Serving HTC and Critical MTC in a RAN Slice

Vincenzo Mancuso[1], Paolo Castagno[2], Matteo Sereno[2], and Marco Ajmone Marsan[1,3]
[1]IMDEA Networks Institute, Madrid, Spain; [2]University of Turin, Italy; [3]Polytechnic of Turin, Italy
vincenzo.mancuso@imdea.org, matteo@di.unito.it, castagno@di.unito.it, ajmone@polito.it

***Abstract*—We consider a slice of a radio access network where human and machine users access services with either high throughput or low latency requirements. The slice offers both eMBB and URLLC service categories to serve HTC (Human-Type Communication) and MTC (Machine-Type Communication) traffic. We propose to use eMBB for both HTC and MTC, transferring machine traffic to URLLC only when eMBB is not able to meet the low latency requirements of MTC. We show that by so doing the slice is capable of providing very good performance to about one hundred MTC users under high HTC traffic conditions. Instead, running time-critical MTC over only eMBB is not doable at all, whereas using URLLC suffices for at most a few tens of devices. Therefore, our approach improves the number of users served by the slice by one order of magnitude, without requiring extra resources or compromising performance. To study system performance we develop a novel analytical model of uplink packet transmissions, which covers both legacy eMBB- or URLLC-based MTC, as well as our compound approach. Our model allows to tune slice parameters so as to achieve the desired balance between HTC and MTC service guarantees. We validate the model against detailed simulations using as an example an autonomous driving scenario.**

I. Introduction

In addition to high performance mobile broadband services for human type communications (HTC), 5G usage scenarios include machine type communications (MTC). Environments that require a mix of services with heterogeneous requirements are challenging to manage, but likely common. It is easy to foresee scenarios with a mixture of HTC and MTC traffic on a same slice or on co-deployed slices, in the very same base station (BS) and coverage area.

In this paper we focus on the coexistence of HTC and time-critical MTC traffic. As motivational example, we study the automotive environment, which requires a service mix with most stringent overall requirements: broadband HTC for infotainment requires high data rates, while critical MTC for automated driving requires very low latency and very high reliability [1]. However, our study is more generally applicable to scenarios such as smart factories, or smart cities.

Latency objectives for time-critical MTC traffic are of the order of few tens of ms, and reliability constraints are of the order of five to six nines. Throughput requirements for broadband services belonging to the HTC class are of the order of a few Mb/s per vehicle. How to support these requirements in 5G radio access networks (RANs) is not yet clear, but several tools are being developed to reach such goal [2]. A first tool is slicing, that allows network operators to create partitions of the RAN resources, which can be allocated in many ways, so as to obtain the desired performance. Another important tool are service categories, in particular eMBB (enhanced mobile broadband) and URLLC (ultra-reliable low-latency communications) [3], [4]. The eMBB service category is an evolution to higher performance of the traditional mobile services, mostly conceived for HTC. The URLLC service category is designed to handle limited numbers of users (up to few tens) with very strict requirements in terms of latency and/or reliability, and it can meet, e.g., the needs of the autonomous driving scenario [5], but can only accommodate few users. Thus, serving large numbers of machines with time-critical applications within a 5G slice remains a challenge.

While results already exist on the analysis of the performance of eMBB and URLLC in a downlink (DL) sliced BS [6], not much is known about uplink (UL). This is critical because the UL behavior in the considered network framework is likely to have more impact on performance, due to the possibility of (i) collisions of user transmissions, including when sent over URLLC, and (ii) using dedicated resources for URLLC while also allowing it to puncture the resources of other services without the coordination of the BS [7]. For UL, we also need to consider the possibility of using eMBB and leveraging the `RRC_CONNECTED` state for multiple, back-to-back network accesses. Such differences make existing downlink models [6], [8], [9] inadequate for our study.

In modeling the targeted MTC-HTC uplink traffic mix, we consider that the system key performance indicators (KPIs) are the throughput of HTC traffic and the fraction of MTC messages reaching their destination within a given latency.

To meet the application constraints on KPIs, we propose to use the eMBB service category for both HTC and MTC users. Furthermore, we propose to transfer critical MTC traffic to URLLC when eMBB is not able to meet the latency requirements. We show that by so doing the 5G RAN slice can provide very good performance for large numbers of HTC and MTC users, up to a few hundreds per base station. This is not possible by just using either eMBB or URLLC for MTC users. We also show that using a single slice for HTC and MTC is much more convenient (and flexible) than using two separate slices with orthogonal resources.

To prove the superiority of our proposed approach, we develop a novel analytical model of uplink transmissions in a base station slice, and validate it with detailed simulations. The performance analysis of BS services is mainly carried out using standard stochastic modeling tools; however, the study of the performance of the BS service processor requires the solution of a system of coupled queues with complex blocking phenomena that represents a generalization of the classical product form for queuing networks [10], and that bears no similarity to existing models of access schemes.

To summarize, the main contributions of this paper are the following: (i) we propose a novel resource management scheme for UL transmissions over eMBB and URLLC in a 5G

slice; (ii) we derive a novel analytical model of the behavior of the 5G UL; (iii) we validate our model against detailed simulations, and prove the accuracy of the analysis in spite of the introduced simplifications; (iv) we study the feasibility of eMBB and URLLC to support realistic numbers of connected machines — using an autonomous vehicles example in a highway scenario — in the presence of HTC traffic on the same slice, gaining interesting insight into system operations and showing that our approach performs better than several possible alternatives.

II. Background and Related Work

5G Radio Resources, Slices and Services. The 5G NR specifications [11] go beyond the pure scheduled OFDMA approach of 4G. With 5G, frames last 10 ms and are organized in 10 subframes, each of which includes a fixed number of slots. The number of symbols transmitted in a slot is normally 14, and the total number of slots per subframe depends on bandwidth. For instance, with a 20 MHz channel, and considering transmission subcarriers spaced 15 kHz apart, it is possible to allocate about 1300 slots per subframe.

According to 3GPP, a network slice instance is "a set of network functions and the resources for these network functions which are arranged and configured, forming a complete logical network to meet certain network characteristics" [12]. As such, applying network slicing in the 5G context allows handling different service classes or tenants like if by means of separate (virtual) networks. Indeed, slices in 5G are meant to serve the purposes of three main service categories: eMBB, mMTC, and URLLC, multiple instances of which can coexist on the same infrastructure in the presence of several tenants and service providers [13], [14]. For us, mMTC is not of interest because it targets massive MTC scenarios. We instead rely on eMBB and URLLC. Furthermore, a slice can offer multiple services, e.g., by using eMBB for regular data traffic and URLLC for urgent warning messages.

5G permits access to transmission resources following two orthogonal paradigms. The first one requires the user to obtain a transmission grant from the gNB (i.e., the 5G BS). The second paradigm is based on a grant-free transmission scheme, which is meant for either sporadic small-size transmissions or for traffic with short latency and high reliability requirements. While a grant-based scheme permits very efficient use of resources, under the full control of the gNB and its scheduler, it can hardly guarantee delays at millisecond timescale. Therefore, it is suitable for all kinds of HTC and for MTC in which latency and reliability are not an issue. In contrast, grant-free access schemes simplify the network access procedure at the cost of sacrificing efficiency in the use of resources. In fact, grant-free transmissions either require a semi-static and thus inefficient allocation of resources, or can collide and need to use conservatively low modulation and coding schemes not previously agreed with the target transmission recipient. This class of schemes suits the needs of URLLC, since grant-free transmissions do not incur connection establishment overheads. However, in order to be able to guarantee high reliability in addition to low delay, URLLC uses resources inefficiently: first, 5G allows URLLC users to transmit over a few symbols per slot, the rest of the symbols remaining unused (thus using "minislots" and wasting resources); second, each URLLC packet transmission can be repeated multiple times [2]. So, the price to pay to deploy a URLLC-based service is that a handful of users consume a large portion of cellular resources with little exchanged data. Indeed, due to lack of coordination, grant-free transmissions can collide with each other and with grant-based transmissions.

