Virtual Resource Allocation for URLLC in MEC-enabled UAVs: A Reliability and Availability Analysis

Marcos Falcão, Caio Souza, Andson Balieiro and Kelvin Dias

[1]Centro de Informática (CIn) – Universidade Federal de Pernambuco (UFPE)
Av. Jorn. Aníbal Fernandes, s/n – Cidade Universitária – Recife – PE – Brazil

{mrmf, cbbs, amb4,kld}@cin.ufpe.br

Abstract. *Unmanned Aerial Vehicle (UAV) communication networks and Multi-access Edge Computing (MEC) will occupy an important position in the future wireless communication system. Unlike regular datacenter environments, MEC can help mobile devices improve computing and communication capabilities, and its combination with UAVs helps to deal with the Line of Sight (LoS) issues, besides allowing node mobility. This work addresses the dynamic resource provisioning in a UAV equipped with MEC resources (MEC-enabled UAV) that provides on demand communication capabilities to Ultra-reliable and Low-latency Communication (URLLC) services. We adopt a Continuous Time Markov Chain (CTMC) to analyze the overall node availability and reliability, while taking into account virtual host setup (repair) delays and failure events for Virtual Network Functions (VNFs) hosted on MEC-enabled UAVs. Our results show that the containerized VNF setup delays critically impact the admission process, whereas reliability is more prone towards VNF failures.*

Resumo. *Redes de comunicação de veículos aéreos não tripulados (UAV) e computação de borda de acesso múltilplo (MEC) ocuparão uma posição importante no sistema de comunicação sem fio do futuro. Diferentemente dos ambientes comuns de datacenter, o MEC pode ajudar os dispositivos móveis a melhorar as capacidades de computação e comunicação, e sua combinação com UAVs ajuda a lidar com os problemas de Linha de Visão (LoS), além de permitir a mobilidade dos nós. Este artigo aborda o provisionamento dinâmico de recursos de um UAV equipado com uma nuvem MEC que fornece capacidades de comunicação/processamento sob demanda para serviços de confiabilidade alta e latência baixa (URLLC). Nós analisamos a disponibilidade e confiabilidade dos nós via cadeia de markov de tempo contínuo (CMTC), considerando o atraso de incialização/reparo e eventos de falha de funções de redes virtuais (VNFs) embarcadas em UAVs com MEC. Nossos resultados mostram que os atrasos de configuração de VNF em contêineres impactam criticamente o processo de admissão, enquanto a confiabilidade é mais afetada pelas falhas.*

1. Introduction

The fifth and sixth generations (5G and 6G) of mobile communication networks are expected to meet several service categories, including URLLC, which encompasses autonomous vehicles and smart industry applications. To do so, MEC and Network Functions

Virtualization (NFV) are put forward together for extending the notion of cloud computing to the network edge, while enabling network function (e.g., Mobility Management Function) decoupling from dedicated hardware, respectively [Sarrigiannis et al. 2020].

It is widely accepted that each additional hop between the User Equipment (UE) and application server represents increased latency and lower reliability [Santoyo-González and Cervelló-Pastor 2018], which indicates that the best possible location for a MEC node is theoretically one hop away from the UE, especially for URLLC services. Hence, recent efforts have put forward the design of densely deployed MEC-NFV ground and air-based nodes (Fig. 1), both of which can serve as edge computing platforms to offload tasks from Internet-of-Things (IoT) devices, cache popular contents to reduce the burden of backhaul networks [Cheng et al. 2018], or act as one or multiple mobile VNFs [Strinati et al. 2020].

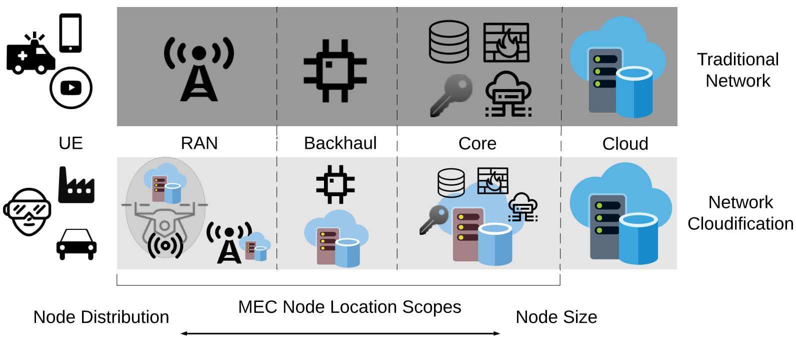


Figure 1. MEC Location Scopes

In terms of practical applications, MEC-enabled UAVs have been considered a promising paradigm to facilitate URLLC, playing a key role in providing service recovery in disaster-stricken regions, enhancing public safety networks or where fixed infrastructure is unfeasible. More recently, since the New Radio (NR) specification in the 3rd Generation Partnership Project (3GPP) Release 17 [3GPP 2019], MEC-enabled UAVs have received considerable attention in terms of traffic management [Lyu et al. 2017], coverage enhancement [Li et al. 2019], improving quality of service [Wang et al. 2019], transmission powers [Zeng et al. 2019], latency minimization [Mozaffari et al. 2019], and network access [Nasir et al. 2019].

Since most efforts are focused on latency and power consumption, computational resource availability and failure resiliency are often neglected in the context of MEC-enabled UAVs [Emara et al. 2021]. As a result, the problem of onboard task execution, accounting for the joint restrictions of physical resources and failure-prone virtual components is still not addressed. Motivated by the above, we assess both the availability and reliability based on the proposed CTMC framework that encompasses a dynamic edge resource allocation process [Falcao et al. 2021]. The CTMC approach is a well-known tool for multiple studies in the telecommunications field and has also been used for evaluating modern service classes such as URLLC [Nielsen et al. 2018].

