig. 5. The maximum number of users to be admitted with $\Delta = 5$ kbits and 100% reliability.

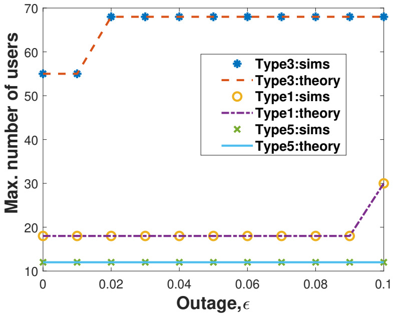

Fig. 6. The maximum number of users that can be admitted with $\Delta = 5$ kbits, $T_{max} = 5$ ms for different reliability.

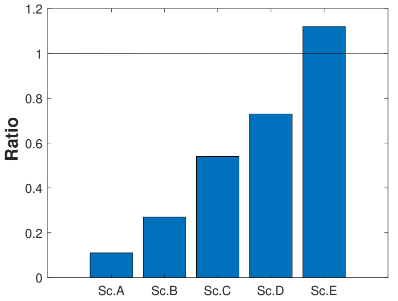

Fig. 7. The decision whether to admit user type 6 for different combinations of T_{max} and Δ, when $\epsilon = 0$ (Heterogeneous users).

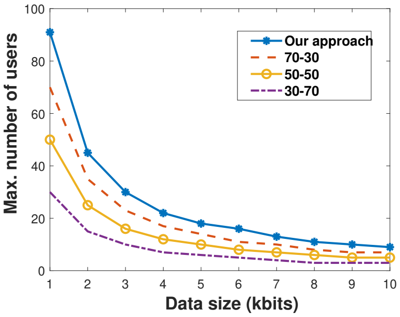

Fig. 8. The number of users to be admitted with our approach and when splitting strictly T_{max} between transmission and processing for users of type 1 with $T_{max} = 5$ ms.

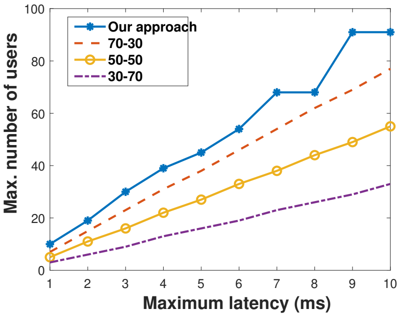

Fig. 9. The number of users to be admitted with our approach and when splitting strictly T_{max} between transmission and processing for users of type 2 with $\Delta = 3$ kbits.

depicts the results. Again, there is a perfect match between theory and simulations. At least 100% more type 4 users can be admitted because the worst-case channel conditions for user type 4 (their lowest CQI is 6) are better than for the other two user types, which experience the worst CQIs of 3 and 1. While it is expected that relaxing the latency would always allow for a higher number of admitted users, this is not the case. This is more emphasized with higher T_{max}, where increasing T_{max} by 1 ms does not always increase n_{max}.

C. Further assessments

The results so far pertain to the case of 100% reliability. We proceed next with investigating the impact of reduced reliability on the number of admitted users (homogeneous case). To that end, we consider user types 1, 3, and 5. The size of the data is $\Delta = 5$ kbits, whereas $T_{max} = 5$ ms. Fig. 6 depicts the results for different outages ϵ. What can be observed first is the higher number of type 3 users that can be admitted for the same reasons as in the scenario corresponding to Fig. 4. The second observation, which is rather surprising, is that the number of users that can be admitted does not increase drastically with the outage ϵ. This is completely different from the case when only the RAN limitations are considered when deciding on how many users to be admitted [14]. The rationale behind this stems from the large number of users receiving service. Namely, for a large n, $\lfloor K/n \rfloor = \lfloor K/(n+1) \rfloor$, and only where a shift down by 1 occurs, there is a jump in n_{max}.

