Resource Management for Multi-Service Coexistence in 5G/6G NFV-MEC Networks

Caio Bruno B. de Souza*, Marcos Rocha de M. Falcão†, Maria G. Lima Damasceno†, Renata K. Gomes Dos Reis†, Andson M. Balieiro†

Centro de Informática (CIn), Universidade Federal de Pernambuco (UFPE), Av. Jornalista Anibal Fernandes, Recife, 50740560, Pernambuco, Brasil.

*Corresponding author(s). E-mail(s): cbbs@cin.ufpe.br;
Contributing authors: mrmf@cin.ufpe.br; mgld@cin.ufpe.br; rkgr@cin.ufpe.br; amb4@cin.ufpe.br;
†These authors contributed equally to this work.

Abstract

Enabling the coexistence of multiple services on the same NFV-MEC network is challenging due to conflicting resource requirements, virtualization overhead, and potential processing failures, all within the strict resource constraints of the NFV-MEC node. Additionally, the critical nature of URLLC services often necessitates service prioritization, which can adversely impact the performance of eMBB applications. This paper addresses these challenges by designing a Continuous-Time Markov Chain (CTMC)-based model that incorporates these features to analyze resource allocation for multiple coexisting services in an NFV-MEC system. Extensive analyses of energy consumption, availability, response time, and memory consumption are conducted across various system configurations. Results reveal that higher loads of URLLC services decrease system availability and increase response times for both service types. The study also finds that an increase in the number of containers does not necessarily lead to a proportional increase in energy consumption, and energy and memory consumption exhibit similar patterns due to their common usage during setup and active processing states. While increasing buffer size slightly improves service availability with minimal impact on energy consumption (as buffered requests do not use resources while in the queue), it negatively affects service response times.

Keywords: MEC, NFV, URLLC, eMBB, 5G/6G Networks, Resource Allocation

1 Introduction

The Fifth Generation (5G) of Mobile Networks was designed to support services with heterogeneous requirements including Ultra-Reliable and Low Latency Communication (URLLC), which focuses on reliability and latency, and enhanced Mobile Broadband (eMBB), comprising bandwidth-hungry applications [1]. Along with 5G Network deployments, the Sixth Generation of Mobile Networks (6G) is expected to expand the 5G services, including Immersive Communication (IC) and Hyper Reliable and Low-latency Communication (HRLLC). The former boosts the e eMBB scenario and provides rich and interactive mobile services to users, including interactions with machine interfaces while the latter extends the URLLC and encompasses applications with more stringent requirements on reliability and latency[2].

To accommodate services with contrasting requirements, Multi-Access Edge Computing (MEC), Network Function Virtualization (NFV), and Network Slicing (NS) are essential in 5G/6G networks [3]. MEC provides computing resources near end users, enabling services and applications to be hosted on MEC nodes. This significantly reduces latency to sub-millisecond levels, which is crucial for URLLC and HRLLC (e.g., autonomous vehicles and remote surgery), and decreases data traffic towards core and external networks. By decoupling network functions from proprietary hardware, NFV allows virtualized network functions (VNFs) to run on generic hardware, offering flexibility and adaptable network resource allocation according to demand dynamics[4]. When integrated with MEC, the advantages of NFV can be extended closer to users, benefiting both URLLC/HRLLC and eMBB/IC services. Moreover, NFV and MEC can be used to enable NS, which allows different services (e.g., URLLC/HRLLC and eMBB/IC) to share the same physical infrastructure by creating multiple customized virtual networks.

Enabling the coexistence of these services on the same NFV-MEC network is challenging since resource allocation must meet conflicting requirements while adhering to the strict resource capacity constraints of the NFV-MEC [1]. Additionally, due to the critical nature of URLLC/HRLLC applications, service prioritization is often adopted to favor them, which may negatively impact the performance of eMBB/IC services. Current studies have addressed service coexistence in 5G/6G networks [1] [5] [6] [7] [8], but they predominantly focus on resource allocation in Radio Access Network (RAN), not covering computational nor cost aspects beyond the RAN. Although there are solutions for resource allocation in the NFV-MEC domain, they often consider fault-free environment [9], instantaneous VNF setup [10] or zero-recovery delay [10] [11], not accounting for the overhead incurred by virtualization technologies. Moreover, many of these solutions only deal with a single type of service [11] [12] [13].

Besides supporting different services, achieving sustainability and energy efficiency are key considerations in 6G networks [14] as they directly impact both the economic and ecological aspects of cellular networks. Energy cost is a significant component of the overall operational expenditure for network operators. Studies [15] indicate that the distributed and widespread nature of MEC nodes leads to notable energy usage. This increased usage, in turn, raises costs, carbon footprint, and energy requirements. Hence, understanding the energy consumption at this level, including the virtualization costs, allows the operators to develop more efficient NFV-MEC systems [16].

This paper fills these gaps by addressing dynamic resource allocation for URLLC/HRLLC and eMBB/IC in the NFV-MEC domain, considering virtualization overhead and potential failures during service processing. Additionally, to meet the strict latency requirements of URLLC/HRLLC services and manage load variations in the NFV-MEC system, service prioritization favoring URLLC/HRLLC and resource scaling upon demand are implemented. By designing a Continuous-Time Markov Chain (CTMC) - based model that incorporates these features, extensive analyses are conducted in terms of energy consumption, availability, response time, and memory consumption, considering various NFV-MEC node and service configurations, such as resource amounts, failure rates, setup times, and service rates. This study may assist operators in properly designing the NFV-MEC system to support both service types and provide insights for developing cost-performance efficient solutions for resource allocation. The results show that higher loads of uRLLC services decrease system availability and increase response times for both types of services. It is also noted that an increase in the number of containers does not necessarily lead to a proportional increase in energy consumption. Additionally, energy and memory consumption exhibit similar patterns due to the common nature of resource usage during setup and active processing states. Regarding the increase in buffer size, it is observed that service availability slightly improves, with minimal impact on the system's energy consumption since buffered requests do not consume resources while in the queue. However, increasing the number of buffer positions negatively impacts service response times.

The remainder of this paper is organized as follows. Section 2 discusses relevant works in the field of NFV-MEC resource allocation. Section 3 describes the proposed CTMC-based model for a single node NFV-MEC, assuming a virtual environment featured with containers that are able to process both URLLC/HRLLC and eMBB/IC requests. Section 4 presents the model validation and a result analysis obtained by extensive discrete-event simulations. Finally, Section 5 provides our concluding remarks and highlights future work directions.

2 Related Work

The coexistence of multiple services with heterogeneous requirements within the same 5G/6G network presents significant challenges in resource allocation and operational cost control, all while concurrently satisfying their Quality of Service (QoS) levels. To address these challenges, various solutions have been proposed. For instance, [5] investigates joint resource allocation for eMBB and URLLC, proposing a sub-optimal solution. In a follow-up study, [1] extends [5] by introducing four new sub-optimal solutions in addition to an optimal one that either ensures a predefined fairness level or guarantees a minimum bandwidth for eMBB users. Both studies focus on bandwidth allocation (resource blocks - RBs) for users and evaluate their solutions in terms of Accumulated Throughput and Spectral Efficiency.

In [6] is formulated a resource block (RB) scheduling optimization problem aimed at maximizing the minimum expected data rate experienced by eMBB users while simultaneously meeting URLLC reliability and response time requirements. The authors decompose the problem into two sub-problems, each addressing RB scheduling

for different user types, and provide solutions for both. They compare their solutions with existing ones from the literature, evaluating them in terms of the cumulative distribution function (CDF) for minimum expected achieved rate (MEAR) and fairness score. Similarly, [8] proposes a joint resource allocation algorithm for eMBB and URLLC users by adopting superposition and puncturing techniques. Their goal is to maximize the Minimum Expected Achieved Rate (MEAR) for eMBB users while ensuring fairness between them and meeting the URLLC QoS requirements. For the superposition technique, the authors apply one-to-one matching theory to compute the pairs of URLLC and eMBB users. Although these works offer valuable solutions with significant results, they are limited to resource allocation within the Radio Access Network. They do not address computational or cost aspects beyond the RAN. These approaches could be integrated into our work, which deals with resource allocation issues in the NFV-MEC domain.

Although there are solutions for resource allocation in NFV-MEC domain, they often consider fault-free environment [9], instantaneous VNF setup [10] or zero-recovery delay [10] [11], without accounting for the overhead incurred by virtualization technologies or the presence of multiple service types. Differently, the authors in [17] study the deployment costs of containerized VNFs (CNFs) in MEC nodes, analyzing latency, energy consumption, and resource demands in terms of memory and CPU usage throughout the VNF lifecycle. They observe that CNF states yield different levels of resource consumption and that state transitions incur significant costs in terms of latency, energy, and resource utilization. The authors provide valuable measurement-based models for all the analyzed metrics as a function of the number of CNFs. In contrast to [17], our work considers different service categories coexisting within the same NFV-MEC node, alongside potential failures and their effects during service processing. Given the significance of Nguyen et al.'s findings on resource consumption, we adopt their models for memory and energy consumption as inputs in our study, demonstrating that these works are complementary.

