Enabling the eMBB and URLLC coexistence in MEC-NFV Networks

Caio Souza [*†], Marcos Falcão[*], Andson Balieiro[*], Tarik Taleb[‡§], Elton Alves[¶]
[*]*Centro de Informática (CIn), Universidade Federal de Pernambuco (UFPE), Recife, Brazil*
[†]*Sidia Institute of Science and Technology , Manaus, Brazil.*
[‡]*Centre for Wireless Communications (CWC), University of Oulu, Oulu, Finland.*
[§]*Ruhr University Bochum, Bochum, Germany.*
[¶]*Faculdade de Engenharia da Computação, Universidade do Sul e Sudeste do Pará, Brazil.*
caio.souza@sidia.com, {amb4,mrmf}@cin.ufpe.br
tarik.taleb@oulu.fi, tarik.taleb@rub.de, eltonalves@unifesspa.edu.br

***Abstract*—The coexistence between enhanced Mobile Broadband (eMBB) and Ultra Reliable Low Latency Communications (URLLC) is challenging in modern communication systems. To support such diversity, Multi-access Edge Computing (MEC) and Network Function Virtualization (NFV) emerge as complementary paradigms that shall offer fine-grained on-demand distributed resources closer to the User Equipment (UE). In this work, we address the combination of MEC, NFV and dynamic virtual resource allocation to overcome the challenge of resource dimensioning in the network edge. A Continuous Time Markov Chain (CTMC) based model was designed to evaluate how requests are managed by the virtualization resources of a single MEC node, with a primary focus on meeting the requirements of both eMBB and URLLC services. Practical factors such as resource failures, service prioritization, and setup (repair) times were integrated into the formulation. Some of our key findings include the idea that higher eMBB arrival rates decrease availability and increase response times, while URLLC availability remains stable, and that the container setup rates and failure rates substantially affect both availability and response times, with higher setup rates enhancing both availability and reducing response times.**

***Index Terms*—Multi-access Edge Computing, Ultra-Reliable Low-Latency Communications, Continuous-Time Markov Chain, Network Function Virtualization, Enhanced Mobile Broadband, Dynamic Resource Allocation**

I. Introduction

The synergy between MEC and NFV is key for advancing the challenge in the coexistence of URLLC and eMBB services, which lies in their divergent requirements, efficient resource allocation, interference minimization, and consistent performance [1]. In particular, while MEC empowers the hosting of virtualized network functions (VNFs) and applications closer to the end-users, reducing the latency and enhancing the overall reliability, NFV facilitates dynamic resource allocation, aligning network capacity with demand fluctuations, besides enabling content to be cached and processed at the network edge, further ensuring rapid response times [2]. While it enables a broad spectrum of use cases, the simultaneous operation of eMBB and URLLC introduces multiple challenges, especially regarding dynamic resource allocation within the MEC-NFV domain to balance the contrasting requirements of different use cases [3]. To enable the coexistence of eMBB and URLLC services, the concept of Network Slice is pivotal. It plays a fundamental role in enabling the shared utilization of physical infrastructure, allowing for the creation of multiple virtual networks.

Although multiple works have addressed the coexistence of different service types in 5G networks, the majority focus on radio resource allocation [3]- [5], leaving a gap on relevant resource provisioning factors. Notably, prior research often presupposes fault-free cloud environments [8] or with instantaneous provisioning times [7] which may not align with the remaining components of the 5G network. Furthermore, most studies often do not consider that other service subcategories may widely differ [9] and neglect, among others, the overhead caused by virtualization. For instance, in the VNF instance boot process, energy is consumed, and resources are allocated, yet services remain unattended, impacting, the response time and making critical services such as URLLC unfeasible.

This paper addresses the combination of MEC, NFV, and dynamic virtual resource allocation within the context of the URLLC and eMBB coexistence. We propose a Continuous Time Markov Chain (CTMC) based model to characterize the dynamic virtual resource allocation for both URLLC and eMBB services and analyze their availability and response times. In addition, practical aspects such as resource failures, service prioritization, and setup (repair) times have been incorporated into the model, as they can incur significant impacts on the 5G applications' requirements. Moreover, the MEC-NFV node model encompasses dynamic scaling capabilities and service prioritization to accommodate the two 5G service categories. Some of our key findings include the idea that higher eMBB arrival rates decrease availability and increase response times, while URLLC availability remains stable. Moreover, the container setup rates and failure rates substantially affect both availability and response times, with higher setup rates enhancing availability and reducing response times. Also, the number of containers emerges as a significant factor, enhancing both availability and response times, while buffer sizes mainly impact response times.