Having considered the homogeneous users case until now, we proceed with evaluating the performance for heterogeneous users. As there is not a high dependency on the number of admitted users on ϵ, we consider the case of 100% reliability. In deciding whether or not to admit a URLLC user, after some other users are already present in the cell, we proceed as follows. We pick user types 1-5, and decide whether user type 6 can be admitted. In the following scenarios there are 3 type 1, 3 type 2 and 3 type 3 users, 2 users of type 4 and type 5. So, in total there are 13 users before the arrival of user type 6. We consider 5 scenarios in terms of Δ and T_{max}:

- Scenario A: $\Delta = 1$ kbits, $T_{max} = 5$ ms;
- Scenario B: $\Delta = 2$ kbits, $T_{max} = 4$ ms;
- Scenario C: $\Delta = 3$ kbits, $T_{max} = 3$ ms;
- Scenario D: $\Delta = 4$ kbits, $T_{max} = 3$ ms;
- Scenario E: $\Delta = 4$ kbits, $T_{max} = 2$ ms;

Fig. 7 shows the results. On the y-axis the ratio of the LHS and RHS of (21) is depicted. As long as this ratio is smaller than 1, user type 6 can be admitted in the cell. Note that for $\epsilon = 0$ condition (23) is never satisfied, so for a user to be admitted (22) must hold. As can be observed from Fig. 7, in Sc. A-D user type 6 is always admitted. The reason is that in these cases the maximum latency is not lower than 3 ms, or the packet size is not large enough, or both requirements are not too restrictive. On the other hand, in Sc. E, we have the case of both large data size and lower latency, hence, resources are not sufficient to admit user type 6.

D. Performance Comparisons

Next, we compare our approach of jointly allocating RAN and computing resources with the approach in which the number of admitted users is decided separately for RAN and edge cloud. The user of interest is type 1, and $T_{max} = 5$ ms. We consider three types of "separate approaches". In the first, exactly 50% of the latency will on transmission and the rest on processing. In the second approach, 30% of the time can be dedicated to transmission and the remainder on processing. Vice versa for the third type - 70% of the time can be spent on transmission and 30% on processing. For the separate-allocations, we determine the maximum number of users that can be admitted in regards to the RAN and computing resources, respectively, and then take the minimum of those two to be the number of users that can be served. For all scenarios, the reliability is 100%. Fig. 8 shows the results as a function of the data size transmitted at once. As can be observed, the number of admitted users is the highest with our approach, outperforming the others by at least 30%. The reason is that one resource can compensate for the other, which is not possible with the separate approaches. The second thing to observe is that from separate approaches $70-30$ performs the best. The rationale behind this is that there are more computing than RAN resources ($M = 500$ vs. $K = 273$). Therefore, the less restricted requirement on the transmission time (higher respective T_{max}) enables admitting more users.

Finally, we compare the three aforementioned "separate approaches" with our combined-resource approach for different T_{max}, when the amount of data transmitted at once by each user is 3 kbits. We consider users of type 2. Fig. 9 illustrates the results. Similar to the previous scenario, our approach outperforms the separate-resource consideration approaches by at least 30%. Also, among the separate approaches, $70-30$ performs the best for the same reasons as previously.

VII. Related Work

Admission control for eMBB traffic is considered in [15], where the channel variability of the users plays a crucial role in determining the maximum number of admitted users. When it comes to mMTC traffic, which is characterized by the least stringent requirements, the corresponding admission policies have been provided in [16]. However, neither [15] nor [16] consider the computation, and the allocation is performed only across PRBs. In [17], the authors consider the network slicing process for the three service types in 5G to determine the optimal amount of slices for each service type in order to satisfy traffic requirements. However, there are no indications about the number of URLLC users that can be admitted.

Further, [18] considers optimized resource allocation and transmission for URLLC users. The resource allocation in [18] is derived for both fixed and adaptive transmission attempt assignments. While reducing the required resources is one of the objectives in [18], both the problem setup and the goal are different from our work. To meet the reliability and strict latency requirements in 5G networks, in [19] the authors propose a periodic resource allocation scheme. Similar to our work, the packet sizes in [19] are assumed constant. However, [19] is limited in scope as the environment is a factory, and admission control is not the objective. Moreover, the computation part is not considered, as opposed to our work.

In [20] and [21], the joint admission of eMBB and URLLC traffic is considered. However, the setup in both of them is different from ours, since their goal is to maximize the number of eMBB users that can be admitted while serving all URLLC users. The main difference to our work stems from the fact that we do not assume to have enough resources to serve all URLLC users, like [20], [21]. Besides, we consider a traffic computation requirement, which makes our scenario more complex as we must perform allocation across two resources.

The work most similar in spirit to ours is [14], where the admission control is performed for URLLC traffic. Similar to our work, in [14] the maximum number of users that can be admitted is determined for homogeneous users, whereas for heterogeneous users the policy for admitting a newly arriving user is provided. However, only the delivery component of the latency is considered in [14], and the computation requirement is completely omitted. On the other hand, here we consider the general case by taking into account the delays caused by both the transmission and computation.