Sustainability and energy efficiency stand as pillars of 6G networks, as they directly impact both the economic and ecological aspects of cellular networks and are linked to the United Nations' Sustainable Development Goals (UN SDGs) for 2030 [14]. Thus, efforts have to be devoted to minimizing environmental impact by optimizing network architectures, reducing power consumption, and developing energy-efficient solutions in different network segments. In this respect, the authors in [18] propose a strategy for balancing task delay requirements and energy consumption in Integrated Sensing and Communications (ISAC)-aided 6G V2X networks using MEC. They aim to minimize queuing latency with long-term latency and energy consumption constraints for data fusion computing tasks and adopt the Lyapunov optimization method. Although their joint computation offloading and resource allocation (JCORA) scheme presents great results, it neglects failure events during task processing and the different service categories coexisting in MEC nodes, which are expected in real 6G networks.

In [19], a Resource-Ability Assisted Service Function Chain (SFC) Embedding and Scheduling algorithm for virtualization-based 6G Networks is proposed. To enhance the SFC embedding and scheduling, the solution selects nodes with strong capabilities and sufficient resources to accommodate SFCs. Results show that the algorithm

achieves a higher SFC acceptance ratio compared to previous methods. [20], in turn, addresses the Virtual Network Function (VNF) placement in Non-Terrestrial Networks (NTNs), considering resource constraints and bandwidth limitations. It focuses on delay-aware VNF placement to support ultra-low delay services and improve resource utilization. The authors propose Linear Programming-based and Hungarian-based solutions to deal with VNF placement problem, achieving superior performance in terms of resource utilization and execution time. Although both works present important findings, they overlook significant factors in their analysis and formulation, including VNF failure, setup time, and energy consumption. These aspects are essential for designing and operating 6G networks that meet reliability, efficiency, and sustainability expectations.

Regarding the dynamic allocation in NFV-MEC nodes, our previous works [13] [12] propose CTMC-based models to analyze NFV-MEC node performance when supporting URLLC services, considering VNF failure and setup/repair time. Additionally, the former encompasses a resource pre-initialization strategy to mitigate the negative effects of VNF failures and setup time [13] in a container-based environment. While the latter [12] evaluates URLLC services running in a hybrid NFV-MEC node that leverages the strengths of both virtual machines and container technologies. It allows a service provider to properly dimension a MEC-enabled UAV node under availability, power consumption, reliability, and latency perspectives. However, this work differs from both by addressing the coexistence of different services in the NFV-MEC node domain and extending the analysis to encompass memory consumption. Additionally, the current paper extends our previous work presented at the SBRC 2024 conference [21] by broadening the scope, incorporating service categories anticipated for 6G networks, offering a more comprehensive related work discussion and model description, analyzing memory consumption, crucial for resource-constrained environments, and evaluating two additional scenarios.

3 System Model

In this study, a single isolated NFV-MEC node is designed to support two service types: eMBB/IC (orange flow) and URLLC/HRLLC (blue flow), as in Fig. 1. Both eMBB/IC or URLLC/HRLLC requests originated from users in the RAN are passed on to the MEC node and handled by containerized VNFs, which are scaled accordingly. The system has a finite number of containers and limited buffer capacity for each service type, so incoming requests may need to wait in a buffer until a container becomes available. Each container operates independently to execute a single VNF, while a central unit manages the admission control of new requests based on resource availability. When resources are available (containers or buffer positions), a request is admitted and either placed in the buffer if all containers are occupied or immediately assigned to an available container. This mechanism ensures efficient resource utilization and effective processing of service requests within the NFV-MEC node.

The system incorporates a dynamic VNF auto-scaling strategy to manage load variations. Before a containerized VNF can begin processing services, it must undergo an initialization process, which introduces a setup delay in addition to memory and

energy consumption. Furthermore, failures may occur during service processing, necessitating a repair time. If a containerized VNF fails, it will be restarted, and the request it was handling will be reassigned to another VNF if available; otherwise, it will be placed back in the service queue with higher priority over new requests. In either case, the processing is restarted to assure reliability.

Given that URLLC/HRLLC services present strict latency requirements, the resource allocation incorporates a prioritization policy that favors this service type as follows: (1) URLLC/HRLLC services are prioritized over eMBB/IC ones; therefore, containers being released or activated are first allocated to URLLC/HRLLC services. (2) If an URLLC/HRLLC service is waiting to be processed and an eMBB/IC service is completed, the released container will be restarted with the VNF for URLLC/HRLLC services. However, if other containers are available, the current container is allocated to the next eMBB/IC service or deactivated if the eMBB/IC buffer is empty. (3) Preemption of the lowest priority eMBB/IC service in processing is not allowed. Fig. 2 illustrates the resource allocation process in the proposed NFV-MEC node, depicted here with a limited capacity to run up to two containers (VNFs) simultaneously for simplicity of illustration.

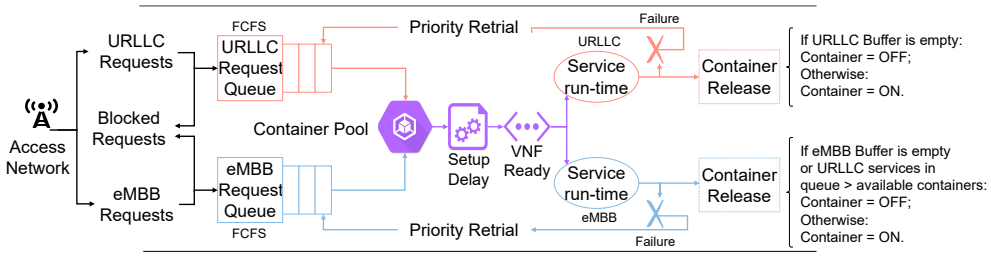


Fig. 1 Edge Node Resource Allocation Flow

In Fig. 2, the first three events are regular service requests, being t0 the 1st eMBB request, $t1$ the 1st URLLC request and $t2$ another eMBB/IC (2nd eMBB). However, since there are only two containers ($CT1$ and $CT2$), only the 1st eMBB and 1st URLLC requests are allocated to each available container ($CT1$ and $CT2$) in $t3$ and $t4$, respectively, while the 2nd eMBB service is placed in a dedicated buffer. This triggers a setup phase due to each container initialization, hence a waiting period is set until the resource is ready; the service is only processed if the setup is successful, such as in $t5$ ($CT1$ finishes setup phase and starts processing the 1st eMBB service) and $t8$ ($CT2$ finishes setup phase and starts processing the 1st URLLC service). Moreover, during the setup intervals, other two URLLC arrivals happen in $t6$ (2nd URLLC) and $t7$ (3rd URLLC), being placed in the URLLC buffer since both containers are currently processing other requests. Up to $t10$, the system is processing an eMBB request in $CT1$ and a URLLC in $CT2$ while holding a single eMBB and two URLLC services in their buffers.

Furthermore in Fig. 2, in $t11$, CT1 completes processing the first eMBB service, becoming available. The same happens in $CT2$ in $t12$, where the first URLLC service

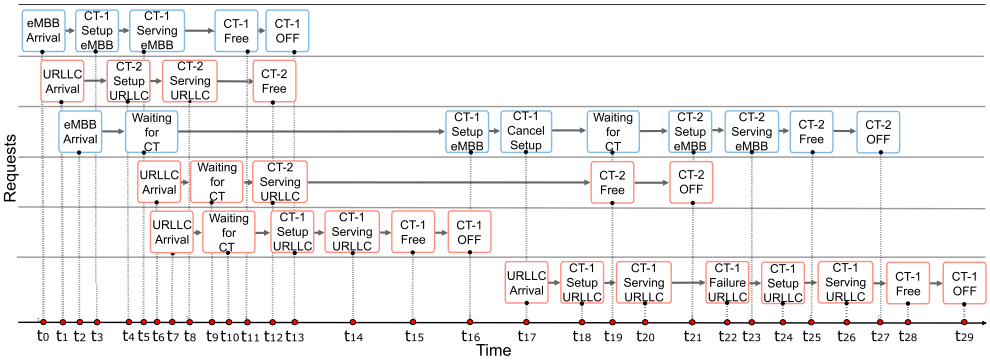


Fig. 2 An Example of Resource Allocation for URLLC and eMBB services in a NFV-MEC System

is completed, however, in this case, $CT2$ uses fast allocation to start serving the second URLLC service that was buffered. On the other hand, in $t13$ $CT1$ is reinitialized to begin serving the third URLLC, which was also buffered. This happens since $CT1$ switches its image and internal components from eMBB to an URLLC service. Only then, in $t14$, $CT1$ starts processing the third URLLC service that ends right after in $t15$. Now in $t16$, again $CT1$ needs to transition from attending an URLLC service to start another eMBB service, which was in the buffer. However, in $t17$, a new URLLC arrival cancels the setup phase for the $CT1$ due to the higher priority given to URLLC services. Hence, in $t18$, $CT1$ begins another setup phase, but this time to address the newly arrived URLLC service (4th URLLC service).

The last set of events in Fig. 2 begins at $t19$, where $CT2$ finishes processing the 2nd URLLC service, while at $t20$, the 4th URLLC service starts to be processed by $CT1$. Furthermore, at $t21$, $CT2$ transitions to begin processing the 2nd eMBB service, which was buffered, incurring a new setup period, only to properly begin processing it at $t23$. At $t22$, a failure occurs during the fourth URLLC service in $CT1$, triggering a new setup period at $t24$. At $t25$, $CT2$ finishes processing the 2nd eMBB service. At $t26$, $CT1$, which experienced a failure and was restarted, begins serving the 4th URLLC service. At $t27$, $CT2$ shuts down since there are no services left to process. At $t28$, $CT1$ finishes processing the last service (4th URLLC), and since there are no services left to process, it also shuts down at $t29$.