A practical key difference between grant-based and grant-free access schemes is that the former has to go through the random access procedure prior to obtaining transmission grants. Moreover, once the service is granted to a user, the user is promoted to the `RRC_CONNECTED` state, i.e., she is assigned a service position in the scheduler of the network processor and can keep using the network until her transmission queue gets empty. For practical implementation limitations, the number of service positions is finite, and service requests in excess to such number are blocked. Moreover, for each eMBB service, the `RRC_TIMEOUT` controls the evolution of user service. This per-user timeout counter is reset after each data packet transmission. Until the timeout expires, the service position remains associated with a user and cannot be shared.

Performance of URLLC Transmissions. Several papers and projects already studied issues related to resource orchestration for network slicing. There exist strong results on the performance of downlink URLLC schedulers, in which the BS controls resource puncturing and can optimize more than one service simultaneously [6], [8]. Several other works focus on multiplexing access schemes for URLLC exploiting either NOMA or grant-free approaches with a low density of active devices [15]–[17]. However, we are interested in the uplink behavior of URLLC with non-negligible probability of contending for limited resources, and in which BS coordination is not possible at all. Hence, we cannot reuse existing models.

Not many papers study the performance of URLLC traffic in the uplink. A system analysis of different transmission procedures (named Reactive, K-Repetition, and Proactive) for UL grant-free transmission of URLLC traffic is presented in [7]. A detailed simulation in a 21-cell scenario shows that grant-free transmissions can provide significantly lower latency at the desired reliability level (taken to be 10^{-5}) with respect to grant-based transmission, even at high network loads. The impact of power control in a similar scenario is investigated in [18], again by simulation. The possibility of using both dedicated and shared resources for the UL transmission of URLLC data is considered in [19], together with the possibility of adopting advanced receivers to resolve collisions. Also in this case, the performance study is based on detailed simulations. A hybrid resource allocation scheme is also considered in [20], that assumes URLLC grant-free transmissions to be repeated on both resources dedicated to each specific end user, or common to groups of users, so as to save channel resources. A simple analytical model of URLLC UL transmissions shows that significant resource saving is possible by means of resource sharing among users.

Our analysis leverages the analytical study presented in [21], which proposes an iterative approach to evaluate the perfor-

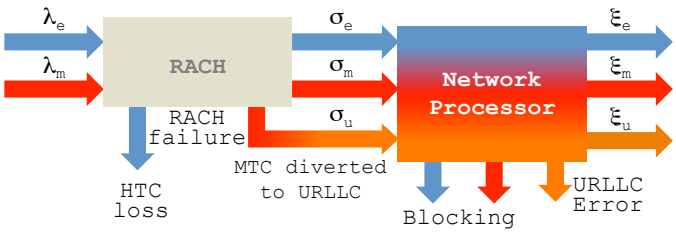

Fig. 1. Overview of a RAN slice with support for HTC and MTC

mance of a RACH that allows to isolate or share pools of access resources. We use the results presented in that paper as a modified plug-in for our model, which is needed to derive RACH throughput and the associated losses. Our model uses those results as input. Moreover, that paper does not account for the presence of URLLC at all, and does not consider the impact of the RRC_CONNECTED state on the performance of the network processor, so that we could not reuse the network processor model of [21] and instead derived a new one. Besides, although the 3GPP recommendations on slicing and virtualization [12] consider the possibility to use multiple slices/services, we have found no analytical work on the exploitation of multiple services for the same traffic type (in our case, we use eMBB and URLLC to serve MTC).

There exists no model for BSs where URLLC and eMBB coexist to provide service to machines and human customers like in the scenario we tackle, and where the resource allocation for HTC and MTC follows a non-orthogonal approach, although the advantages of non-orthogonal resource allocation for different service categories are discussed in [14].

III. A SLICE WITH COEXISTING HTC AND MTC TRAFFIC

We consider a gNB offering a slice in which the RAN operator provides service for HTC broadband devices and for several MTC devices with mild to stringent delay constraints. Hence we focus on eMBB and URLLC service categories. Mixing service categories is useful for scenarios like autonomous driving, in which humans access the network for infotainment, while connected vehicles report kinetic information to a cruise control system. Broadband devices generate HTC traffic, which is characterized by large volumes of data moved over short time intervals. MTC produces small packets sent at a quasi-periodic pace, and with relatively low latency and high reliability requirements. For instance, an autonomous driving unit would have to send up to a few tens of position update messages per second to be able to maintain a distance of a few meters from other vehicles and road edges [22]. MTC traffic can be carried by eMBB (with no latency or reliability guarantee), URLLC (only for few users) or a mix of the two, aiming at low latency and high reliability for larger numbers of users.

We study a specific slice that supports the eMBB and URLLC service categories to build coexisting HTC and MTC services, as depicted in Fig. 1. Traffic using a grant-based access scheme enters the system through the RACH subsystem. In our case, RACH resources are partitioned and allocated to more than one service category instance. We refer the reader to [23] for details on RACH operation, while a thorough study of its performance with multiple slices is reported in [21].

As shown in Fig. 1, when the slice load grows and approaches congestion, RACH access requests are prone to collisions and failures, which trigger new attempts after a random backoff interval (not explicitly shown in Fig. 1; retries occur within the RACH subsystem). eMBB grant-based service requests can fail due to an excessive number of unsuccessful attempts or because of an application timeout that triggers the drop of the request. Successful RACH attempts lead the service request to a second subsystem, the network processor, which is responsible for assigning transmission resources and deals with data transmission within a service. While eMBB requests are filtered by the sliced RACH subsystem, URLLC service requests go directly to the network processor. So, we propose that at least part of unattended eMBB requests, the part that corresponds to MTC devices, can turn into URLLC requests. This is done to serve those MTC processes that have deadlines, although they only resort to inefficient grant-free URLLC transmissions when the deadline approaches and service delivery becomes urgent. Therefore, MTC transmissions have access to multiple service categories: eMBB for initial service access attempts, plus URLLC for urgent service delivery of requests that failed on the RACH within a timeout.

For eMBB, originated by either HTC or MTC devices, the only possibility to transmit data consists in entering the RRC_CONNECTED state and receive service. However, since the gNB can only keep a limited number of terminals in the RRC_CONNECTED state, requests can be blocked, as shown in Fig. 1. In contrast, URLLC requests cannot be blocked, since they are grant-free, although their associated transmissions can fail due to collisions and decoding errors.

HTC traffic uses long backoff intervals between RACH attempts, in the order of hundreds of milliseconds, while MTC traffic can try several times the random access procedure with short backoff intervals spaced a few milliseconds apart; furthermore, MTC traffic is quasi-periodic and cannot tolerate more than a few tens of milliseconds for the initial network access procedure [23]. MTC requests are diverted to URLLC after a timeout expires, and will transmit without further trying to obtain transmission grants.