The remainder of this paper is organized as follows. The related work is described in Section 2. Section 3 describes the CTMC-based system model and the formulations for reliability and availability. Numerical Analysis of a single MEC-enabled UAV node serving critical URLLC applications are conducted in Section 4. Section 5 concludes this work.

2. Related Work

The architectural alternative of MEC-enabled UAVs allows great improvements towards latency requirements, however, unlike traditional ground MEC nodes, MEC-enabled UAVs are usually powered by limited battery which highly impacts its onboard computational capacity. Hence, several studies proposed different architecture and analytical models comprising optimal use of UAV infrastructure.

In [Jeong et al. 2018], the authors attempted to host cloudlets on a UAV so as to minimize energy consumption at the UE while optimizing data rates and the UAV's trajectory under latency constraints. Similarly, in [Hu et al. 2019], the authors minimize the power consumption by jointly optimizing computation offloading and trajectory design. In [Xiong et al. 2019], the authors proposed UAV-aided edge servers for heterogeneous IoT devices aiming at minimizing the overall energy consumption. In [Zhang et al. 2019], the authors used stochastic optimization tools for energy consumption minimization while optimizing the UAV's trajectory. In [Costanzo et al. 2020], a dynamic communication and computation allocation strategy is proposed for selecting the ideal altitude and minimizing the energy consumption while concomitantly satisfying latency constraints. In [Yang et al. 2019], the authors propose a MEC network over multiple UAVs, focusing on minimizing the total power consumption. More recently, the authors in [Bekkouche et al. 2020] explored the performance of UAVs as edge nodes for Aerial Control System (ACS) function hosting, which is responsible for controlling and orchestrating an UAV fleet, focusing on UAVs as backhaul and core network equipment. Finally, in [Wang et al. 2021] a multi-agent deep reinforcement learning based trajectory control algorithm was proposed for jointly maximizing the fairness among UEs and UE-load of each UAV, as well as minimizing the energy consumption. Table 1 compares our work to the existing literature.

Table 1. Summary of Existing Related Work

Work	MEC	Single UAV	Multiple UAVs	Access Network	Computing Resources
[Jeong et al. 2018]	✓	✓	✗	✓	✗
[Hu et al. 2019]	✓	✓	✗	✓	✗
[Xiong et al. 2019]	✓	✓	✗	✓	✗
[Zhang et al. 2019]	✓	✓	✗	✓	✗
[Costanzo et al. 2020]	✓	✓	✗	✓	✗
[Yang et al. 2019]	✓	✗	✓	✓	✗
[Bekkouche et al. 2020]	✓	✓	✗	✓	✗
[Wang et al. 2021]	✓	✗	✓	✗	✗
This Work	✓	✓	✗	✗	✓

It is noticeable how previous works prioritized power management mechanisms based on either trajectory optimization or in Radio Access Network (RAN) improvements, not accounting for the computational standpoint, where two practical points play a key role: (1) virtualization technology and the (2) on-demand resource activation (scaling). With regard to the first, although NFV has traditionally been implemented over

Virtual Machines (VMs), it is widely accepted that containers (CTNs) are the most cost-effective solution in terms of physical resource utilization, besides having lower instantiation overheads [Morabito 2015], which impacts on the latter, specially to support VNF scaling for URLLC services. 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, especially for critical applications.

Motivated by this gap, we adopt and analyze a hybrid VM-containerized onboard infrastructure that leverages the best of both: the VM's strong isolation and the flexibility of containers, while considering how commonly neglected assumptions such as the virtual resource setup (repair) delays and failures can impact communication constraints for URLLC services.

3. System Model

The role of UAVs on enhancing 5G/6G networks can be widely diverse, with UAVs acting as radio, core Network Functions (NFs), edge cloud servers or backhaul equipment [Bekkouche et al. 2020]. The benefits of co-locating these capabilities on a single UAV include improvements on service latency and signaling overhead, while the limitations are mainly related to availability and reliability.

UAV-enabled NFs complement ground core network, facilitating cooperation. Core NFs on UAVs can reduce latency while ensuring stable connectivity, also providing a means of collecting local environment data for enriching network analytics, e.g., for Quality of Service (QoS) assurance. Given the natural energy limitations on UAVs, core NFs that hold user registration data, policy and authentication are not suitable to be hosted on UAVs to avoid losing critical data. Typical core NFs that can be hosted on UAVs include Control plane functions e.g., Access and Mobility Function (AMF), Session Management Function (SMF), and Network Exposure Function (NEF).

In this work, a single isolated MEC-enabled UAV node is evaluated (Fig. 2), where URLLC requests originated from UEs are processed by onboard VNFs, which can be scaled to cope with intensive request periods.

3.1. Key Assumptions

In order to process URLLC requests, a MEC-enabled UAV node first must receive the request and then execute an onboard VNF to process it. The result is then sent back to the mobile user. This work focuses on computing capacity-related issues, thus we assume that mobile users are within the UAV's coverage area and that radio resources are allocated appropriately. Radio resource allocation and UAV coverage management have been thoroughly discussed in [Wang et al. 2021].

