

Faster Together: The Case for the Integration of Critical MTC and HTC

Vincenzo Mancuso, Paolo Castagno, Matteo Sereno, and Marco Ajmone Marsan

CORRESPONDING AUTHOR: Paolo Castagno (e-mail: paolo.castagno@unito.it).

V. Mancuso (vincenzo.mancuso@ieee.org) is with University of Palermo, Italy, and IMDEA Networks Institute, Spain; P. Castagno (paolo.castagno@unito.it) and M. Sereno (matteo.serenio@unito.it) are with University of Turin, Italy; Marco Ajmone Marsan (marco.ajmone@imdea.org) is with IMDEA Networks, Spain; V. Mancuso and M. Sereno are also with CNIT, Italy.

This work has been supported by the the RESTART Program, financed by the Italian government with the resources of the Italian Recovery, Transformation and Resilience Plan – Mission 4, Component 2, Investment 1.3, theme 14 “Telecommunications of the future” (PE00000001 - program “RESTART”, CUP D93C22000910001). V. Mancuso and M. Ajmone Marsan have been supported by project TUCAN6-CM (TEC-2024/COM-460), funded by CM (ORDEN 5696/2024)

ABSTRACT We investigate a future cellular network scenario where many human and machine terminals respectively need broadband connectivity (eMBB— enhanced mobile broadband—service), and low latency and high reliability in data transfer (cMTC— critical machine type communications— service). Following an *unconventional* approach not yet supported by standards, we propose (a) sharing resources between eMBB and cMTC uplink services and (b) letting machine traffic use eMBB services when and where not otherwise needed by human traffic. We study the system by means of a novel analytical model that characterizes throughput, failure probability and latency distributions for uplink human and machine traffic types. We use simulations to validate our model. In particular, we unveil the impact of redundancy in cMTC transmissions, which makes it possible to serve, in a single cell, a few hundred machine terminals with high reliability (up to “six nines”), with very low latency (tens of ms) and with negligible impact on human terminal traffic. Our results show that our unconventional service architecture is able to self-adjust to traffic conditions and, while in principle it can be enhanced with load-based optimization in the allocation of resources, it could work near-optimally in a traffic-load-oblivious management framework.

INDEX TERMS Machine traffic; cMTC; 6G; RAN slicing; Performance evaluation.

I. Introduction

Supporting communication services with heterogeneous requirements in the same radio cell is challenging yet necessary. The challenge is exacerbated by the potential traffic mix composed by *human*- and *machine*-type communications (HTC and MTC, respectively). In this context, usage scenarios proposed for the sixth generation of cellular networks (6G) separately address high throughput mobile broadband services for HTC, and massive or critical operational contexts for MTC. The current approach leaves unaddressed the problem of supporting large volumes of machine applications and devices with low latency and high reliability requirements, in the presence of human traffic. This is critical, e.g., in smart factories and smart cities, or in intelligent transportation systems [1]. Target objectives are stringent and contrasting, in the order of tens of milliseconds and 99.999% reliability for critical MTC, whereas possibly coexisting HTC devices would require tens of Mb/s or more. Given a

deployed radio access network (RAN) with radio slices [2], [3], how to support this class of latency and reliability requirements for multiple slices is not yet clear; however, to address the challenge, new approaches to slice allocation and management are sprouting [4]. They are all based on the definition of 3GPP service categories, which is common to 5G and 6G technologies. For HTC, the target is to rely on the *enhanced mobile broadband* service (eMBB). For MTC, in case of critical communications, the target is to use the *critical MTC* service (cMTC) initially referred to as *ultra reliable and low latency communications* (URLLC) [5]–[8]. Nonetheless, existing approaches to cMTC are limited to support high reliability and low delay only for very few users, and, especially in the uplink, consume potentially large fractions of RAN resources [9].

Most of the available approaches to the analysis of cMTC/URLLC focus on downlink traffic and the possibility to *puncture* eMBB transmissions to make room for cMTC

as soon as critical traffic becomes available at the base station [10], [11]. These works are very useful but unfortunately cannot be extended to the uplink case since, differently from the downlink, cMTC transmissions cannot be controlled and strictly coordinated by the base station (BS, the gNB). Indeed, the resulting lack of coordination is potentially harmful and menaces to kill the efficiency of RANs, because uplink traffic is not negligible [12] and not necessarily sporadic, especially in some IoT applications [13]. Hence, network operators might have to face the following dilemma when designing uplink radio slices: shall we make room for a decent number of critical MTC devices or rather give priority to HTC interactive users?

In this article, we argue that RAN resources can be allocated in a way that supports large numbers of critical MTC flows without compromising uplink broadband access for HTC users, thereby enabling higher throughput and lower latency. We build on some key results and intuitions from our previous work [14], [15] to design a novel approach that can meet the requirements of many critical MTC users while being agnostic of MTC and HTC traffic loads. Our study is motivated by applications such as control of machine operations (typically automated or remote driving) where data traffic consists in uplink sequences of packet transmissions to report the machine and environment status and expect to obtain timely commands, with high reliability. In those applications, packet sequences start and end when a controlled machine moves in and out of a radio cell.

We start from the observation that using cMTC in uplink (i) might lead to collisions, including with scheduled traffic, but this effect can be negligible if cMTC traffic is limited; however, (ii) MTC might *also* receive dedicated physical resources, to avoid excessive collisions; (iii) we also argue that one might further consider the possibility of establishing uplink eMBB connections for machine devices, instead of concentrating all machine traffic on a cMTC slice. Hence, the problem is how to use physical resources: which part should go to cMTC only, which one exclusively to eMBB, and which one to both. Accordingly, in the article, we model the coexistence of HTC and critical MTC in uplink, to identify which class of allocations is more desirable.

Despite the intrinsic complexity of implementing MTC flows over multiple services, prior works have shown that those classes of approaches can produce substantial gains. In particular, our previous work [15], for the first time, showed the potentialities of using eMBB as primary transmission vector for MTC, and cMTC as backup that extends reliability and reduces latency constraints for a broad population of critical MTC devices; more recently, the authors of [8] have proposed to use cMTC for critical MTC traffic and eventually resort to eMBB for reliable retransmissions over resources allocated through *piggybacking* messages carried by the ACKs to cMTC transmissions. These approaches are actually not incompatible, as one might want to use the former scheme, based on eMBB, to increase the number of

customers in the system and boost the reliability of pure cMTC through the adoption of the latter scheme, for the portion of traffic that uses cMTC.

To compromise between performance and efficiency, we define a simple heuristic policy to use eMBB for MTC until latency and reliability constraints of the specific applications are at risk of being violated—which can occur due to the activity of machines as well as of humans that share the same BS—and then divert MTC to the more conventional cMTC approach. We use eMBB as a primary service for MTC, due to the assumption that uplink machine traffic is neither low nor sporadic. We further propose to use redundancy and coding in machine traffic that ends up using cMTC transmissions, protect part of those transmissions with dedicated resources, and enable the reuse of HTC resources for cMTC transmissions, although potentially causing collisions and triggering (limited) HTC retransmissions. We call this policy *CRITICAL Machine Service ON 3 Bandwidth parts* (CRIMSON-3B).

Our scheme and its analysis are validated through numerical simulations of 3GPP-compliant uplink access and transmission mechanisms, demonstrating that a cell could support many more critical MTC devices than with current approaches. The improvement over potential alternatives such as those outlined in 3GPP standards [16] is dramatic, up to two orders of magnitude higher in the number of supported devices, with negligible impact on HTC uplink performance.

A. Contribution

Summarizing, the article's contribution encompasses the following key aspects: (i) designing a new unconventional class of resource management schemes for 3GPP uplink; (ii) modeling the effects of coexistence of eMBB and non-sporadic cMTC across dedicated and shared physical uplink resources; (iii) presenting a comprehensive performance evaluation of uplink throughput and loss incurred by legacy and unconventional approaches to the coexistence of eMBB and cMTC with strict latency and reliability constraints; (iv) demonstrating that using both eMBB and cMTC is not only desirable but also necessary to support realistically high numbers of connected machine devices (tens of machines per MHz) with a few-millisecond latency and extremely high reliability (above 0.99999), in the presence of predominant HTC traffic (80% or more of uplink capacity); (v) delivering a model-based approach to numerically evaluate network slicing configuration alternatives with low complexity, with no need to resort to heavy simulations.

II. Background

3GPP specifies three main service categories to handle traffic: eMBB for broadband traffic, mMTC for massive machine traffic, and cMTC, formerly known as URLLC, for mission-critical machine-type traffic. Current standards and practices also specify how to allocate portions of physical

resources in a semi-static way, using the so-called *bandwidth parts*, i.e., contiguous sets of physical resource blocks with their numerology parameters, allocated to specific services or groups of users [17]. Currently adopted recommendations describe how to deploy one or multiple instances of each service category within the same radio coverage, meant to support coexisting services and providers [18]–[21]. In particular, 3GPP 5G New Radio (NR) specifications [22] identify two orthogonal paradigms for admission to uplink transmission resources: eMBB and mMTC use transmission grants assigned by the base station (i.e., the gNB in 5G and 6G) after a negotiation phase that lasts a few milliseconds; instead, cMTC uses a grant-free scheme, with either an inefficient semi-static allocation of transmission opportunities or immediate random access (a non-orthogonal multiple access scheme—NOMA [23]) with potential collisions between multiple uplink users [24]. Using NOMA has scarce efficiency because, to proactively manage collisions and channel impairments, it resorts to conservative transmission schemes, avoids sender-receiver negotiations, and does not necessarily use all available symbols in a resource block (it uses “minislots” instead); NOMA transmitters can even resort to send multiple copies of the same message, a basic form of forward error correction [4], [25].

III. Related Work

The integration of service categories like cMTC together with grant-based service categories for uplink traffic has garnered significant research attention due to the different sets of Quality of Service (QoS) requirements covered by these two types of service categories. To ensure clarity and generality, this section uses terms “grant-free” and “grant-based” to describe these traffic types, although other names can be found in the literature (e.g., URLLC instead of cMTC).

Efforts to address the coexistence of grant-free and grant-based traffic, or just the improvement of efficiency of grant-free access, have primarily focused on multiplexing in the context of network slicing, scheduling with retransmissions and replicas, and resource allocation strategies (e.g., see [26] for a recent survey of results investigating the coexistence between eMBB and URLLC/cMTC traffic). Below, we review key contributions, highlighting their strengths, limitations, and how they compare to our proposed resource management framework, which uniquely integrates grant-free and grant-based traffic.

a: Multiplexing and Slicing Approaches

Gerasin *et al.* [27] propose multiplexing schemes that optimize device transmission power to address static resource contention scenarios. However, their approach lacks mechanisms for dynamically reallocating MTC traffic between grant-free and grant-based services, as required in our framework.

Yang *et al.* [28] highlight innovations in 5G NR, including preemption indications and enhanced power control, to enable dynamic multiplexing of grant-free and grant-based traffic. These methods aim to ensure latency and reliability for grant-free traffic while maintaining cost efficiency. Nevertheless, their survey does not provide a comprehensive model for evaluating performance across varying conditions.

Doğan *et al.* [23] propose a NOMA scheme combining OFDM and index modulation (OFDM-IM) to mitigate collisions between grant-free and grant-based traffic. Their approach, leveraging fractional subcarrier activation, achieves 99.999% reliability within 1 ms and reduces latency. However, despite its promise, this method introduces significant protocol modifications, diverging from the non-invasive strategies preferred in our framework.

Feng *et al.* [29] propose a two-timescale Lyapunov-based RAN slicing algorithm to balance power consumption and service quality. Lee *et al.* [30] take a distributed approach to RAN slicing by dividing frequencies into shared and dedicated regions, optimizing resource allocation through game-theoretic models. Similarly, Alsenwi *et al.* [31] and Santos *et al.* [32] focus on radio slicing techniques to separate grant-free and grant-based traffic, respectively for downlink and uplink. They employ optimization-aided Deep Reinforcement Learning to maximize grant-based throughput while ensuring grant-free reliability. While these strategies improve coexistence by relying on slicing, unlike our proposal, they lack a mechanism to adapt RAN slices to dynamic traffic fluctuations, limiting their application in highly variable environments.

Bairagi *et al.* [33] and Interdonato *et al.* [34] explore puncturing mechanisms for downlink transmissions, controlled by the gNB via preemptive scheduling, to manage coexistence of eMBB and cMTC. They focus on minimizing disruptions to eMBB traffic. While effective, puncturing can be efficiently applied only in downlink, whereas in uplink puncturing turns into a mere erasure of eMBB signals at the gNB, when the gNB is able to detect the energy of unscheduled cMTC transmissions while receiving eMBB frames. In addition, puncturing/erasure often results in spectral inefficiency, both in downlink and uplink, a drawback mitigated in our opportunistic resource-sharing framework by means of an unconventional approach to the use of eMBB and cMTC service categories to support MTC applications jointly rather than individually.

b: Redundant Grant-Free and HARQ-Based Scheduling Schemes

Jacobsen *et al.* [35] analyze uplink grant-free transmissions, emphasizing reliability improvements through repetition schemes. They demonstrate that reactive HARQ strategies meet stringent latency targets, but, differently from our work, their work is limited to static scenarios and does not address dynamic interactions between grant-free and grant-based traffic.

