Dynamic Resource Allocation for URLLC in UAV-Enabled Multi-access Edge Computing

Marcos Falcão, Caio Souza, Andson Balieiro, and Kelvin Dias
Centro de Informática (CIn)
Universidade Federal de Pernambuco (UFPE)
Recife, Brazil
{mrmf,ccbs,amb4,kld}@cin.ufpe.br

***Abstract*—In the context of Ultra-reliable Low Latency Communications (URLLC), the concepts of Multi-access Edge Computing (MEC), Network Function Virtualization (NFV), and Unmanned Aerial Vehicle (UAV) emerge as complementary paradigms that shall offer fine-grained on-demand distributed resources closer to the User Equipment (UE) and strong Line-of-Sight (LoS) paths between UAV and ground transmission nodes. However, compromise between onboard computation resource allocation and the URLLC requirements becomes challenging since UAVs are limited due to their size, weight, and power, and the virtualization adds extra overhead, which imposes a burden on the conventional Network Functions (NFs). This work proposes a NFV-MEC over UAV model based on Continuous-time Markov Chain (CTMC), with an embedded virtual resource scaling scheme for dynamic resource allocation (DRA). It also extensively analyzes the NFV-MEC architecture's virtualization layer, including node availability and power consumption, besides the URLLC conflicting reliability and latency metrics. The designed model allows analyzing how the main underlying virtualization parameters impact the critical services in a single NFV-MEC over a UAV node, assisting the network operator in proper node dimensioning and configuration.**

***Index Terms*—UAV, MEC, URLLC,NFV,CMTC**

I. Introduction

The architectural movement toward Multi-access Edge Computing (MEC) enables the deployment of core network functions and cloud applications closer to the User Equipment (UE). It avoids the burden imposed by multiple network components, equipment, and protocols that build the path between regular application servers and UE [1]. MEC generally allows UEs to offload the tasks to MEC servers at the network edge, benefiting both operator and UEs by reducing end-to-end service latency and increasing service capacity. The computation infrastructure in the edge is similar to that of data centers but on a smaller scale, which can be efficiently orchestrated by Network Function Virtualization (NFV) [2].

To meet the demands of device density, position, and service types, a highly decentralized edge node infrastructure becomes necessary [3]. This implies a high deployment cost often associated with physical location expenses, hardware, software, and energy consumption [4]. Compared to fixed MEC infrastructure, MEC-enabled Unmanned Aerial Vehicles (UAVs) yield an efficient and flexible solution to meet the dynamic demands of ultra-dense networks [5]. Owing to the mobility, UAVs enable edge servers closer to UEs, establishing Line-of-Sight (LoS), which assures the best connectivity conditions towards and consequently higher transmission rates and reliability [6]. On the other hand, the main challenges rely on the limited resource and battery life, which leads to the importance of efficient resource dimensioning and allocation.

Recently, a great effort has been dedicated to the field of MEC-enabled UAVs, especially on trajectory and energy optimization [7], [8]. However, little has been discussed about the impact of the computing subsystem [9] on MEC-UAV node dimensioning and applications, primarily when virtualization is used to provide computing resources since it brings critical practical points to be considered. For instance, containers are cost-effective for resource utilization, and present lower startup overhead [10], being suitable to support VNF scaling for URLLC services. However, they are still not mature compared to VMs, which may affect the service reliability and availability. In addition, studies that address resource allocation in terrestrial MEC nodes ignore the time overhead for virtual resource instantiation and failure recovery or computational power degradation of parallel VMs on the same physical node. Besides, they do not consider metrics such as reliability, availability, power consumption, and response time [11]–[13].

This paper proposes a NFV-MEC over an UAV model based on Continuous-time Markov Chain (CTMC), with an embedded virtual resource scaling scheme for dynamic resource allocation (DRA). It allows analyzing how the main underlying virtualization parameters impact the critical services in an NFV-MEC over a UAV node and thus assisting the network operator in the proper node dimensioning and configuration for URLLC, considering not only the latency and reliability but also power consumption and availability. The remainder of this paper is organized as follows. Section II presents related studies. Section III describes the system model, basic assumptions, proposed model, and derived performance metrics. Model validation and extensive analysis are presented in Section IV. Finally, Section V concludes this work.

This work was supported by the CNPq Project No. 433142/2018-9, CNPq Research Productivity Fellowship (Grant No. 312831/2020-0) and Pernambuco Research Foundation (Grant No. IBPG-0096-1.03/16).