The network processor allocates transmission resources and reserves RRC_CONNECTED service positions to grant-based traffic, but can also keep shared pools of resources.

We consider that requests entering the network processor generate one or more packet transmissions, interleaved with short idle intervals (*think time*), and eventually leave the system either because users have no more data to send or because they handover to another cell due to their mobility patterns. Thus, access requests correspond to flows with a finite average duration. This duration models the time spent in the RRC_CONNECTED state for grant-based traffic, or the volume of data associated with a grant-free access requests.

Note that the network processor models the uplink data service in the RAN, and it accounts for slot scheduling and transmission, and for collisions due to URLLC activity. To model it, we consider that each service category instance has dedicated transmission resources. However, we also consider that, to enforce reliability, as proposed in [24], URLLC requests are mapped onto r_u packet replicas transmitted over the

TABLE I
NOTATION AND CELL PARAMETERS USED IN THE ANALYTICAL MODEL

Description	*Notation*
Subframe duration	τ
URLLC replicas over shared or dedicated resources	r_e, r_u
Arrival rate	λ_e, λ_m, λ_u
Arrival rate at network processor	σ_e, σ_m, σ_u
Request output rate	ξ_e, ξ_m, ξ_u
Average number of packets contained in an uplink message (packets per access request)	n_e, n_m, n_u
Probability that a packet does not conclude an uplink message (for geometrical distribution of bursts)	p_{re}, p_{rm}, p_{ru}
Probability of success on the RACH	s_e, s_m
Number of slots per subframe	c_e, c_m, c_u
Number of slots to transmit a packet	k_e, k_m, k_u
Probability that a slot is used for URLLC, conditional on y request arrivals (shared or dedicated)	$a_e(y)$, $a_u(y)$
Average (unconditional) number of slots used by URLLC (shared or dedicated)	b_e, b_u
Per-slot URLLC failure probability	π_e, π_u
URLLC packet failure probability over shared or dedicated resources	f_e, f_m

dedicated URLLC slots, plus r_e packet replicas over the eMBB resources dedicated to HTC. URLLC traffic is immediately transmitted using slots picked at random, with the following constraints: (i) all URLLC traffic is served in the subframe following the one in which it arrives, so to guarantee low latency, and (ii) URLLC replicas originated by the same terminal do not collide. If at least one replica is successfully received, then the URLLC user counts a success.

The goal of the system under evaluation is to serve as many HTC and MTC users as possible while keeping the per-packet failure probability below a given threshold. The performance of this system is not straightforward to assess, because of the intertwined relation between the traffic accessing different service category instances. The reason behind this behavior is threefold: (i) the traffic experienced on URLLC depends on the overflow of MTC traffic from eMBB; (ii) the overflow of MTC depends on the presence of eMBB traffic in the network processor, due to the fact that service positions available at the gNB for admitting devices to the `RRC_CONNECTED` state can be, at least partially, shared among services; and (iii) URLLC traffic is partially replicated over HTC resources allocated to eMBB, so that it slows down HTC traffic and changes the turnover rate of `RRC_CONNECTED` users.

IV. ANALYSIS OF THE SYSTEM

With reference to Fig. 1, we are interested in modeling HTC and MTC flows of requests in the slice, and in particular at the network processor. Table I summarizes the used notation. We use index e to denote resources, or use of resources, of the eMBB service associated with HTC, m for the eMBB service for MTC, u for URLLC, and x to refer to any of the above, where no further specification is needed. We denote by λ_x the rate of service requests in the system, by σ_x is the rate of requests that reach the network processor, and by ξ_x the average number of requests per unit time accepted and served by the network processor. Furthermore, every request brings in a number of packets with average n_x, and we assume that the number of packets sent after network access is geometrically distributed with parameter p_{rx}. For simplicity, we restrict the analysis to the case $\lambda_u = 0$, i.e., in absence of traffic sent directly to URLLC without attempting to access eMBB first.

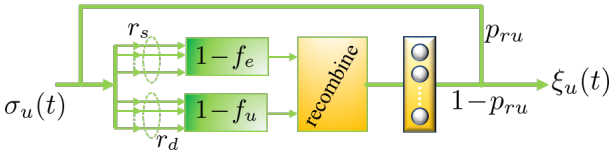

Fig. 2. Schematic representation of URLLC dynamics with packet repetitions over two sets of network slice resources. URLLC repetitions are recombined at the receiver and count as a single packet. The number of packets in a message is geometrically distributed, which is modeled with a probabilistic feedback.The infinite server after the recombine block models the inter-arrival time between packets.

A. Traffic Flows

Arrivals at the network processor. For grant-based traffic, the rate of arrivals at the network processor depends on the RACH operation, which is analyzed as described in [21]. Due to lack of space, we omit here the details, but we remark that the probability that a request successfully leaves the RACH is a function of the loads of HTC and MTC traffic, and of the slice configuration in terms of dedicated and shared resources. We denote by s_x such success probabilities, which account for the multiple RACH attempts and timeouts that can be imposed on the RACH operation on a per-service basis. For URLLC, we model the rate of arrivals at the network processor as the overflow of MTC requests that fail on the RACH. Thus, the incoming traffic intensities to the network processor are

$$\sigma_e = s_e\lambda_e; \quad \sigma_m = s_m\lambda_m; \quad \sigma_u = (1 - s_m)\,\lambda_m. \qquad (1)$$

Allocation of slots to services. In the network processor, we denote by c_x the number of slots per subframe reserved to each service category instance. However, we recall that URLLC can also transmit packet replicas on the c_e slots assigned to eMBB for HTC. Accordingly, the network processor operates as a set of three coupled queues: the first queue serves eMBB packets for HTC, the second queue serves MTC traffic that obtained a service grant, and the third queue serves grant-free transmissions over the URLLC service. Since URLLC can transmit on HTC resources, we further model the impact of such transmissions as a linear decrease of HTC capacity, proportional to the activity on URLLC. If we indicate with $b_e \leq c_e$ the average number of slots used by URLLC over the resources of the eMBB service of HTC, the average number of slots available for HTC becomes $c_e - b_e$. Computing b_e is therefore a key task in our analysis, and will be tackled in the next subsection jointly with the derivation of the collision probability and failure probability experienced over URLLC.

B. Average utilization of slots for URLLC

With an approximation that will be validated later, we model URLLC arrivals at the network processor as Poisson with rate σ_u. Assuming that the number of packets transmitted after accessing the network is geometrically distributed with parameter p_{ru}, each arrival corresponds to $n_u = \frac{1}{1-p_{ru}}$ packets, on average. Thus, since each MTC packet served by URLLC is replicated r_u times on dedicated URLLC slots and r_e times on resources shared by HTC and URLLC (see Fig. 2),

we can compute the distribution of the number of slots used by URLLC and its average.