The remainder of this paper is organized as follows. Section II describes a CTMC-based model for a single node NFV-

MEC, assuming a virtual environment featured with containers that process both URLLC and eMBB requests. Section III presents the model validation and a result analysis obtained by extensive discrete-event simulations. Finally, Section IV provides concluding remarks and highlights future directions.

II. System Model

We assess the performance of a MEC node, where both eMBB and URLLC requests (packets) originated from UEs are processed by the RAN, passed to the MEC node and are handled by containerized VNFs, which are scaled accordingly. This model was designed in isolation from the RAN, Core, and Central Cloud, hence, the only uncertainty is due to the virtual components themselves, i.e., setup, failure, and repair events. The system consists of a finite amount of containers and buffer positions that can be allocated to each type, with each VNF running equally and independently on a single container, and where a centralized control unit determines the request admission. An admission occurs if there are enough resources (available containers or buffer positions). If so, each request may be processed or queued.

In order to cope with sudden load variations, a dynamic VNF auto-scaling strategy was embedded into our formulation. Thus, before the proper processing stage, the containerized-VNF must be initialized, which incurs a delay (setup time). In addition, failures may take place during service and its respective repair time is also incorporated into our model. In this case, the containerized VNF is restarted, and the request is either reallocated to another available container or, if there are no available resources, it is placed back in its respective service queue with higher priority than new requests. In both cases, the service processing is restarted.

We also adopt service prioritization as follows: (1) if there are both URLLC and eMBB services to be served, URLLC services have higher priority, thus, the containers that are being released or activated are allocated first to URLLC services. (2) In the case where there is a URLLC service waiting in queue for available resources and an eMBB service has been completed, the released container is restarted to be used by the URLLC service. However, if there are other available containers, the current one will be allocated to a sequential eMBB service or deactivated if the eMBB queue is empty. (3) The preemption of the lower-priority service (eMBB) that is being processed is not allowed.

The system is modeled using an M/N/c/k+K queue with two user types, prioritization, failure, initialization time, FCFS service discipline, and a limited buffer for each user type. k and K represent the maximum number of URLLC and eMBB users in the system, respectively. The model states are denoted by the tuple (i, j, l, m), where $i, j, l, m \in N$, with i and j representing the number of URLLC and eMBB services, and l and m the number of active containers for each user category, with $l + m$ being smaller or equal to the maximum number of containers (c). The service arrivals follow a Poisson process with rate λ_u for URLLC services and λ_e for eMBB. The service processing is provided by the c available containers, with an exponentially distributed service time with rates μ_u for URLLC and μ_e for eMBB. Similarly, the failure occurrence and container initialization time follow exponential distributions with rates γ and α, respectively. Fig. 1 summarizes all possible CTMC transitions and states, with its respective parameters. By computing the steady-state probabilities (π) of the model, useful metrics may be derived to analyze the system performance as follows.