3.1 A CMTC-Based Model

The NFV-MEC system is modeled using an M/N/c/k+K queue with First-Come, First Served (FCFS) service discipline, two user types, prioritization, limited buffers for each user type, failure, and resource setup/repair time. The model states are denoted by (i, j, l, m), where $i, j, l, m \in N$ with i and j being the number of URLLC/HRLLC and eMBB/IC services in the system, respectively, and l and m the mean number of active containers for each user type, with $l + m$ being less than or equal to the maximum number of containers (c). Service request arrivals follow a Poisson process with rates λ_U for URLLC/HRLLC services and λ_E for eMBB/IC services. The c

available containers provide service processing with exponentially distributed service times, with rates μ_U for URLLC/HRLLC and μ_E for eMBB/IC. Similarly, the container initialization and failure occurrence times follow exponential distributions with rates α and γ, respectively. The maximum system capacity for URLLC/HRRLC and eMBB/IC users are denoted by k and K, respectively. Fig. 3 summarizes the transitions and states within the proposed system, along with their respective parameters, enabling the calculation of steady-state probabilities ($\pi_{i,j,l,m}$) as outlined in [12] and the derivation of key performance metrics as described in Section 3.2.

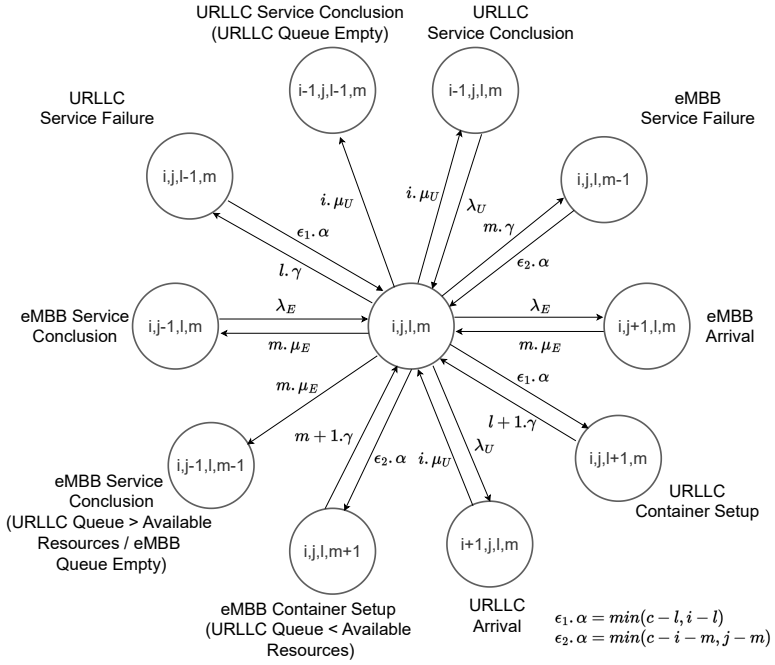


Fig. 3 Generic CTMC State Transition Diagram

3.2 Derived Metrics

In NFV-MEC networks, service processing at the MEC server is strongly tied to resource availability, response time, and memory consumption while power management is crucial for network operation costs [22].

3.2.1 Availability

Placing services closer to users can significantly reduce latency, but the resource limitations of edge nodes may constrain their service capacity and, consequently, their availability. When capacity is exceeded, incoming service requests must be forwarded to neighboring nodes or central cloud. This not only introduces uncertainty regarding

latency but also increases the risk of service denial, which can be even more detrimental. Therefore, it is imperative to analyze the availability of NFV-MEC nodes for each service type. In our model, availability (A) is defined as the system's ability to accept new service requests, which depends on having at least the minimum amount of functional and accessible VNFs or buffer positions available. Furthermore, due to service prioritization, the availability for each service type—URLLC (A_U) and eMBB (A_E)—is represented by Eqs. 1 and 2, respectively, which express the complementary probability of the system reaching its maximum capacity for each service type.

$$A_U = 1 - \sum_{j=0}^{K} \sum_{l=0}^{c} \sum_{m=0}^{min(c-l,j)} \pi_{k,j,l,m} \tag{1}$$

$$A_E = 1 - \sum_{i=0}^{k} \sum_{m=0}^{c} \sum_{l=0}^{min(c-m,i)} \pi_{i,K,l,m} \tag{2}$$

3.2.2 Response Time

The response time is crucial for URLLC due to its strict latency requirements. It is also important for eMBB applications, as low response times enhance the user experience for services like online gaming and video streaming. In this work, response time is defined as the interval between the service's admission at the NFV-MEC node and its completion, including any VNF (container) initialization and recovery overhead. Eqs. 3 and 4 denote the response time for URLLC and eMBB services, respectively, which are calculated as the ratio between the mean number of URLLC (or eMBB) users in the NFV-MEC system and the rate of admitted URLLC (or eMBB) users.

$$T_U = \frac{\sum_{i=0}^{k} \sum_{j=0}^{K} \sum_{l=0}^{min(c,i)} \sum_{m=0}^{min(c-l,j)} i\pi_{i,j,l,m}}{\lambda_U A_U} \tag{3}$$

$$T_E = \frac{\sum_{i=0}^{k} \sum_{j=0}^{K} \sum_{l=0}^{min(c,i)} \sum_{m=0}^{min(c-l,j)} j\pi_{i,j,l,m}}{\lambda_E A_E} \tag{4}$$

3.2.3 Power Consumption

Computational power consumption is a critical factor that service providers must consider during system dimensioning. In our formulation, the average power consumption (PC) is calculated by totaling the consumption of virtual resources (containers) in each operational state, including setup and busy, as shown in Eq. 5. These components are determined using Eqs. 6 and 7 from [17], where $CTsetup$ and $CTbusy$ represent

the number of containers in the setup and busy states, respectively, as defined by Eqs. (9) and (8). The first equation estimates the average number of containers in the busy state by iterating over each system state under service load, varying the combination of each container type from 0 to the number of services in a particular category or the maximum resources available. Additionally, Eq. (9) calculates the average number of containerized VNFs in the setup state by iterating over states where the number of active services exceeds the total number of available resources for each service category. It is important to note that PC accounts for the power consumption associated with managing both service types and is expressed in Watts.

$$PC = 112 + \Delta P_{setup} + \Delta P_{busy} \tag{5}$$

$$\Delta P_{setup} = 0.455 CT_{setup} - 0.00224 (CT_{setup})^2 \tag{6}$$

$$\Delta P_{busy} = 7.23 CT_{busy} \tag{7}$$

$$CT_{busy} = \sum_{i=0}^{k} \sum_{j=0}^{K} \sum_{l=0}^{min(c,i)} \sum_{m=0}^{min(c-l,j)} (l+m) \pi_{i,j,l,m} \tag{8}$$

$$CT_{setup} = \sum_{i=0}^{k} \sum_{j=0}^{K} \sum_{l=0}^{min(c,i)} \sum_{m=0}^{min(c-l,j)} min((c-l-m), (i+j-l-m)) \pi_{i,j,l,m} \tag{9}$$

3.2.4 Memory Consumption

Given the ongoing consideration of the resource constraints of edge nodes, it is essential to carefully analyze the resource consumption of services running in the NFV-MEC system to ensure optimal node configuration. In this paper, memory consumption (M) is also assessed for both service types, taking into account the VNF (container) setup and active (processing) stages. Eq. 10, as described in [17], outlines a linear relationship for memory consumption (in MB), where $CTbusy$ and $CTsetup$ are defined by Eqs. 8 and 9, respectively.

$$M = 3360 + 30.9 CT_{busy} + 5.62 CT_{setup} \tag{10}$$

4 Result Analysis

Extensive discrete-event simulations (Figs. 4-15) were performed using a Coloured Petri Net-based simulator [23] to validate the model's analytical results, with lines representing analytical findings and markers indicating simulation outcomes. We define four scenarios that simultaneously assess the influence of pairs of parameters on the system performance. The first scenario (Section 4.1) involves variations in container amounts (c) and buffer positions for eMBB (K). Similarly, configurations with different numbers of containers (c) and buffer sizes for URLLC (k) are analyzed in the second scenario (Section 4.2). These scenarios aim at the impact of adopting systems with different capacities in terms of parallel processing and service admission on both eMBB and URLLC services, as well as on power and memory consumption. The last two scenarios assess the combined effects of service rates and buffer size settings for URLLC (Section 4.3) and eMBB services (Section 4.4), illustrating how enhancements in service request processing speed and the system's capacity to admit each user type can positively impact the system's overall functionality.

For all scenarios, the URLLC service arrivals (λ_U) were varied from 2.5 to 25 requests/ms in order to analyze the system performance under different URLLC loads. Unless stated otherwise, the baseline values for failure (γ) and setup rates (α) were set to 0.001 and 1 unit/ms, respectively, in accordance with [24]. The remaining parameters are summarized in Table 1. The following sections present the average results for each scenario. Each data point is based on 10 simulation instances, with each instance consisting of 2,700,000 simulation steps and processing 2,200,000 services. The simulations were executed on a computer with an Intel Core i7-9750H 6-core processor (4.50 GHz), 16 GB of RAM, and running Windows 10. The Bootstrap method [25] was applied, with the resample size set at 30 and the number of (re)samplings set to 1000.