We first consider the average number of slots per packet used by URLLC. With c_e and c_u available slots per subframe, and k_u slots needed to transmit a packet, the probabilities of a URLLC transmission over a slot of either eMBB or URLLC, conditional on the arrival of y requests, are:

$$a_x(y) = 1 - (1 - (k_u r_x)/c_x))^y, \quad x \in \{e, m\}, \tag{2}$$

since slots are selected uniformly at random and independently. Therefore, $c_e a_e(y)$ is the average number of slots used by URLLC over HTC's eMBB resources when y URLLC requests arrive at the network processor. Similarly, $c_u a_u(y)$ is the average conditional slot utilization over dedicated URLLC resources. With Poisson arrivals of intensity $n_u \sigma_u \tau$ packets per subframe, where τ is the subframe duration, the unconditional average number of slots used for URLLC over HTC is:

$$b_e = \sum_{y=0}^{\infty} \frac{(n_u \sigma_u \tau)^y}{e^{n_u \sigma_u \tau}\, y!} a_e(y) c_e = c_e \Big(1 - e^{-n_u \sigma_u \tau k_u r_e / c_e}\Big). \tag{3}$$

Similarly, URLLC uses on average the following number of slots out of its dedicated pool:

$$b_u = c_u \left(1 - e^{-n_u \sigma_u \tau k_u r_u / c_u}\right). \tag{4}$$

C. Distribution of URLLC slot utilization

We use a recursive formulation to derive the distribution of resources used by URLLC over either HTC's eMBB resources or resources dedicated to URLLC. We formulate the method for the former case, the latter being formally identical.

Let $\omega(\kappa, \ell)$ denote the number of configurations with ℓ users on URLLC using κ resources for their transmissions, which is less than the sum of resources used by individual users in a subframe, due to collisions.

The range of possible values for κ depends on ℓ. In particular, for $\ell = 1$ we have that $\kappa = k_u r_e$. For $\ell > 1$, κ belongs to the interval $\{k_u r_e, k_u r_e + 1, \ldots, k_u r_e \ell\}$.

In the following we provide a formula that allows the computation of the number of configurations $\omega(\kappa, \ell)$ by using expressions of the number of configurations computed for $\ell - 1$. With $\ell = 1$, $\omega(\kappa, 1)$ is equal to the number of possible choices of $k_u r_e$ objects (the slots used by r_e replicas) among c_e available objects (the available slots):

$$\omega(\kappa, 1) = \binom{c_e}{k_u r_e}. \tag{5}$$

The number of configurations $\omega(\kappa, \ell)$ can be computed from the number of configurations with $\ell - 1$ URLLC devices and $\kappa - i$ occupations with $i \in \{0, \cdots, k_u r_e\}$. Indeed, $\omega(\kappa, \ell)$ is the sum of all terms with $\ell - 1$ users on URLLC, each term multiplied by the number of configurations that use i resources when a user is added, and hence at most with $k_u r_e$ new slots:

$$\omega(\kappa, \ell) = \sum_{i=0}^{k_u r_e} \omega(\kappa - i, \ell - 1) \binom{\kappa - i}{k_u r_e - i} \binom{c_e - \kappa + i}{i}, \tag{6}$$

where the term $\binom{\kappa - i}{k_u r_e - i}$ accounts for transmissions colliding on resources already selected by other users, and $\binom{c_e - \kappa + i}{i}$ is the number of possible choices among the unused resources. Finally, the conditional probability of using κ slots when y requests occur, denoted by $q_e(\kappa, y)$, can be written as:

$$q_e(\kappa, y) = \omega(\kappa, y) \Big/ \sum_{i=k_u}^{y\, k_u r_e} \omega(i, y). \tag{7}$$

Once the probability mass function of y is known,[1] the distribution of κ is the weighted sum of (7).

D. URLLC failure probability

A URLLC transmission fails if all its replicas fail due to undecodable collisions. Furthermore, we assume (as normally done in the literature) that the collision between MTC's URLLC and HTC's eMBB transmissions on a given slot can be decoded by MTC with probability $1 - \alpha$, and cannot be decoded by HTC.

For the case of transmissions over URLLC resources, the probability that a user accesses a slot is $\frac{k_u r_u}{c_u}$ and the distribution of the number of users that access a slot is binomial with population $\mathcal{A}$, where $\mathcal{A}$ represents the packets to transmit in a subframe,[2] and is approximated with a Poisson distribution with average $\sigma_u n_u \tau$, because each request brings in n_u packets to transmit, and τ is the subframe duration. Therefore, the joint distribution of the number $\mathcal{N}$ of users that access a slot with $\mathcal{A}$ packets to transmit in a subframe is

$$\begin{aligned} p_{\mathcal{N},\mathcal{A}}(x, y) &= p_{\mathcal{N}|\mathcal{A}}(x, y)\, p_{\mathcal{A}}(y) = \\ &= \binom{y}{x} \left(\frac{k_u r_u}{c_u}\right)^x \left(1 - \frac{k_u r_u}{c_u}\right)^{y-x} \frac{(\sigma_u n_u \tau)^y}{y!} e^{-\sigma_u n_u \tau}. \end{aligned} \tag{8}$$

Taking the average over $\mathcal{A}$, we obtain the distribution of the number of active users per slot:

$$p_{\mathcal{N}}(x) = \frac{(\sigma_u n_u \tau k_u r_u / c_u)^x}{x!} e^{-\sigma_u n_u \tau k_u r_u / c_u}. \tag{9}$$

Therefore, the number of transmitting users per slot is Poisson as well, and we can compute the collision probability as follows: a user observes a collision if at least one more user accesses the same slot. Hence, the per-slot collision probability π_u is expressed as

$$\pi_u = \frac{1 - p_{\mathcal{N}}(0) - p_{\mathcal{N}}(1)}{1 - p_{\mathcal{N}}(0)}. \tag{10}$$

Finally, considering the adoption of r_u replicas for each packet, the failure probability over dedicated resources is:

$$f_u = \left(1 - (1 - \pi_u)^{k_u}\right)^{r_u}. \tag{11}$$

For URLLC transmissions over HTC resources, in addition to collisions of URLLC slots, it is also possible that URLLC collides with HTC. Specifically, an HTC transmission collides on a slot used by URLLC with probability b_e/c_e, and uses $n_e k_e$ slots per packet. Therefore, a flow ξ_e of successfully served HTC requests uses $\xi_e n_e k_e / (1 - b_e/c_e)$ slots out of the

[1] The probability mass function of the requests arriving from independent users can be also estimated empirically or approximated with a Poisson distribution when the number of sources is large.

[2] Since we work at subframe level, we can safely assume that the packets come all from different users.

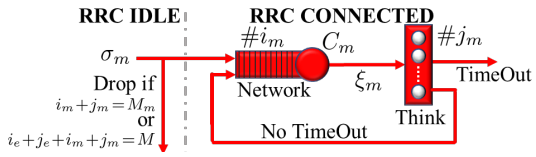

Fig. 3. Queuing network modeling the network processor for MTC

available c_e slots per subframe. Thus, the per-packet URLLC failure probability, with r_e replicas, is

$$f_e = \left(1 - \left((1-\pi_e)\left(1 - \alpha\frac{\xi_e n_e k_e}{c_e - b_e}\right)\right)^{k_u}\right)^{r_e}, \qquad (12)$$

where π_e is derived like π_u, and the factor $\left(1 - \alpha\frac{\xi_e n_e k_e}{c_e - b_e}\right)$ accounts the fact that a collision with HTC transmissions results in a lost slot for URLLC with probability α. The computation of ξ_e requires the analysis of the network processor and will be tackled in the next subsection. The overall success probability of URLLC packets is simply given by $1 - f_e f_u$.

E. Network processor model

Since the gNB has only M positions for concurrent service instances in the `RRC_CONNECTED` state, when a new arriving request finds the M positions busy, it is rejected.