The use of NFV in UAV environments provides flexibility to adapt the functions offered by MEC-enabled UAVs to the specific service requirements and accommodate deployment scenarios according to the resource demands, by incorporating new VNFs or scaling when necessary [Borja et al. 2018]. In this respect, NFV is said to leverage the use of physical hardware by allowing seamless elastic resource provisioning, which is a key aspect when dealing with capacity-restricted UAV nodes. Since onboard VNFs should use limited resources, while minimizing energy expenditures, the adoption of a dynamic VNF scaling mechanism may prove to be beneficial.

3.2. Computing Model

In our framework, each VNF runs independently on a single microvisor/microkernel–based VM [Fautrel et al. 2019] or container, with VMs executing continually while containers are activated/deactivated upon demand. A centralized control unit accounts for blocking or admitting requests, only activating containers if all available VMs are processing requests. On the other hand, the containerized VNF activation is fulfilled in two major steps: kernel image initialization and function launch, assumed to be a single transition interval (setup time), during which power and computing resources are in use but the request is not being processed.

Active container-hosted VNFs may fail during request processing, implying in a service migration to an available VM/container or a reset (new setup period), with progress being lost only in the latter. In practice, recovery intervals are highly dependant on failure type; for instance, some software crashes can be immediately fixed by the host in few microseconds, while others may take longer (usually few milliseconds) to reboot device and/or VNF. We have opted for mapping only the worst-case since the proposed framework deals exclusively with URLLC flows, knowing, however, that other traffic types may be added and thus multiple failure situations can be mapped accordingly. Finally, immediately after a VNF ends processing the URLLC request and there is no other remaining, the VNF instance can switch to a low power consumption state (denoted as OFF state) if hosted by a container or remain active if located in a VM. For being significantly smaller than the setup (repair) magnitudes, in this work the shutdown delay is ignored as in [Kaur et al. 2017]. The above process is summarized in Fig. 2.

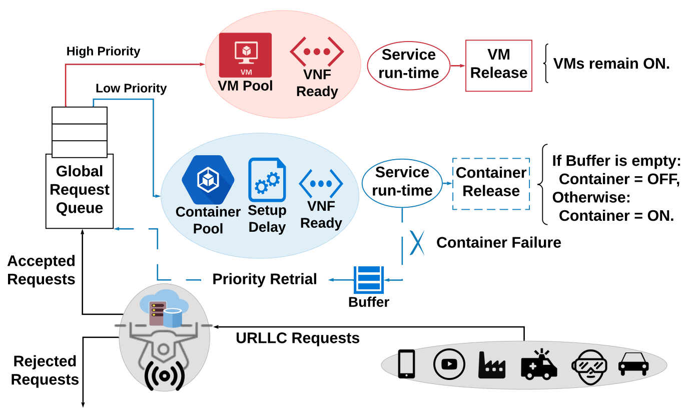


Figure 2. System Model Diagram

3.3. Analytical Model

The system comprises a single MEC-enabled UAV with a simultaneous capacity of n VMs, c containers and k URLLC services, where $k = n + c$. The service request follows a Poisson process with rate λ (requests/ms) and server capacities of one service per unit time, which are exponentially distributed (service rate μ) for both VM-based and container-hosted VNFs. It is currently accepted that URLLC control applications are likely to fit a isochronous packet trace, i.e., a superposition of deterministically spaced and sporadic packet streams, with each part contributing to the overall traffic, being appropriately modeled as a Poissonian process [Anand and de Veciana 2018]. Furthermore, container setup times and failures are also exponentially distributed respectively with rates α and γ, although there is currently no literature to support that when they are exponential random variables [Ren et al. 2016]. With respect to the attendance order, a typical First Come First Served (FCFS) queue was assumed for new requests but prioritization for retrial requests (due to failures) are accepted.