II. Related Work

Previous studies proposed different solutions for using UAV as network infrastructure. In [7], the authors address the energy consumption minimization on both UE and UAV while optimizing the horizontal trajectory of the vehicle. In [8], a communication and computation allocation strategy is proposed to select the ideal altitude and minimize the UAV energy consumption under latency constraints. [14] adopts UAVs as backhaul and core network (CN) equipments and proposes MEC-enabled networks over multiple UAVs to minimize the power consumption. More recently, [15] explored UAVs acting as radio, CN, and edge clouds (ECs) but its testbed focused on the analysis of UAVs as ECs hosting aerial control system functions in which both the travel plan and computational usage were considered to reduce energy consumption. Although they bring important outcomes, these studies just focus on the latency or energy consumption minimization, neither covering reliability or availability issues nor considering virtualization overheads that may impact the URLLC.

Other works have focused solely on MEC-related computational resource issues, encompassing problems such as resource placement, scheduling, node dimensioning, and Dynamic Resource Allocation (DRA) [16]. For instance, the authors in [17] design a greedy-based search algorithm to find the minimum number of MEC servers considering delay and workload budget. In [18], a model based on queuing theory was proposed with an optimization problem to identify the number of virtual resources to maximize the task execution capacity using the first fit strategy to solve it. Even addressing different problems in MEC environment, previous works neglect virtualization overheads (e.g., resource setup time, failure recovery, and performance degradation of multiple VMs on the same physical node) that may cause violation of URLLC constraints or energy budget of the UAV.

While there has been significant attention paid to latency and energy consumption aspects of MEC-enabled UAVs based on trajectory optimization, computing infrastructure resilience, virtualization overhead, and resource availability have received far less attention even though these are of paramount importance for resource dimensioning/allocation, especially considering critical applications. Motivated by this gap, we propose a CMTC model that allows a service provider to properly dimension a MEC-enabled UAV node under availability, power consumption, reliability and latency perspectives, offering a complementary solution to those previous works. Besides, extensive analysis is carried out considering the 3GPP standard for URLLC when selecting parameter values to explore. In our work, two practical computational concepts play a crucial role: (1) virtualization technology and (2) on-demand resource activation (scaling). With regards to the first, although NFV has traditionally been implemented over VMs, it is widely accepted that containers are the most cost-effective solution in terms of physical resource utilization, besides having smaller instantiation overheads [10], which impacts the latter, especially to support VNF scaling. However, they are still not mature compared to VMs, i.e., there are multiple security risks involved in containerization since they share a single kernel, which may affect both availability and reliability [21].

III. System Model

A. Computing Model

Each on-board VNF runs equally and independently on a single microkernel-based VM [22] or container that shares a common physical machine (PM), with VMs executing uninterrupted while containers are scaled upon demand. A central control unit determines request admission, only activating containerized VNFs when all VM-hosted ones are busy. The containerized VNF activation comprises two phases: initializing the kernel image and launching the specified function, which is interpreted as setup time, during which power and resources are consumed but no service is processed.

Following [23], we have considered that the parallel-operating VMs can deteriorate each other's processing time. On the other hand, the container's average performance is generally superior than the VM's and comparable to that of the PM with regards to multiple features. Thus, for the case of a single VM deployment, the task's execution rate is μ services/time unit, however, to account for the VM overhead, the task's execution rate is modeled taking the total number of VMs (see (1)), where d is the computation degradation factor. This results in μ_{VM} being a monotone decreasing function of d, whereas the μ_C is constant since the container performance overhead is negligible.

$$\mu_{VM} = \frac{\mu}{(1+d)^{(n-1)}} \tag{1}$$

Active containerized VNFs may suffer failures during attendance, which implies either a service migration to an available VM/container or a repair (triggering a new setup period), with progress being lost only in the latter case. In general, repair times will depend on the failure type; for instance, a software component crash can be quickly fixed by the host in a few microseconds, while others may take a few milliseconds to reboot device and VNF. Since the model deals with critical flows, only the worst-case scenario is considered. Lastly, as soon as an operative VNF finishes processing and there are no remaining requests, the VNF instance can either be powered down together with the host container or remain active if hosted by a VM. The shutdown delay is ignored for being significantly smaller than the setup (repair) magnitudes [24]. Fig. 1 summarizes the above process.