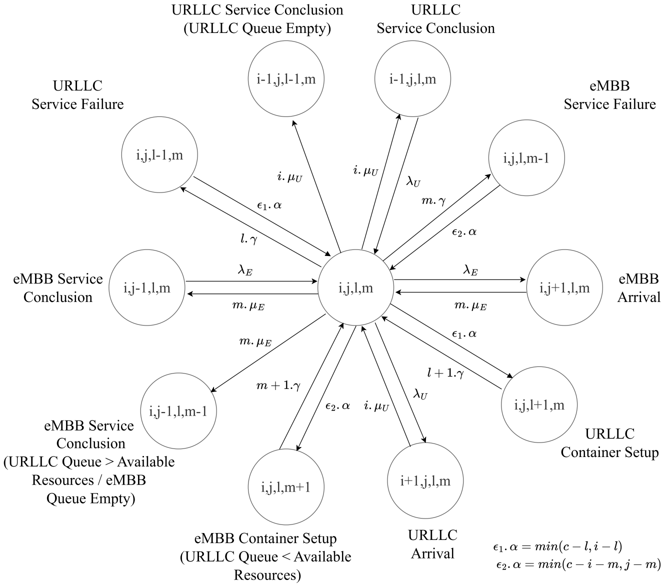


Fig. 1. Generic CTMC State Diagram

A. Availability

The MEC and NFV combination has been widely acknowledged for its potential to reduce latency and enhance reliability by placing virtualized network functions and applications closer to the UE. However, the limited resources of edge nodes impose constraints on their service capacity, which is typically known as availability. Consequently, when the maximum capacity is reached, two primary alternatives emerge: forward the exceeding flow to a neighboring MEC node or redirect it to the central cloud [10]. These alternatives involve establishing a new route comprising multiple intermediate hops, which can introduce significant uncertainty concerning latency and reliability. As a result, it is essential to analyze the availability of edge nodes. In our model, the MEC availability refers to the system's ability to offer the minimum amount of functional and accessible VNFs or buffer positions. In addition, due to the service prioritization, the MEC node availability is segmented in terms of each service type, i.e., URLLC (A_U) and eMBB (A_E) respectively, as in Eqs. 1 and 2, which are obtained by summing the probabilities of all states except those representing full capacity for each type of service.

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

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

B. Response Time

Response time assumes a crucial role in URLLC applications, although it is also relevant to the eMBB. Recognizing that the significance may vary depending on the service category, the response time for each category has been defined. It is defined as the interval between the arrival of the service (at the MEC-NFV node) and its conclusion, which includes any configuration/restarting times if these events occur. The Eqs. 5 denote response times for URLLC and eMBB services, which are calculated by dividing the average number of online services into each category as in Eqs. 3 and 4 and their respective admission rates in the MEC-NFV node.

$$\overline{U}_U = \sum_{i=0}^{k} \sum_{j=0}^{K} \sum_{l=0}^{min(c,i)} \sum_{m=0}^{min(c-l,j)} i\pi_{i,j,l,m} \quad (3)$$

$$\overline{U}_E = \sum_{i=0}^{k} \sum_{j=0}^{K} \sum_{l=0}^{min(c,i)} \sum_{m=0}^{min(c-l,j)} j\pi_{i,j,l,m} \quad (4)$$

$$T_U = \frac{\overline{U}_U}{\lambda_U A_U}; T_E = \frac{\overline{U}_E}{\lambda_E A_E} \quad (5)$$

III. Validation and Analysis

The analytical results (lines) were validated against discrete-event simulations (markers) (Figs. 2a-4d), using a Coloured Petri Nets-based simulator. The main parameters were set following a subset of the 3GPP Release 16 (TR 38.824) [11]. In the first scenario (Section III-A), the impact of each user type on each other by adopting multiple eMBB request rates (λ_E) was evaluated. Each subsequent scenario simultaneously assesses the influence of a pair of parameters, container setup rate (α) and failure rate (γ) in Section III-B, which may represent hardware and software improvements that reduce the time in which network functions are made available to process services, and the use of components with different reliability to provide the service, respectively. Finally, the variation of URLLC service rate (μ_U) and eMBB service rate (μ_E) in Section III-C, with aims at illustrating how enhancements in service request process speed (e.g., achieved through the utilization of advanced processing units and optimized algorithms) can positively impact the system's overall functionality. In all scenarios, the arrival of the URLLC services (λ_U) ranged from 2.5 to 25 requests/ms to analyze 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 [12]. The remaining parameters are in Table I. The results represent the average of each metric, considering 10 simulation instances, with 2700000 steps and 2200000 services attended each, and a confidence interval of 95%.

A. Effects of the eMBB load (λ_E)

This scenario evaluates the impacts of the eMBB service request arrival rate, which ranges from 5 up to 30 arrivals/ms. The resulting curves represent the adopted eMBB loads, where the blue curves (light and dark) correspond to small loads (5 and 10, respectively), green and yellow to medium loads (15 and 20, respectively), and red and orange to higher loads (25 and 30, respectively).