Similarly, Liu *et al.* [36] introduce an analytical framework for contention-based grant-free access, evaluating HARQ schemes like K-repetition and Proactive. While their framework provides valuable insights, it does not consider dynamic transitions between traffic types.

Shashin *et al.* [37] propose adaptive algorithms for shared resource allocation, optimizing replicas and modulation schemes for grant-free traffic. Their results show improved capacity, yet the dual-use potential of grant-free and grant-based traffic for MTC remains unexplored in their work.

c: Hybrid Resource Allocation and Analytical Models

Kotaba *et al.* [38] investigate shared diversity resources to improve efficiency for uplink grant-free traffic. Their evaluation demonstrates near-reference reliability with reduced resource usage. Ghanem *et al.* [39] propose finite blocklength transmission for grant-free systems, incorporating partial time overlap to minimize delays. While effective, these frameworks do not dynamically adapt to traffic demands of grant-free and grant-based traffic.

With a different approach and purpose, Li *et al.* [40] propose energy-efficient mechanisms for grant-free traffic, incorporating frequency-hopping and proactive dropping to reduce power consumption. Despite significant reliability gains, their framework does not fully integrate hybrid traffic scenarios, limiting flexibility.

Mirri *et al.* [41] propose a multi-user detection scheme for grant-free machine users with massive and time critical applications, which can improve decoding performance substantially. Although that work does not address the coexistence of HTC and MTC traffic, and it considers workloads fundamentally different from what addressed in this work, their proposed scheme could be incorporated into our approach to further enhance performance.

d: Combined Use of Grant-Based and Grant-Free Access

A paper that is worth mentioning, because, like us, it considers critical and massive MTC, is [8]. Similarly to what we do, in that paper the authors propose the exploitation of both grant-based and grant-free access. However, contrary to what we do, they do not consider the coexistence of multiple traffic types (i.e., MTC and HTC) and suggest to first try a (coded) grant-free access for critical MTC, and in case this is not successful, to resort to grant-based access (or to a second grant-free round). Transmissions succeeding with grant-free access are able to meet a tighter latency deadline, but suffer a lower reliability. On the contrary, transmissions using grant-based access achieve higher reliability but a looser latency deadline, and require modifications in message formats and ACK schemes to avoid passing through RACH channels. A comparison of the performance of the two schemes is not necessary, because the proposal of [8] describes mechanisms to improve the reliability of cMTC/URLLC and could be nested in our proposal, resulting in further improvements, at the expense of added complexity. Moreover, the proposal

TABLE 1: Comparison to SoTA proposals that can be used in uplink and key contributions in this work

Work	Service selection	eMBB and cMTC	HTC and MTC	Latency-critical	Reliability-critical	Large MTC populations	Modeling	Resource use efficiency	Diverse scenarios
Mirri <i>et al.</i> [8]	✓			✓	✓		✓		
Gerasin <i>et al.</i> [27]		✓		✓				✓	
Feng <i>et al.</i> [29]		✓		✓			✓	✓	
Santos <i>et al.</i> [32]		✓		✓	✓		✓	✓	
Jacobsen <i>et al.</i> [35]							✓		
Liu <i>et al.</i> [36]				✓		✓	✓		
Kotaba <i>et al.</i> [38]							✓		
Ghanem <i>et al.</i> [39]				✓			✓		
Li <i>et al.</i> [40]				✓	✓	✓	✓		
Mirri <i>et al.</i> [41]				✓	✓	✓	✓	✓	
Zhou <i>et al.</i> [42]				✓	✓		✓		
Mahmood <i>et al.</i> [43]								✓	
CRIMSON-3B (our)	✓	✓	✓	✓	✓	✓	✓	✓	✓

of [8] is specifically designed for sporadic traffic and evaluated in such context, while ours is more broadly designed for (short) MTC packet streams, and evaluated with this traffic load (which might result from automated/remote driving or platooning applications). At a macroscopic level we can however observe that both approaches result in the possibility of providing latency and reliability guarantees to hundreds of machine terminals within a cell. This is a clear indication of the fact that mixing grant-based and grant-free access for critical MTC traffic can offer interesting advantages over classical approaches.

e: Comparisons and Key Contributions of This Study

Most existing studies in the literature do not apply to the uplink case. They do not consider the coexistence of HTC and MTC workloads either, although some of them consider IoT traffic and the coexistence of eMBB and cMTC. With the exception of [8], available works do not provide an analytical framework that evaluates both grant-free and grant-based performance while dynamically selecting the service type for MTC based on deadlines and constraints [28], [42].

As summarized in Table 1, which only includes studies that can be applied to the uplink case, our study addresses this gap by proposing:

- **Dynamic service selection:** Unlike static models like [27], [29], [32], our framework dynamically transitions critical and non-sporadic MTC traffic from grant-free to grant-based services, prioritizing grant-based until constraints necessitate grant-free use, while other

works exist which consider sporadic MTC transmissions and give priority to grant-free access [8].

- **Integration of HTC and MTC:** Existing frameworks only focus on separated traffic categories, possibly in the IoT context. Instead, our proposal explicitly integrates HTC and MTC traffic over shared radio resources, ensuring efficient coexistence.
- **Latency and reliability constraints:** While most of available works consider the presence of latency constraints [8], [27], [29], [32], [35], [38], [40]–[43], and only a few consider as well reliability [8], [32], [41]–[43], very few works also propose solutions for large machine populations [40], [41], although we go beyond and can guarantee hard latency bounds and high reliability—according to application’s requirements—for truly massive numbers of machine devices.
- **Analytical and Simulative Modeling:** Our framework generalizes queuing models to capture complex blocking phenomena between grant-free and grant-based traffic, extending beyond classical models [39], [40], while other works in Table 1 mostly rely on physical transmission models.
- **Resource Efficiency:** By dynamically sharing resources, our scheme avoids under-utilization observed in static slicing approaches [27], [29] while being able to cope with six-nines reliability requirements for uplink grant-free traffic, which had been achieved in the past only for sporadic traffic patterns [32], [43].
- **Diverse Scenario Validation:** Analytical and simulative results in smart highway and dense urban environments demonstrate the robustness of our framework, outperforming legacy approaches and proving the superiority of our paradigm for very different operational scenarios, while other works typically focus on simple and not diversified evaluation scenarios.

IV. CRIMSON-3B: making HTC and critical MTC faster together

We consider a BS that serves HTC broadband devices; it also provides connectivity to machine devices monitoring and controlling a critical cyber-physical system (e.g., a swarm of robots or UAVs, or vehicles that need driving assistance). HTC broadband devices clearly require eMBB, whereas machines involved in control systems have latency and reliability requirements, which can only be provided by under-loaded eMBB slices or cMTC. Indeed, it is important to remark that eMBB cannot provide strict guarantees under medium-high loads and cMTC can only serve small numbers of terminals. Notice also that other service types like mMTC can be effective for mission-critical services [44] but are totally unsuitable for time-critical traffic [45].

For HTC traffic, we therefore consider only eMBB, whereas for MTC we propose a mix of cMTC and eMBB. This is an unconventional approach to the use of service categories in 3GPP networks, and we are aware of the

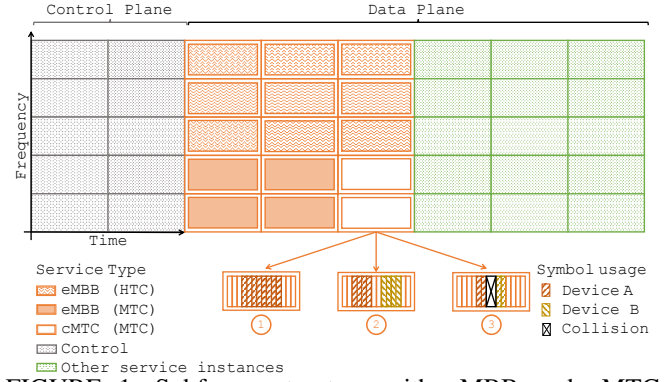


FIGURE 1: Subframe structure with eMBB and cMTC allocations. cMTC minislots can have success in a slot (1) even when multiple devices use the same slot (2), or collide (3).

technical difficulties that could arise once one might want to deploy our proposal. Still, our study is meant to demonstrate the technical convenience of mixing cMTC and eMBB for the sake of coexistence of HTC and MTC, making them scale in terms of devices that can be served by the same BS, therefore making HTC and critical MTC perform faster together, both in terms of throughput and in terms of latency.

In a nutshell, our proposal for critical MTC handling works as follows. Critical MTC packet flows, for which latency and reliability constraints are defined in application’s requirements, are initially considered for grant-based transmission on a dedicated eMBB bandwidth part. This first attempt is typically successful in light and medium system load, so that the latency deadline can easily be satisfied. When the system traffic is high, the grant allocation may be slow, so that packet delays risk exceeding the prescribed maximum latency. When this happens, the MTC packet flow is moved to cMTC, and uses both the cMTC bandwidth part as well as some resources of the HTC eMBB bandwidth part, that are stolen from HTC to transmit packet replicas, to meet the reliability requirements of MTC flows also when diverted to cMTC.

We call our proposal CRIMSON-3B because it splits critical MTC traffic over a BS bandwidth part dedicated to cMTC, plus two additional bandwidth parts allocated to eMBB services, one for MTC only, and the other for HTC, with MTC stealing resources without control and with a random pattern.

A. cMTC as a fast backup for MTC cell access

Fig. 1 depicts an example of generic 3GPP frame structure, with data and control planes. In the example, a portion of resources is allocated to HTC over eMBB. Another portion of resources, disjoint from the previous, is allocated to MTC over eMBB. Eventually, MTC traffic that uses cMTC with NOMA, is transmitted over a third disjoint set of resources. However, cMTC specifications do not mandate the use of

disjoint resource sets for them, as they might reuse resources allocated to other service instances [46].

In general, as numerically shown later in the paper, machine devices could use eMBB as far as the load of the service is low, and cMTC could be used to skip lengthy grant-based procedures when latency-critical packets approach their deadline, as proposed in [15]. Which resources cMTC should use is an open problem. A simple approach consists in dedicating a pool of resources to cMTC, as in Fig. 1. Another consists in letting cMTC transmissions reuse physical resources allocated to other service instances. Our claim is that a hybrid approach is desirable, but it requires the deployment of some workarounds and protocol extensions in order to work robustly. Besides, we will explore the performance of different options that are either aligned with existing standard or, as proposed in the literature, require continuous re-tuning of system parameters.

Specifically, by following the intuition of [15] we propose to use cMTC for MTC only when eMBB is not viable because of failures in accessing the BS via the random access channel (RACH). Thus, in our unconventional proposal, critical traffic belonging to MTC would resort to use cMTC as an overflow of traffic from another service instance using eMBB (cf. Fig. 2). In other words, a MTC user that needs to access a high reliability delay-critical service, first attempts to establish a connection using eMBB resources. If resources are not granted in a timely fashion, the user resorts to cMTC.

In addition, we study the impact of minislot configurations in NOMA transmissions for 3GPP, and explore pros and cons of introducing redundancy in NOMA transmissions at two stages: by means of transmitting replicas (as suggested, e.g., in [37]) and by coding with redundancy MTC messages over multiple cMTC transmissions.

B. Uplink packet transmission modalities

As shown in Fig. 2, HTC devices offer a traffic of λ_h arrivals per second (arr/s), out of which only a portion σ_h is admitted into service, using eMBB. HTC requests which do not obtain a transmission grant are simply lost, thus $\sigma_h \leq \lambda_h$ and the fraction of lost HTC traffic is $1 - \sigma_h/\lambda_h$.

For MTC traffic, which we assume to be critical in terms of both latency and reliability, the offered load is λ_m , and such quantity is entirely served by either eMBB (σ_{me} arr/s will use transmission grants) or cMTC (σ_{mc} arr/s will use NOMA to transmit). In this case, $\sigma_{me} + \sigma_{mc} = \lambda_m$.

A network processor handles the two eMBB instances at the BS. It maps uplink physical transmission resources onto service instances and handles *service positions* for established HTC or MTC connections, i.e., positions reserved for grant-based traffic. However, the number of concurrently active connections is limited because they require scheduling and handling state information.