B. Analytical Model

The system comprises a single MEC-enabled UAV comprising a maximum capacity of K services that are served by up to n VMs and c containers, with $K \geq n + c$, which implies a queue (q) that is limited to $K - (n + c)$ services. Service requests follow a Poisson process with rate λ (requests/ms) and server capacities of one service with an exponentially distributed service rate of μ_C for containerized VNFs, whereas for VMs, μ_{VM} is given by (1). Control applications are likely

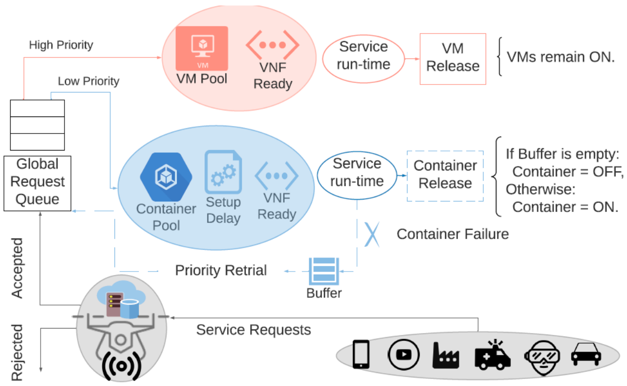


Fig. 1. Failure & Repair model diagram

to fit a regularly spaced packet trace (isochronous), i.e., a superposition of deterministically spaced and sporadic packet streams, where each contributes to a portion of the overall traffic, which might be modeled as a Poisson [26].

Container setup/repair times and failures are also exponentially distributed with rates α and γ, respectively. A regular first come first served queue was assumed for new requests with prioritization for retrial. We assume a standalone deployment and the system is modeled as an M/M/n+c/K queue (Fig. 2) with setup time and failure. The feasible state space is given by $\Omega = (i, j, k) \mid 0 \leq i \leq n,\ 0 \leq j \leq c$, and $0 \leq k \leq K$, with $i + j \leq k$ and $i, j, n, c, k, K \in \mathbb{Z}^+$. Each state ($i$,$j$,$k$) denotes the number of services allocated to VMs (i), containers (j) and the total number of services in the system (k), respectively. Furthermore, the steady-state probabilities $\pi(i, j)$ are extracted from the solution of a linear system formed by the normalization condition (2) and balance equations of all states of the model. States with the same color in Fig. 2 follow the same type of balance equation (omitted due to the paper size limit). Please consider $(i, j, k) \in \Omega$ in all equations to follow.

$$\sum_{i=0}^{n} \sum_{j=0}^{c} \sum_{k=i+j}^{K} \pi(i, j, k) \tag{2}$$

C. Performance Metrics

In MEC-enabled networks, task execution at the MEC server is strongly tied to resource availability, failure resilience and response time [20], while power management is crucial for UAV equipment. In this respect, we consider the steady-state analysis of the CTMC under study, followed by the derivation of the system's Availability (A), Reliability (R), Power Consumption (C) and Response Time (T).

1) Availability (A): In our work, Availability is the system's ability to offer the minimum amount of functional and accessible VNFs. In particular, a VNF instance is considered available if at least one of its constituents (VM-hosted or containerized) remains accessible. In brief, the MEC-enabled UAV node Availability (A) (3) is obtained by the probability sum of all states except those representing full capacity.

$$A = 1 - \sum_{i=0}^{n} \sum_{j=0}^{c} \pi_{i,j,K} \tag{3}$$

2) Reliability (R): The designed model also evaluates the Reliability (R) being given by (4), which combines the admitted flow $\lambda * A$ with the effective failure rate in the entire node, i.e., it denotes the probability that a service is served without experiencing failures while being processed by MEC VNFs.

$$R = 1 - \frac{\gamma}{\lambda * A} \sum_{j=1}^{c} j \left[\sum_{i=0}^{n} \sum_{k=1}^{K} \pi_{i,j,k} \right] \tag{4}$$

3) Power Consumption (C): The computational power consumption is an important component of the operational costs and must be considered by the service provider for resource planning to address cost-performance trade off, mainly in UAV nodes, which are battery-oriented devices. In our framework, power consumption (C) is formed from the combination of the mean number of virtual resources and energy consumption values (P) for each virtualization technology (VM and Container) and operating states (Idle, Setup and Busy), as shown in (10). The mean number of VMs and containers (CT) in each state is described in (5)-(9). Since VMs are constantly powered, no equation for its setup state is described. The notation used to express the energy consumption of each technology and state are summarized in Table I.