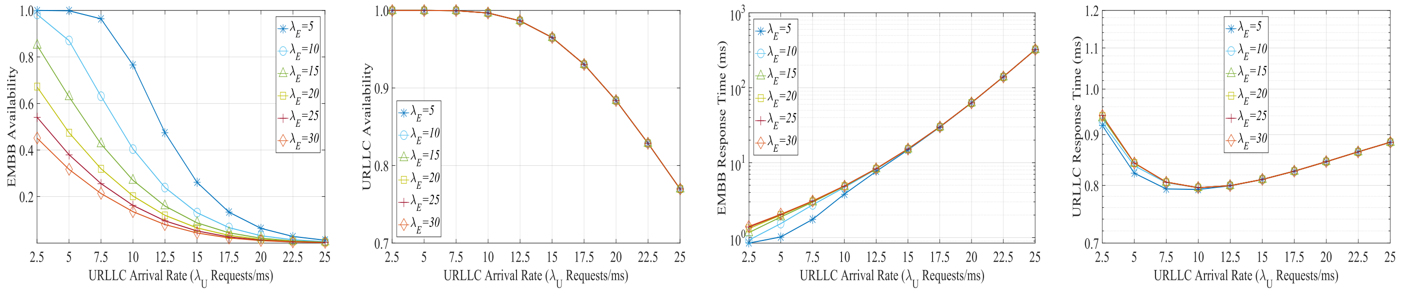
Figs. 2a-2b depict the Availability as strictly decreasing curves. Notably, the Availability for eMBB users (Fig. 2a) displays a greater disparity among the configurations, whereas the results for URLLC users (Fig. 2b) exhibit overlapping patterns. This aligns with our expectations, given that the URLLC category is accorded higher priority over eMBB, rendering the eMBB arrival rate (λ_E) inconsequential for the URLLC Availability. Conversely, in Fig. 2a, eMBB users contend for unoccupied containers. As the curves represent varying eMBB user loads, the overall eMBB Availability fluctuates, with higher values corresponding to curves indicating lower eMBB arrival rates (e.g., $\lambda_E = 5$ and $\lambda_E = 10$). Consequently, the curves in Fig. 2a have a more pronounced decline compared to those in Fig. 2b, as the former is influenced by both eMBB and URLLC arrival rates while the latter is solely influenced by the URLLC arrival rate. Moreover, the eMBB Availability (Fig. 2a) converges to zero at $\lambda_U = 22.5$, whereas the URLLC's (Fig. 2b) remains above 80% at the same point. These findings appear reasonable for the majority of future service categories, but are considered suboptimal for URLLC standards.

Regarding the Response Time (Figs. 2c-2d), significant disparities can be observed, starting with the employed scale. In Fig. 2c, the Response Time for eMBB users exhibits a wide range of values spanning from 1 ms up to 300 ms. In contrast, Fig. 2d depicts a considerably narrower range, with the Response Time for URLLC users ranging from 0.8 ms to 0.94 ms, these values indicate that across all load scenarios assessed in this configuration, the latency requirements for delivering all URLLC services listed in Table II are consistently met.

Despite these distinctions, the curves in both figures exhibit substantial overlap across the majority of the evaluated points, ultimately converging to the same final value. However, the key distinction lies in their respective behaviors. In Fig. 2c, the curves demonstrate a monotonically increasing trend, while Fig. 2d displays a sudden drop in the Response Time for URLLC users until $\lambda_U = 10$. Beyond this, all curves resume an upward trajectory, converging to 0.89 ms at $\lambda_U = 25$, which is lower than the initial value of approximately 0.94 ms at $\lambda_U = 2.5$. This unexpected behavior can be attributed to the container setup delay, during which requests await the container loading completion. Consequently, all curves experience a decrease in Response Time from $\lambda_U = 2.5$ to $\lambda_U = 10$, followed by a steady increase. However, the Response Time values do not reach the same levels as at $\lambda_U = 2.5$, as all containers have already been initialized. Additionally, in Fig. 2d, slight variations are observed in $\lambda_U = 2.5$ to $\lambda_U = 7.5$, attributed to the presence of eMBB users. These users also

TABLE I
EXPERIMENT SETS

Section	Varying Parameters	λ_E	α	γ	μ_U	μ_E	C	K	k
III-A	λ_E	5,10,15,20,25,30	1	10^{-3}	2	2	10	20	20
III-B	α, γ	10	1,2,4	10^{-2}, 10^{-3}	2	2	10	20	20
III-C	μ_U, μ_E	10	1	10^{-3}	1,2,4	1,2	10	20	20



(a) eMBB Availability (b) URLLC Availability (c) eMBB Response Time (d) URLLC Response Time

Fig. 2. Effects of the eMBB load (λ_E)

TABLE II
EXAMPLES OF EMBB AND URLLC APPLICATIONS.

