Noname manuscript No.
(will be inserted by the editor)

An Analytical Framework for URLLC in Hybrid MEC Environments

Marcos Falcao · Caio Bruno Souza · Andson Balieiro · Kelvin Dias

Received: date / Accepted: date

Abstract The conventional mobile architecture is unlikely to cope with Ultra-Reliable Low-Latency Communications (URLLC) constraints, being a major cause for its fundamentals to remain elusive. Multi-access Edge Computing (MEC) and Network Function Virtualization (NFV) emerge as complementary solutions, offering fine-grained on-demand distributed resources closer to the User Equipment (UE). This work proposes a multipurpose analytical framework that evaluates a hybrid virtual MEC environment that combines VMs and Containers strengths to concomitantly meet URLLC constraints and cloud-like Virtual Network Functions (VNF) elasticity.

Keywords Multi-access Edge Computing · Ultra-Reliable Low-Latency Communications · Network Function Virtualization. · Continuous-Time Markov Chains

1 Introduction

Ultra-Reliable Low-Latency Communications (URLLC) require Multi-access Edge Computing (MEC) as one of the enabling technologies to concomitantly fulfill the extremely low-latency and high reliability requirements [1]. Both latency and reliability can be significantly enhanced by placing application/core functions in close proximity to the User Equipment (UE). Nevertheless, MEC itself opens new challenges such as guaranteeing service level agreements (SLAs) with low capacity nodes while keeping infrastructure costs to a minimum, despite its highly distributed nature.

In such a challenging environment, the underlying virtualization entities play a key role on resource provisioning. In general, communication infrastructure provisioning must be designed under two perspectives: (a) application re-

Marcos Falcao, Caio Bruno Souza, Andson Balieiro and Kelvin Dias
Centro de Informática, Universidade Federal de Pernambuco, Brasil
E-mail: {mrmf, cbbs, amb4, kld}@cin.ufpe.br

quirements and (b) financial cost. In particular for URLLC, both can be highly impacted by a number of virtualization-related (e.g., operation overheads, security risks) and edge node configurations aspects (e.g., node availability). For instance, the edge node cost from the service provider perspective can be affected if node infrastructure is overestimated, focusing solely on keeping maximum URLLC application performance or may lower costs at an expense of breaking several SLAs in case of underestimation. In this respect, to optimize MEC resources for URLLC, it is imperative for an operator to analyze the impact of these aspects on both performance and infrastructure cost.

Existing studies have approached MEC resource provisioning issues for the fifth generation (5G) and beyond 5G (B5G) of mobile communication networks in much the same way as conventional data centers, i.e., ignoring phenomenons that are irrelevant in the latter scenario but that may impact the first. For example, the virtualization layer setup time overhead [3] or the failure aspects [4] are rarely assessed in large scale systems since there is no significant impact upon the existing application and due to the resource amount, fact that does not hold for small MEC nodes serving URLLC. In this work, we exploit a Continuous Time Markov Chain (CTMC) framework that incorporates the main assumptions regarding a hybrid virtual platform for MEC nodes and is suitable for several classes of problems such as dimensioning, edge node placement, and dynamic resource provisioning for URLLC. We also derive relevant performance metrics to provide means for building optimal provisioning strategies and better understand the relationship between several parameters under various scenarios, including multiple MEC node sizes, virtual resource types, setup/failure rates, and traffic loads.

The remainder of this paper is organized as follows. Section 2 presents the related work advancements in the field of MEC. Section 3 describes the system model encompassing the basic assumptions, and Section 4 depicts the proposed framework. Validation and numerical results are evaluated in Section 5, and Section 6 concludes this work.

2 Related Work

There is no current consensus on the size, computational power, virtualization technology nor on the best location of MEC nodes [5]. In fact, MEC has not yet been clearly defined; neither functionally nor physically, and since URLLC offers more challenges due to its stringent requirements and lack of real-world data compared to other 5G categories, multiple works fail to provide adequate considerations in order to evaluate MEC-NFV-enabled networks for URLLC. This section first describes the challenges addressed by previous authors and the underlying virtual environment assumptions concerning the MEC-NFV infrastructure.

2.1 Optimization Targets

A body of existing works on MEC-related computational resource issues encompasses multiple classes of problems. The most common can be categorized in resource/service placement, edge node dimensioning, and dynamic resource allocation (DRA) [6].