$$\overline{VM}_{idle} = \sum_{i=0}^{n} (n - i) \left[\sum_{j=0}^{c} \sum_{k=0}^{K} \pi_{i,j,k} \right] \tag{5}$$

$$\overline{VM}_{busy} = \sum_{i=0}^{n} i \left[\sum_{j=0}^{c} \sum_{k=0}^{K} \pi_{i,j,k} \right] \tag{6}$$

$$\begin{aligned} \overline{CT}_{idle} &= \sum_{j=0}^{c} (c - j) \sum_{i=0}^{n} \pi_{i,j,k}, \text{ with } k = i + j \\ &+ \sum_{j=0}^{c} \sum_{i=0}^{n} \sum_{k=i+j+1}^{K} (c - j - min(k - i - j, c - j))\pi_{i,j,k} \end{aligned} \tag{7}$$

$$\overline{CT}_{setup} = \sum_{j=0}^{c} \sum_{k=n+j+1}^{K} min(k - n - j, c - j)\pi_{n,j,k}. \tag{8}$$

$$\overline{CT}_{busy} = \sum_{j=1}^{c} j \left[\sum_{i=0}^{n} \sum_{k=j+i}^{K} \pi_{i,j,k} \right]. \tag{9}$$

$$\begin{aligned} C &= P_{idle}^{VM}\ \overline{VM}_{idle} + P_{busy}^{VM}\ \overline{VM}_{busy} \\ &+ P_{idle}^{CT}\ \overline{CT}_{idle} + P_{setup}^{CT} \overline{CT}_{setup} + P_{busy}^{CT}\ \overline{CT}_{busy} \end{aligned} \tag{10}$$

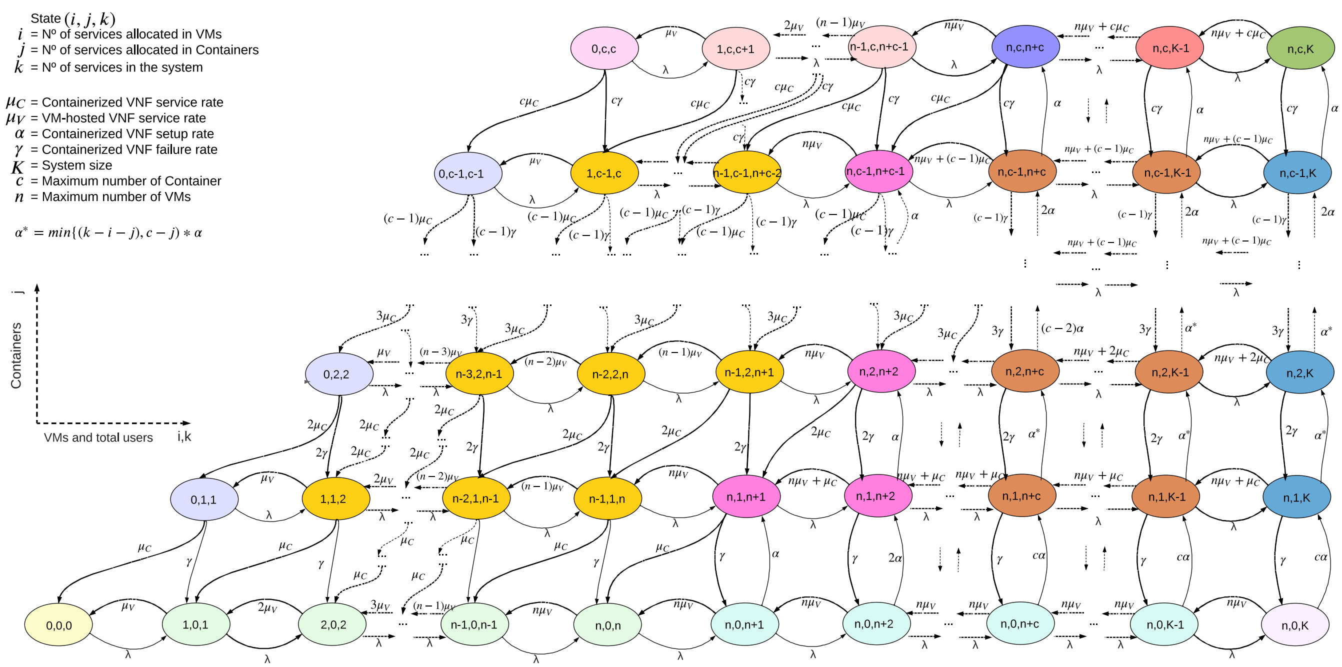


Fig. 2. Space state diagram

4) Response Time (T): We define the Response Time (T) of a VNF that processes the service as the interval between the service arrival on the edge node and its processing time, including the containerized VNF setup restart times if these events are triggered. The Mean Response Time is obtained by calculating the mean number of services in the system and the mean number of accepted services (11).