We assume that the gNB allocates dedicated portions of bandwidth to each service category instance, which are equally shared by the instances of that service that are in state `RRC_CONNECTED`. This corresponds to the allocation of c_x slots to each one of the three service category instances. On the contrary, to enhance flexibility in the use of resources, services can compete for positions in the network processor. In particular, out of the M positions, we reserve k_e (resp. k_m) to HTC (resp. MTC). The remaining positions are shared. Hence, the maximum number of concurrent services of the eMBB service instance of HTC (resp. MTC) is equal to $M_e = M - k_m$ (resp. $M_m = M - k_e$).

The dynamics of service requests arriving at the network processor for the eMBB service instance associated with MTC, are described by the queuing network in Fig. 3. The network processor for the eMBB service of HTC is similar, although its data service capacity is $c_e - b_e$, which accounts for the activity on URLLC.

The customers in this queuing network represent service requests arriving from the RACH. We assume these arrivals can be described as a Poisson process with rate σ_m (we will validate this assumption against simulations). In this queuing network we identify two stations labelled as `Network` and `Think`, with i_m and j_m customers, respectively. `Network` is a processor sharing queue with service rate equal to c_m, which models the sharing of transmission slots across active services. The infinite-server station labelled as `Think` represents the idle (or thinking) time of active services. When the time spent in this station ends, customers are routed back to `Network` if the instance of their think time was shorter than `RRC_TIMEOUT`, and they leave the system otherwise.

Arriving customers can be dropped for two different conditions: $i_m + j_m = M_m$ or $i_m + j_m + i_e + j_e = M$. The former condition blocks service requests when the number of active services in the service instance is equal to its maximum allowed M_m (local constraint), while the latter blocks new arrivals when the sum of the active services in the network processor (i.e., overall in the slice) is equal to M.

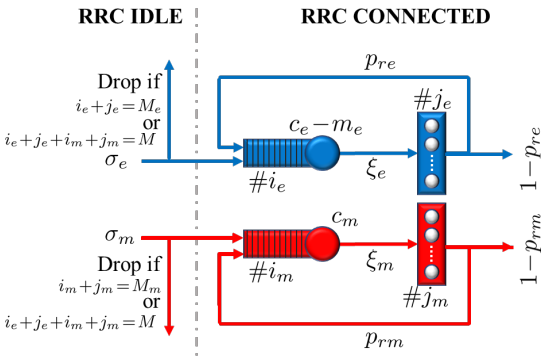

Fig. 4. Network processor for the slices MTC and eMBB

Note that the described queuing system does not admit a product form solution [10] because of time-dependent routing at the exit of the `Think` station and because of blocking. Let us first focus on customer routing. If we denote by T_o the value of `RRC_TIMEOUT`, and by μ_m the rate of the exponential distribution of the time spent in the `Think` station, we can modify the queuing network model described so far by replacing the distribution of the infinite server station `Think` by an exponential distribution with service rate μ_m, truncated at T_o, with average $E[T_{\min}] = p_{rm}/\mu_m$, where $p_{rm} = 1 - e^{\mu_m T_o}$ is the probability that a new request is generated before the expiry of the `RRC_TIMEOUT`. With this transformation we obtain a probabilistic routing in the queuing network, at the cost of a mild approximation (we will validate also this approximation in the next section).

While the minimum between an exponential service time and a timeout exhibits a distribution that is not suitable for the product form solution of the queuing network, it can be well approximated by means of a distribution with rational Laplace transform, making the modified service distribution of the `Think` station suitable for product form. The same approach applies to non-exponential service time distributions at the station `Think`: this is for example the case of quasi-periodic updates in autonomous driving scenarios, as we will use in simulation experiments.

Disregarding the interaction between HTC and MTC through URLLC, the service capacity of HTC is c_e. However, the puncturing of URLLC reduces the capacity available for HTC. We reflect this in our model by setting the HTC capacity to $c_e - b_e$, where b_e is an average. We will validate also this simplification by comparing the model results with the simulation outcomes.

We now come to blocking due to local and global constraints, considering that we have two coupled networks of queues (see Fig. 4). Those constraints introduce a dependence between queues that is similar to the mutual interdependence among different classes or networks that was already studied in several settings (see [25], [26], and [27]).

The computation of the performance measures requires the use of specialized algorithms that account for the peculiarities of this class of queuing networks. In our analysis we use the

algorithm for the normalization constant proposed in [26] to compute the blocking probability of the network processor.

V. Performance evaluation

Our goal in this section is to validate the simplifying assumptions introduced in the development of the model, to study the impact of the main system parameters (traffic generated by HTC users, number of MTC users, URLLC replication schemes), and to prove the superiority of our proposed approach based on leveraging both eMBB and URLLC for MTC, with respect to simpler network management approaches.

For these purposes, we consider a 5G gNB located along a highway, with coverage of 1 km in both directions. Vehicles of different types move at an average speed equal to 100 km/h, so that the average time spent in the cell is about 1.2 minutes. We consider one isolated slice of the gNB, in which resources are allocated to three service instances, one eMBB (we refer to it as e) for HTC traffic, and one eMBB (m) plus one URLLC (u) for MTC traffic. HTC traffic can correspond to voice, or messaging, or infotainment used by vehicle passengers, while MTC traffic originates from autonomous driving applications (for example, we can think of truck platooning).

Service e in the uplink direction is used by HTC for transmission of packets with average size equal to 100 kb, that request access at random times. The amount of traffic generated by each user is defined by the duration of the think time between the end of a packet transmission and the request for the next packet transmission. With average think time equal to 100 ms, the HTC traffic generated by one vehicle is up to 1 Mb/s, while with average think time equal to 20 ms, it is up to 5 Mb/s. The performance indicator for HTC traffic is in terms of throughput.

Service m is used by MTC for the transmission of data necessary for autonomous driving (e.g., vehicle position, speed, acceleration, etc.) organized in 1 kb data units that request access quasi-periodically, once every at least[3] 60 ms (that corresponds to 1.6 meters at 100 km/h). Hence, the MTC traffic generated by one vehicle is up to 10 kb/s. If an MTC RACH request does not obtain service from the gNB resources of m within 50 ms, and risks missing the MTC latency deadline, which is set to 60 ms, the request is transferred to service u, where the data units corresponding to the request are transmitted using the URLLC procedures (as long as the vehicle remains in the cell). The performance indicator for MTC traffic is in terms of throughput of the requests meeting the 60 ms latency deadline, that is set between 99% and 99.999% of the offered MTC traffic, corresponding to miss probabilities between 10^{-2} and 10^{-5}.

The gNB uses 20 MHz for uplink transmissions, with frames of 10 ms and subframes of 1 ms. According to 5G numerology, about 1300 slots are available per ms with that bandwidth. Of those, 750 are exclusively allocated to service e, 350 to service m, and 200 to service u. RACH access opportunities exist every 10 ms for requests using eMBB. Of the 54 RACH preambles, 30 are reserved for service e, 10 for service m, and the remaining 14 are shared between the two eMBB services (remember that URLLC does not access the RACH). The maximum number of allowed RACH transmissions before failure is 10 for service e and 3 for service m. The gNB can accommodate a maximum of 200 users in the `RRC_CONNECTED` state. Of those 200 positions, 50 are exclusively reserved for service e, 50 for service m, and the remaining 100 are shared. Thus, the allocation of these resources among services is not orthogonal. We consider a high modulation efficiency for service e, equal to 8 bits per symbol, and a lower efficiency for services m and u, equal to 4 bits per symbol. Data units that reach the URLLC service are transmitted a total of 4 times for improved reliability. We will look at the case 2+2 (i.e., $r_u = 2$ transmissions on the dedicated resources of u, and $r_e = 2$ transmissions that puncture the resources of service e), and at the case 1+3 (i.e., $r_u = 1$ and $r_e = 3$). A collision between URLLC transmissions on a minislot of the dedicated resources of u implies a loss of both contents. Instead, thanks to a higher transmission power of URLLC, a collision on the resources of e implies a loss of the eMBB content with probability 1, and a loss of the URLLC content with probability $\alpha = 0.01$.