Instead, cMTC traffic uses NOMA over the set of resources assigned to a cMTC service instance and does not enforce any device coordination for scheduling transmis-

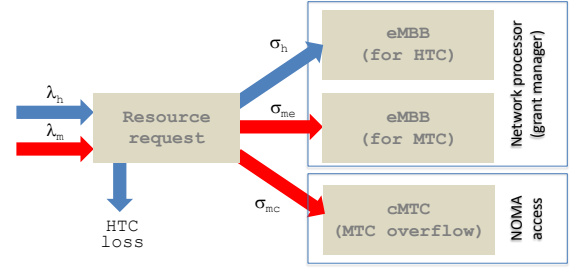


FIGURE 2: Overview of a RAN slice in our proposal for HTC and MTC

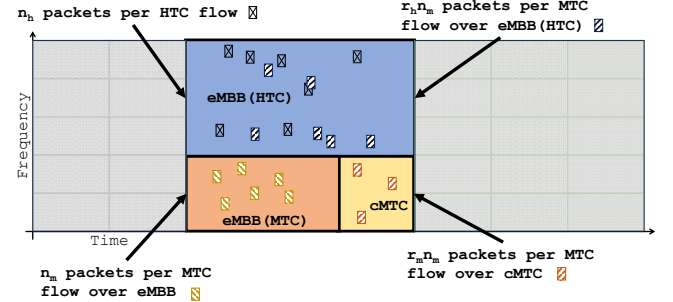


FIGURE 3: Slice allocation for HTC and MTC, and packet distribution over resources in our proposal

sions. According to common guidelines for latency-critical NOMA transmissions, we assume that the cMTC traffic sent by a machine device is transmitted using a random set of minislots, which only wait for the next subframe to start. Also, we assume that no collisions will occur between transmissions originated by the same device, while distinct devices can collide because they do not coordinate (cf. Fig. 1).

Fig. 3 reports an example of resource allocation between slices and packet scheduling in HTC and MTC according to our proposal. As reported in the figure, it is possible to observe collisions between cMTC and eMBB uplink transmissions when physical resources are allocated to service instances in a non-orthogonal way. This situation is possible, according to 3GPP recommendations, only for cMTC reusing eMBB resources. Thus, we assume that cMTC can transmit over two portions of physical resources: one portion that is exclusively dedicated to the cMTC service instance, plus the portion of resources allocated to the instance of eMBB that serves HTC. Notice that it would not make sense to let cMTC transmit also over the resources of eMBB associated with MTC.

In case of collisions between cMTC minislots, we consider that all involved transmissions are lost even if minislots only partially overlap. Instead, in case of collision between a minislot of cMTC and a slot of HTC, we consider that the HTC transmission is lost for sure, while the cMTC transmission can still be decoded with probability $1 - \alpha$ (i.e., we assume that cMTC is transmitted with higher power

than eMBB and the gNB can detect cMTC energy and erase eMBB signals, as suggested in [19]). We do so because cMTC transmissions, being critical, are typically set to use higher power than eMBB transmissions.

To further protect MTC traffic, we adopt the proposal of [37], [43], which conforms to the specifications for configured grant uplink access [24], to send replicas of cMTC packets in the same subframe. We propose to map cMTC requests onto r_m replicas over a cMTC bandwidth part, plus r_h replicas over the bandwidth part allocated to the eMBB service assigned to HTC, with possible collisions and interference. It is enough that at least one of the $r_m + r_h$ replicas goes through and is correctly decoded at the BS to count for a cMTC success.

Here, a packet might not fit a single minislot, so that the success of a replica depends on the reception of multiple minislots. To improve robustness, we therefore introduce further redundancy within each replica, exploiting for example fountain codes [47]. By so doing, in the case of redundancy $R > 0$, a cMTC transmission of length k symbols is actually transmitted in $(1 + R)k$ symbols, mapped onto $(1 + R)k/\gamma$ minislots, where γ is the number of symbols per minislot adopted. Thus, if enough minislots are decoded, such that k of the coded symbols are decoded, the replica is successfully decoded.

C. Uplink Traffic Flows

Users generate flows with finite duration. Each uplink access request carries a random sequence of transmissions to the BS, alternating with *think time* intervals. After each transmission, p_h (resp. p_m) indicates the probability that an HTC (resp. MTC) flow has more data to send. This means that, on average HTC flows produce $n_h = 1/(1 - p_h)$ packets and MTC flows produce $n_m = 1/(1 - p_m)$ packets.

This traffic pattern is characteristic of cyber-physical applications. For example, vehicles equipped with automated or assisted driving systems commonly generate periodic transmissions of vehicle-status updates to the application controller. The controller, in turn, issues updated directives and control commands, which must be delivered reliably and within strict timeliness constraints.

D. Slicing

The described system and flows can be naturally mapped onto not necessarily standard network slices. Thus, HTC over eMBB as well as MTC over eMBB can be easily seen as standard 3GPP slice instances. MTC over cMTC would also be a standard 3GPP slice, although using a mix of resources for NOMA transmissions—some of which dedicated and others allocated to HTC—is an *unconventional* approach not yet taken into account in existing 3GPP recommendations on slicing. However, the approach is not forbidden by recommendations, which leave the operator free to assign or reassign bandwidth parts to different services. Likewise, although using multiple service instances for the same slice

is possible in principle, that solution is not accounted for in recommendations, and therefore using eMBB as primary service instance for critical MTC traffic and cMTC as backup is also far from a conventional interpretation of existing recommendations and existing deployments.

In practice, our proposal does not require changing the way BSs announce slice resources, nor changing any control or data plane protocol, and its implementation would be even simpler than other multi-service approaches proposed to serve a class of traffic through modified data and control frames (e.g., [8]). Instead, our proposal requires minimal configuration and software adjustments, i.e.: (i) forcing the reuse of physical resources over multiple slices, which surely has pros and cons, but was formally viable in 3GPP already since 5G specifications, (ii) instructing cMTC on the number of replicas to send over two bands, which is supported by 3GPP specifications [48], and (iii) enforcing the possibility of moving a service request (and hence a flow) from a service instance to another, which is already possible because 3GPP devices are allowed to connect to multiple slices at the same time, and would require only a software update at terminal level.

V. Analysis of CRIMSON-3B operation

We analyze the operation of CRIMSON-3B, whose high-level abstraction is represented in Fig. 2. We focus on throughput, latency and number of supported devices for HTC and critical MTC, with the notation of Table 2.

The system described in the previous section includes two traffic classes, three service instances over two service categories, a network processor with two coupled queues, and a subset of users (machine devices) transmitting with NOMA and generating interference to another subset of users (humans). The analysis of the system is cumbersome because of the intertwined structure of subsystems within the cell. In particular, the portion of MTC traffic that ends up using cMTC depends on RACH performance and on blocking at the network processor, the latter being a function of the speed at which network processor queues can be served, which, in turn, is affected by the interference caused by cMTC transmissions. In addition, the necessary level of redundancy enforced in cMTC is tied to the level of activity of the network processor and the availability of resources, which in turn depends on the volume of cMTC transmissions and the interference that they cause to HTC device, hence the number of retransmissions caused to HTC.

To analyze system performance, in what follows we rely on previously published results for some components of the analysis, in particular on [14] for what concerns the computation of RACH performance, and on [15], [49] for the computation of blocking probability and latency for the coupled queues at the network processors, serving HTC and MTC flows with transmission grants. This article derives how to analyze what feeds network processor and NOMA

TABLE 2: Notation

Description	Notation	Unit
Subframe duration	τ	s
Arrival rate	λ_h, λ_m	requests/s
Packet length	L_h, L_m	bits
Average number of packets per access request	n_h, n_m	-
Spectral efficiency of the modulation and coding scheme	η_h, η_m	b/symbol
Symbols per slot (minislot length)	$\gamma, \gamma_{\max}$	symbols
Redundancy of coding for cMTC replicas (redundancy bits per data bit)	R	-
Number of cMTC replicas	r_h, r_m	-
Number of slots to transmit a packet	k_h, k_{me}, k_{mc}	slots
Number of slots per subframe	$\ell_h, \ell_{me}, \ell_{mc}$	slots
Probability of success on the RACH	s_h, s_m	-
Service positions available	F, F_h, F_m	-
Average number of slots used by cMTC per subframe	b_h, b_m	slots
Average number of slots used by HTC, counting retransmissions	t	slots
Average number of minislots that do not collide in a slot when j are present	$T(j)$	minislots
Distribution of minislots accessing the same slot when j cMTC transmissions access a subframe	$Q_h(\kappa j), Q_m(\kappa j)$	-
cMTC success probability per slot	$S_{mc}^{(\text{slot},h)}, S_{mc}^{(\text{slot},m)}$	-
cMTC success probability per replica	$S_{mc}^{(h)}, S_{mc}^{(m)}$	-
cMTC success probability per packet	S_{mc}	-
Rate of served flows	$\sigma_h, \sigma_{me}, \sigma_{mc}$	requests/s
Throughput	$\xi_h, \xi_m, \xi_{me}, \xi_{mc}$	b/s

processes in the described system, and how HTC and MTC traffics interplay, as a function of several service parameters.

Next, we characterize the traffic over the different service categories, derive expressions for the utilization of RAN resources, and for the performance of HTC and critical MTC flows.

A. Traffic characterization

The rate of flow access requests is λ_h for HTC traffic and λ_m for critical MTC. However, the RACH operation determines the amount of grant-based traffic that actually reaches the network processor. We denote by s_h and s_m the success probabilities of access request at the RACH, obtained through the analysis of [14], under the offered load conditions of our system. Hence, traffic intensities for

requests at the network processor or using NOMA are: $s_h \lambda_h$ for HTC, $s_m \lambda_m$ for MTC served by eMBB and $(1 - s_m) \lambda_m$ for MTC served by means of cMTC. However, the network processor can block requests, due to lack of BS resources to handle the scheduling of transmission grants. Hence, only $\sigma_h \leq s_h \lambda_h$ is actually admitted into service for HTC, and the rest is lost. Likewise, part of the MTC requests $s_m \lambda_m$ will be blocked, but we assume that they will be redirected to cMTC, to safeguard the critical applications behind them. As already mentioned, we denote by σ_{me} and σ_{mc} the number of MTC requests per second that will access the network processor or will use cMTC, respectively.

Computing the values of σ_h , σ_{me} and σ_{mc} requires the analysis of the behavior of the network processor under the interference caused by NOMA transmissions. We therefore move to compute the utilization of resources due to NOMA before going back to the network processor.

B. Resource utilization

The BS implements a RAN with three bandwidth parts. ℓ_h slots per subframe are allocated to HTC flows, whereas ℓ_{me} and ℓ_{mc} slots are allocated to MTC over eMBB and MTC over cMTC, respectively. Since in CRIMSON-3B cMTC can also use NOMA over the HTC part, the effective capacity of HTC used by HTC in the network processor decreases proportionally to NOMA transmissions, occupying $b_h \leq \ell_h$ slots, on average. Hence, HTC is left with $\ell_h - b_h$ slots per subframe, on average.

To compute b_h , we follow the approach of [15], except here we account for cMTC coding and for the fact that cMTC transmissions only use minislots instead of entire slots. Specifically, in the presence of a fountain code with redundancy R , a transmission is decoded if a fraction $\frac{1}{1+R}$ of its symbols are not collided. One cMTC replica with redundancy R therefore uses k_{mc} slots:

$$k_{mc} = \left\lceil \frac{(1+R)L_m}{\eta_m \gamma} \right\rceil,$$

where L_m is the packet size in bits, η_m is the transmission efficiency in bits/symbol and γ is the number of symbols used in a minislot by cMTC transmissions.

With the above-defined quantities, the results of [15] still formally apply, as resumed in the following proposition (whose proof, with slightly different notation, can be found in [15]).

Theorem 1:

The average numbers of slots picked by cMTC with flows of n_m packets requiring k_{mc} minislots each and arriving at rate σ_{mc} are:

$$b_h = \ell_h \left(1 - e^{-n_m \sigma_{mc} \tau k_{mc} r_h / \ell_h} \right), \text{ shared with HTC,} \quad (1)$$

$$b_m = \ell_{mc} \left(1 - e^{-n_m \sigma_{mc} \tau k_{mc} r_m / \ell_{mc}} \right), \text{ dedicated to cMTC;} \quad (2)$$

where τ is the subframe duration.

Notice that, in a subframe, the ensemble of cMTC transmissions uses $n_m \sigma_{mc} \tau k_{mc}$ minislots per replica, on average.

Resource utilization due to HTC transmissions depends on collisions with cMTC, because HTC is not time-critical and can use HARQ (hybrid ARQ) techniques and retransmit undecoded packets. Now, since HTC attempts to transmit $\sigma_h n_h k_h \tau$ slots per subframe and retransmits upon collisions, and since the collision probability with cMTC is b_h / ℓ_h , HTC accesses $t \geq \sigma_h n_h k_h \tau$ slots per subframe, such that

$$\left(1 - \frac{b_h}{\ell_h}\right) \cdot t = \sigma_h n_h k_h \tau. \quad (3)$$

The above result can be summarized as follows.