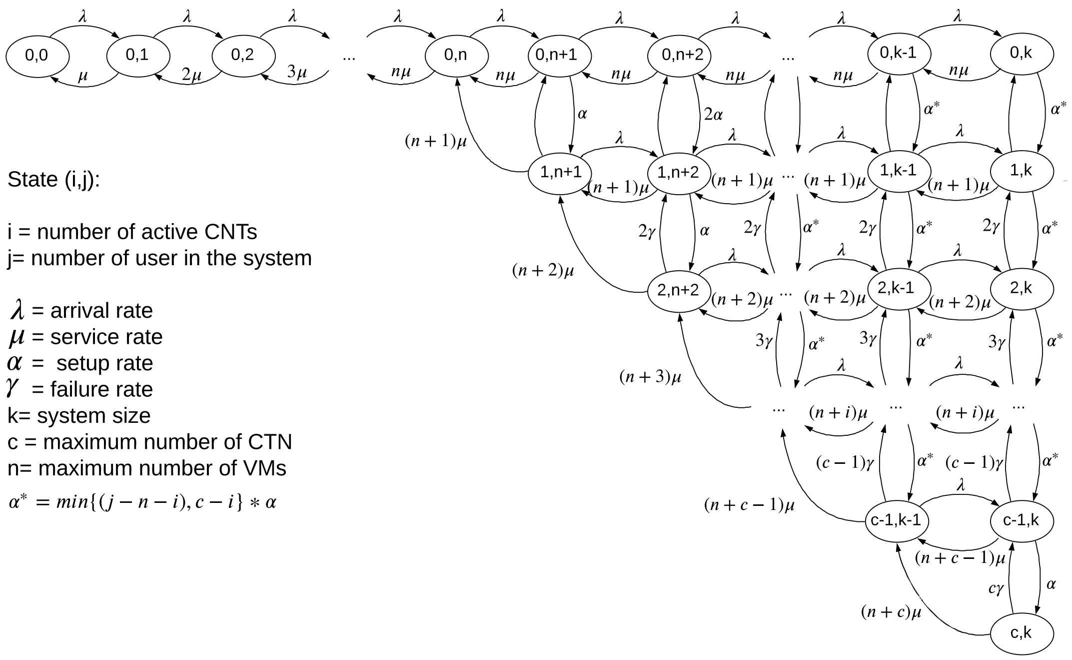


Figure 3. CTMC State Diagram

In this respect, we apply queueing theory for quantitative analysis. We assume that the URLLC networks are standalone deployment and the system is modeled as an M/M/k/k queue with setup time and failure, $n, c, k > 0$ and $n+c = k$ (no buffer allowed). The feasible state space is given by $\Omega = (i, j) \mid 0 \le i \le c,\ 0 \le j \le k$, provided that $i \le j - n$, with $i, j, n, c, k \in \mathbb{Z}^+$. Each state (i,j) designates the number of active containers and URLLC services, respectively. Since VMs are always active, regardless of being busy or idle, the states $(0, j)$ with $0 \le j \le n$ indicate that only VMs are processing the current load, whereas states with $j \ge n$ imply that in addition to all available VMs, containers are processing the extra load. Fig. 3 presents the state transition diagram of our system. Considering $(i, j) \in \Omega$ for all equations to follow, the steady-state probabilities $\pi(i, j)$ can be obtained from the linear system formed by the normalization condition (Eq. 1) and balance equations (Eqs. 2-10) in Table 2.

$$\sum_{\Omega} \pi_{i,j} = 1. \tag{1}$$

The balance-equation for the empty system $(0, 0)$ can be expressed in Eq. 2. For states where no container is active $(i = 0)$, there is at least one service $(j > 0)$ and there are enough VMs to process the admitted requests, the following will apply: Eq. 3 if $j < n$ and $n > 1$ or Eq. 4 if $j = n$. However, if the ongoing services overcome the existing VM limit and the system allows at least two containers and two services ($j > n, c > 1$ and $k > 1$), then only Eq. 5 applies. Finally, other border states are represented by Eq. 6, i.e., no container is on ($i = 0, j > 0$) and request limit k is reached ($j = k$ and $c > 0$).

$$\lambda\pi_{0,0} = \mu\pi_{0,1} \tag{2}$$

$$(\lambda + j\mu)\pi_{0,j} = \lambda\pi_{0,j-1} + (j+1)\mu\pi_{0,j+1} \tag{3}$$

$$(\lambda + n\mu)\pi_{0,n} = (\lambda\pi_{0,n-1}) + (n\mu\pi_{0,n+1}) + ((n+1)\mu\pi_{1,n+1}) \tag{4}$$

$$(\lambda + (min(c, j-n)\alpha) + n\mu)\pi_{0,j} = (\lambda\pi_{0,j-1}) + (n\mu\pi_{0,j+1}) + (\gamma\pi_{1,j}) \tag{5}$$

$$((min(c, k-n)\alpha) + n\mu)\pi_{0,k} = (\lambda\pi_{0,k-1}) + (\gamma\pi_{1,k}) \tag{6}$$

For the states where the number of active containers varies between $0 < i < c$, Eq. 7 only applies if the amount of active containers and VMs is equal to the number of active services ($j = i + n, j < k$ and $c > 1$), otherwise, if the active service number is bounded by that value and the limit k ($i + n < j < k$ and $k > n + 2$), then Eq. 8 is applicable. The remaining states covered by Eq. 9 are limited by k ($j = k$ and $c > 1$).

$$(\lambda + (n+i)\mu + i\gamma)\pi_{i,j} = ((n+i)\mu\pi_{i,j+1}) + (\alpha\pi_{i-1,j}) + (n+i+1)\mu\pi_{i+1,j+1} \tag{7}$$

$$\begin{aligned}&(\lambda + (n+i)\mu + ((min(c, j-n) - i)\alpha + i\gamma))\pi_{i,j}\\ &= \lambda\pi_{i,j-1} + ((min(c, j-n) - (i-1))\alpha\pi_{i-1,j}) + ((n+i)\mu\pi_{i,j+1}) + ((i+1)\gamma\pi_{i+1,j})\end{aligned} \tag{8}$$

$$\begin{aligned}&((n+i)\mu + ((min(c, k-n) - i)\alpha + i\gamma)\pi_{i,k}\\ &\qquad = ((min(c, k-n) - (i-n))\alpha\pi_{i-1,k}) + (\lambda\pi_{i,k-1}) + ((i+1)\gamma\pi_{i+1,k})\end{aligned} \tag{9}$$

Lastly, Eq. 10 refers to the set of states where all containers are active ($i = c$) and the service limit ($j = k$) is reached.