$$T = \frac{1}{\lambda * A} \sum_{k=0}^{K} k \left[\sum_{i=0}^{n} \sum_{j=0}^{c} \pi_{i,j,k} \right] \tag{11}$$

IV. Validation and Analysis

The scenario of interest is composed of one semi-static MEC-enabled UAV which able to increase its coverage area. Intuitively, the higher the altitude, the larger is the coverage offered by the platform and the lower is the chance of shadowing effects, thus the service arrival rate may increase or decrease accordingly [8]. The analytical results were validated against discrete-event simulations (Figs. 3-4), where the lines denote the analytical and the markers represent simulation results. For the main parameters, we have followed a subset of the 3GPP Release 16 (TR 38.824), in which the service rates μ and μ_C are 1 (1 service/ms), whereas μ_{VM} is given by (1).

Each scenario simultaneously evaluates the impact of a pair of parameters: Fig. 3 Multiple VM Amounts (n) and Overhead Degradation Factor (d) and Fig. 4 multiple container amounts (c) and failure rates (γ), with the service arrivals ranging from 1 up to 100 requests/ms, setup rate (α) and queue size equal to 1 unit/ms [24] and 10, respectively. In addition, unless otherwise stated the baseline values for failure (γ) was 0.001 [24]. In terms of power consumption for VMs and containers for different operation states, we adopted the values from the network intensive experiment in [10], which is summarized in Table I. The remaining parameters can be found in Table II.

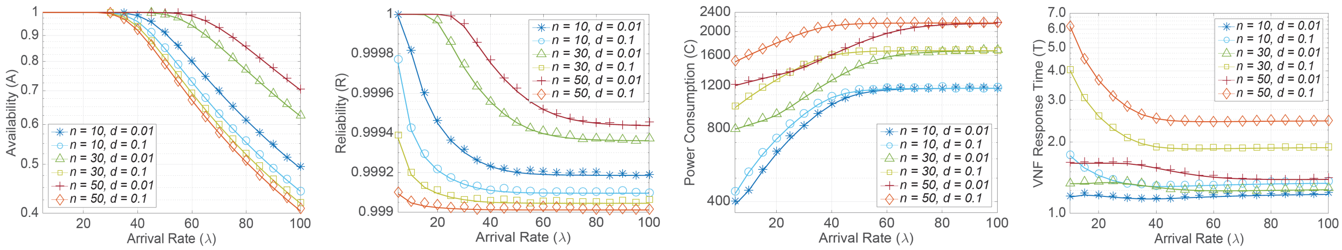
TABLE I
Notation and Power Consumption Values

Virtualization	State	Status	Symbol	Value
VM-hosted	Idle	ON	P_{idle}^{VM}	20W
VM-hosted	Busy	ON	P_{busy}^{VM}	25W
Containerized	Idle	SLEEP	P_{idle}^{CT}	4W
Containerized	Setup	ON	P_{setup}^{CT}	8W
Containerized	Busy	ON	P_{busy}^{CT}	23W

A. Multiple VMs (n) and Overhead Degradation Factor (d)

In Figs. 3a-3b both Availability and Reliability have similar configuration disposal, however with large magnitude. In Fig. 3a, the combined increase of n and d implies a less available system due to a lower VM service rate. This also means that containers are likely to be more required, meaning that failures happen more often, leading to a less reliable system (Fig. 3b). Comparing the Availability for configurations (($n = 10$), ($n = 30$)) and (($n = 30$), ($n = 50$)) that share $d = 0.01$, the absolute difference is 11.5% and 9.1%, respectively, whereas in Fig. 3b the difference among the Reliability from the same curves is much smaller in absolute terms, which would only be relevant for specific services such as URLLC [25].