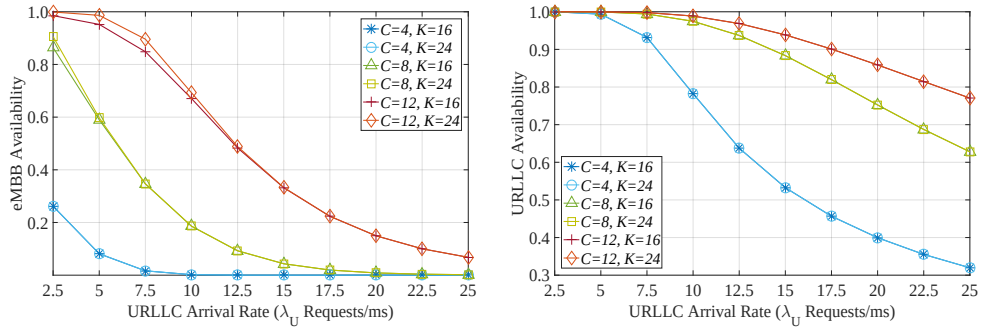
Table 1 Adopted Parameters and Scenarios

Section	λ_E	α	γ	μ_U	μ_E	C	K	k
4.1	10	1	10^{-3}	2	2	4,8,12	16,24	16
4.2	10	1	10^{-3}	2	2	4,8,12	16	16,24
4.3	10	1	10^{-3}	1,2,4	2	10	16	16,24
4.4	10	1	10^{-3}	2	1,2,4	10	16,24	16

4.1 Impact of container quantity (c) and eMBB buffer size (K)

To begin our analysis, the first scenario examines how different NFV-MEC node configurations, specifically the number of containers (c) and buffer sizes for eMBB services (K), affect system behavior. Figs. 4a and 4b show that increasing the number of containers positively impacts the availability of both service types. The highest availability is achieved with 12 containers (black and orange curves), followed by the configuration with $c = 8$ (green and yellow curves). For instance, when λ_U equals 10 in Fig. 4a, eMBB availability is about 20% with 8 containers, whereas with $c = 12$, it reaches 69%, representing an absolute difference of 49%. On the other hand, the alternatives

analyzed for eMBB buffer size showed minimal impact on eMBB availability, suggesting that higher values than those tested could influence eMBB availability. However, the adopted configurations still play a role as they affect the response time, which will be discussed later. A similar behavior occurs for the URLLC availability (Fig. 4b), where the number of containers has a pronounced impact, while the buffer size has a modest effect, resulting in overlapping curves.



(a) eMBB Availability (b) URLLC Availability

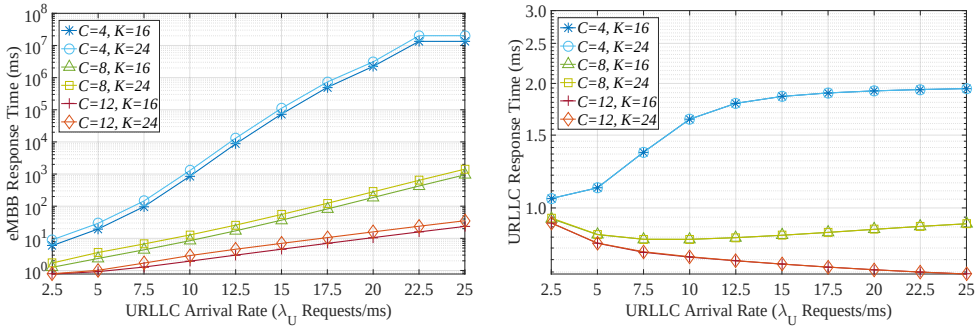
Fig. 4 Effects of the number of containers (c) and eMBB buffer sizer (K) on availability

A larger buffer for eMBB services leads to an increased eMBB response time, as shown in Fig. 5a, because it allows more eMBB service requests to wait in line for processing, resulting in longer queuing delays. Conversely, adopting more containers reduces this queuing time and, consequently, the eMBB response time. An underestimated number of containers may result in excessively long response times, making non-prioritized applications unfeasible. For instance, NFV-MEC node configurations with $c = 4$ (light and dark blue curves) can support Smart Office applications that require a maximum latency of 10 ms [26] coexisting with URLLC services with a load of up to $\lambda_U = 2.5$. In contrast, the other configurations can achieve this coexistence even for higher λ_U values, such as 12.5.

The eMBB buffer size variation has minimal influence on URLLC response time, as shown in Fig.5b. It is evident that the response time is predominantly impacted by the number of containers. We observe that only configurations with $c = 8$ and $c = 12$ can support robotic applications, which require a latency of 1 ms, a latency not achieved by setups with 4 containers, even at the lowest analyzed λ_U value of 2.5. However, NFV-MEC nodes comprising 4 containers provided a response time lower than 2ms for all analyzed λ_U values, making them feasible for Intelligent Transportation Systems, which tolerate latency from 10 to 100 ms [27].

An interesting aspect can be observed in Fig. 5b (left part) regarding the configurations with $c = 8$ (green and yellow curves) and $c = 12$ (red and orange curves). Initially, their response times decrease, but for $c = 8$, this behavior changes, with the response time returning to its initial value when λ_U reaches 25. This is likely due to the container setup time for those processing eMBB services or turned off but needing to be reconfigured to handle URLLC services. This behavior takes place at low URLLC loads, with λ_U ranging from $2,5$ to 10, which is when eMBB services have a higher

chance of accessing computing resources. Conversely, a reduction in URLLC response time is observed within a given interval of λ_U, reflecting the higher amount of ready containers to process URLLC services, which reduces the container reconfiguration.



(a) eMBB Response Time (b) URLLC Response Time

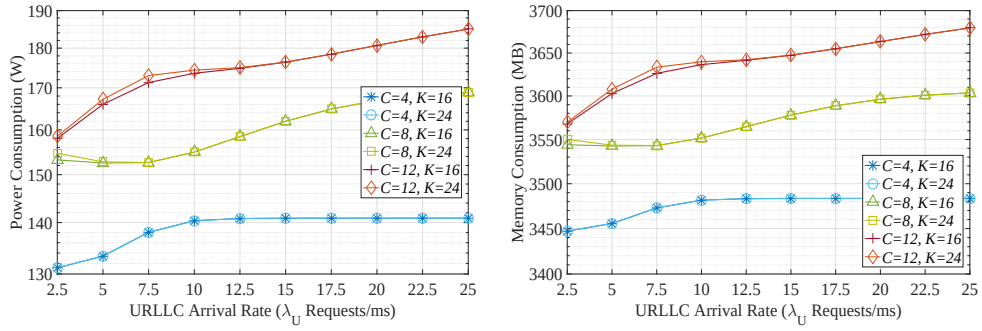
Fig. 5 Effects of the number of containers (c) and eMBB buffer sizer (K) on Response Time

In contrast to response time, an increase in the number of containers boosts power consumption, as illustrated in Fig. 6a. For example, with $\lambda_U = 10$, doubling the number of containers from 4 to 8 results in a 70% increase in energy consumption. However, this increase in power consumption does not scale proportionally with the number of containers because the service load may not be sufficient to keep all containers active and processing services continuously.

It is also observed in Fig. 6a that varying the eMBB buffer size, while keeping the number of containers constant, results in only minimal differences in power consumption. A larger buffer leads to slightly higher power consumption because more users are waiting for processing, which prevents containers from being turned off or reinitialized. This keeps the containers in a high state of consumption for a longer period. These findings highlight the importance of balancing performance and energy efficiency in NFV-MEC node design. Ensuring sufficient computing resources to meet service demands while avoiding excessive power consumption is crucial. This has direct implications for operators aiming to optimize resource provisioning strategies and minimize operational costs.

When comparing the energy consumption (6a) and memory consumption (6b) graphs, we can see that despite the different values, both present similar behaviors. This phenomenon occurs due to the similar nature of these consumptions, in which memory consumption presents higher values for the active and in-process state of the containers, as well as energy consumption. In the VNF setup process, a similar phenomenon occurs, because at this moment the system will move the data necessary for the execution of the VNF to the memory gradually, until the total value of memory consumed by the VNF in the processing state is reached. During the execution time of this process, the average memory consumption of the VNF is naturally lower than its total consumption in the active processing state, similar to what occurs with energy consumption during the setup process.

The 6b graph shows a memory consumption variation of 233 MB between the lowest consumption of the evaluated configurations ($c = 4, \lambda_U = 2.5$) and the highest consumption ($c = 12, \lambda_U = 25$), showing that improvements in the amount of available processing resources along with the growing service demand of users can significantly impact the operation of a small NFV-MEC node. These results emphasize the need for optimized resource management in NFV-MEC environments, especially in memory- and energy-constrained scenarios (e.g., extreme edge nodes and unmanned aerial vehicles). While increasing the number of containers enhances service availability, it also raises memory usage and power consumption. This trade-off calls for intelligent scaling strategies, such as dynamic provisioning and load balancing, to prevent resource overallocation during low-demand periods. Moreover, memory consumption is a key factor in determining the cost and feasibility of NFV-MEC deployments. Excessive memory usage in scenarios with hardware limitations can cause performance bottlenecks and drive up operational costs by requiring more expensive, high-capacity hardware.