$$((n+c) + i\gamma)\mu\pi_{c,k} = (\alpha\pi_{c-1,k}) \tag{10}$$

Table 2. Balance Equation Description

No.	Equation	States	Condition(s)	Meaning
(2)	$\lambda\pi_{0,0} = \mu\pi_{0,1}$	$(0,0)$	n/a	*Empty system*
(3)	$(\lambda + j\mu)\pi_{0,j} = \lambda\pi_{0,j-1} + (j+1)\mu\pi_{0,j+1}$	$(0,j)$	$(0 < j < n)$ and $(n > 1)$	*All services running on VMs*
(4)	$(\lambda+n\mu)\pi_{0,n} = (\lambda\pi_{0,n-1}) + (n\mu\pi_{0,n+1}) + ((n+1)\mu\pi_{1,n+1})$	$(0,j)$	$(j = n)$	*Existing VMs match service load*
(5)	$(\lambda + (min(c, j - n)\alpha) + n\mu)\pi_{0,j} = (\lambda\pi_{0,j-1}) + (n\mu\pi_{0,j+1}) + (\gamma\pi_{1,j})$	$(0,j)$	$(n < j < k)$, $(c > 1)$ and $(k > 1)$	*Service load surpasses VM capacity; Containers are setting up*
(6)	$((min(c,k - n)\alpha) + n\mu)\pi_{0,k} = (\lambda\pi_{0,k-1}) + (\gamma\pi_{1,k})$	$(0,j)$	$(j = k)$	*Similar to (5) but with Full system;*
(7)	$(\lambda+(n+i)\mu+i\gamma)\pi_{i,j} = ((n+i)\mu\pi_{i,j+1}) + (\alpha\pi_{i-1,j})$	(i,j)	$(0 < i < c)$, $(j = n+i)$ and $(c > 1)$	*All VMs are busy and some Containers are serving;*
(8)	$(\lambda + (n + i)\mu + ((min(c, j - n) - i)\alpha + i\gamma))\pi_{i,j} = ((min(c, j - n) - (i - 1))\alpha\pi_{i-1,j}) + ((n + i)\mu\pi_{i,j+1}) + ((i + 1)\gamma\pi_{i+1,j})$	(i,j)	$(0 < i < c)$, $(n+i < j < k)$ and $(k > n+2)$	*Similar to (7) but there are containers setting up*
(9)	$((n + i)\mu + ((min(c, k - n) - i)\alpha + i\gamma)\pi_{i,k} = ((min(c, k - n) - (i - n))\alpha\pi_{i-1,k}) + (\lambda\pi_{i,k-1}) + ((i+1)\gamma\pi_{i+1,k})$	(i,k)	$(0 < i < c)$, $(k > n+2)$ and $(c > 1)$	*Similar to (8) but with full system*
(10)	$((n + c)\mu + c\gamma)\pi_{c,k} = (\alpha\pi_{c-1,k})$	(c,k)	n/a	*All resources are being used and the system is full*

3.4. Performance Metrics

Recent advancements on the field of URLLC push forward to the adoption of a MEC-NFV environment for multiple network parts, e.g., core network and application functions, bringing cloud features to the extreme opposite side of the network (from a location point of view). This phenomenon allows reduced Latency as well as increased Reliability for such sensitive traffic. On the other hand, MEC-enabled UAV nodes are particularly restricted in terms physical computational resources and energy consumption levels, which highly impact its availability. In other words, if maximum node capacity is reached either due to the high demand or to low battery levels, requiring the node to power off part of its already scarce resources [Li et al. 2021], the node must forward the excessive flow to a neighbor (ground/aerial node) or to the central cloud [Chen et al. 2020], both of which incurs on a new challenge since multiple intermediate hops might be introduced, causing some degree of uncertainty towards reliability. Thus, guarantying maximum MEC-enabled UAV node availability and reliability becomes specially relevant and challenging considering the URLLC environment.

3.4.1. Availability

In our framework, the Availability (A) quantifies the probability that an incoming URLLC request finds a vacant VM-hosted or containerized VNF at the MEC-UAV node, being given by Eq. 11, which is obtained by the probability sum of all states except those of full capacity.

$$A = 1 - \sum_{i=0}^{c} \pi_{i,j} \text{ with } j = k. \tag{11}$$

3.4.2. Reliability

Another relevant metric that quantifies the degree of successful task execution. In our system, the Reliability (R) is the probability that a URLLC service is served without experiencing failures in the MEC-enabled UAV, given by Eq. 12, which combines the admitted flow (λ*A) with the overall failure rate (effective).

$$R = 1 - \frac{1}{\lambda * A} \sum_{i=1}^{c} \sum_{j=1}^{k} i\gamma\pi_{i,j}. \tag{12}$$

4. Results

This section analyzes the proposed metrics focusing on a single MEC-enabled UAV node in terms of VMs and containers, serving critical URLLC applications. Following a subset from the 3GPP Release 16 (TR 38.824), the service time is set to 1ms (1 service/ms) while service arrivals range from 1 to 5 requests/ms. Unless otherwise stated, the baseline for failure (γ) and setup rates (α) are 0.001 and 1 unit/ms, respectively [Kaur et al. 2017]. The total number of VNFs in the following experiments is restricted to 10 units, which denotes the capacity of a small MEC-enabled UAV, although the proposed model is not limited to a specific amount of VNFs. The default network parameters are summarized in Table 3 while alternative configurations are indicated in Figs. 4-6.

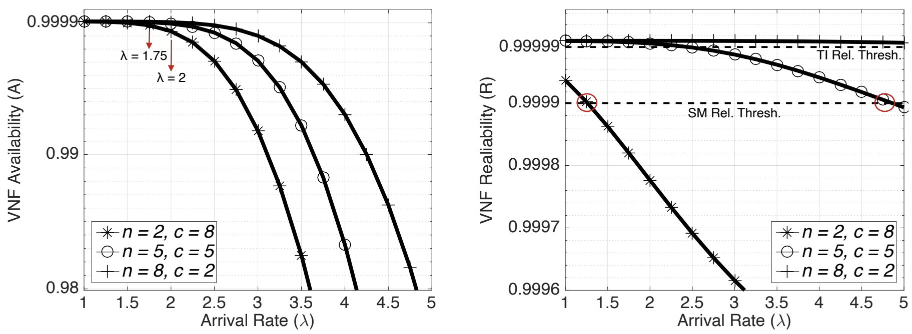
Table 3. Default Parameter Settings

Parameter	Values
Max. number of VNFs (k)	10
URLLC request rate (λ)	$[1, 5]$
URLLC service rate (μ)	1
Container setup rate (α)	1
Container failure rate (γ)	0.001

Three scenarios are evaluated in terms of multiple VM-hosted/containerized VNF arrangements, and improved setup/failure rates. The analytical model is validated against extensive discrete-event simulations in Figs. 4-6, where the analytical model is denoted by lines whereas markers represent simulation results.