Theorem 2:

The effective utilization of HTC is

$$t = \ell_h \frac{\sigma_h n_h k_h \tau}{\ell_h - b_h}, \quad (4)$$

and the probability that HTC uses any slot (accessed or not by cMTC) is t / ℓ_h .

Eventually, MTC traffic over eMBB is not affected by cMTC collisions and therefore the number of slots used in each subframe by MTC over dedicated eMBB resources is simply $\sigma_{me} n_m k_{me} \tau$, where $k_{me} = \left\lceil \frac{L_m}{\eta_m \gamma_{\max}} \right\rceil$.

C. Performance analysis: Grant-based services

A BS can only accommodate up to F flows at a time, serving them with transmission grants. Additional requests are dropped. As in [15], we assume that F_h positions of the network processor are reserved to HTC, F_m are reserved to critical MTC, and $F - F_h - F_m$ positions are allocated on a “first come, first served” policy. Our model is general and also allows, for instance, for the evaluations of classic network processors with a common pool of resources for all service instances (when $F_h = F_m = 0$) or with fully separated resources (when $F_h + F_m = F$).

Apart for service positions, the network processor has available service resources for its two queues. The MTC queue has exactly ℓ_{me} slots per subframe, fitting $\eta_m \gamma_{\max}$ bits each, where $\gamma_{\max}$ indicates how many symbols a slot can carry. Instead, although there are ℓ_h slots per subframe in the resource region allocated to HTC, only $\ell_h - b_h$ are *useful* slots, on average, the rest being occupied by cMTC. For the analysis of the coupled queues at the network processor, we use the normalization algorithm of [49] using as input parameters (i) the average HTC arrival rate after RACH, $s_h \lambda_h$; (ii) the average HTC service rate $\eta_h \gamma_{\max} (\ell_h - b_h) / \tau$, computed with a spectral efficiency η_h that is typically greater than that of machine devices, η_m ; (iii) the average MTC arrival rate after RACH, $s_m \lambda_m$ and (iv) the corresponding service rate $\eta_m \gamma_{\max} \ell_{me} / \tau$; and (v) network processor parameters F , F_h , and F_m .

What we propose is a numeric approach to feed the general algorithm defined in [49]. Running the algorithm

yields the blocking probability and the throughput of the two coupled queues, one for HTC (lost requests) and one for MTC (in which case, flows are redirected to cMTC). We have indicated with σ_h the flow-level throughput of HTC, i.e., the number of HTC requests per second that are not blocked, which corresponds to a throughput, expressed in bits per second, equal to

$$\xi_h = \sigma_h n_h L_h, \quad (5)$$

if L_h is the size of HTC packets.

For MTC, σ_{me} indicates the flow-level rate over eMBB, which corresponds to a throughput (in bit per second) equal to:

$$\xi_{me} = \sigma_{me} n_m L_m. \quad (6)$$

Here, of course, we are tacitly assuming that the network processor is stable and does not introduce any loss, e.g., it does not drop any packets and all packets of accepted flows are eventually served.

We remark that relying on averages as we do here carries some approximations, which might happen to be critical especially for the case of cMTC resource consumption of HTC resources, impacting the service time of HTC packets at the network processor. However, we will validate the model and assess the importance of our approximations through realistic simulations in the performance evaluation section.

D. Performance analysis: NOMA-based service

For cMTC, we need to evaluate the success or failure of minislots, combined in packet replicas, and evaluate the success of at least one replica to be able to analyze the successes and throughput at the decoder. We indicate with S_{mc} and ξ_{mc} respectively the packet success probability (i.e., the success of at least one replica) and the throughput (in bit per second, discounting redundancy).

To compute S_{mc} and ξ_{mc} , let's start from the consideration that, with RACH loss and blocking of MTC at the network processor, the load of cMTC can be quantified as

$$\sigma_{mc} = (1 - s_m) \lambda_m + (s_m \lambda_m - \sigma_{me}). \quad (7)$$

Based on the above quantity, and assuming that the number of machine devices contributing to σ_{mc} is large enough (e.g., at least a dozen), we can model the probability that j cMTC transmissions occur in a subframe through a Poisson distribution $P_m(j)$, with average equal to $\sigma_{mc} n_m \tau$.

Therefore, if $S_{mc}^{(h)}(j)$ and $S_{mc}^{(m)}(j)$ indicate the probabilities of success of independently transmitted replicas when j transmissions are ongoing, with replicas sent respectively over HTC or dedicated cMTC resources, we can use the total probability formula for S_{mc} and ξ_{mc} :

$$S_{mc} = 1 - \sum_{j=1}^{\infty} \frac{P_m(j)}{1 - P_m(0)} \left(1 - S_{mc}^{(h)}(j)\right)^{r_h} \left(1 - S_{mc}^{(m)}(j)\right)^{r_m}, \quad (8)$$

$$\xi_{mc} = \frac{L_m}{\tau} \sum_{j=1}^{\infty} j P_m(j) \left(1 - \left(1 - S_{mc}^{(h)}(j)\right)^{r_h} \left(1 - S_{mc}^{(m)}(j)\right)^{r_m}\right). \quad (9)$$

The first expression is derived as the complement of the probability that no replica is successfully decoded, given that at least one transmission occurs, hence the term $1 - P_m(0)$ at the denominator of (8). Besides, each packet of length L_m (before adding redundancy and replicas) sent in an interval τ contributes to throughput proportionally to the success probability of at least one of its replicas, conditional to the transmission of j packets in τ , hence the expression in (9).

We still need to evaluate the per-replica conditional success probabilities $S_{mc}^{(h)}(j)$ and $S_{mc}^{(m)}(j)$. Now, with fountain codes, a replica is successful if at least $\lceil \frac{L_m}{\eta_m \gamma} \rceil \leq k_{mc}$ of its slots are successful. If j independent transmissions are ongoing in the subframe, using minislots chosen uniformly at random, the replica success probability over the HTC and cMTC bandwidth parts are simply expressed as the complementary cumulative distribution of slot successes out of k_{mc} trials and individual success probability $S_{mc}^{(\text{slot}, x)}(j)$, for $x \in \{h, m\}$, the latter expressing the per-slot conditional success probabilities of cMTC transmissions. This is a binomial distribution, hence we have:

$$S_{mc}^{(x)}(j) = \sum_{\kappa = \lceil \frac{L_m}{\eta_m \gamma} \rceil}^{k_{mc}} \binom{k_{mc}}{\kappa} \left(S_{mc}^{(\text{slot}, x)}(j) \right)^\kappa \left(1 - S_{mc}^{(\text{slot}, x)}(j) \right)^{k_{mc} - \kappa}. \quad (10)$$

Therefore, the quest for the expression of the cMTC packet success probability turns into deriving the success probability in each single slot over the bandwidth part shared with HTC, $S_{mc}^{(\text{slot}, h)}$, and over the bandwidth part dedicated to cMTC, $S_{mc}^{(\text{slot}, m)}$. The following theorem shows how to compute such probabilities.

Theorem 3:

If $T(\kappa)$ is the average number of cMTC minislots that fit into a slot without colliding with other cMTC minislots, when κ transmissions use that slot,¹ $Q_x(\kappa|j)$, $x \in \{h, m\}$, are the probabilities that $\kappa \leq j$ cMTC minislots use the same slot when j transmissions are active in a subframe over either HTC or dedicated cMTC resources, and α is the probability that a cMTC symbol is lost after a collision with an HTC symbol, the average rates of successful minislots over HTC and cMTC dedicated resources are:

$$S_{mc}^{(\text{slot}, h)}(j) = \frac{\ell_h}{j r_h k_{mc}} \sum_{\kappa=1}^j Q_h(\kappa|j) T(\kappa) \left(1 - \alpha \frac{\sigma_h n_h k_h \tau}{\ell_h - b_h} \right); \quad (11)$$

$$S_{mc}^{(\text{slot}, m)}(j) = \frac{\ell_{mc}}{j r_m k_{mc}} \sum_{j=1}^{\kappa} Q_m(\kappa|j) T(\kappa), \quad (12)$$

¹ $T(\kappa)$ is combinatorial, considering that each of the κ active transmissions sends γ adjacent symbols chosen at random within the slot.

where

$$Q_h(\kappa|j) = \binom{j}{\kappa} \left(\frac{r_h k_{mc}}{\ell_h} \right)^\kappa \left(1 - \frac{r_h k_{mc}}{\ell_h} \right)^{j-\kappa}, \quad (13)$$

$$Q_m(\kappa|j) = \binom{j}{\kappa} \left(\frac{r_m k_{mc}}{\ell_{mc}} \right)^\kappa \left(1 - \frac{r_m k_{mc}}{\ell_{mc}} \right)^{j-\kappa}. \quad (14)$$

Proof:

The probability that a cMTC packet uses any available slot is $\frac{r_h k_{mc}}{\ell_h}$ or $\frac{r_m k_{mc}}{\ell_{mc}}$, and since slots are chosen uniformly at random by j independent agents, a binomial random variable describes how many cMTC transmissions will use a given slot, which leads to (13) and (14).

The sum of products of the form $Q_x(\kappa|j) T(\kappa)$, $x \in \{h, m\}$, is the average number of minislots that successfully fit into a slot when j cMTC transmissions occur in a subframe. Thus, the average number of minislots that do not collide with other minislots is obtained by further multiplying by the total number of slots (either ℓ_h or ℓ_{mc}). To obtain (11) and (12), we next need to normalize with respect to the total number of transmitted minislots (either $j r_h k_{mc}$ or $j r_m k_{mc}$). This leads to (12) for cMTC transmissions that use dedicated resources. Instead, in the case of cMTC transmissions that use HTC resources, we have to discount the collisions between cMTC and HTC with a factor α . To do so, we need to consider that the probability that HTC occupies any slot accessed by cMTC minislots can be computed from (4) and is equal to $\frac{t}{\ell_h} = \frac{\sigma_h n_h k_h \tau}{\ell_h - b_h}$. Therefore, $\alpha \cdot \frac{\sigma_h n_h k_h \tau}{\ell_h - b_h}$ is the joint probability that any slot accessed by cMTC is also accessed by HTC and the collision results in a loss also for cMTC. One minus such a quantity is the probability that HTC does not impair cMTC minislots, which therefore needs to be included in (11). This concludes the proof. ■

Since flows generated by machine devices either go to eMBB or cMTC, MTC achieves a total throughput given by

$$\xi_m = \xi_{me} + \xi_{mc}. \quad (15)$$

E. Computation flow

Carrying on the analysis described in this section requires going through the following sequence of steps:

- 1) For the chosen system configuration and offered traffic, compute the RACH performance with [14], to obtain the minimum load of network processor ($s_h \lambda_h$ for the HTC queue and $s_m \lambda_m$ for the MTC queue) and NOMA transmissions by accounting for RACH losses only, i.e., $(1 - s_m) \lambda_m$;
- 2) Compute the resource utilization of cMTC over HTC, b_h , using (1) with an initial estimate of σ_{mc} equal to $(1 - s_m) \lambda_m$;
- 3) With the output of RACH and the initial value computed for b_h , evaluate blocking at the network processor numerically with [49], to update the MTC overflow towards cMTC, using (7), and update b_h accordingly;

- 4) Iterate on the numerical evaluation of network processor and cMTC load until convergence, using sequentially the algorithm in [49] and then equations (7) and (1);
- 5) Compute throughput ξ_h and ξ_{me} (in bits per second) of HTC and MTC over dedicated eMBB from the output of the network processor in the last iteration;
- 6) Compute the throughput (and packet success probability) of MTC as described in Section D, starting from (13)-(14) to be plugged in (11)-(12), whose values must be replaced in (10), which in turn must be used in (8) and (9) jointly with the Poisson distribution $P_m(j)$ computed on the average value $\sigma_{mc}n_m\tau$, where σ_{mc} was computed with (7) in the last iteration.
- 7) Evaluate (15) to obtain the cumulative MTC throughput.

Theorem 4:

The convergence of iterations at step 4) is guaranteed.

Proof:

Iterations converge because they undergo a negative feedback loop. Specifically, increases of b_h imply a higher overflow of MTC from eMBB to cMTC, hence free resources at network processor, resulting in lower blocking and less MTC traffic overflow, which eventually turns into a reduction of b_h . Similarly, it is easy to see that the decrease of b_h leads to higher MTC utilization levels on eMBB resources, which turns into less overflow and hence causes a reduction on b_h . ■

The validation of the model and its assumptions described so far for HTC and MTC operations, jointly with the performance evaluation of our proposal as compared to alternative schemes is proposed next.

VI. Validation and Performance evaluation

In this section we report numerical tests and realistic simulations that we conducted to validate the model and assess the performance of CRIMSON-3B and the sensitivity to its configuration parameters.

We use two reference scenarios: a *smart highway* with 5G gNBs [50] spaced every 2 km, and a *dense urban* scenario [51].