(a) Power Consumption (b) Memory Consumption

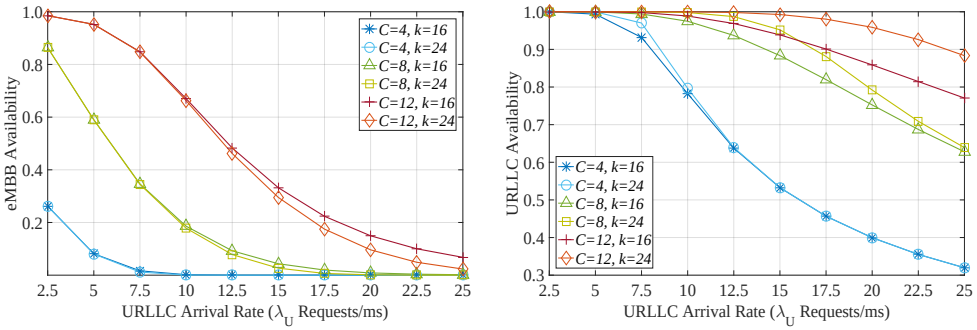
Fig. 6 Effects of the number of containers (c) and eMBB buffer sizer (K) on Resource Consumption

4.2 Impact of container quantity (c) and URLLC buffer size (k)

Continuing the evaluation of NFV-MEC node configurations, this experiment focuses on the impact of container quantity (c) and URLLC buffer size (k). While the previous scenario examined eMBB buffer configurations, this one analyzes how variations in URLLC buffer sizes influence system availability. Figs. 7a and 7b present the NFV-MEC node availability for eMBB and URLLC services, respectively, considering different configurations in terms of the number of containers (c) and URLLC buffer size (k). These results resemble the previous scenario; however, the curves show an inverted order in Fig. 7a, with larger buffers corresponding to lower availability. This inversion is particularly noticeable in configurations with 12 containers (orange and red curves). The curve with the smaller URLLC buffer (red, $k = 16$) achieves higher eMBB availability than the orange one ($k = 24$). This occurs because adopting a larger URLLC buffer increases the system's capacity to admit URLLC service requests, leading to more resource usage by this service, especially as the URLLC load increases. In

contrast, in Fig. 4b, the configuration with the largest URLLC buffer($k = 24$, orange) presents higher availability than the red one ($k = 16$). This is expected, as having more URLLC buffer positions, even though they are passive elements, allows a greater number of URLLC services to be admitted into the system.

Expanding the URLLC buffer enhances the system's ability to accommodate URLLC requests but also increases resource competition, potentially affecting eMBB service availability. This trade-off underscores the need for service differentiation strategies in MEC. For instance, if a MEC system prioritizes URLLC applications such as autonomous driving or industrial automation, larger buffer sizes may be justified despite their impact on eMBB availability. Furthermore, these findings provide valuable insights for capacity planning and resource orchestration in NFV-MEC deployments. Operators should dynamically adjust buffer sizes to ensure both service types achieve acceptable performance levels while preventing resource starvation for eMBB applications.



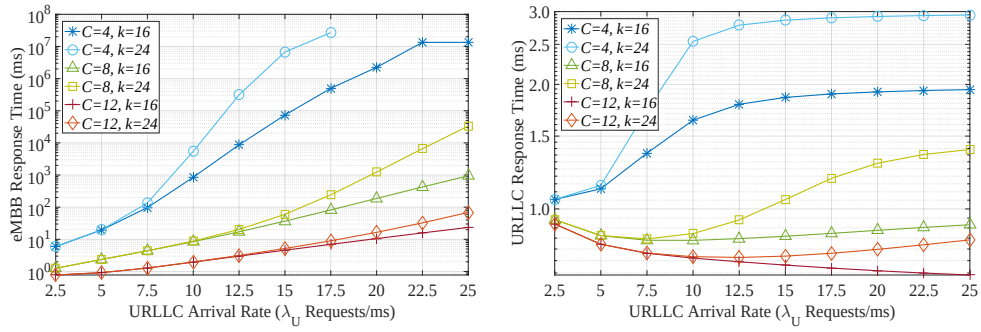
(a) eMBB Availability (b) URLLC Availability

Fig. 7 Effects of the number of containers (c) and URLLC buffer sizer (k) on availability

Regarding the response time, a larger buffer size negatively impacts the response time of both service types, as shown in Figs. 8a and 8b. However, this impact can be mitigated by increasing the number of containers in the NFV-MEC system. In Fig. 8a, for all configurations, the eMBB response time remains below 10 ms when λ_U is 2.5, even though the curves exhibit different growth patterns. For instance, configurations with 4 containers show a more pronounced increase in response time compared to those with 12 containers. Therefore, a more effective solution to improve eMBB response time is to adopt more containers, especially when the system is under high URLLC loads.

Analyzing the configurations, we observe that NFV-MEC nodes with 8 and 12 containers can support virtual and augmented reality applications, which demand a maximum latency of 8 ms [28], even under high λ_U values, such as 10. In contrast, setups with $c = 4$ can only accommodate these services when the URLLC load is low, such as at $\lambda_U = 2.5$. In general, adopting larger URLLC buffers degrades eMBB response time, as evidenced by comparing the curves with $k = 24$ and $k = 16$. Additionally, the setup with $c = 4$ and $k = 24$ (light blue curve) results in significantly

longer eMBB latency due to the combination of high URLLC load ($\lambda_U = 17, 5$) and higher admissibility of URLLC services.



(a) eMBB Response Time

(b) URLLC Response Time

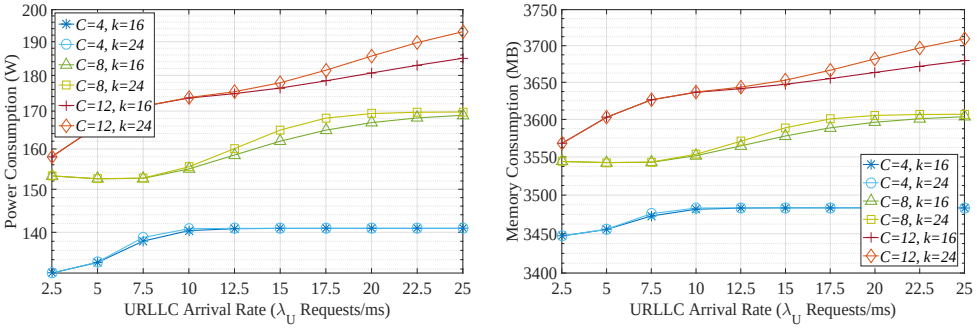
Fig. 8 Effects of the number of containers (c) and URLLC buffer sizer (k) on Response Time

Due to service prioritization, the URLLC response time range is narrower and presents lower values compared to eMBB, as shown in Fig. 8b. However, the curves (configurations) exhibit different behaviors, with some being strictly ascending while others go through both ascending and descending stages. Despite these variations, the order of the curves remains consistent with the previous experiment (Fig. 8a). Similar behavior is expected with minor shifts for λ_U values beyond those tested.

For configurations with 4 containers (light and dark blue curves), it is likely that their capacity is reached more quickly than with other configurations, leading to longer latency as the buffer becomes utilized. However, even for these configurations, the URLLC response time remains stable and at an acceptable level of 3 ms, which is reasonable for supporting fabric automation, which requires a maximum latency of 10 ms [27]. For setup with $c = 12$ and $k = 16$ (red curve), we observe a strictly decent behavior of the response time as the URLLC load increases. This reduction in response time may be attributed to the prompt handling of arriving URLLC services that encounter containers ready for processing, as they have completed the configuration stage. This minimizes the waiting time of these services in the buffer. Additionally, with a smaller buffer size ($k = 16$), fewer service requests wait in line for processing, which lowers URLLC response time and explains the difference compared to the configuration with $k = 24$ (orange graph).

In terms of power consumption (see Fig. 9a), a similar pattern to the previous scenario (Section 4.1) is observed: a higher number of containers results in increased power consumption, while the URLLC buffer size generally has minimal impact on power consumption. This behavior is particularly evident under low URLLC loads, where pairs of curves overlap. Since service requests in the queue contribute negligible power consumption, larger buffers show only minor variations in power usage. However, this variation becomes more noticeable as the system approaches its total capacity, leading to containers being continuously active and thus spending longer periods in the highest power consumption state.

Memory consumption is directly impacted by the provision of more processing resources, as we can see in Fig. 9b. On the right side of the graph, when the highest service loads in the system are evaluated, an increase to 8 containers for the scenarios with $c = 4$ (blue and light blue curves) results in memory consumption up to 123 MB higher.



(a) Power Consumption (b) Memory Consumption

Fig. 9 Effects of the number of containers (c) and URLLC buffer sizer (k) on Resource Consumption

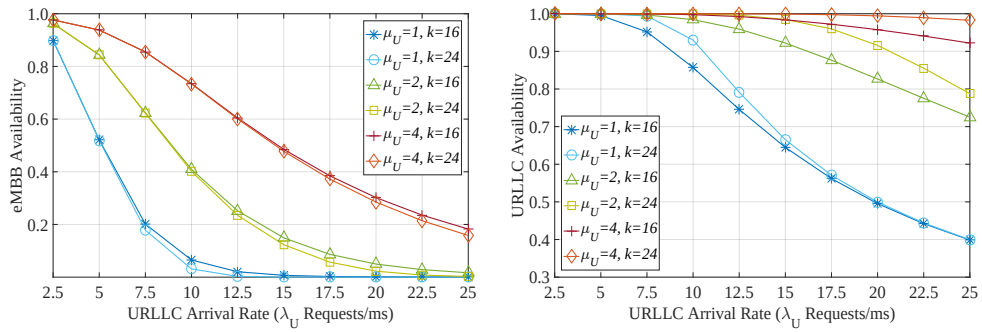
4.3 Impact of URLLC service rate (μ_U) and URLLC buffer size (k)

Expanding on the previous scenarios, this experiment shifts the focus to URLLC-specific parameters. Here, we analyze how variations in the URLLC service rate (μ_U) and buffer size (k) influence the NFV-MEC system performance, particularly in terms of resource availability and service prioritization. As observed in Fig. 10a, a higher URLLC service rate leads to higher eMBB availability. This is because faster URLLC service processing frees up computing resources more quickly, making them available for other services, which indirectly contributes to improved eMBB availability. This effect is particularly evident when the URLLC arrival rate is 12.5 requests/ms, and the URLLC buffer size (k) is 24, where eMBB availability increases by 43% when the URLLC service rate doubles from 2 to 4 services/ms. It is also noted that increasing the URLLC buffer size negatively impacts eMBB availability, especially at high URLLC arrival rates. A larger URLLC buffer size means that more service requests will wait in the queue to be processed, increasing the time URLLC services dominate the processing resources and prolonging the wait time for eMBB services. moreover, for very high URLLC arrival rates, eMBB availability tends to zero due to the prioritization favoring URLLC services during resource scheduling.