4.1. Multiple VM/Container Arrangements

The impact of varying VM/container ratios is registered in Figs. 4a-4b. The results reveal that configurations with smaller VM/containers ratios are more likely to underperform with regard to both availability and reliability, which is somehow expected since containerized VNFs are less stable compared to VM-hosted VNFs. In particular, for both metrics,



(a) Availability **(b)** Reliability

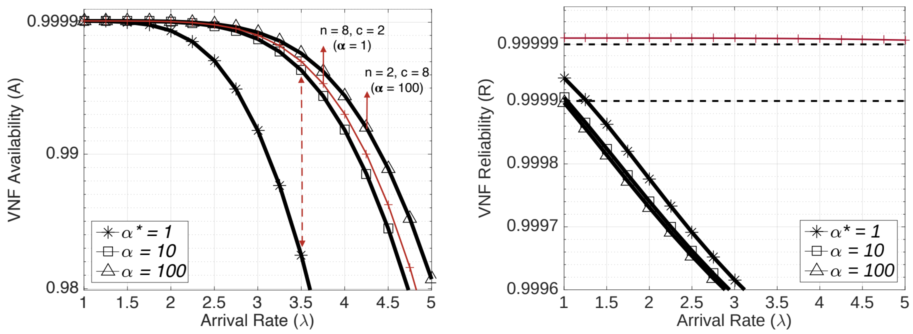
Figure 4. MEC node Availability and Reliability for multiple VM/container arrangements

we expected similar results between configurations at least until $\lambda = 2$, since theoretically, this is the load in which all tested configurations have enough VMs $n = (2, 5, 8)$. However, in Fig. 4a the curves start to differentiate at $\lambda = 1.75$ with larger discrepancies becoming clear as λ increases. As soon as all VMs occupied by URLLC requests, new incoming requests are sent to containerized VNFs. Yet, containers require a startup time, which affects node availability since, although it is not yet processing, it no longer counts as an available resource. This interval causes the service limit on the MEC-enabled UAV node to be reached.

In Fig. 4b the MEC-enbaled UAV configuration with 2 VMs kicks off with considerably lower reliability, breaking the Smart Manufacturing (SM) threshold at $\lambda = 1.25$ and drifting away rapidly in comparison to other configurations. Differently from the availability, the reliability standards are determined for URLLC, depending on the application. For instance, the Smart Manufacturing (SM) and Transport Industry (TI) thresholds are 99.99% and 99.999%, respectively (3GPP Rel-16).

In Fig. 4b, the only fairly compatible arrangement with all depicted applications for every λ is the configuration with 8 VMs and 2 containers. Curiously, the set with 5 VMs and 5 containers rapidly degrades even though the maximum λ value matches with the VM number, breaking two application thresholds. In particular, considering URLLC applications, even relatively few requests experiencing setup delays and one or multiple failures can cause capacity issues, leading to unavailability and lower reliability as more containerized VNFs are needed. On the other hand, building a platform formed only by VMs would likely become costly in terms of power consumption since VMs are not fast enough to be dynamically scaled for URLLC applications.

A possible solution is to adjust the VM/container ratio according to the demand, i.e., an operator can select configurations with more containers for low demands and vice-versa. For instance, considering only the reliability in Smart Manufacturing applications (Fig. 4b), the MEC-enabled UAV configuration with 2 VMs and 8 containers could be used if $\lambda < 1.25$, whereas if $\lambda < 4.75$ a more balanced set (5 VMs and 5 containers) could take place and finally, the arrangement with 8 VMs, 2 containers would only become available if $\lambda > 4.75$.



(a) Availability **(b)** Reliability

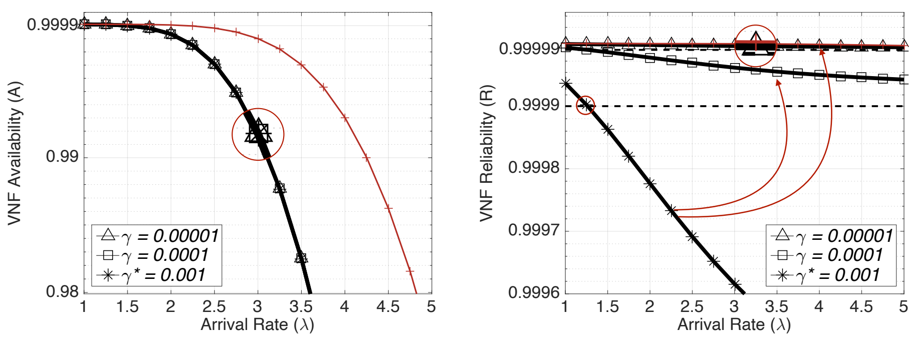
Figure 5. MEC node Availability and Reliability for improved setup rates

4.2. Improved Setup Rates

A single configuration with 2 VMs and 8 containerized VNFs was adopted with varying setup rates (α) in this section. A larger α means smaller container setup delays, i.e., more VNF instances become available per unit time. In the following lines, the impact of improved α is assessed, considering ten and one-hundred times the original baseline values while keeping the same failure rates.