In the *smart highway*, vehicles move at 120 km/h, with average time spent in a cell equal to one minute. HTC serves infotainment used by vehicle passengers. Connection requests arrive uniformly at random in time and last 60 seconds, on average, with packet size $L_h=100$ kb and think-time equal to either 20 or 100 ms, hence generating between 1 and 5 Mb/s per vehicle.

Critical MTC traffic carries data necessary for autonomous driving (e.g., vehicle position, speed, acceleration, etc.). It sends 1-kb packets quasi-periodically, once every about 60 ms. Hence, one vehicle is responsible for up to 16 kb/s of critical MTC traffic, sending about 1000 packets per flow.

Critical MTC traffic not allocated through the RACH within 50 ms, is transferred to cMTC to avoid further delays before the second packet is ready to be sent.

We consider a BS with a total of about 20 MHz for uplink, using numerology $\mu = 0$, i.e., with subframes lasting $\tau=1$ ms and 1300 slots per subframe. The three bandwidth parts comprise 750 slots for HTC, 350 for MTC over eMBB, and 200 for MTC over cMTC.

Access requests for eMBB can be sent to the RACH every 10 ms, using a total of 54 RACH preambles, out of which 30 are reserved for HTC, 10 for MTC, and the remaining 14 are shared. The RACH cannot acknowledge more than 13 requests per RACH access opportunity. Devices can attempt RACH access up to 10 times for HTC and 3 for MTC, after which they give up on eMBB. After a RACH failure, the device takes a random backoff which is a geometrically distributed number of intervals between RACH access opportunities. The average backoff is 2 for MTC (i.e., 20 ms) and 25 for HTC (i.e., 250 ms). The RACH access is limited by a timeout set to 50 ms for MTC and 5 s for HTC. For MTC, we consider a power ramping offset of 2, i.e., the transmission power of MTC RACH requests is higher than for HTC. Following the standard, the maximum latency of the BS in acknowledging a RACH request is 11 ms [52], which prevents devices to transmit RACH requests in any two consecutive RACH opportunities.

For what concerns flows which are accepted by the RACH, they access a network processor with the limits that $F_h=50$ serving positions are reserved for HTC and $F_m=50$ are for MTC.

We consider a high modulation efficiency for HTC, $\eta_h=8$ bits per symbol, while for MTC we set $\eta_m=4$ bits per symbol.

Flows that instead are diverted to cMTC use r_m transmission on the dedicated cMTC bandwidth part, and r_h transmission over the HTC bandwidth part. We test different values of r_m and r_h .

If a collision occurs on the HTC bandwidth part, we assume that the HTC slot is lost whereas the cMTC minislot remains decodable with probability $1-\alpha$, with α set to either 0.01 or 0.1. We test various minislot sizes, from 2 to 14 symbols, and pre-encode packets with a redundancy factor up to 10.

We will also consider a dense urban scenario in which HTC receives 1000 slots per subframe, MTC over eMBB receives 200 slots, whereas the cMTC service instance is allotted 100 slots. The different partition of resources accounts for the fact that in the dense urban scenario HTC users are slower (i.e., they move at 10 m/s) and cells are smaller (i.e., there is a handover every 200 m or 20 s, about three times more often than in the smart highway scenario). Per-flow traffic characteristics remain unchanged from the smart highway scenario, although the volume of HTC traffic becomes higher.

In what follows, we primarily analyze throughput performance and losses, because they characterize the number of devices that can be connected and the reliability of the offered service, respectively. Instead, we do not show latency figures because successfully delivered messages, by definition, respect the application's timeout; hence, throughput and loss figures also indicate how well the network manages to meet application's latency constraints and support safe and timely MTC operation, the distribution of latency being irrelevant for the MTC application.

A. Benchmarks

In the numerical evaluation presented in what follows, we test CRIMSON-3B and evaluate legacy/SoTA schemes and two variants of our scheme. The considered SoTA schemes are as follows:

- “PMT” (Pure Machine Type)– In this case, MTC traffic uses only NOMA transmissions (i.e., it does not use eMBB) and transmits r_m copies of the packet over dedicated cMTC resources [35];
- “PBB” (Pure eMBB)– MTC traffic only uses the eMBB service category with grants for transmissions; this is the baseline approach without cMTC [53];
- “LBB” (Load Balanced eMBB)– MTC only uses eMBB over a set of dedicated slots, the size of the set being adjusted as offered traffic varies, so as to balance the load of HTC and MTC slices [54].

The two variants of CRIMSON-3B, in which we remove some of the proposed features to test their importance, are as follows:

- “CRIMSON-2B” (CRIMSON over 2 Bandwidth parts)– In this case, MTC transmits using NOMA, although some r_h replicas are sent via cMTC over HTC resources, whereas r_m replicas are sent over dedicated cMTC resources;
- “CRIMSON-1B” (CRIMSON over 1 Bandwidth part)– MTC uses both granted transmissions over eMBB and r_m NOMA replicas over cMTC on the same resources of MTC eMBB, i.e., no collision is possible with HTC resources while cMTC might interfere with other MTC packets sent over eMBB.

B. Simulation

For simulation aiming at the validation of our model of the operations of the network processor we could not use off-the-shelf simulators because they do not support switching between multiple service categories. Advanced uplink cMTC transmissions with coded packets to be split into multiple minislots is not supported either. We thus developed a Matlab simulator that works on a per-subframe basis and that does not force the distribution of critical MTC traffic overflow and arrival rates at the network processor to be Poisson. It does not use exponential think times either, but rather uses application-specific distributions. Moreover, the simulator

allowed us to evaluate the coupling of critical MTC and HTC at the network processor on a per-packet time scale, while the model resorts to average behaviors.

Our Matlab simulator receives as input the output of the RACH model we had previously developed and validated in [55]. The simulator implements the allocation of resources to HTC and MTC, and the NOMA operations of cMTC. It operates a random allocation of slots (for eMBB transmissions) and minislots (for cMTC) according to the tested policies for handling HTC and MTC traffic. cMTC collisions are implemented considering that eMBB transmissions occupy adjacent portions of the frame, while each cMTC minislot is selected uniformly at random among the ones that are not already used by the same uplink transmitter.

Each failed eMBB transmission is immediately reattempted in the next available subframe. This is implemented in the simulator by handling each connection as an independent transmission buffer from which bits are extracted at the beginning of each subframe to populate eMBB slots.

Arrivals are Poisson as regards the first packet of a connection, while the total number of packets in a connection is geometrically distributed. MTC packet interarrivals within each flow are quasi-deterministic; i.e., their spacing is a uniform random variable with up to 10% deviation from the mean value. HTC think time is a truncated exponential with maximum value equal to five times the mean. The user population in the simulator is finite, and each user is marked as connected through a RACH grant (i.e., using eMBB), connected through cMTC, or idle. Poisson arrivals of connections are associated to idle users, so that no user can have more than one active connection at each time.

For MTC flows that resort to cMTC, each packet is coded with a fountain code, then split into an integer number of minislots, and sent over multiple replicas. A packet replica is decoded after the correct reception of enough of its minislots, according to the properties of fountain codes.

C. Model Validation

Fig. 4 compares model results and simulation estimates. The figure refers to the case of CRIMSON-3B with $r_m=1$ and $r_h=3$ when high HTC load is injected (80% of what can be served with 750 radio resources per subframe), and HTC think time 100 ms, in the smart highway scenario.

The surface in Fig. 4 (a) reports the values of the MTC throughput ξ_{mc} over cMTC versus the number of symbols per minislot and the number of redundancy bits R per each data bit (0 means no redundancy, 1 means 100% redundancy, and so on). Surfaces in Fig. 4 (b) and (c) report throughput values for MTC over eMBB (ξ_{me}), and for HTC (ξ_h), respectively.

Simulation results are reported in the same 3D plots. Each simulation point was run multiple times, until the 95% confidence intervals of all throughput metrics became lower than 1% of the respective averages, with a maximum

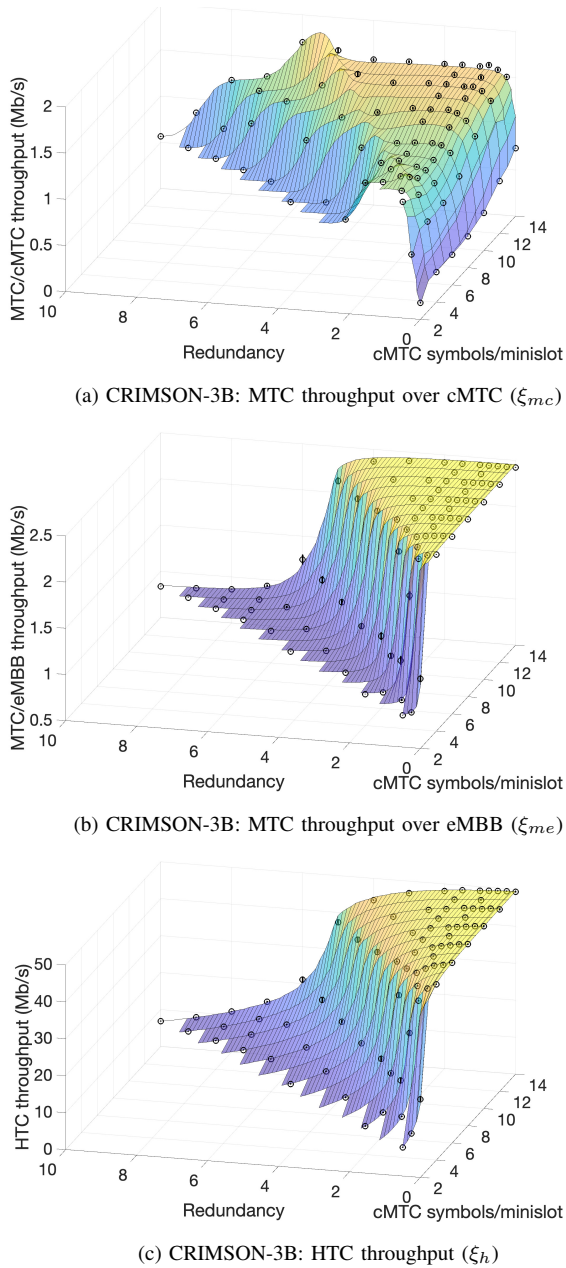


FIGURE 4: Throughput obtained with CRIMSON-3B ($r_m = 1$, $r_h = 3$) with 80% HTC traffic, 100 ms HTC think time, 4 arrivals/s MTC load, $\alpha=0.01$ (smart highway scenario). Circles and bars mark mean and 95% confidence intervals (simulations).

of 100 repetitions. Each circle in the figure represents the average yielded by simulations, and a (very short) vertical bar indicates the associated confidence interval at 95% confidence level. We can first of all observe that simulation results match very well the surfaces representing analytical results for all parameter combinations. Similar good matches between simulation and analysis were observed for many more parameter values, and are not reported here for the sake of brevity. This proves the excellent accuracy of our

analytical model, so that in the rest of this section we will only discuss analytical results.

Observing the surfaces in Fig. 4 we see that using a high number of symbols per minislot is beneficial for both HTC and MTC throughput. This is due to the fact that shorter minislots cause higher numbers of cMTC transmissions, resulting in more frequent collisions with HTC and other cMTC traffic, although some inter-cMTC collisions can be avoided when less than half of the slot symbols are used in each cMTC transmission. For this reason, in the rest of this section we will show results for the case of 14 symbols per minislot, which is the maximum number. In addition, we see that a limited amount of coding redundancy (around 50%) has a very positive impact on the MTC throughput over cMTC, with marginal reductions of the HTC throughput. This behavior is readily explained by the fact that increasing redundancy delivers diminishing marginal gains while causing traffic to rise linearly.