Work	Use Case	Latency	Category
[1]	Smart Transportation	10 - 100 ms	URLLC
[1]	Robotics/Telepresence	1 ms	URLLC
[13]	FWA	4 ms	eMBB
[14]	8K Video Streaming	20 ms	eMBB

contribute to container (re)initializations, when an eMBB request is completed and immediately followed by a URLLC request, triggering a new initialization process, which explains the small differences among the curves in this interval.

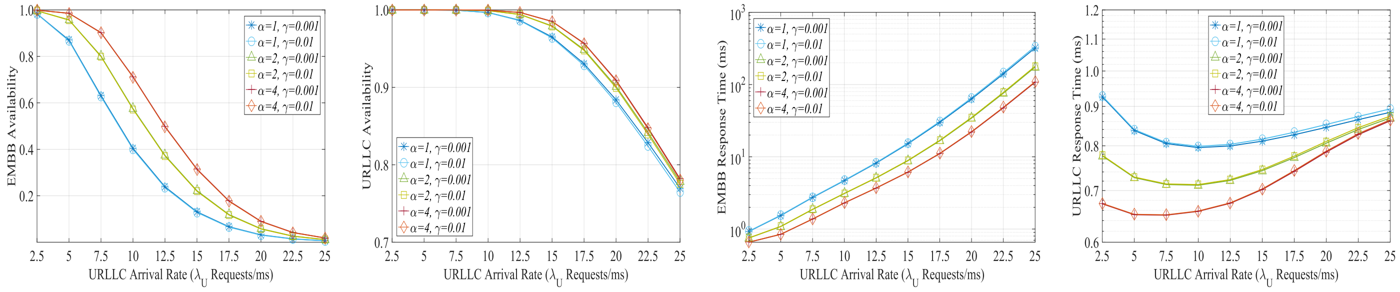
B. Effects of the container setup rate (α) and service failure rate (γ)

This section evaluates the impact of varying the container setup rate (α) in combination with changes in the service failure rate (γ). The availability of eMBB services (Fig. 3a), exhibited significant variations among the curves with different setup rates ($\alpha = 1$, $\alpha = 2$, and $\alpha = 4$), while overlapping with configurations having the same setup rate but different failure rates. Notably, the absolute differences in availability reached up to 30% for $\lambda_U = 10$ when comparing the $\alpha = 1$ (light and dark blue) and $\alpha = 4$ (red and orange) configurations. Higher container setup rates were observed to result in increased availability and reduced user waiting times in the buffer. Interestingly, the experiment revealed that even when the service failure rate was increased by a factor of ten, it did not significantly impact the availability for eMBB users, which can be attributed to the buffer's capacity to accommodate failed service requests. Moreover, consistent with the previous scenario, the availability for eMBB applications diminished rapidly across all configurations, in contrast to the URLLC availability shown in Fig. 3b, which experienced a comparatively smaller impact due to its higher priority.

Regarding the availability for URLLC users (Fig. 3b), it was observed that the container setup rate (α) had a relatively minor impact compared to the eMBB case. Specifically, the differences in availability among the curves with different α values were limited to approximately 2% at $\lambda_U = 15$, when comparing the $\alpha = 1$ (light and dark blue) and $\alpha = 4$ (red and orange) configurations. As for the impact of different failure rates, a more pronounced difference was noted when compared to the eMBB case in Fig.3a, where overlapping occurred. For the URLLC, container failures produced a slight difference among the curves with the same α, making it possible to distinguish between, for instance, the light and dark blue curves. In other words, the URLLC is significantly more sensitive to the failure rate than the eMBB.

When examining the eMBB Response Time (Fig. 3c), it becomes apparent that a higher container setup rate leads to a reduced response time, as expected. Initially, since there is little competition for resources between eMBB and URLLC users, the difference between the evaluated configurations is of a few milliseconds. However, as the URLLC request arrival rate intensifies, this disparity becomes more pronounced. The increasing URLLC arrival rate creates a higher demand for resources, and since it has a higher priority, the eMBB requests are interrupted, either restarting service in another container or waiting in the buffer for available resources, causing the eMBB response time to be more affected. In such cases, only configurations with a α of 4 have the capacity to handle high-resolution video streaming services, which demand a latency of under 20 milliseconds [14] when λ_U reaches 20 arrivals per millisecond. It was also noticeable that the failure rate had little impact in this experiment, which explains the pair of overlapped curves with the same values of α.