As expected, a higher URLLC service rate combined with a larger URLLC buffer results in increased URLLC availability, as shown in Fig. 10b. For instance, improving the URLLC service rate from 1 to 2 requests/ms while maintaining a buffer size of 24 results in a 41% boost in URLLC availability, similar to the increase achieved by doubling the number of available containers from 4 to 8, as previously depicted in Fig. 7b. Conversely, while higher URLLC service rates and larger buffer sizes both

contribute to improved availability, the effect is more pronounced in configurations with higher service rates. Comparing URLLC service rates of 2 requests/ms ($\mu_U = 2$) in Fig. 10b with configurations having 8 containers in Fig. 7b, we have that increasing the URLLC buffer from 16 to 24 yields a 9% improvement in availability in the former case, compared to a 4% improvement in the latter.

Faster service rates for URLLC improve its performance while also reducing contention for shared resources, indirectly benefiting eMBB applications. This denotes that investing in high-performance virtualization infrastructures, such as accelerated computing platforms with specialized hardware (e.g., Field-Programmable Gate Arrays or Data Processing Units), enables network operators to build a more efficient and adaptable multi-service ecosystem that meets the stringent requirements of 5G applications.



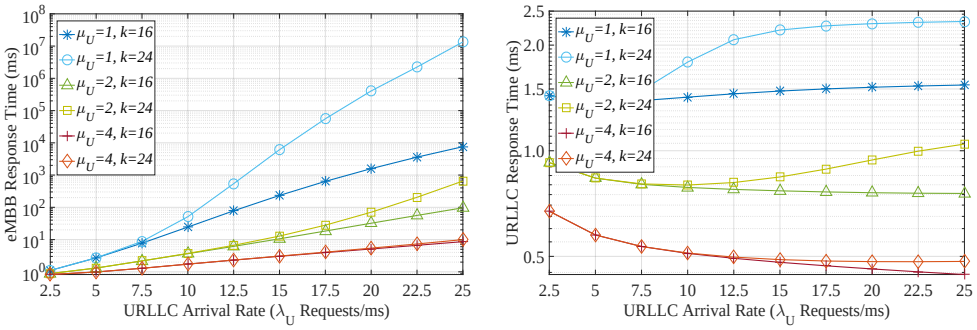
(a) eMBB Availability (b) URLLC Availability

Fig. 10 Effects of the URLLC service rate (μ_U) and URLLC buffer sizer (k) on availability

Similarly to eMBB availability, the eMBB response time is also indirectly affected by adopting higher URLLC service rates, as depicted in Fig. 11a. Although the processing time of eMBB services is not affected by higher URLLC service rates, experience shorter waiting times in the buffer before accessing the containers, which decreases the overall eMBB response time. However, larger URLLC buffer sizes mean that a greater number of URLLC requests occupy the processing resources, reducing the time available for eMBB requests. This increased demand for resources by URLLC services leads to longer queueing delays for eMBB requests, consequently increasing their response time. For a URLLC arrival rate of 20 requests/ms and buffer size of 16, results show that doubling the URLLC service rate from 1 to 2 reduces the eMBB response time from 1.58s to 32ms, thereby supporting eMBB applications such as Fixed Wireless Access (FWA) and 8K Video Streaming [29]

Analyzing the URLLC response time in Fig. 11b, it is observed that increasing URLLC service rates improves URLLC performance, similar to the improvements seen when the number of containers was boosted in the previous scenario (Section 4.2). Additionally, larger buffer sizes lead to longer latency for configurations with the same URLLC service rate, but this effect is mitigated by higher URLLC service rates. Comparing configurations with 12 containers in Fig. 8b (right part) and the best ones

in Fig. 11b, we observe a 21.6% reduction in response time in the former when an 8-position larger buffer is adopted, while only a 9% reduction is achieved in the latter with the same improvement. This demonstrates that it may be preferable for system operators to invest in improving the VNF processing speed for URLLC users (e.g., through software optimization) if the cost of adding new containers is similar.



(a) eMBB Response Time

(b) URLLC Response Time

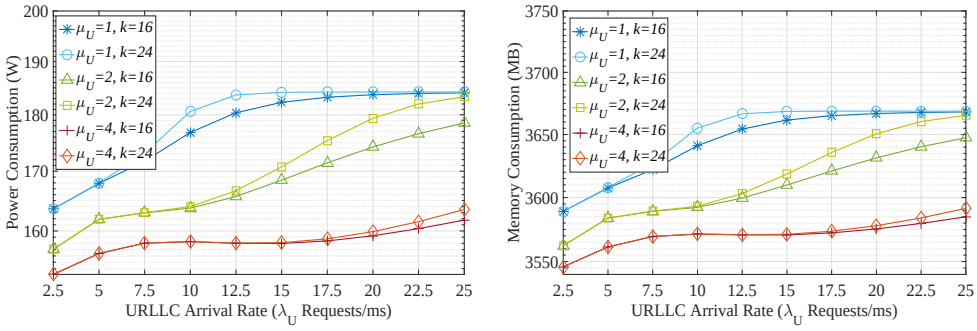
Fig. 11 Effects of the URLLC service rate (μ_U) and URLLC buffer sizer (k) on Response Time

Regarding power consumption, as shown in Fig. 12a, higher URLLC service rates are associated with lower power costs across all URLLC loads. However, for configurations with the same URLLC service rate, larger URLLC buffers consume more power. This is because the increased system availability provided by larger buffers results in longer periods where containers remain in the processing state, which has higher energy consumption. Unlike in Section 4.2, where adding more containers leads to higher power consumption, boosting URLLC service rates actually reduces consumption. This is evident in Fig. 9a, where, with a URLLC buffer size of 24, a URLLC request arrival rate of 22.5 requests/ms, and an increase in the number of containers from 8 to 12, power consumption rises by 11.8%. In contrast, in Fig. 12a, doubling the URLLC service rate from 2 to 4 while keeping the same arrival rate and buffer size reduces power consumption by 11.3%, while also offering better URLLC service.

As with energy consumption, higher URLLC service rates result in lower memory consumption, while larger buffer sizes for this service category result in a small increase in this consumption, as illustrated in Fig. 12b. When comparing configurations with a URLLC service rate equal to 4 requests/ms (red and orange curves) with configurations with a rate equal to 1 request/ms (blue and light blue curves), we can observe a memory consumption of approximately 2.7% lower when the URLLC arrival rate is 15 requests/ms.

4.4 Impact of eMBB service rate (μ_E) and eMBB buffer size (K)

Following the analysis of the URLLC service rate and buffer size, this last experiment examined how similar factors applied to eMBB services influence the NFV-MEC



(a) Power Consumption (b) Memory Consumption

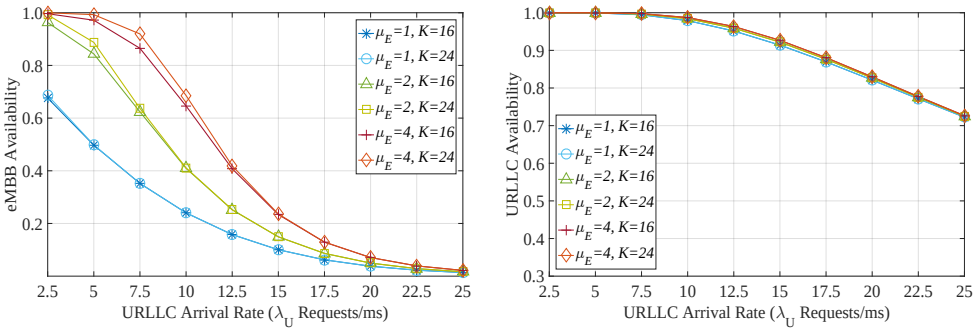
Fig. 12 Effects of the URLLC service rate (μ_U) and URLLC buffer sizer (k) on Resource Consumption

system performance. This experiment explores the effects of varying the eMBB service rate (μ_E) and buffer size (K) on eMBB availability, considering the interactions between URLLC and eMBB traffic. The impact of varying the eMBB service rate along with the eMBB buffer size on eMBB availability is illustrated in Fig. 13a. Results show that higher eMBB service rates provide better eMBB availability, especially when the system's eMBB load (10 requests/ms) is greater than the URLLC's. However, as the URLLC load increases, the availability of all configurations tends to zero, as the improvements achieved by larger eMBB buffers are essentially canceled out by URLLC service prioritization. This contrasts with the results in Fig. 10a, where an improved URLLC service rate brings noticeable differences, particularly under high URLLC loads.