Quite odd is the shape of the surfaces and in particular the bump that can be observed in the surface of the MTC throughput over cMTC. This is due to multiple contrasting and intertwined factors. First, all surfaces stop when the number of minislots required to transmit all cMTC replicas of a single transmission exceeds what is available. Higher values of redundancy and/or less symbols per minislot would not be feasible. This is why the surfaces cover smaller and smaller ranges of redundancy values as the number of symbols per slot diminishes. Second, while it is not surprising that the cMTC throughput shows a maximum relative to the redundancy level, the appearance of a bump was quite unexpected. On the one hand, redundancy helps only until the amount of required cMTC minislots generates too many collisions between different cMTC packets, after which, the impact of collisions becomes predominant. This explains the shape of the surface on the rightmost and leftmost part of Fig 4 (a). On the other hand, a side effect of increasing the occupancy of resources with cMTC, consists in increasing the MTC overflow from eMBB to cMTC. This happens because cMTC replicas that collide with HTC cause HTC retransmissions, hence make HTC connections reside for longer time in the network processor (remember that the average number of packets per connection is the same for all HTC connections). In turn, this means that MTC flows find the access to network processor resources more challenging, hence overflow with higher probability. As shown in Fig. 4 (b), the throughput of MTC over eMBB suffers a sudden “change of phase” as a function of redundancy. At that point in which the MTC/eMBB throughput collapses, a large mass of MTC traffic overflows to cMTC while the collision probability is still far from 1, and causes a sudden rise of cMTC throughput. However, the effect of collisions takes over quite soon, and yields the bump shape of Fig. 4 (a). The increase of overflow counteracts the increase of collision probability only “during” the change of phase of Fig. 4 (b), after which, only collisions increase,

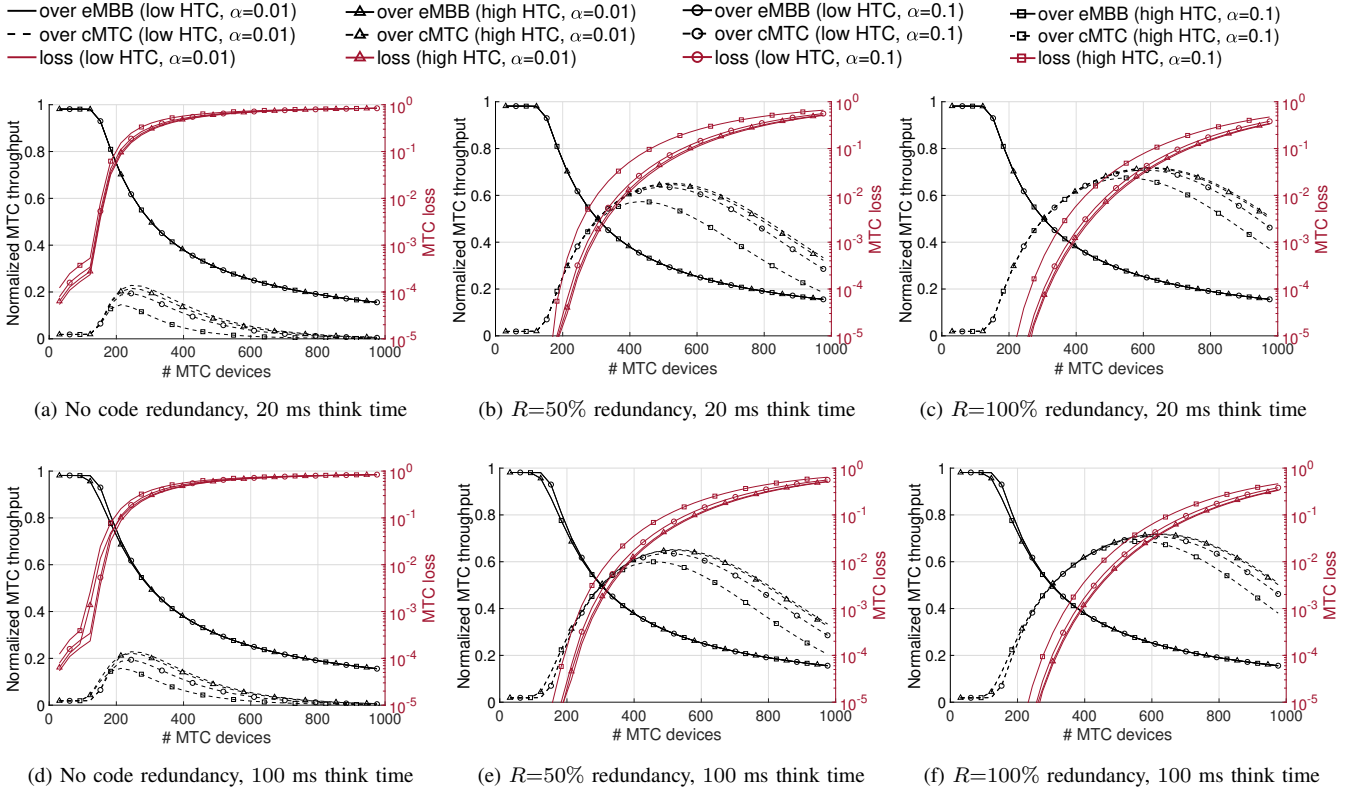


FIGURE 5: MTC throughput components (black curves, left y-axis) and normalized loss (red curves, right y-axis) of CRIMSON-3B (with $r_m = 1$ and $r_h = 1$ replicas) with 14 cMTC symbols/minislot (smart highway scenario).

which explains the change of slope in Fig. 4 (a) as the redundancy increases after reaching the tip of the bump. Note also that eMBB has reserved resources for MTC, so that the throughput cannot go below the floor value of Fig. 4 (b). The impact of cMTC activity over HTC is somehow smoother than what observed for MTC. In fact, Fig. 4 (c) shows that cMTC progressively slows down HTC, with an acceleration in the area that corresponds to the rapid change of phase of MTC/eMBB (hence to the bump area).

D. MTC Throughput Components

We now analyze the throughput components of MTC with CRIMSON-3B. Without loss of generality, we keep using the smart highway scenario to draw numerical examples.

In Fig. 5 we show the components of the MTC throughput (black curves, left linear vertical axis) normalized to its offered load, as well as the MTC throughput loss (red curves, right log vertical axis) versus the number of MTC terminals in the network. Results are shown for low and high HTC traffic (i.e., 20% and 80% HTC offered load, computed with reference to the capacity achievable with 750 slots/ms), short and long think time of HTC (i.e., 20 ms and 100 ms) and for MTC decoding probability in the case of collision over HTC equal to either 0.9 or 0.99 (i.e., $\alpha = \{0.1, 0.01\}$). The number of symbols per slot is taken to be 14, and we use CRIMSON-3B with $r_h = r_m = 1$ cMTC replicas. Moving from left to right in each line of the figure, the three plots show respectively

the cases of no redundancy, and $R=50\%$ or $R=100\%$ code redundancy.

Since we are interested in the capacity of the BS with CRIMSON-3B, in terms of running devices, the figure reports losses over the right axis, whose acceptable level is specific of each application (some require no more than 0.1%, others cannot tolerate more than 10^{-5}).

In Fig. 5, we can see only small differences between the cases of 20 and 100 ms think time, and even the impact of α is not substantial, although obviously a higher MTC decoding probability and a lower HTC traffic increase the throughput of cMTC. On the contrary, the impact of redundancy is striking. Going from 0 to 50% redundancy, the number of sustainable MTC users (i.e., the number of users that corresponds to an MTC loss close to 10^{-3}) roughly doubles from about 150 to about 300. Further increasing redundancy to 100%, the number of sustainable MTC users grows to about 400.

This is surprisingly good performance, especially if we compare to simpler resource managements schemes, using for example no eMBB for MTC traffic (CRIMSON-2B scheme), or just eMBB for MTC traffic (PBB scheme). Results are reported in Fig. 6, where we see a clear advantage of our proposed resource management scheme for MTC throughput, with only marginal reductions in HTC throughput for very high numbers of users. The figure also shows that adding more replicas is not necessarily advantageous.

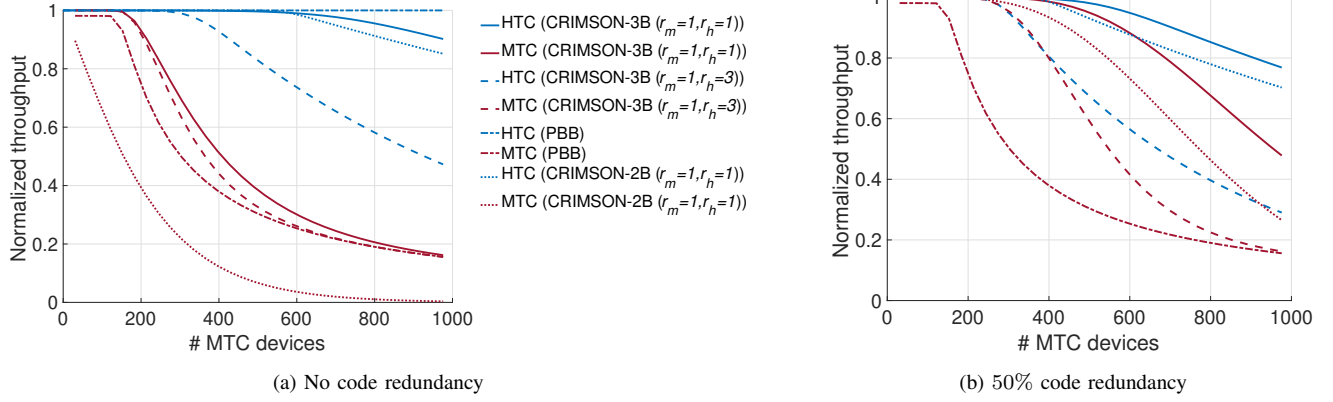


FIGURE 6: Comparison between CRIMSON-3B, eMBB baseline and alternative schemes, with 20 ms HTC think time, 80% HTC load, $\alpha = 0.01$, 14 cMTC symbols per minislot and $R=50\%$ redundancy in the encoding of cMTC (smart highway scenario).

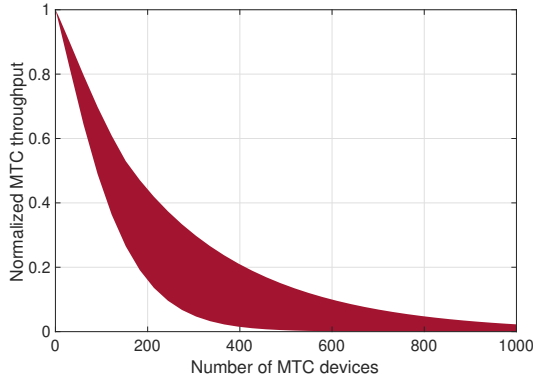


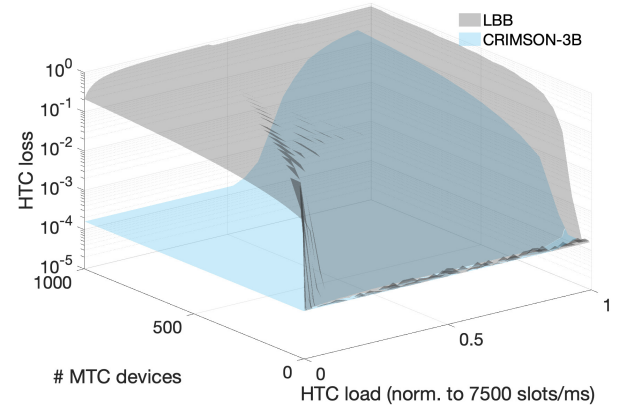
FIGURE 7: Achievable performance of CRIMSON-2B (using only cMTC for MTC) with 1 to 3 repetitions on HTC resources and 1 to 4 repetitions on dedicated cMTC resources, and cMTC dedicated bandwidth occupying up to 1000 slots/ms (out of 1300 total slots/ms), with high HTC load (80% of the capacity at 750 slots/ms). Traffic parameters as in the highway scenario.

In fact, CRIMSON-3B with $r_m=r_h=1$ works better than with $r_m=1$ and $r_h=3$. This result is consistent with those observed for different redundancy levels, as the number of replicas also contributes to redundancy, thereby yielding progressively diminishing marginal gains while causing cMTC traffic and HTC retransmissions to increase linearly with the number of replicas.

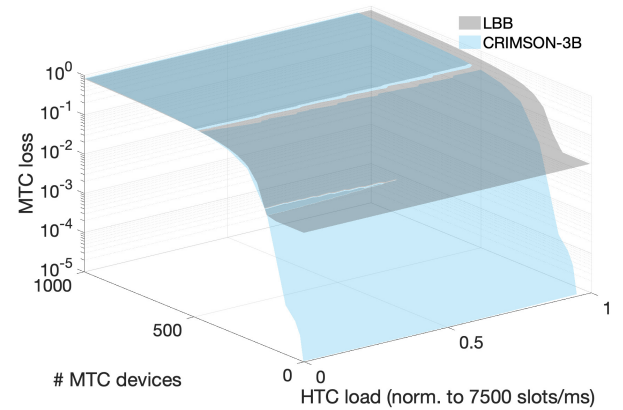
Considering the results in Fig. 5, in the following we will mostly consider $R=50\%$ code redundancy and 20 ms HTC think time, having in mind that with higher code redundancy and HTC think times, performance (in terms of the maximum number of MTC devices) is somewhat better.

E. Alternative approaches

Fig. 7 shows that MTC traffic sent exclusively on cMTC cannot fulfill the requirements of time critical applications as



(a) HTC performance with LBB and CRIMSON-3B



(b) MTC performance with LBB and CRIMSON-3B

FIGURE 8: Comparison between Load balancing over eMBB (LBB) and our scheme CRIMSON-3B with $r_h = 2$ and $r_m = 1$. Traffic parameters as in the highway scenario.