A. Model Validation

The first numerical results are shown in Fig. 5, where we report a comparison between the results obtained with our network processor model (see Fig. 1) and those produced by a detailed simulator developed in Matlab. The curves in Fig. 5 report the number of dedicated and shared slots used per frame (left vertical axis), as well as the failure probabilities for MTC and HTC (right vertical axis) in the case of 2+2 URLLC transmissions and low-load HTC traffic with think time value equal to 100 ms. For simulation experiments we did not use standard simulators (like, e.g., ns3 or OMNET) due to the presence in our proposal of very innovative features that are not yet available in such tools. The main differences between the model and the simulator are in the fact that in the simulator the overflow traffic from service m to u is not Poisson. Similarly, none of the flows of requests that leave the RACH and enter the Network processor is Poisson in the simulator. Moreover, the interval between the end of an MTC packet transmission and the next one in the simulator is constant, equal to 60 ms, while in the model we assume a negative exponential distribution with average 60 ms. Moreover, in the simulator, the coupling between MTC's URLLC and HTC's eMBB at the network processor is evaluated on a per-subframe basis, counting the actual utilization (while the model uses the average) of resources due to the activity of HTC and MTC in the eMBB service e. Simulation results are reported together with their confidence intervals at 99% confidence level, but intervals are barely visible in the figure because they are quite small. We can see that the match among simulation and analytical results is extremely good. For this reason, in the rest of this section we only report model results.

B. Performance with One Slice Serving HTC and MTC

In Fig. 6 we report the curves of the throughput of MTC traffic, normalized to the offered load, versus the total number of MTC users in the considered slice. Looking at these

[3]The 60 ms are counted from the end of the previous transmission, so that the average interarrival time of requests is longer than 60 ms.

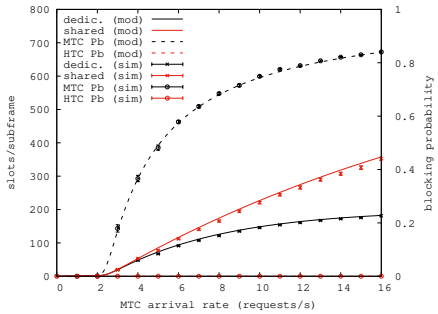

Fig. 5. Validation with 2+2 retransmissions and with HTC think time equal to 100 ms

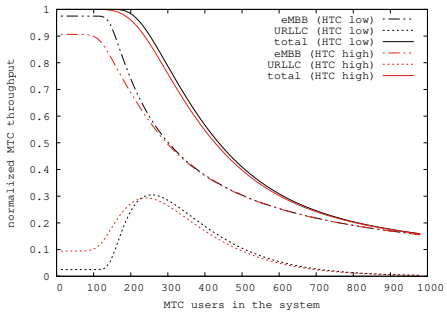

Fig. 6. MTC performance and breakdown with 2+2 repetitions and HTC think time of 20 ms

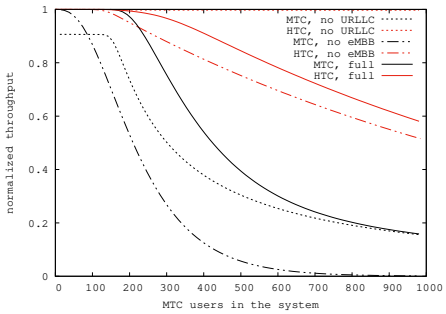

Fig. 7. MTC and HTC performance under multiple MTC schemes with 2+2 repetitions and HTC think time of 20 ms (high HTC traffic)

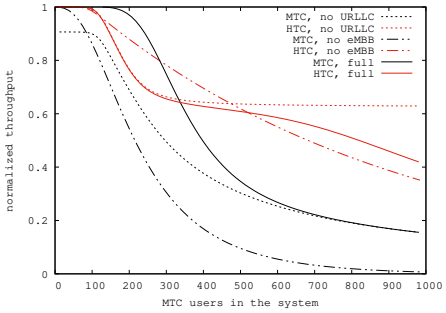

Fig. 8. MTC and HTC performance under multiple MTC schemes with 1+3 repetitions and HTC think time of 100 ms (high HTC traffic)

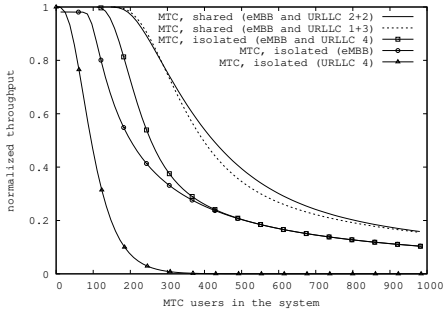

Fig. 9. MTC performance with a single slice ("MTC, shared") and with a stand-alone slice for each traffic type ("MTC, isolated").

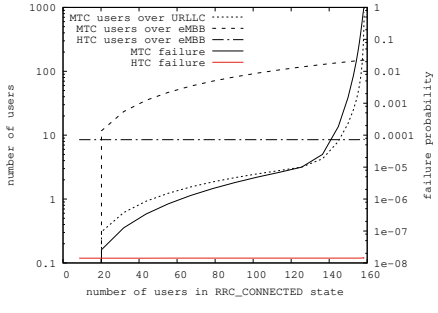

Fig. 10. Users in the system and failure probability of MTC with 2 + 2 repetitions and HTC think time of 20 ms (low HTC traffic)

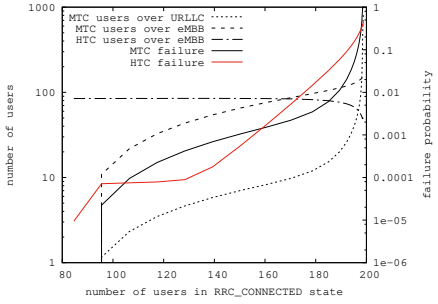

Fig. 11. Users in the system and failure probability of MTC with 2 + 2 repetitions and HTC think time of 20 ms (high HTC traffic)

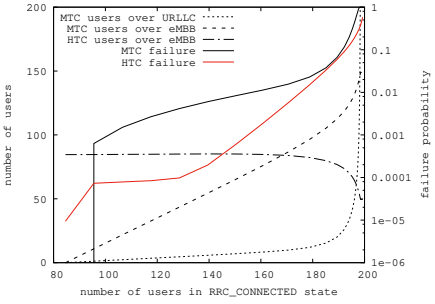

Fig. 12. Users in the system and failure probability of MTC with 1 + 3 repetitions and HTC think time of 100 ms (high HTC traffic)