In Fig. 3c, both the maximum number of VMs (n) and overhead degradation factor (d) greatly impacted power consumption, but in different ways. Since VMs are not dynamically scalable, it was expected that the increase in n would also increase overall power consumption, regardless of λ. On the other hand, the overhead degradation factor (d) clearly impacts



(a) Availability (b) Reliability (c) Power Consumption (d) Response Time

Fig. 3. Multiple VMs (n) and Overhead Degradation Factor (d)

TABLE II
EXPERIMENT SETS

Section	Parameters	n	c	d	γ
IV - A	n, d	10,30,50	40	10^{-2},10^{-1}	10^{-3}
IV - B	c, γ	10	40,60,80	10^{-2}	10^{-3},10^{-2}

the non-saturated system, and particularly in this scenario, has only impacted configurations with higher n values (30 and 50) respectively. For $\lambda \to 100$, the curves with the same n overlap, which evinces a saturated system, i.e., all resources are processing services, although each pair of curves has different throughput due to the different values for d. With respect to the differences in power consumption, there is a maximum gap of 500W between the two curves with $n = 30$ and $d = 0.01$, and $n = 30$ and $d = 0.1$, respectively, at $\lambda = 35$, whereas the maximum difference in power consumption between the curves with $n = 50$ and $d = 0.01$, and $n = 50$ and $d = 0.1$, respectively, is approximately 600W at the same reference point. This difference is quite significant since each of them has the same VM amount. Another particularity is observed when comparing the curves with $n = 30$ and $d = 0.1$, and $n = 50$ and $d = 0.01$ for $\lambda = 20$ 50, which had similar performance, although the former has 20 fewer VMs than the latter when considering the total number of resources. In brief, in addition to accounting for the fact that the impact of d is potentiated by n, higher d values also make the overall service rate lower, leaving the VMs in Busy mode for longer periods, besides forcing the system to activate containers more often.

Unlike power consumption, changes in n have very little impact on the response time (Fig. 3d) compared to the results of the curves with different values of d. This indicates that the positive effects of increasing n can be mitigated depending on the value of d. In general, there are multiple nuances involving each parameter, service load, and performance metrics that might conflict, highlighting the relevance of an adequate dimensioning process. For instance, the response time almost doubles when comparing the two curves with $n = 50$. Moreover, considering availability and reliability, the curve with $n = 50$ and $d = 0.01$ had the best performance, whereas for power consumption, it was one of the worst.

B. Multiple Containers (c) and Failure Rates (γ)

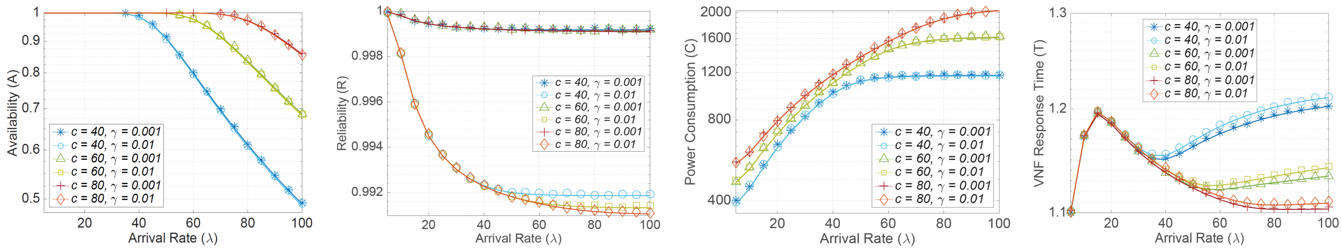
A larger γ means smaller intervals between successive containerized VNF failures. In contrast to α, this parameter improves the system as it gets smaller. Thus, γ was enhanced by factors of 10 and 100. In addition, containers are not prone to significant overhead issues, besides, there is no correlation between the container amount and its individual service rate. For this resource type, failures are isolated events. In this scenario, we have reduced and fixed the total amount of VMs ($n = 10$) to highlight the impact of c and γ. Thus, for small $\lambda < 10$ most services are processed in VMs, not having a notable influence on the results.

It was found that γ values have little influence on Availability (Fig. 4a), Power Consumption (Fig. 4c), and even Response Time (Fig. 4d) within the assumed ranges. This does not mean that the parameter has no impact on these metrics, but rather that γ would need to be at least of the same order as α in order to have a significant effect on the system's capacity due to more frequent container failures. In contrast, the container amount c had a significant impact on all three metrics, particularly for high values of λ where the maximum differences were observed. As for Reliability (Fig. 4b), the failure rate γ remains one of its key components and hence significantly impacts the curves, while c has little impact. In addition, compared to Reliability, the maximum absolute difference between curves is much higher (0.007).