With regards to the Response Time of URLLC users (Fig. 3d), the container setup rate has a more pronounced impact compared to the previous scenario in Fig. 2c, where the only varying parameter was λ_E. This is particularly evident at the initial stages of the curves when containers are predominantly powered off or allocated to the eMBB users. During this period, the low arrival rate of URLLC services translates to



(a) eMBB Availability

(b) URLLC Availability

(c) eMBB Response Time

(d) URLLC Response Time

Fig. 3. Effects of the container setup rate (α) and service failure rate (γ)

shorter waiting times for a container to become available, reducing the overall response time. However, as the URLLC service arrival rate increases, this disparity diminishes, ultimately converging towards the end of the curves when the majority of containers are occupied by URLLC services.

Furthermore, it is noteworthy that a higher failure rate leads to an increase in the response time, since the failure occurrence becomes more frequent, especially for higher λ_U values, impacting the service time due to the need for container resets. However, similarly to the Availability in Fig. 3b, this remains relatively insignificant compared to the differences caused by altering the setup rate. This results in a more distinguishable difference among the pair of curves that were overlapping (e.g., light and dark blue). Finally, as the curves approach the system's capacity, a greater number of containers remain active to accommodate the incoming service requests, resulting in a temporary decline in the response time. Nevertheless, as resource competition intensifies within the URLLC service category, the response time gradually escalates once again and all curves tend to converge around 0.9 ms. At this point, all configurations remain capable of providing service to robotic and telepresence systems, which require a latency of 1 ms [1].

C. Effects of the URLLC service rate (μ_U) and eMBB service rate (μ_E)

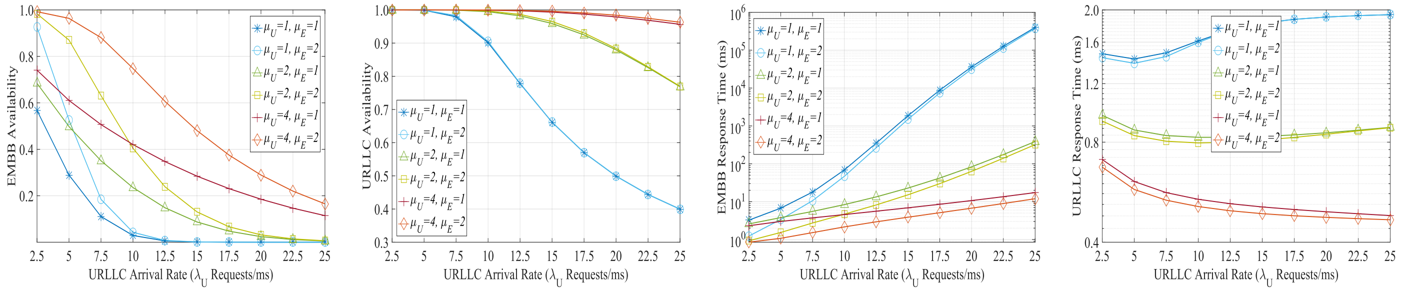
This section aims to assess the influence of different service rates on each user type, specifically the URLLC service rate (μ_U) and the eMBB service rate (μ_E). Fig. 4a illustrates that a higher eMBB service rate leads to increased availability for this service category, particularly in the leftmost region of the graph. For configurations with the same μ_U values, the curve with $\mu_E = 2$ exhibits higher availability compared to those with $\mu_E = 1$. For example, at $\lambda_U = 7.5$, the configuration with ($\mu_U = 2$, $\mu_E = 1$) demonstrates an availability of 38%, while its counterpart ($\mu_U = 2$, $\mu_E = 2$) exhibits 62%, representing a significant difference of 24%. However, this effect diminishes as the URLLC arrival rate increases, resulting in convergence at the rightmost part of the graph. Moreover, a higher URLLC service rate implies less time spent by these requests monopolizing the resources, leading to greater availability. This explains why configurations with $\mu_U = 1$ and $\mu_U = 4$ are shifted to the left and right, respectively, compared to the adopted baseline ($\mu_U = 2$).