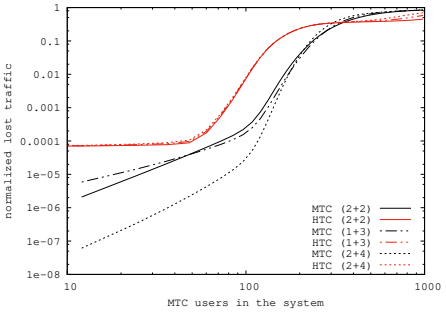

Fig. 13. Failure in traffic delivery over MTC and HTC with URLLC replication schemes (high HTC traffic, 20 ms think time)

performance parameters allows us to obtain an indication on the maximum number of autonomous vehicles that can be supported in the slice with satisfactory performance using our approach. Acceptable performance is achieved when, e.g., the overall MTC throughput is at least 99.9% of the offered MTC traffic, so that the probability for MTC transmissions to miss their 60 ms deadline does not exceed 10^{-3}. Solid lines refer to the total throughput, while dot-dashed lines refer to the throughput on service m, and dotted lines refer to the throughput on service u. Black curves refer to low HTC traffic (generated by an average number of 8.4 HTC users with think time equal to 20 ms, and corresponding to 10% of the capacity of service e, i.e., of 750 slots per ms), and red curves refer to high HTC traffic (90% of the capacity of service e, i.e., 75.6 HTC users on average). Fig. 6 shows results for the case of 2+2 URLLC transmissions, and think time equal to 20 ms for HTC, but results for the case of 1+3 URLLC transmissions, and think time equal to 100 ms for HTC are very similar. The total MTC throughput is close to 1 (above 0.99) until the number of MTC users exceeds 165 (with high HTC traffic) or 190 (with low HTC traffic). These numbers reduce respectively to 135 and 170 users if we seek for MTC throughput above 99.9%. However, if we consider ultra-high reliability, in which the normalized throughput must be guaranteed above 99.999% (deadline miss probability for MTC lower than 10^{-5}), the number of MTC users cannot exceed just a few units in case of high HTC traffic and up to 145 in case of low HTC traffic, in the setups we have tested. As expected, with low HTC traffic, numbers are higher than with high HTC traffic, but the difference is not drastic, if reliability requirements are not extreme. The main role of URLLC is to provide the throughput that the eMBB service e is not able to offer to MTC traffic, until collisions of URLLC packets cannot sustain the load. From this we can conclude that the number of autonomous vehicles that can be supported in the slice with satisfactory performance using our approach is of the order of 150-200,

depending on the amount of HTC traffic.

C. Performance when MTC Uses Only One Service or Both

The next set of results is reported in Figs. 7 and 8, where we plot the normalized throughput of both HTC and MTC traffic (in red and black, respectively) versus the total number of MTC users in the slice. Three cases are considered for MTC traffic: (i) MTC uses either the two services m and u, or (ii) just the eMBB service m (using also the resources of u), or (iii) just URLLC, through service u (with all the resources of m and u used by u). The purpose of showing these results is to prove the superiority of our choice of using both services m and u to support the autonomous driving application in the cell we consider. Fig. 7 shows results for the case of 2+2 URLLC transmissions, and think time equal to 20 ms for HTC, while Fig. 8 shows results for the case of 1+3 URLLC transmissions, and think time equal to 100 ms, always for high HTC traffic.

Note that the solid black curves in Fig. 7 are the same as in Fig. 6. As regards MTC traffic, we clearly see that for both figures the use of both m and u services is a must in order to admit a reasonable number of users with acceptable QoS. If only the grant-based access service m is used, the throughput loss is of the order of 9%, even for quite low numbers of MTC users in the slice. If only the grant-free access service u is used, performance dramatically degrades even for very low numbers of MTC users, and our model reveals that URLLC alone cannot sustain more than about 30 to 35 MTC users with a normalized throughput higher than 99% (note that these values match the ones reported by 3GPP for the number of admissible URLLC users within a cell).

The performance of HTC is quite different in the two figures. Whereas in Fig. 7, regardless of the handling of MTC traffic, the normalized throughput of HTC degrades slowly and regularly, in Fig. 8 we observe a significant drop in the HTC performance between 100 and 300 MTC users when MTC uses service m. This is due to the competition between the two eMBB services e and m for the shared positions in the `RRC_CONNECTED` state. On the contrary, the further decrease that we see beyond 300 MTC users, except for the case in which URLLC is not used, is due to collisions because of URLLC transmissions.

D. Performance with or without Sharing

In Fig. 9 we compare the throughput achieved by MTC traffic when the partial resource sharing described and evaluated so far is enforced (curves labeled as "MTC shared"), and in the simpler cases in which MTC and HTC use isolated slices (curves labeled as "MTC isolated"). In both cases MTC uses m and/or u services. The total amount of resources is the same in the two scenarios. However, in the case of isolated slices, the resources statically assigned to MTC are: 550 slots, 15 RACH preambles and 100 positions in the `RRC_CONNECTED` state. Thus, HTC is assigned 750 slots, 39 RACH preambles and 100 positions in the `RRC_CONNECTED` state. This allocation follows the proportion of dedicated resources in the "shared" case evaluated so far, except the "isolated" approach does not leave any shared pool of resources. Due to this separation of resources between MTC and HTC traffic, in the scenario with isolated slices, the puncturing of URLLC (with 4 packet copies) insists on MTC traffic. The superiority of the shared slice approach is quite clear from numerical results, for any value of minimum acceptable throughput. In addition, using for MTC only one service yields worse performance, especially if the selected service is URLLC.

E. Failure Probabilities

The next set of results comprises Figs. 10 to 12. They report the average number of HTC users on service e, and of MTC users on services m and u, versus the total number of users in the `RRC_CONNECTED` state in the gNB. Also reported are the curves of the HTC failure probability and of the MTC failure probability (counted on service u only, since failures on service m are not real failures, but just transfers to service u). Failure probabilities refer to the right vertical axis, while numbers of users refer to the left vertical axis. The three figures report results for low and high HTC traffic, for HTC think time of 20 and 100 ms, and for 2+2 and 1+3 URLLC transmissions, as indicated in labels and captions in the figures.

In low HTC traffic cases (the plot with 1+3 transmissions and 100 ms think time is very similar to the one shown in Fig. 10), we can see that the HTC failure probability is negligible, and the MTC failure probability remains below 10^{-5} until about 125 users in the `RRC_CONNECTED` state. The number of HTC users in the `RRC_CONNECTED` state is low, much less than the number of reserved positions (less than 10 w.r.t. 50). The number of MTC users in the `RRC_CONNECTED` state over service m grows well over 100, consuming all of its reserved positions (50) and practically all those that are shared (100). The number of MTC users over service u remains low until 130-140 users in the `RRC_CONNECTED` state, and then explodes, causing a surge in the MTC failure probability. These figures tell us that with low HTC traffic (10% of the reserved capacity) it is possible to serve about 130 vehicles implementing autonomous driving, with the desired QoS.

On the contrary, with high HTC traffic (90% of the reserved capacity), Figs. 11 and 12 tell us a different story. With 2+2 transmissions, at URLLC loss probability around 10^{-5}, the number of MTC users on services m and u is of the order of a few units. Instead, with 1+3 transmissions, we can have about 10 users, almost all in m. Both cases are not acceptable for the considered scenario. High numbers of MTC users can be obtained only with quite high values of the MTC failure probability, which is also not acceptable for autonomous driving and applications alike. It must be observed that the HTC failure probability in this case grows very high, especially when the number of URLLC users becomes high, as expected due to collisions. This tells us that, as expected, the system can offer good performance to MTC traffic if the load due to HTC traffic is low. This can be guaranteed with appropriate HTC user admission control policies.