In Fig. 4d, the Response Time spikes for $\lambda = 10$, which corresponds to the fixed maximum amount of VMs ($n = 10$). From this point on, the containers are turned on, which explains the sudden spike due to the setup time and, moreover, the sudden drop when part of these resources are available and processing incoming services. Lastly, in each pair of curves, there is a slight increase in the Response Time at different λ, which is explained by the resource limit and further queue activation, which is not a processing unit. Compared to the Response Time in Fig. 3d, it is noticeable that the maximum difference between curves is much less significant (0.1 ms).

V. CONCLUSION

Wireless communication networks are transitioning from pure communication to service enablers in multiple verticals, composing a system that dynamically adapts to the evolving landscape. For instance, the edge node requires not only high availability and reliability, but also low-latency for autonomous decision making, while also coping with resource constraints. This work analyzed the impact of virtualization layer parameters on URLLC applications in the context of UAV-Enabled Multi-access Edge Computing. In this respect, a dynamic resource provisioning scheme was created considering practical



(a) Availability (b) Reliability (c) Power Consumption (d) Response Time

Fig. 4. Multiple Container Amounts (c) and Failure Rates (γ)

assumptions such as resource failures, setup/repair periods, and processing overhead. Hence, it allows a network operator to configure and dimension the node taking into account classic URLLC performance metrics (Availability, Reliability, and Response Time), and the power consumption (for UAV nodes), while also admitting customization, such as the VM/container interference assumptions. As a future direction, other network segments (e.g., RAN) and the UAV trajectory can be incorporated in order to perform joint parameter optimization (e.g., transmit powers) and computation resources (such as CPU cycles) to find the most favorable compromise between energy consumption, latency, and performance.

References

[1] R. Ali, Y. B. Zikria, A. K. Bashir, S. Garg and H. S. Kim. "URLLC for 5G and Beyond: Requirements, Enabling Incumbent Technologies and Network Intelligence", in IEEE Access, vol. 9, pp. 67064-67095, 2021, doi: 10.1109/ACCESS.2021.3073806.

[2] B. Ali, M. A. Gregory and S. Li, "Multi-Access Edge Computing Architecture, Data Security and Privacy: A Review, " in IEEE Access, vol. 9, pp. 18706-18721, 2021, doi: 10.1109/ACCESS.2021.3053233.

[3] M. C. Filippou et al., "Multi-Access Edge Computing: A Comparative Analysis of 5G System Deployments and Service Consumption Locality Variants", in IEEE Communications Standards Magazine, vol. 4, no. 2, pp. 32-39, June 2020, doi: 10.1109/MCOMSTD.001.1900034.

[4] A. Santoyo-González and C. Cervelló-Pastor, "Edge Nodes Infrastructure Placement Parameters for 5G Networks", 2018 IEEE Conference on Standards for Communications and Networking (CSCN), 2018, pp. 1-6, doi: 10.1109/CSCN.2018.8581749.

[5] R. Islambouli and S. Sharafeddine, "Optimized 3D Deployment of UAV-Mounted Cloudlets to Support Latency-Sensitive Services in IoT Networks", in IEEE Access, vol. 7, pp. 172860-172870, 2019, doi: 10.1109/ACCESS.2019.2956150.

[6] X. Li, H. Yao, J. Wang, X. Xu, C. Jiang and L. Hanzo, "A Near-Optimal UAV-Aided Radio Coverage Strategy for Dense Urban Areas," in IEEE Transactions on Vehicular Technology, vol. 68, no. 9, pp. 9098-9109, Sept. 2019, doi: 10.1109/TVT.2019.2927425

[7] J. Zhang et al., "Stochastic Computation Offloading and Trajectory Scheduling for UAV-Assisted Mobile Edge Computing," in IEEE Internet of Things Journal, vol. 6, no. 2, pp. 3688-3699, 2019, doi: 10.1109/JIOT.2018.2890133.

[8] F. Costanzo, P. D. Lorenzo and S. Barbarossa, "Dynamic Resource Optimization and Altitude Selection in Uav-Based Multi-Access Edge Computing," ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020, pp. 4985-4989, doi: 10.1109/ICASSP40776.2020.9053594.

[9] F. Zhou, Y. Wu, R. Q. Hu and Y. Qian, "Computation Rate Maximization in UAV-Enabled Wireless-Powered Mobile-Edge Computing Systems," in IEEE Journal on Selected Areas in Communications, vol. 36, no. 9, pp. 1927-1941, Sept. 2018, doi: 10.1109/JSAC.2018.2864426.