From the perspective of URLLC user availability (Fig. 4b), it is observed that the eMBB service rate (μ_E) has an insignificant impact on this performance metric, resulting in overlapping curves. Conversely, higher URLLC service rates ($\mu_U = 2$ and $\mu_U = 4$) lead to greater availability as the requests are serviced more rapidly. For instance, at $\lambda_U = 20$, configurations with $\mu_U = 1$ (light and dark blue) exhibit an availability of approximately 50%, while those with $\mu_U = 2$ (green and yellow) achieve around 88%, i.e., a substantial difference of 48%.

Regarding the eMBB response time (Fig. 4c), a higher service rate for this category, represented by configurations where $\mu_E = 2$ (light blue, yellow, and orange lines), results in shorter response times compared to their respective counterparts with $\mu_E = 1$ (dark blue, green, and red lines). However, the performance difference between the two curves with $\mu_U = 1$ (light and dark blue) and the two curves with $\mu_U = 2$ (green and yellow) is minimal. Notably, the performance difference becomes more pronounced for configurations with $\mu_U = 4$ (red and orange lines). These configurations consistently maintain the eMBB response time below 100 ms throughout the experiment, a threshold considered crucial for multiple eMBB applications such as the FWA service.

Fig. 4d further reveals that a higher service rate for eMBB users, represented by configurations with $\mu_E = 2$ (light blue, yellow, and orange lines), also leads to shorter URLLC response times compared to their respective counterparts with $\mu_E = 1$ (dark blue, green, and red lines). This is attributed to eMBB requests spending less time occupying containers, which are then reinitialized to handle incoming URLLC requests. However, in most cases, this difference is below 0.1 ms and may not be significant even for URLLC applications. Conversely, an increase in the URLLC service rate ($\mu_U = 1$, $\mu_U = 2$, and $\mu_U = 4$) results in shorter response times for this service category, with a more substantial impact. For example, at $\lambda_U = 10$, the orange curve ($\mu_U = 4$, $\mu_E = 2$) shows a response time of approximately 0.5 ms, whereas the yellow curve ($\mu_U = 2$, $\mu_E = 2$) exhibits 0.8 ms. This 0.3 ms difference is significant for URLLC applications, as some require a response time of 1.2 ms or less, while others, such as Robotics and Telepresence, demand at most only 1 ms [1].

In configurations where $\mu_U = 1$ (light and dark blue lines), an interesting behavior is observed in Fig. 4d. As the URLLC request arrival rate approaches the processing



(a) eMBB Availability

(b) URLLC Availability

(c) eMBB Response Time

(d) URLLC Response Time

Fig. 4. Effects of the URLLC service rate (μ_U) and eMBB service rate (μ_E)

capacity, a decrease in the response time for this service category is observed. This is attributed to URLLC containers spending more time active and less time in the setup state, thereby reducing the impact of this component. However, shortly thereafter, there is an increase in the response time due to competition for processing resources within the same service category, resulting from a larger number of URLLC requests waiting in the buffer. This behavior is also present in configurations with $\mu_U = 2$ and $\mu_U = 4$, but for larger $\lambda_U > 25$ values, which are not represented in this figure.

IV. Conclusions and Future Directions

This work delved into the dynamics of virtual resource allocation in MEC-NFV architecture that cater to both URLLC and eMBB services. The adopted approach leveraged a CTMC-based model that incorporated practical factors such as resource failures, service prioritization, and setup (repair) times, since they can incur significant impacts on the 5G applications' requirements. Numerical results showcase insights on how multiple parameter variations such as eMBB/URLLC service arrival rates and container setup/failure rates can impact both eMBB and URLLC services, considering their respective thresholds. In brief, the proposed model serves as a valuable tool for comprehending the operational dynamics of the MEC-NFV-based 5G network when catering to diverse service categories. As future directions, we advocate for the exploration of multiobjective formulations for resource allocation problems within the MEC-NFV paradigm, with a special emphasis on accommodating the coexistence of eMBB and URLLC services. Additionally, we propose investigating the development of cost-effective solutions to further optimize resource allocation strategies, addressing the evolving demands of 5G networks.