F. Different URLLC Replication Schemes

Since we have observed that adequate MTC performance require the use of URLLC to guarantee reliability in traffic delivery, we finally compare different URLLC replication schemes in Fig. 13. The figure shows the fraction of traffic that

is not delivered (i.e., the complement to one of the normalized throughput) for MTC and HTC vs the number of MTC users. HTC traffic is high and HTC think time is short (20 ms), which corresponds to the most challenging setup among the cases studied in this paper. In addition to the 2+2 and 1+3 replication schemes used for the performance evaluation described so far, here we add the case 2+4 of 2 URLLC replicas over dedicated resources (service u) and 4 replicas over the resources of HTC (service e). The curves depicted in the figure show that if the total number of replicas is constant (cases 2+2 and 1+3) the performance of both HTC and MTC is comparable. Using more resources for URLLC, as in the case with 2+4 transmissions per URLLC packet, the loss of MTC can be significantly reduced. For instance, with a loss target below 10^{-5}, it is possible to support just 20 to 30 MTC users with the 2+2 and 1+3 replication schemes, and up to about 100 MTC users with the 2+4 scheme. This remarkable gain is obtained at quite a small cost for HTC. Indeed, the increase in loss of MTC traffic is marginal, especially considering that HTC traffic does not need to target loss rates much below 10^{-3}, and it becomes noticeable only for very high numbers of MTC users, where the MTC performance is bad, so that operating at those numbers of MTC users is not desirable. We can conclude that the proposed approach is capable of offering good performance to MTC traffic also when HTC traffic is high, by adequately managing the number of replicas of URLLC transmissions that puncture HTC resources.

VI. Conclusions

We presented a detailed analytical model for the performance evaluation of uplink transmissions in a gNB slice supporting HTC and MTC traffic through two eMBB and one URLLC service instances. We applied our performance model to a highway scenario supporting autonomous driving as well as infotainment services. Numerical results show that the simultaneous use of one eMBB service and URLLC for MTC is mandatory to achieve the QoS required by autonomous driving services. In addition, they indicate that, with the considered system parameters, the maximum number of autonomous vehicles that can be supported is of the order of one hundred; a number that can be suitable for an early phase of autonomous driving adoption. Finally, results show that for the system to operate properly, with the desired QoS for MTC, e.g. for the autonomous driving application, the load induced by HTC, e.g., by infotainment services, must be kept low, unless URLLC traffic is allowed to transmit several packet replicas over HTC's eMBB resources.

Acknowledgements

V. Mancuso was supported by the Ramon y Cajal grant RYC-2014-16285 from the Spanish Ministry of Economy and Competitiveness. This work was partially supported by the EU 5GROWTH project (Grant No. 856709), and by the Region of Madrid through the TAPIR-CM project (S2018/TCS-4496).

References

[1] 3rd Generation Partnership Project, "Service requirements for the 5G system; (Release 15)," 3GPP TS22.261 v16.3.0, Apr. 2018.

[2] Z. Li, M. Uusitalo, H. Shariatmadari, and B. Singh, "5G URLLC: Design Challenges and System Concepts," in ISWCS, Aug 2018.

[3] Recommendation ITU-R M.2083, "IMTVision – Framework and overall objectives of the future development of IMT for 2020 and beyond)," Tech. Rep., Sep. 2015.

[4] "Service requirements for the 5G system; (Release 15)," 3GPP TR 38.913, Aug. 2017.

[5] "Study on NR Vehicle-to-Everything (V2X)," 3GPP, TR 38.885 Release 16 V16.0.0, March 2019.

[6] A. Anand, G. de Veciana, and S. Shakkottai, "Joint Scheduling of URLLC and eMBB Traffic in 5G Wireless Networks," IEEE/ACM Transactions on Networking, vol. 28, no. 2, p. 477–490, Apr. 2020.

[7] R. Abreu, J. Thomas, K. Pedersen, G. Berardinelli, and P. Mogensen, "System Level Analysis of eMBB and Grant-Free URLLC Multiplexing in Uplink," in IEEE VTC Spring, Jun. 2019.

[8] Chih-Ping Li, Jing Jiang, W. Chen, Tingfang Ji, and J. Smee, "5G ultra-reliable and low-latency systems design," in EuCNC, 2017.

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

[10] F. Baskett, K. M. Chandy, R. R. Muntz, and F. G. Palacios, "Open, Closed, and Mixed Networks of Queues with Different Classes of Customers," Journal of ACM, vol. 22, no. 2, pp. 248–260, Apr. 1975.

[11] "System Architecture for the 5G System," 3GPP TS 23.501 Version 15.2.0, 2018.

[12] "Study on management and orchestration of network slicing for next generation network," 3GPP, 3GPP TR 28.801 Version 15.1.0 - Release 15, 2018.

[13] X. Foukas, G. Patounas, A. Elmokashfi, and M. K. Marina, "Network Slicing in 5G: Survey and Challenges," IEEE Communications Magazine, vol. 55, no. 5, pp. 94–100, May 2017.

[14] 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, pp. 55 765–55 779, 2018.

[15] I. Gerasin, A. Krasilov, and E. Khorov, "Flexible Multiplexing of Grant-Free URLLC and eMBB in Uplink," in IEEE PIMRC, 2020.

[16] S. Doğan, A. Tusha, and H. Arslan, "NOMA With Index Modulation for Uplink URLLC Through Grant-Free Access," IEEE Journal of Selected Topics in Signal Processing, vol. 13, no. 6, pp. 1249–1257, 2019.

[17] W. Yang, C. Li, A. Fakoorian, K. Hosseini, and W. Chen, "Dynamic URLLC and eMBB Multiplexing Design in 5G New Radio," in IEEE CCNC, 2020.

[18] R. Abreu, J. Thomas, K. Pedersen, G. Berardinelli, K. Z. Istvan, and P. Mogensen, "Power control optimization for uplink grant-free URLLC," in IEEE WCNC, 2018.

[19] N. H. Mahmood, R. Abreu, R. Böhnke, M. Schubert, G. Berardinelli, and T. H. Jacobsen, "Uplink grant-free access solutions for urllc services in 5g new radio," in ISWCS, 2019.

[20] Z. Zhou, R. Ratasuk, N. Mangalvedhe, and A. Ghosh, "Resource Allocation for Uplink Grant-Free Ultra-Reliable and Low Latency Communications," in IEEE VTC Spring, 2018.

[21] V. Mancuso, P. Castagno, M. Sereno, and M. A. Marsan, "Slicing cell resources: The case of HTC and MTC coexistence," in IEEE INFOCOM, 2019.

[22] F. Dressler, F. Klingler, M. Segata, and R. Lo Cigno, "Cooperative Driving and the Tactile Internet," Proceedings of the IEEE, 2018.

[23] "Evolved Universal Terrestrial Radio Access (E-UTRA); Medium Access Control (MAC) protocol specification," 3GPP TS 36.321 V13.1.0, April 2016.

[24] N. H. Mahmood, R. Abreu, R. Böhnke, M. Schubert, G. Berardinelli, and T. H. Jacobsen, "Uplink Grant-Free Access Solutions for URLLC services in 5G New Radio," in ISWCS, 2019.

[25] A. Hordijk and N. Van Dijk, "Networks of queues with blocking," in Performance, Oct 1981.

[26] S. S. Lam, "Queuing Networks with Population Size Constraints," IBM Journal of Research and Development, vol. 21, pp. 370–378, 1977.

[27] W. Henderson, D. Lucic, and P. G. Taylor, "A net level performance analysis of stochastic Petri nets," Australian Mathematical Society, vol. 31, no. 2, pp. 176–187, Oct. 1989.