[10] R. Morabito, "Power Consumption of Virtualization Technologies: An Empirical Investigation," IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC), 2015, pp. 522-527, doi: 10.1109/UCC.2015.93.

[11] L. Yala, P. A. Frangoudis and A. Ksentini, "Latency and Availability Driven VNF Placement in a MEC-NFV Environment," IEEE Global Communications Conference, doi: 10.1109/GLOCOM.2018.8647858.

[12] Ma, S., Chen, X., Li, Z. et al. "Performance Evaluation of URLLC in 5G Based on Stochastic Network Calculus," in Mobile Netw Appl 26, 1182–1194,2021 doi: 10.1007/s11036-019-01344-1.

[13] N. Kherraf, H. A. Alameddine, S. Sharafeddine, C. M. Assi and A. Ghrayeb, "Optimized Provisioning of Edge Computing Resources With Heterogeneous Workload in IoT Networks," in IEEE Transactions on Network and Service Management, vol. 16, no. 2, pp. 459-474, June 2019, doi: 10.1109/TNSM.2019.2894955.

[14] Z. Yang, C. Pan, K. Wang and M. Shikh-Bahaei, "Energy Efficient Resource Allocation in UAV-Enabled Mobile Edge Computing Networks," in IEEE Transactions on Wireless Communications, vol. 18, no. 9, pp. 4576-4589, Sept. 2019, doi: 10.1109/TWC.2019.2927313.

[15] O. Bekkouche, K. Samdanis, M. Bagaa and T. Taleb, "A Service-Based Architecture for Enabling UAV Enhanced Network Services," in IEEE Network, vol. 34, no. 4, pp. 328-335, July/August 2020, doi: 10.1109/MNET.001.1900556.

[16] Z. Cheng, Q. Wang, Z. Li and G. Rudolph, "Computation Offloading and Resource Allocation for Mobile Edge Computing," IEEE Symposium Series on Computational Intelligence, 2019, pp. 2735-2740, doi: 10.1109/SSCI44817.2019.9003106.

[17] S. Lee, S. Lee and M.K. Shin, "Low Cost MEC Server Placement and Association in 5G Networks," International Conference on Information and Communication Technology Convergence, 2019, pp. 879-882, doi: 10.1109/ICTC46691.2019.8939566.

[18] M. Emara, H. ElSawy, M. C. Filippou and G. Bauch, "Spatiotemporal Dependable Task Execution Services in MEC-Enabled Wireless Systems" IEEE Wireless Communications Letters, vol. 10, no. 2, pp. 211-215, 2021, doi: 10.1109/LWC.2020.3024749.

[19] 3GPP, "System Architecture for the 5G System (5GS)," Technical Spec. (TS) 23.501 v15.9.0, Mar. 2020.

[20] Kekki S. et al, "MEC in 5G networks," in ETSI White Paper, no. 28, pp. 1–28, 2018.

[21] T. Zhang, H. Qiu, L. Linguaglossa, W. Cerroni and P. Giaccone, "NFV Platforms: Taxonomy, Design Choices and Future Challenges," IEEE Transactions on Network and Service Management, vol. 18, no. 1, pp. 30-48, 2021, doi: 10.1109/TNSM.2020.3045381.

[22] T. Fautrel, L. George, F. Fauberteau and T. Grandpierre, "An hypervisor approach for mixed critical real-time UAV applications," IEEE International Conference on Pervasive Computing and Communications Workshops, 2019, pp. 985-991, doi: 10.1109/PERCOMW.2019.8730705.

[23] I. Mavridis and H. Karatza, "Combining containers and virtual machines to enhance isolation and extend functionality on cloud computing,"Future Generation Computer Systems, vol 94, no. 1, pp 674-696, 2019, doi: 10.1016/j.future.2018.12.035.

[24] 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," IEEE Wireless Communications, vol. 24, no. 3, pp. 48-56, June 2017, doi: 10.1109/MWC.2017.1600427.

[25] M. Falcao, C.B. Souza, A. Balieiro and K. Dias, "An analytical framework for URLLC in hybrid MEC environments," Journal of Supercomputing, 2021, doi: 10.1007/s11227-021-03945-8

[26] A. Anand and G. de Veciana, "Resource Allocation and HARQ Optimization for URLLC Traffic in 5G Wireless Networks,"IEEE Journal on Selected Areas in Communications, vol. 36, no. 11, pp. 2411-2421, Nov. 2018, doi: 10.1109/JSAC.2018.2874122.