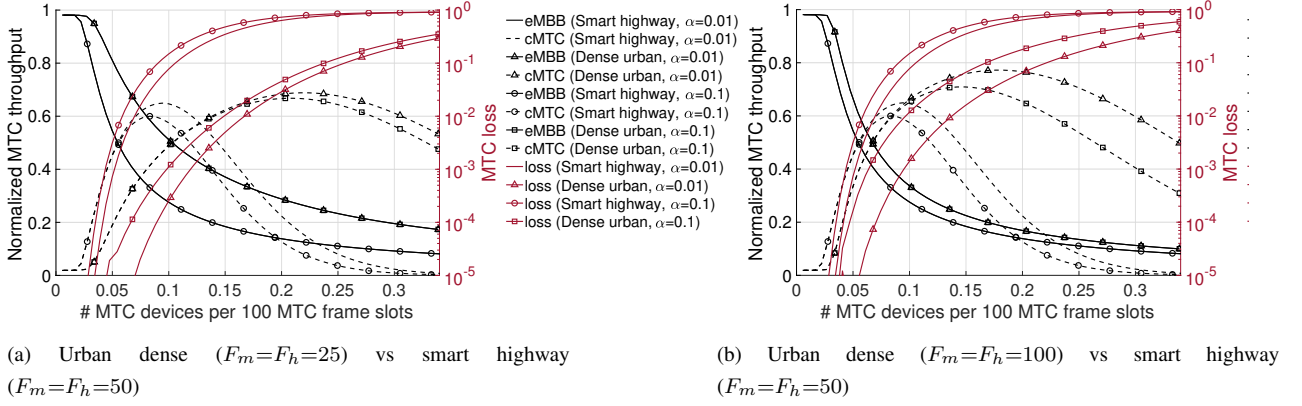


FIGURE 9: Comparison of MTC performance breakdown in the smart highway and the dense urban scenarios, with $R=50\%$ code redundancy, 100 ms HTC think time, 80% HTC load. Black curves report normalized throughput (left y-axis) whereas red curves report losses (right y-axis).

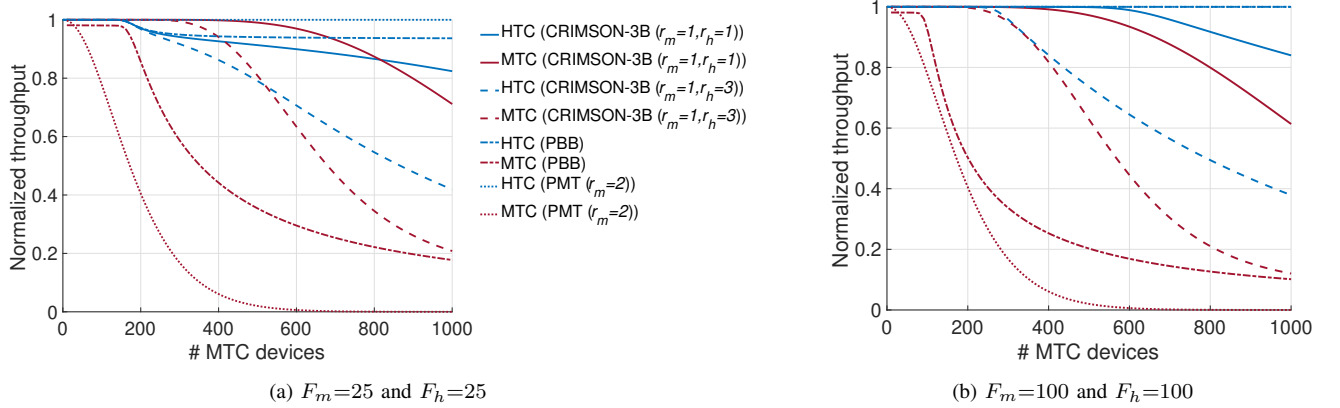


FIGURE 10: Comparison between CRIMSON-3B and baseline configurations, with 20 ms HTC think time, 80% HTC load, $\alpha = 0.01$, 14 cMTC symbols per minislot and $R=50\%$ redundancy in the encoding of cMTC packets (dense urban scenario with $F_m=25$ and $F_h=25$ in plot (a) and with $F_m=100$ and $F_h=100$ in plot (b)).

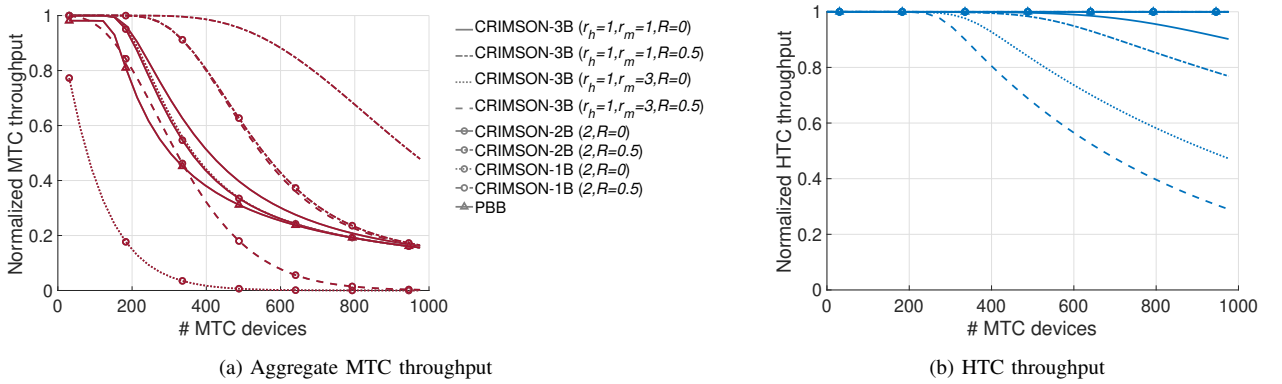


FIGURE 11: Aggregate throughput with CRIMSON-3B for 20 ms HTC think time, 80% HTC load, $\alpha=0.01$ and 14 cMTC symbols per minislot (smart highway scenario). Numbers in parentheses indicate cMTC replicas and the code redundancy adopted.

soon as a handful of devices are present. The figure considers the specifications presented above for the highway scenario. The area shown in the figure represents the set of possible values for the MTC throughput when the number of cMTC repetitions varies from 1 to 3 on the shared bandwidth and from 1 to 4 in the reserved MTC bandwidth, with reserved MTC bandwidth up to 100 slots/ms, and HTC offered traffic set to 80% of the capacity that corresponds to 750 slots/ms. Here we do not add coding redundancy to MTC packets.

Fig. 8 illustrates the performance obtained with pure load balancing (no cMTC, only two eMBB slices, one for HTC and one for MTC). The figure also reports results for CRIMSON-3B when one cMTC replica is sent over dedicated resources and 2 over HTC resources. From the comparison, it becomes clear that load balancing cannot boost MTC performance by relying on using eMBB only, except with a very high number of machine devices, when loss is however too high for both of the schemes considered in the figure. Moreover, in so doing, load balancing is not able to protect HTC traffic, not even when the number of machine devices is very small.

F. Performance in different scenarios

CRIMSON-3B is not specifically meant to work in smart highway scenarios like the ones considered in the above subsections. Fig. 9 compares the performance of our proposal in the smart highway scenario to those in a dense urban scenario, where the distribution of radio resources is more in favor of HTC. To have a fair comparison between schemes in which radio resources are differently available, the abscissa of the figure reports the number of MTC devices present in the system per each 100 radio slots of a subframe.

The number of dedicated connections at the network processor here is set to either $F_h = F_m = 25$ or $F_h = F_m = 100$. The former setting allows the sharing of most of the resources of the network processor between HTC and MTC. In the latter setting, no sharing is possible.

We can see that results for the dense urban scenario are qualitatively aligned to the ones for the smart highway scenario. Interestingly, however, the dense urban scenario is more favorable to MTC devices; this is due to the fact that HTC flows are much longer and tend to occupy more resources in the smart highway scenario. The importance of sharing network processor resources can be appreciated by comparing the two plots. When sharing is allowed, the number of MTC devices that can be supported is significantly larger.

Fig. 10 shows that the impact of CRIMSON-3B on HTC can be neglected until several hundreds of MTC devices are active in case of no shared network processor resources, while with $F_h = F_m = 25$, HTC is not affected only until about 180 to 190 MTC devices (without redundancy). Here, PMT (with two replicas) would be able to cope with a handful of machine users, while not using cMTC at all would preclude any high reliability guarantees to MTC. Also in this case,

the number of cMTC replicas over the resources of HTC cannot increase arbitrarily and $r_h = 1$ works optimally, as numerically verified in our experiments.

G. Different cMTC Redundancy

To further investigate the impact of redundancy on MTC and HTC throughput, in Fig. 11 we plot the aggregate MTC and HTC throughput in the smart highway scenario with our CRIMSON-3B scheme in the case of 80% HTC load with 20 ms HTC think time, $\alpha = 0.01$ and 14 cMTC symbols per minislot. The considered values of redundancy are 0 and 50%, and the cMTC replication schemes used are $r_m = r_h = 1$ as well as $r_m = 1$ and $r_h = 3$. The plots of the aggregate MTC throughput show that a large difference exists between the case of no redundancy and the cases with redundancy. If we also look at the plots of the HTC throughput, we see that (obviously) an increased redundancy in cMTC transmissions reduces the HTC throughput, and the reduction is more pronounced with more replicas on HTC resources, since each one of the 3 repetitions that overlap HTC resources carry the increased redundancy. In contrast, using cMTC “politely”, i.e., without interfering with HTC, is far from optimal.

H. Different cMTC Schemes

To better appreciate the QoS level of both MTC and HTC traffic as a function of the number of MTC terminals, in Fig. 12 we report the fraction of lost traffic for both MTC (chart a) and HTC (chart b) for various schemes. Considering only the CRIMSON-3B with $r_h = r_m = 1$, we see that with 300 MTC terminals the fraction of lost traffic is around 0.1% for both HTC and MTC. The plots in 12 (a) clearly show that CRIMSON-3B with $r_h = r_m = 1$ is by far the best option to obtain good performance for MTC traffic, in comparison to the alternatives of using eMBB only, of using cMTC politely, and also of different numbers of replicas of cMTC packets. As regards the number of replicas, we considered different options, from one replica on cMTC resources and no transmission on HTC resources to two replicas on cMTC resources and four replicas on HTC resources. The replica combination $r_m = r_h = 1$ is the one that achieves the most efficient use of resources in the studied scenario while transmitting further replicas (in particular over HTC) becomes progressively less beneficial. In general, the impact on cMTC is impressive, allowing the system to handle hundreds of devices instead of just a few of them. The combined use of code redundancy, replicas, and reuse of resources—as we have proposed—can boost performance dramatically and can allow the system to cope with high MTC reliability requirements of the order of 5 or 6 nines. At the same time, HTC is barely affected, e.g., with 200 active MTC devices.

VII. Conclusions

In this article we introduced a new approach to the coexistence of MTC and HTC, when MTC traffic is generated

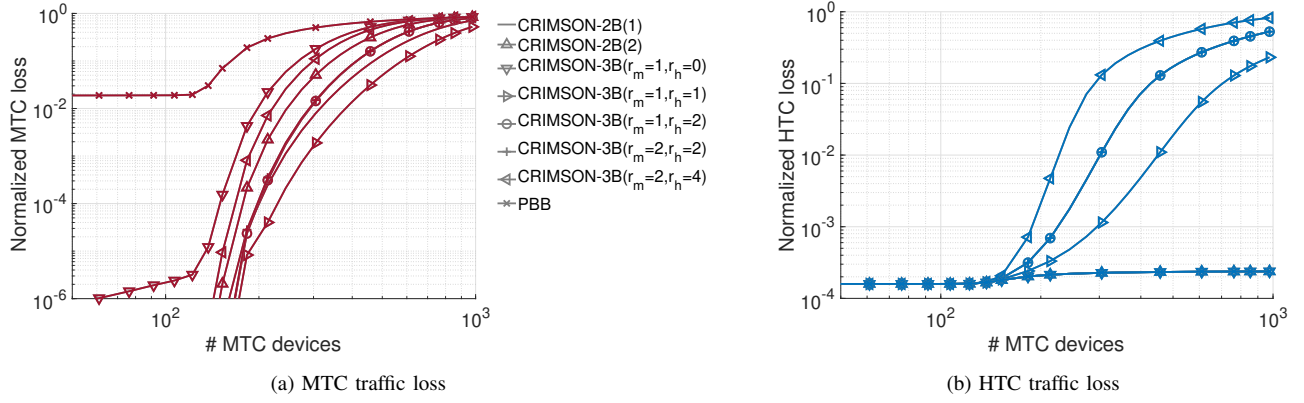


FIGURE 12: Impact of the number of replicas on traffic loss with 20 ms HTC think time, 80% HTC load, $R=50\%$ code redundancy in cMTC transmissions, $\alpha = 0.01$ and 14 cMTC symbols per minislot (smart highway scenario).

by applications such as remote driving, and consists in sequences of packet transmissions to report status information and receive the corresponding commands timely and with high reliability, according to application-specific requirements. Our approach, CRIMSON-3B, is able to manage large numbers of MTC devices generating latency-critical MTC traffic as well as HTC terminals with high throughput requirements. CRIMSON-3B is an unconventional approach to radio resource slicing, with the use of two eMBB and one cMTC service category instances, with MTC traffic that can overflow from eMBB to cMTC, and with cMTC transmission replicas that can interfere with HTC.