It is also observed that larger eMBB buffer sizes only increase the eMBB availability when the URLLC arrival rate is lower than the eMBB arrival one. For higher URLLC arrival rates, availability is conditioned only by the remaining throughput of the available containers. Comparing the system improvement approaches presented in scenario 4.3 with those of this experiment, we can observe that within the adopted prioritization policy, improving the URLLC processing resources results in better eMBB availability results than those obtained by improving the resources for this service category. This behavior can be observed mainly when comparing the curves with a URLLC service rate of 4 requests/ms ($\mu_U = 4$) from Fig. 10a with the curves from this experiment with an eMBB service rate of 4 requests/ms ($\mu_E = 4$), in which the curves with URLLC improvements present an average availability 2% higher than the average availability with eMBB improvements.

The final set of results in Fig. 13b illustrates the impact of varying the eMBB service rate along with different eMBB buffer sizes on URLLC availability. Owing to the service prioritization policy, variations in the eMBB service rate have negligible impact on URLLC availability, with the difference between the best configuration (eMBB service rate of 4 requests/ms, $\mu_E = 4$) and the worst ($\mu_E = 1$) not exceeding 1.5% availability. Additionally, it is evident that the eMBB buffer size does not influence URLLC availability, regardless of the eMBB service rate used. The minor variation observed in URLLC availability is primarily due to the time required for eMBB requests to release processing resources currently in use. Once released, the

number of pending eMBB requests in the queue becomes irrelevant, as a new container to handle eMBB requests is provisioned only when all URLLC requests have been allocated and there are available processing resources. Consequently, the eMBB buffer size has no significant effect on URLLC availability.



(a) eMBB Availability (b) URLLC Availability

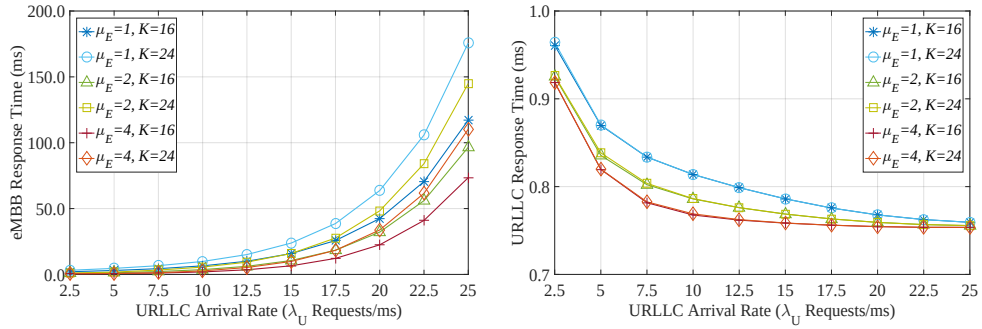
Fig. 13 Effects of the eMBB service rate (μ_E) and eMBB buffer sizer (K) on availability

The results from obtained from the service rate and eMBB buffer size variations are presented in Fig. 14a. Higher eMBB service rates considerably improve response times, even for configurations with larger buffer sizes for this service category. This improvement is mainly evidenced when the URLLC arrival rates reach their maximum value calculated in the experiment (25 requests/ms), in which the environment configuration with an eMBB service rate of 4 requests/ms ($\mu_E = 4$) and eMBB buffer size 16 presents a response time 24% lower than the configuration with $\mu_E = 2$ and eMBB buffer of the same size, reaching an average response time of 73.48 ms, which is capable of serving a good part of the applications consumed by eMBB users. Even so, these improvements have lower eMBB response time results than the results presented in the experiment 4.3, which in its configuration with a URLLC service rate of 4 requests/ms with the same buffer settings presented a reduction of 86.44% in eMBB response time reaching values of 9.96 ms.

The impact of varying the eMBB service rate and buffer size on URLLC response time is depicted in Fig. 14b. It was observed that higher eMBB service rates lead to a reduction in URLLC response time, a trend that is particularly pronounced at the start of the curves when the demand for eMBB services exceeds that of URLLC. This is because, as eMBB services occupy processing resources for a shorter duration, URLLC services spend less time waiting for the container to become available. Additionally, the eMBB buffer size appears to have no effect on URLLC response times. Despite the variations observed, all experiments resulted in URLLC response times below 1 ms.

In applications such as smart cities or industrial automation, eMBB and URLLC services must coexist efficiently. For example, in autonomous vehicle control (URLLC) and video streaming (eMBB), the system must prioritize low-latency communication for critical functions while maintaining acceptable performance for bandwidth-intensive applications like video streaming or cloud computing. Higher eMBB service rates help reduce response times, which directly benefits applications requiring quick

data access. Moreover, eMBB buffer sizes should be carefully adjusted based on traffic type and system load. While larger buffers may accommodate more eMBB traffic, they can negatively affect URLLC services by increasing the wait time for critical applications. As demonstrated in the results, systems should be designed to avoid excessive queuing and ensure that URLLC services retain priority access to resources.



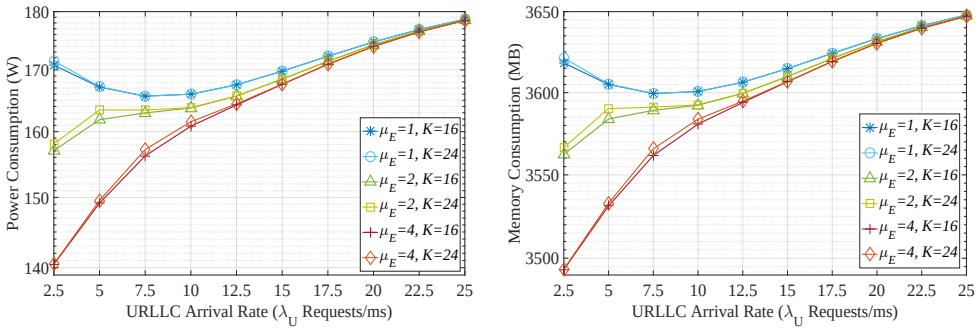
(a) eMBB Response Time (b) URLLC Response Time

Fig. 14 Effects of the eMBB service rate (μ_E) and eMBB buffer sizer (K) on Response Time

The average energy consumption of multiple node configurations with vary the eMBB service rate and buffer positions is presented in Fig. 15a. It is evident that during the initial segments of the curves, when the URLLC arrival rate has not yet surpassed the eMBB arrival rate ($\lambda_E = 10$), a higher eMBB service rate results in lower energy consumption. This is particularly noticeable when comparing the configuration with an eMBB service rate of two requests/ms ($\mu_E = 2$) and an eMBB buffer size of 16 with the configuration using $\mu_E = 4$ and the same buffer size, at the point where the URLLC arrival rate equals 2.5. In this case, the configuration with the higher eMBB service rate demonstrates a 10.57% reduction in energy consumption. A larger eMBB buffer size implies that more users will queue to be served, thereby reducing the time during which containers are turned off or in setup. Additionally, at the initial segments of the curves, two distinct behaviors can be observed: an increase for $\mu_E > 1$ and a slight decrease when $\mu_E = 1$. The decreasing behavior occurs because, within this range of URLLC arrival rates, processing resources are predominantly allocated for extended periods to serving eMBB requests. Consequently, with each new URLLC service arrival, a processing resource must be reinitialized to accommodate the request, resulting in longer setup times (which have lower energy consumption) and shorter processing times (with higher energy consumption). Although improvements in the eMBB service rate may lead to a reduction in energy consumption, the results of Experiment 4.3 revealed an average energy consumption of 3.41% lower for configurations with higher URLLC service rates.

Regarding the impacts of the variation in the eMBB service rate associated with improvements in the buffer size of this service category on memory consumption, we can observe in Fig. 15b that the same reasons that generate variations in energy consumption also generate similar variations in memory consumption. For low URLLC service loads (left side of the graph), higher eMBB service rates imply lower memory

consumption, reaching a minimum value of approximately 3.5 GB. As the URLLC demand increases, the curves tend to converge due to the dominance of this service category in the system, reaching a maximum common consumption of 3.64 GB in the highest service load evaluated in the experiment.



(a) Power Consumption (b) Memory Consumption

Fig. 15 Effects of the eMBB service rate (μ_E) and eMBB buffer sizer (K) on Resource Consumption

5 Conclusion

This work proposed a CTMC-based framework for the virtualization layer of a NFV-MEC node designed to address resource allocation challenges in the context of 5G/6G networks. Focusing on the coexistence of eMBB/IC and URLLC/HRLLC services, we explored the impact of various system parameters, such as the number of containers, buffer sizes, and service prioritization, on key performance metrics, including availability, response time, power consumption, and memory usage. In addition, often overlooked factors such as virtual resource setup delays, failures, and computational power degradation were also taken into account, highlighting their critical impact on communication constraints.

Our findings indicate that increasing the number of computational resources enhances service availability and reduces response times. However, buffer size can negatively impact response times when it varies with the amount of available resources. Conversely, improving service rates for the URLLC category leads to better response times and availability for both service categories, compared to solutions that focus on increasing computational resources. Energy and memory consumption exhibit similar patterns due to the similar nature of resource use during setup and active processing states. While adding more computational resources improves performance, it also directly increases resource consumption. In contrast, improving the URLLC service rate results in lower resource consumption. Therefore, if implementation costs for both approaches are similar, it is economically more advantageous for operators to enhance URLLC service rates in production. Furthermore, the analysis highlights the critical trade-offs between resource allocation, service prioritization, and energy efficiency,

showing the importance of optimizing system configurations to balance these conflicting demands, particularly in scenarios involving high service loads and stringent latency requirements.