In Fig. 5a the availability significantly increases as α improves, implying a throughput increase since the remaining parameters are fixed. It was noted that the curve in which $\alpha = 100$ surpasses the one depicted by the configuration with 8 VMs, 2 containers ($\alpha = 1$, red line with crosses) from the previous experiment, even though there are four times more stable VMs. Despite that, Fig. 5b reveals the opposite: interestingly, for smaller setup delays the reliability decreases, pushing the configuration away from the SM and TI reliability thresholds. This occurs since shorter setup delays lower the probability of a service to be processed in a VM, i.e., containerized VNFs become more available. This indicates that isolated improvements in the containerized VNF setup rate may concomitantly aid the admission process and become a burden to the admitted URLLC flows.

4.3. Improved Failure Rates



(a) Availability **(b)** Reliability

Figure 6. MEC node Availability and Reliability for improved failure rates

Similarly to the previous experiment, a fixed configuration with 2 VMs and 8 containerized VNFs was adopted, but with varying failure rates (γ). A larger γ means smaller intervals between successive containerized VNF failures. In Fig. 6a, it was noted that, although distinct failure rate magnitudes were tested, the availability remains almost unchanged in relation to the baseline configuration. On the contrary, the failure rate reduction has a deep effect on the edge reliability (Fig. 6b), allowing the configuration to be suited for the Smart Manufacturing threshold when $\gamma = 0.0001$ and even the Transport Industry threshold when $\gamma = 0.00001$, besides showing similar behavior compared to the configuration with 8 VMs, 2 containers ($\gamma = 0.001$, red line with crosses).

These findings suggest what level of improvement that containerized VNFs must achieve to meet specific requirements, recalling that both software and hardware share relevance towards this aspect [Lal et al. 2017]. This clearly indicates that the containerized VNF setup delays critically impact the admission process, whereas reliability is more reactive towards VNF failures. Besides, the joint analysis of availability and reliability reveals the amount of traffic that would experience uncertainty by being dodged to other MEC nodes besides the effective failure rate for the admitted flow, both of which should be considered by a mobile operator when opting for a given infrastructure set.

5. Conclusion

This paper introduced a dynamic resource allocation framework for MEC-enabled UAV nodes leveraging NFV, which allows mobile service providers to address multiple problems such as node dimensioning. To yield the framework practical, we characterize the MEC-enabled UAV node with the following assumptions: a hybrid virtualization environment with continuously powered VMs and dynamically scaled failure-prone containerized VNFs. In particular, we evaluated the combined impact of setup (repair) delays and failures by individually quantifying node Availability and Reliability, which revealed that individual containerized VNF setup/failure rate improvements may not positively impact the overall performance in every case. Ongoing works include power consumption and end-to-end latency analysis and model extension to cover other features such as different users (e.g., enhanced Mobile Broadband and URLLC) sharing the node resources. Another extension is to generalize the VNF setup time, since there is no literature to support that when they are exponential random variables.

Referências

[3GPP 2019] 3GPP (2019). *3GPP TS22.125 v17.1.0, Unmanned Aerial System (UAS) support in 3GPP, stage 1 (release 17).*

[Anand and de Veciana 2018] Anand, A. and de Veciana, G. (2018). Resource allocation and harq optimization for urllc traffic in 5g wireless networks. *IEEE Journal on Selected Areas in Communications*, 36(11):2411–2421.

[Bekkouche et al. 2020] Bekkouche, O., Samdanis, K., Bagaa, M., and Taleb, T. (2020). A service-based architecture for enabling uav enhanced network services. *IEEE Network*, 34(4):328–335.

[Borja et al. 2018] Borja, N., Sanchez-Aguero, V., Vidal, I., and Valera, F. (2018). Adaptable and automated small uav deployments via virtualization. *Sensors*, 18:4116.

[Chen et al. 2020] Chen, Z., Xiao, N., and Han, D. (2020). Multilevel task offloading and resource optimization of edge computing networks considering uav relay and green energy. *Journal of Applied Sciences*, 10:2592–2601.

[Cheng et al. 2018] Cheng, N., Xu, W., Shi, W., Zhou, Y., Lu, N., Zhou, H., and Shen, X. (2018). Air-ground integrated mobile edge networks: Architecture, challenges, and opportunities. *IEEE Communications Magazine*, 56(8):26–32.

[Costanzo et al. 2020] Costanzo, F., Lorenzo, P. D., and Barbarossa, S. (2020). Dynamic resource optimization and altitude selection in uav-based multi-access edge computing. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4985–4989.

[Emara et al. 2021] Emara, M., ElSawy, H., Filippou, M. C., and Bauch, G. (2021). Spatiotemporal dependable task execution services in mec-enabled wireless systems. *IEEE Wireless Communications Letters*, 10(2):211–215.

[Falcao et al. 2021] Falcao, M., Souza, C., Balieiro, A., and Dias, K. (2021). An analytical framework for urllc in hybrid mec environments. *Journal of Supercomputing*.

[Fautrel et al. 2019] Fautrel, T., George, L., Fauberteau, F., and Grandpierre, T. (2019). An hypervisor approach for mixed critical real-time uav applications. In *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pages 985–991.