The analysis of the system is instrumental to the correct setup of CRIMSON-3B, and required the development of a detailed analytical model for resource allocation in the uplink data plane, for which no alternative analytical derivation was available. Numerical results show that cMTC is key to boost the performance of MTC, and it is convenient to let cMTC reuse resources allocated for HTC transmissions. They also indicate that the use of coding for grant-free transmissions is extremely beneficial, while the number of replicas should be conveniently limited to only a few. The maximum number of machine devices that can be supported with our approach is of the order of a few hundreds, one order of magnitude more than is achievable with traditional and state-of-the-art traffic management approaches. The analysis derived in this manuscript is useful for practitioners and network operators because it enables fast evaluation of uplink network slicing configuration alternatives, at low computational cost and with no need for heavy simulations.

The results in this paper help identify avenues for the effective coexistence within a RAN slice of traffic flows of different nature, with disparate QoS requirements.

REFERENCES

- [1] 3GPP, "Service requirements for the 5G system; (Release 15)," TS22.261 v16.3.0, Apr. 2018.
- [2] —, "Study on management and orchestration of network slicing for next generation network," TR 28.801 Version 15.1.0 - Release 15, 2018.
- [3] S. D'Oro, F. Restuccia, and T. Melodia, "The slice is served: Enforcing radio access network slicing in virtualized 5G systems," in *Proc. of IEEE INFOCOM*, May 2019.
- [4] Z. Li, M. Uusitalo, H. Shariatmadari, and B. Singh, "5G URLLC: Design challenges and system concepts," in *Proc. of ISWCS*, Aug 2018.
- [5] Recommendation ITU-R M.2083, "IMT vision – framework and overall objectives of the future development of IMT for 2020 and beyond," Sep. 2015.
- [6] 3GPP, "Service requirements for the 5G system; (release 15)," TR 38.913, Aug. 2017.
- [7] A. Pourkabirian, M. S. Kordafshari, A. Jindal, and M. H. Anisi, "A vision of 6G URLLC: Physical-layer technologies and enablers," *IEEE Communications Standards Magazine*, vol. 8, no. 2, pp. 20–27, 2024.
- [8] A. Mirri, L. Valentini, I. Leyva-Mayorga, M. Chiani, E. Paolini, and P. Popovski, "Coded random access schemes for critical mMTC with multiple latency deadlines," *IEEE Transactions on Communications*, vol. 73, no. 11, pp. 10662–10675, 2025.
- [9] Q. Song, F.-C. Zheng, and J. Luo, "URLLC over slow fading channels: K-repetition versus multi-connectivity," *IEEE Open Journal of Vehicular Technology*, vol. 6, pp. 1280–1286, 2025.
- [10] Chih-Ping Li, Jing Jiang, W. Chen, Tingfang Ji, and J. Smeets, "5G ultra-reliable and low-latency systems design," in *Proc. of EuCNC*, 2017.
- [11] 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.
- [12] R. Abreu, J. Thomas, K. Pedersen, G. Berardinelli, and P. Mogensen, "System level analysis of eMBB and grant-free URLLC multiplexing in uplink," in *Proc. of IEEE VTC Spring*, Jun. 2019.
- [13] L. Iiyambo, G. Hancke, and A. M. Abu-Mahfouz, "A survey on NB-IoT random access: Approaches for uplink radio access network congestion management," *IEEE Access*, vol. 12, pp. 95487–95506, 2024.
- [14] V. Mancuso, P. Castagno, M. Sereno, and M. Ajmone Marsan, "Slicing cell resources: The case of HTC and MTC coexistence," in *Proc. of IEEE INFOCOM*, 2019.
- [15] —, "Serving HTC and critical MTC in a RAN slice," in *Proc. of WoWMoM 2021*.
- [16] 3GPP, "Physical layer enhancements for NR ultra-reliable and low latency communication (URLLC)," RP-191584, Jun. 2019.
- [17] N. Domènech, S. M. Darroudi, F. Cañellas, D. Camps, S. Rodriguez, A. Gonzalez, O. Lazaro, and T. Ventura, "On the application of 5GNR bandwidth parts to enable remote operations in industrial metrology," in *Proc. IEEE WCNC*, 2025.
- [18] 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.
- [19] 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. 55765–55779, 2018.

- [20] J. Janković, Ž. Ilić, A. Oračević, S. M. Ahsan K., and R. Hussain, "Effects of differentiated 5G services on computational and radio resource allocation performance," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 2226–2241, 2021.
- [21] S. Bakri, P. A. Frangoudis, A. Ksentini, and M. Bouaziz, "Data-driven RAN slicing mechanisms for 5G and beyond," *IEEE Transactions on Network and Service Management*, vol. 18, no. 4, pp. 4654–4668, 2021.
- [22] 3GPP, "System architecture for the 5G system," TS 23.501 Version 15.2.0, 2018.
- [23] 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.
- [24] ETSI, "5G; NR; physical layer procedures for data (3GPP TS 38.214 version 17.1.0 release 17)," TS 138 214 V17.1.0 (2022-08), May 2022.
- [25] Q. Song, C. She, and F. Zheng, "Optimization of repetition scheme for URLLC with diverse reliability requirements," in *Proc. of IEEE VTC Spring*, 2022.
- [26] R. Kumar, D. Sinwar, and V. Singh, "QoS aware resource allocation for coexistence mechanisms between eMBB and URLLC: Issues, challenges, and future directions in 5G," *Computer Communications*, vol. 213, pp. 208–235, 2024.
- [27] I. Gerasin, A. Krasilov, and E. Khorov, "Flexible multiplexing of grant-free URLLC and eMBB in uplink," in *Proc. of IEEE PIMRC*, 2020.
- [28] W. Yang, C. Li, A. Fakoorian, K. Hosseini, and W. Chen, "Dynamic URLLC and eMBB multiplexing design in 5G New Radio," in *Proc. of IEEE CCNC*, 2020.
- [29] L. Feng, Y. Zi, W. Li, F. Zhou, P. Yu, and M. Kadoch, "Dynamic resource allocation with ran slicing and scheduling for URLLC and eMBB hybrid services," *IEEE Access*, vol. 8, pp. 34 538–34 551, 2020.
- [30] D.-S. Lee, C.-S. Chang, R. Zhang, and M.-P. Lee, "Resource allocation for URLLC and eMBB traffic in uplink wireless networks," *Performance Evaluation*, vol. 161, p. 102353, 2023.
- [31] M. Alsenwi, N. H. Tran, M. Bennis, S. R. Pandey, A. K. Bairagi, and C. S. Hong, "Intelligent resource slicing for eMBB and URLLC coexistence in 5G and beyond: A deep reinforcement learning based approach," *IEEE Transactions on Wireless Communications*, vol. 20, no. 7, pp. 4585–4600, 2021.
- [32] E. J. D. Santos, R. D. Souza, and J. L. Rebelatto, "Rate-splitting multiple access for URLLC uplink in physical layer network slicing with eMBB," *IEEE Access*, vol. 9, no. 20, pp. 163 178–163 187, 2021.
- [33] A. K. Bairagi, M. S. Munir, M. Alsenwi, N. H. Tran, S. S. Alshamrani, M. Masud, Z. Han, and C. S. Hong, "Coexistence mechanism between eMBB and URLLC in 5G wireless networks," *IEEE Transactions on Communications*, vol. 69, no. 3, pp. 1736–1749, 2021.
- [34] G. Interdonato, S. Buzzi, C. D'Andrea, L. Venturino, C. D'Elia, and P. Vendittelli, "On the coexistence of eMBB and URLLC in multi-cell massive MIMO," *IEEE Open Journal of the Communications Society*, vol. 4, pp. 1040–1059, 2023.
- [35] T. Jacobsen, R. Abreu, G. Berardinelli, K. Pedersen, P. Mogensen, I. Z. Kovacs, and T. K. Madsen, "System level analysis of uplink grant-free transmission for URLLC," in *Proc. of IEEE GLOBECOM Workshops*, 2017, pp. 1–6.
- [36] Y. Liu, Y. Deng, M. El Kashlan, A. Nallanathan, and G. K. Karagiannis, "Analyzing grant-free access for URLLC service," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 3, pp. 741–755, 2021.
- [37] A. Shashin, A. Belogae, A. Krasilov, and E. Khorov, "Adaptive parameters selection for uplink grant-free URLLC transmission in 5G systems," *Computer Networks*, vol. 222, p. 109527, 2023.
- [38] R. Kotaba, C. Navarro Manchón, T. Balercia, and P. Popovski, "Uplink transmissions in URLLC systems with shared diversity resources," *IEEE Wireless Communications Letters*, vol. 7, no. 4, pp. 590–593, 2018.
- [39] W. R. Ghanem, V. Jamali, Q. Zhang, and R. Schober, "Joint uplink-downlink resource allocation for OFDMA-URLLC MEC systems," in *Proc. of IEEE GLOBECOM*, 2020, pp. 1–7.
- [40] K. Li, P. Zhu, Y. Wang, F.-C. Zheng, and X. You, "Joint uplink and downlink resource allocation toward energy-efficient transmission for URLLC," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 7, pp. 2176–2192, 2023.
- [41] A. Mirri, D. Forlivesi, R. Schiavone, R. Garelo, M. Chiani, and E. Paolini, "A flexible scheme for critical mMTC," in *Proc. of IEEE RTSI*, 2024.
- [42] Z. Zhou, R. Ratasuk, N. Mangalvedhe, and A. Ghosh, "Resource allocation for uplink grant-free ultra-reliable and low latency communications," in *Proc. of IEEE VTC Spring*, 2018.
- [43] 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 *Proc. of ISWCS*, 2019.
- [44] S.-D. Wang, H.-M. Wang, Z. Li, and V. C. M. Leung, "Joint space-time sparsity based jamming detection for mission-critical mMTC networks," *IEEE Transactions on Communications*, vol. 71, no. 2, pp. 1226–1238, 2023.
- [45] O. Elgarhy, L. Reggiani, M. M. Alam, A. Zoha, R. Ahmad, and A. Kusik, "Energy efficiency and latency optimization for IoT URLLC and mMTC use cases," *IEEE Access*, vol. 12, pp. 23 132–23 148, 2024.
- [46] T.-K. Le, U. Salim, and F. Kaltenberger, "An overview of physical layer design for ultra-reliable low-latency communications in 3GPP releases 15, 16, and 17," *IEEE Access*, vol. 9, pp. 433–444, 2021.
- [47] D. J. MacKay, "Fountain codes," *IEEE Proceedings-Communications*, vol. 152, no. 6, pp. 1062–1068, 2005.
- [48] ETSI, "Digital cellular telecommunications system (phase 2+) (GSM); universal mobile telecommunications system (UMTS); LTE; 5G; release 16 description; summary of rel-16 work items (3GPP TR 21.916 version 16.2.0 release 16)," TS 121 916 V16.2.0 (2022-07), Jul. 2022.
- [49] S. S. Lam, "Queueing Networks with Population Size Constraints," *IBM Journal of Research and Development*, vol. 21, pp. 370–378, 1977.
- [50] J. Akhter, R. Hazra, A. Mihovska, and R. Prasad, "A novel resource sharing scheme for vehicular communication in 5G cellular networks for smart cities," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 3, pp. 5848–5856, 2024.
- [51] 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, pp. 905–929, 2020.
- [52] ETSI, "LTE; evolved universal terrestrial radio access (E-UTRA); medium access control (MAC) protocol specification (3GPP TS 36.321 version 17.1.0 release 17)," TS 136 321 V17.1.0 (2022-08), Aug. 2022.
- [53] A. Mahmood, L. Beltramelli, S. Fakhru Abidin, S. Zeb, N. I. Mowla, S. A. Hassan, E. Sisinni, and M. Gidlund, "Industrial IoT in 5G-and-beyond networks: Vision, architecture, and design trends," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 6, pp. 4122–4137, 2022.
- [54] E. Gures, I. Shayea, S. A. Saad, M. Ergen, A. A. El-Saleh, N. M. S. Ahmed, and M. Alnakhli, "Load balancing in 5G heterogeneous networks based on automatic weight function," *ICT Express*, vol. 9, no. 6, pp. 1019–1025, 2023.
- [55] P. Castagno, V. Mancuso, M. Sereno, and M. Ajmone Marsan, "Why your smartphone doesn't work in very crowded environments," in *Proc. of WoWMoM*, 2017.