For future work, we plan to extend the framework by incorporating additional network components, such as RAN elements, and exploring the impact of real-world factors like mobility and varying service demands. This could further refine the performance metrics and provide deeper insights into the holistic operation of NFV-MEC systems in next-generation networks. Finally, another direction involves conducting a comparative numerical analysis with existing models in the literature to emphasize their distinct characteristics, highlighting the semantic differences between our model, previous approaches, and real systems.

Declarations

Funding. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001 and by the UFPE/ Propesqi via Edital Nº 06/2024.

Author Contribution. All authors were actively involved in the conception and design of the study. Caio Souza took the lead in the overall development of the article, overseeing the study design, conducting the research, analyzing the data, and writing the manuscript. Working in close collaboration with him, Marcos Falcão contributed to the formulation of the methodology, participated in data collection and analysis, and co-authored specific sections of the text. Maria Damasceno and Renata Reis played a key role in shaping the theoretical foundation of the work, conducting a comprehensive literature review, and ensuring the study was grounded in relevant prior research with a clear and well-structured scientific argument. Providing overarching guidance, Andson Balieiro contributed conceptual input and strategic direction throughout the project while critically reviewing, editing, and co-writing significant parts of the manuscript, particularly the introduction, discussion, and conclusion.

Conflict of interest. The authors declare no competing interests.

References

[1] Al-Ali, M., Yaacoub, E.: Resource allocation scheme for embb and urllc coexistence in 6g networks. Wireless Networks. **29**(6), 2519–2538 (2023) https://doi.org/10.1007/s11276-023-03328-2

[2] Liu, R., Lin, H., Lee, H., Chaves, F., Lim, H., Sköld, J.: Beginning of the journey toward 6g: Vision and framework. IEEE Communications Magazine **61**(10), 8–9 (2023) https://doi.org/10.1109/MCOM.2023.10298069

[3] Bolettieri, S., Bui, D.T., Bruno, R.: Towards end-to-end application slicing in multi-access edge computing systems: Architecture discussion and proof-of-concept. Future Generation Computer Systems **136**, 110–127 (2022) https://doi.org/10.1016/j.future.2022.05.027

[4] Setayesh, M., Bahrami, S., Wong, V.W.S.: Resource slicing for embb and urllc services in radio access network using hierarchical deep learning. IEEE Transactions on Wireless Communications **21**(11), 8950–8966 (2022) https://doi.org/10.1109/TWC.2022.3171264

[5] Al-Ali, M., Yaacoub, E., Mohamed, A.: Dynamic resource allocation of embb-urllc traffic in 5g new radio. In: 2020 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), pp. 1–6 (2020). https://doi.org/10.1109/ANTS50601.2020.9342844

[6] Bairagi, A.K., Munir, M.S., Alsenwi, M., Tran, N.H., Alshamrani, S.S., Masud, M., Han, Z., Hong, C.S.: Coexistence mechanism between embb and urllc in 5g wireless networks. IEEE Transactions on Communications **69**(3), 1736–1749 (2021) https://doi.org/10.1109/TCOMM.2020.3040307

[7] Kim, Y., Park, S.: Calculation method of spectrum requirement for imt-2020 embb and urllc with puncturing based on m/g/1 priority queuing model. IEEE Access **8**, 25027–25040 (2020) https://doi.org/10.1109/ACCESS.2020.2971223

[8] Prathyusha, Y., Sheu, T.-L.: Coordinated resource allocations for embb and urllc in 5g communication networks. IEEE Transactions on Vehicular Technology **71**(8), 8717–8728 (2022) https://doi.org/10.1109/TVT.2022.3176018

[9] Li, W., Jin, S.: Performance evaluation and optimization of a task offloading strategy on the mobile edge computing with edge heterogeneity. The Journal of Supercomputing **77**(8) (2021) https://doi.org/10.1007/s11227-021-03781-w

[10] Tong, Z., Zhang, T., Zhu, Y., Huang, R.: Communication and computation resource allocation for end-to-end slicing in mobile networks. 2020 IEEE/CIC International Conference on Communications in China (ICCC), 1286–1291 (2020) https://doi.org/10.1109/ICCC49849.2020.9238794

[11] Yala, L., Frangoudis, P.A., Ksentini, A.: Latency and availability driven vnf placement in a mec-nfv environment. In: 2018 IEEE Global Communications Conference (GLOBECOM), pp. 1–7 (2018). https://doi.org/10.1109/GLOCOM.2018.8647858

[12] Falcao, M., Souza, C., Balieiro, A., Dias, K.: An analytical framework for urllc in hybrid mec environments. The Journal of Supercomputing **78** (2022) https://doi.org/10.1007/s11227-021-03945-8

[13] Souza, C., Falcao, M., Balieiro, A., Dias, K.: Modelling and analysis of 5g networks based on mec-nfv for urllc services. IEEE Latin America Transactions **19**(10), 1745–1753 (2021) https://doi.org/10.1109/TLA.2021.9477275

[14] Imoize, A.L., Adedeji, O., Tandiya, N., Shetty, S.: 6g enabled smart infrastructure for sustainable society: Opportunities, challenges, and research roadmap. Sensors

21(5) (2021) https://doi.org/10.3390/s21051709

[15] Joshi, B.: Breaking the energy curve: 5G energy efficiency - ericsson. Ericsson (2019). https://www.ericsson.com/en/blog/2019/2/breaking-the-energy-curve-5g-energy-efficiency

[16] Abrol, A., Jha, R.K.: Power optimization in 5g networks: A step towards green communication. IEEE Access **4**, 1355–1374 (2016) https://doi.org/10.1109/ACCESS.2016.2549641

[17] Nguyen, K., Simonovski, F., Loh, F., Hossfeld, T., Thanh, N.H.: Investigation of container network function deployment costs in the edge cloud. 2024 27th Conference on Innovation in Clouds, Internet and Networks (ICIN), 9–16 (2024)

[18] Liu, Q., Luo, R., Liang, H., Liu, Q.: Energy-efficient joint computation offloading and resource allocation strategy for isac-aided 6g v2x networks. IEEE Transactions on Green Communications and Networking **7**(1), 413–423 (2023) https://doi.org/10.1109/TGCN.2023.3234263

[19] Cao, H., Du, J., Zhao, H., Luo, D.X., Kumar, N., Yang, L., Yu, F.R.: Resourceability assisted service function chain embedding and scheduling for 6g networks with virtualization. IEEE Transactions on Vehicular Technology **70**(4), 3846–3859 (2021) https://doi.org/10.1109/TVT.2021.3065967

[20] Yue, Y., Tang, X., Yang, W., Zhang, X., Zhang, Z., Gao, C., Xu, L.: Delay-aware and resource-efficient vnf placement in 6g non-terrestrial networks. In: 2023 IEEE Wireless Communications and Networking Conference (WCNC), pp. 1–6 (2023). https://doi.org/10.1109/WCNC55385.2023.10118893

[21] Souza, C., Falcao, M., Damasceno, M., Reis, R., Balieiro, A.: Aprovisionamento de recursos para serviços urllc e embb em redes mec-nfv: Uma análise baseada em ctmc. In: Anais do XLII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, pp. 714–727. SBC, Porto Alegre, RS, Brasil (2024). https://doi.org/10.5753/sbrc.2024.1465 . https://sol.sbc.org.br/index.php/sbrc/article/view/29830

[22] Kekki, S., Featherstone, W.: Mec in 5g networks. ETSI White Paper (28), 1–28 (2018)

[23] Shahidinejad, A., Ghobaei-Arani, M., Esmaeili, L.: An elastic controller using colored petri nets in cloud computing environment. Cluster Computing, 1–27 (2019)

[24] Kaur, K., Dhand, T., Kumar, N., Zeadally, S.: Container-as-a-service at the edge: Trade-off between energy efficiency and service availability at fog nano data centers. IEEE Wireless Communications **24**(3), 48–56 (2017) https://doi.org/10.1109/MWC.2017.1600427

[25] Singh, K., Xie, M.: Bootstrap: A Statistical Method (2008). http://www.stat.rutgers.edu/home/mxie/rcpapers/bootstrap.pdf Accessed 2015-06-01

[26] Stallings, W.: 5G Wireless: A Comprehensive Introduction. Addison-Wesley, Boston, MA 02116, US (2021). https://books.google.fi/books?id=-V4OzgEACAAJ

[27] Siddiqui, M.U.A., Abumarshoud, H., Bariah, L., Muhaidat, S., Imran, M.A., Mohjazi, L.: Urllc in beyond 5g and 6g networks: An interference management perspective. IEEE Access **11**, 54639–54663 (2023) https://doi.org/10.1109/ACCESS.2023.3282363

[28] Raca, D., Leahy, D., Sreenan, C.J., Quinlan, J.J.: Beyond throughput, the next generation: a 5g dataset with channel and context metrics. In: Proceedings of the 11th ACM Multimedia Systems Conference. MMSys '20, pp. 303–308. Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3339825.3394938 . https://doi.org/10.1145/3339825.3394938

[29] Sugito, Y., Iwasaki, S., Chida, K., Iguchi, K., Kanda, K., Lei, X., Miyoshi, H., Kazui, K.: Video bit-rate requirements for 8k 120-hz hevc/h.265 temporal scalable coding: experimental study based on 8k subjective evaluations. APSIPA Transactions on Signal and Information Processing **9**, 5 (2020) https://doi.org/10.1017/ATSIP.2020.4