[Hu et al. 2019] Hu, Q., Cai, Y., Yu, G., Qin, Z., Zhao, M., and Li, G. Y. (2019). Joint offloading and trajectory design for uav-enabled mobile edge computing systems. *IEEE Internet of Things Journal*, 6(2):1879–1892.

[Jeong et al. 2018] Jeong, S., Simeone, O., and Kang, J. (2018). Mobile edge computing via a uav-mounted cloudlet: Optimization of bit allocation and path planning. *IEEE Transactions on Vehicular Technology*, 67(3):2049–2063.

[Kaur et al. 2017] Kaur, K., Dhand, T., Kumar, N., and Zeadally, S. (2017). 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.

[Lal et al. 2017] Lal, S., Ravidas, S., Oliver, I., and Taleb, T. (2017). Assuring virtual network function image integrity and host sealing in telco cloue. In *2017 IEEE International Conference on Communications (ICC)*, pages 1–6.

[Li et al. 2021] Li, W.-T., Zhao, M., Wu, Y.-H., Yu, J.-J., Bao, L.-Y., Yang, H., and Liu, D. (2021). Collaborative offloading for uav-enabled time-sensitive mec networks. *EURASIP Journal on Wireless Communications and Networking*.

[Li et al. 2019] Li, X., Yao, H., Wang, J., Xu, X., Jiang, C., and Hanzo, L. (2019). A near-optimal uav-aided radio coverage strategy for dense urban areas. *IEEE Transactions on Vehicular Technology*, 68(9):9098–9109.

[Lyu et al. 2017] Lyu, J., Zeng, Y., Zhang, R., and Lim, T. J. (2017). Placement optimization of uav-mounted mobile base stations. *IEEE Communications Letters*, 21(3):604–607.

[Morabito 2015] Morabito, R. (2015). Power consumption of virtualization technologies: An empirical investigation. In *2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC)*, pages 522–527.

[Mozaffari et al. 2019] Mozaffari, M., Taleb Zadeh Kasgari, A., Saad, W., Bennis, M., and Debbah, M. (2019). Beyond 5g with uavs: Foundations of a 3d wireless cellular network. *IEEE Transactions on Wireless Communications*, 18(1):357–372.

[Nasir et al. 2019] Nasir, A. A., Tuan, H. D., Duong, T. Q., and Poor, H. V. (2019). Uav-enabled communication using noma. *IEEE Transactions on Communications*, 67(7):5126–5138.

[Nielsen et al. 2018] Nielsen, J. J., Liu, R., and Popovski, P. (2018). Ultra-reliable low latency communication using interface diversity. *IEEE Transactions on Communications*, 66(3):1322–1334.

[Ren et al. 2016] Ren, Y., Phung-Duc, T., Yu, Z.-W., and Chen, J.-C. (2016). Design and analysis dynamic auto scaling algorithm (dasa) for 5g mobile networks.

[Santoyo-González and Cervelló-Pastor 2018] Santoyo-González, A. and Cervelló-Pastor, C. (2018). Edge nodes infrastructure placement parameters for 5g networks. In *2018 IEEE Conference on Standards for Communications and Networking (CSCN)*, pages 1–6.

[Sarrigiannis et al. 2020] Sarrigiannis, I., Ramantas, K., Kartsakli, E., Mekikis, P.-V., Antonopoulos, A., and Verikoukis, C. (2020). Online vnf lifecycle management in an mec-enabled 5g iot architecture. *IEEE Internet of Things Journal*, 7(5):4183–4194.

[Strinati et al. 2020] Strinati, E. C., Barbarossa, S., Choi, T., A.Pietrabissa, Giuseppi, A., Santis, E. D., Vidal, J., Becvar, Z., Haustein, T., N.Cassiau, Costanzo, F., Kim, J., and Kim, I. (2020). 6g in the sky: On-demand intelligence at the edge of 3d networks. *ETRI Journal*, 42(5):643–657.

[Wang et al. 2019] Wang, L., Che, Y. L., Long, J., Duan, L., and Wu, K. (2019). Multiple access mmwave design for uav-aided 5g communications. *IEEE Wireless Communications*, 26(1):64–71.

[Wang et al. 2021] Wang, L., Wang, K., Pan, C., Xu, W., Aslam, N., and Hanzo, L. (2021). Multi-agent deep reinforcement learning-based trajectory planning for multi-uav assisted mobile edge computing. *IEEE Transactions on Cognitive Communications and Networking*, 7(1):73–84.

[Xiong et al. 2019] Xiong, J., Guo, H., and Liu, J. (2019). Task offloading in uav-aided edge computing: Bit allocation and trajectory optimization. *IEEE Communications Letters*, 23(3):538–541.

[Yang et al. 2019] Yang, Z., Pan, C., Wang, K., and Shikh-Bahaei, M. (2019). Energy efficient resource allocation in uav-enabled mobile edge computing networks. *IEEE Transactions on Wireless Communications*, 18(9):4576–4589.

[Zeng et al. 2019] Zeng, Y., Wu, Q., and Zhang, R. (2019). Accessing from the sky: A tutorial on uav communications for 5g and beyond. *Proceedings of the IEEE*, 107(12):2327–2375.

[Zhang et al. 2019] Zhang, J., Zhou, L., Tang, Q., Ngai, E. C.-H., Hu, X., Zhao, H., and Wei, J. (2019). Stochastic computation offloading and trajectory scheduling for uav-assisted mobile edge computing. *IEEE Internet of Things Journal*, 6(2):3688–3699.