Acknowledgment

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001 and by the National Council for Scientific and Technological Development (CNPq)- Project No. 433142/2018-9. It is also a product of the Project Model 5G, conducted by the Institute of Science and Technology - SIDIA, in partnership with Sansung Eletrônica da Amazonia LTDA (Informatics Law No. 8387/91, Art. 39 of Decree 10.21/2020).

References

[1] M. U. A. Siddiqui, H. Abumarshoud, L. Bariah, S. Muhaidat, M. A. Imran and L. Mohjazi, “URLLC in Beyond 5G and 6G Networks: An Interference Management Perspective,” in IEEE Access, vol. 11, pp. 54639-54663, 2023.

[2] M. Setayesh, S. Bahrami and V. W. S. Wong, “Resource Slicing for eMBB and URLLC Services in Radio Access Network Using Hierarchical Deep Learning,” in IEEE Transactions on Wireless Communications, vol. 21, no. 11, pp. 8950-8966, Nov. 2022.

[3] A. K. Bairagi et al., “Coexistence Mechanism Between eMBB and uRLLC in 5G Wireless Networks,” in IEEE Transactions on Communications, vol. 69, no. 3, pp. 1736-1749, March 2021.

[4] T. Zhang, H. Qiu, L. Linguaglossa, W. Cerroni and P. Giaccone, “NFV Platforms: Taxonomy, Design Choices and Future Challenges,” in IEEE Transactions on Network and Service Management, vol. 18, no. 1, pp. 30-48, March 2021.

[5] Y. Kim and S. Park, “Calculation Method of Spectrum Requirement for IMT-2020 eMBB and URLLC With Puncturing Based on M/G/1 Priority Queuing Model,” in IEEE Access, vol. 8, pp. 25027-25040, 2020.

[6] W. Li and S. Jin, “Performance evaluation and optimization of a task offloading strategy on the mobile edge computing with edge heterogeneity,” The Journal of Supercomputing, vol. 77, no. 11, pp. 12486-12507, 2021.

[7] Z. Tong, T. Zhang, Y. Zhu and R. Huang, “Communication and Computation Resource Allocation for End-to-End Slicing in Mobile Networks,” 2020 IEEE/CIC International Conference on Communications in China (ICCC), Chongqing, China, 2020, pp. 1286-1291.

[8] W. Li and S. Jin, “Performance Evaluation and Optimization of a Task Offloading Strategy on the Mobile Edge Computing with Edge Heterogeneity,” The Journal of Supercomputing, vol. 77, no. 11, pp. 12486-12507, Nov. 2021.

[9] T. Liu, L. Fang, Y. Zhu, W. Tong and Y. Yang, “A Near-Optimal Approach for Online Task Offloading and Resource Allocation in Edge-Cloud Orchestrated Computing,” in IEEE Transactions on Mobile Computing, vol. 21, no. 8, pp. 2687-2700, 1 Aug. 2022.

[10] I. Sarrigiannis, K. Ramantas, E. Kartsakli, P. -V. Mekikis, A. Antonopoulos and C. Verikoukis, ”Online VNF Lifecycle Management in an MEC-Enabled 5G IoT Architecture,” in IEEE Internet of Things Journal, vol. 7, no. 5, pp. 4183-4194, May 2020, doi: 10.1109/JIOT.2019.2944695.

[11] 3GPP. System architecture for the 5g system (5gs). White Paper, 2020.

[12] K. Kaur, T. Dhand, N. Kumar and S. Zeadally, “Container-as-a-Service at the Edge: Trade-off between Energy Efficiency and Service Availability at Fog Nano Data Centers,” in IEEE Wireless Communications, vol. 24, no. 3, pp. 48-56, June 2017.

[13] M. N. Mahdi, A. R. Ahmad, Q. S. Qassim, H. Natiq, M. A. Subhi, and M. Mahmoud, “From 5G to 6G Technology: Meets Energy, Internet-of-Things and Machine Learning: A Survey,” Applied Sciences, vol. 11, no. 17, p. 8117, Aug. 2021.

[14] Y. Sugito et al., “A Study on the Required Video Bit-rate for 8K 120-Hz HEVC/H.265 Temporal Scalable Coding,” 2018 Picture Coding Symposium (PCS), San Francisco, CA, USA, 2018, pp. 106-110.