VIII. Conclusion

In this paper, we considered the admission control of URLLC users with computation requirements. We obtained the admission policies for both homogeneous and heterogeneous users, taking into account traffic parameters and channel conditions. For homogeneous users, we determined the maximum number of users that can be admitted in the cell, whereas for heterogeneous URLLC users the explicit inequality the newly arriving user needs to satisfy was provided, given the set of users (with different channel conditions) already being served. We ran simulations on a 5G dataset. In the future, we plan to consider the problem of admission control for mMTC traffic with computation requirements.

REFERENCES

[1] P. Popovski, K. F. Trillingsgaard, O. Simeone, and G. Durisi, "5G wireless network slicing for eMBB, URLLC, and mMTC: A communication-theoretic view," *IEEE Access*, vol. 6, 2018.

[2] M. Bennis, M. Debbah, and H. V. Poor, "Ultra-reliable and low-latency wireless communication: Tail, risk, and scale," *Proceedings of the IEEE*, vol. 106, no. 10, 2018.

[3] A. Goldsmith, *Wireless communications*. CUP, 2005.

[4] S. E. Elayoubi, S. B. Jemaa, Z. Altman, and A. Galindo-Serrano, "5G RAN slicing for verticals: Enablers and challenges," *IEEE Communications Magazine*, vol. 57, no. 1, 2019.

[5] G. Ku and J. M. Walsh, "Resource allocation and link adaptation in LTE and LTE Advanced: A tutorial," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, 2015.

[6] ETSI, "5G NR overall description: 3GPP TS 38.300 version 15.3.1 release 15." www.etsi.org, 2018. Technical specification.

[7] J. Navarro-Ortiz, P. Romero-Diaz, S. Sendra, P. Ameigeiras, J. J. Ramos-Munoz, and J. M. Lopez-Soler, "A survey on 5G usage scenarios and traffic models," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, 2020.

[8] S. Boyd and L. Vandenberghe, *Convex optimization*. CUP, 2004.

[9] S. A. Goldman and K. J. Goldman, *A practical guide to data structures and algorithms using Java*. Chapman and Hall/CRC, 2007.

[10] https://github.com/uccmisl/5Gdataset.

[11] D. Raca, D. Leahy, C. J. Sreenan, and J. J. Quinlan, "Beyond throughput, the next generation: A 5G dataset with channel and context metrics," in *Proc. of ACM MMSys*, 2020.

[12] F. Mehmeti and T. La Porta, "Analyzing a 5G Dataset and Modeling Metrics of Interest," in *Proc. of IEEE MSN*, 2021.

[13] C. Rublein, F. Mehmeti, M. Towers, S. Stein, and T. La Porta, "Online resource allocation in edge computing using distributed bidding approaches," in *Proc. of IEEE MASS*, 2021.

[14] F. Mehmeti and T. La Porta, "Admission control for URLLC users in 5G networks," in *Proc. of ACM MSWiM 2021*.

[15] F. Mehmeti and T. La Porta, "Admission control for consistent users in next generation cellular networks," in *Proc. of IEEE ICC*, 2019.

[16] F. Mehmeti and T. La Porta, "Admission control for mMTC traffic in 5G networks," in *Proc. of ACM Q2SWinet 2021*.

[17] H. Chien, Y. Lin, C. Lai, and C. Wang, "End-to-end slicing with optimized communication and computing resource allocation in multi-tenant 5G systems," *IEEE Tran. on Veh. Tech.*, vol. 69, no. 2, 2020.

[18] H. Shariatmadari, S. Iraji, Z. Li, M. A. Uusitalo, and R. Jäntti, "Optimized transmission and resource allocation strategies for ultra-reliable communications," in *Proc. of IEEE PIMRC*, 2016.

[19] Y. Han, S. E. Elayoubi, A. Galindo-Serrano, V. S. Varma, and M. Messai, "Periodic radio resource allocation to meet latency and reliability requirements in 5G networks," in *Proc. of IEEE VTC*, 2018.

[20] V. N. Ha, T. T. Nguyen, L. B. Le, and J. Frigon, "Admission control and network slicing for multi-numerology 5G wireless networks," *IEEE Networking Letters*, vol. 2, no. 1, 2020.

[21] N. U. Ginige, K. B. Shashika Manosha, N. Rajatheva, and M. Latva-aho, "Admission control in 5G networks for the coexistence of eMBB-URLLC users," in *Proc. of IEEE VTC-Spring*, 2020.