Placement and scheduling in MECs consider applications with two or more components, one typically running on a cloud (MEC or central cloud) and the other on the UE. Regarding this class of problems, [7] proposed a placement scheme as a simple optimization problem with two conflicting objectives, namely minimizing access latency and maximizing service availability. In [8], the authors consider a more dynamic standpoint, i.e., the data placement needs to be adapted to serve time-varying demands, while considering system stability and operation cost under communication, computation, and storage constraints.

Edge node dimensioning usually decides on the computational resource characteristics based on a given traffic load, e.g., the total number of servers, processing capacity, and storage. In [9], a delay and pricing model to supply equitable resources to UEs and minimize network delay and price was suggested. Similarly, [10] designs an algorithm to find the minimum number of MEC servers considering both delay and workload budget. In [11], the authors propose an analytical framework to identify the optimal number of virtual resources to maximize the task execution capacity.

Lastly, DRA relates to resource provisioning optimization given both the maximum edge node dimensions and the expected traffic load range, which allows a Service Provider to adjust the existing computational resources dynamically. [12] proposed a DRA algorithm that minimizes the end-to-end delay while ensuring the minimum service rate and maximum reliability. Another example can be found in [13], which investigated a DRA approach accounting to minimize edge Service Level Agreement (SLA) violations and maximize the serving users.

In some cases, a single formulation can be applied to multiple problem classes. For instance, in [14] the author jointly solves 1) a MEC dimensioning sub-problem, 2) an application placement sub-problem, and 3) a workload assignment sub-problem. Besides, some works propose multipurpose formulations in order to provide guidelines for the design of the URLLC architecture, such as in [15], where the authors do not target a specific problem, but rather provide a general formulation for investigating how to optimize queue-related parameters to reduce the delay for URLLC (e.g., Arrival and Service Rate).

The existing performance evaluation research is focused on specific implementations to fulfill optimization problems. Unlike most previous works, we propose a multipurpose analytical framework that encompasses a flexible and realistic analysis that is compatible with the MEC-NFV environment and URLLC, besides enabling service providers to rapidly tune multiple parameters to satisfy the Quality of Service (QoS) requirements for a particular URLLC application.

2.2 Edge Position

Under the MEC paradigm, edge nodes can significantly differ in their deployment location. Fig. 1 describes the traditional communication path from UE to a service hosted in a central cloud, which includes: the backhaul, core network (CN) and cloud host. The three alternatives exemplify the edge placement variants that could emerge to handle 5G-related applications. While the first commercial deployments follow the Far variant, the increasing uncertainty brought by each additional intermediate hop may deeply affect the performance (blue line) for critical URLLC applications. In contrast, fine-grained server distribution is known to increase the overall infrastructure cost (red line) and management complexity [5].

While some authors place computation capacities within Radio Access Network (RAN) sites, others prefer a farther away location, similarly to centralized data centers but introducing new components and inter-working procedures to ensure better performance. In [10] and [12], the analysis covers the full path from the RAN cluster to the MEC node, including some core functions and application layer besides considering the edge servers placed exclusively on the Near Scope (Fig. 1). Similarly, [13] and [9] also consider the Near Scope solely but excludes the RAN analysis focusing only on their optimization schemes for MEC resources. On the contrary, [7] assumes a more flexible approach, allowing an operator to either install the MEC servers close or far away from the end-users. Finally, the analytical framework in [11] allows a high degree of flexibility since there are separate models for the RAN and MEC, i.e., the RAN model's output flow is one of the inputs to the MEC model. Table 1 summarizes this classification.

Concerning edge node positioning, the work closest to ours is [11]. In summary, both frameworks offer no restrictions towards the total Edge node size nor any underlying considerations that compel the edge node position to a certain location from the UE (Near, Mid, or Far). Hence, it is possible to shift the edge node position towards the UE (Near Scope) or central cloud (Far Scope), alter the total number of resources and other parameters (e.g., processing capacity), and still evaluate URLLC traffic behavior inside the edge node in isolation from the RAN and Core network. Besides, provided formulations for other intermediate network subparts it is possible to evaluate ent-to-end requirements.

2.3 Virtual Host Types and Assumptions

MEC is frequently put forward considering virtualization instead of legacy physical equipment. However, the literature consistently brought regular data center architectural approaches as if it could also be applied seamlessly and without greater modifications to the mobile environment, e.g., [16]. For this reason, some works such as [8], [10], [13], [12] and [15] are agnostic towards a

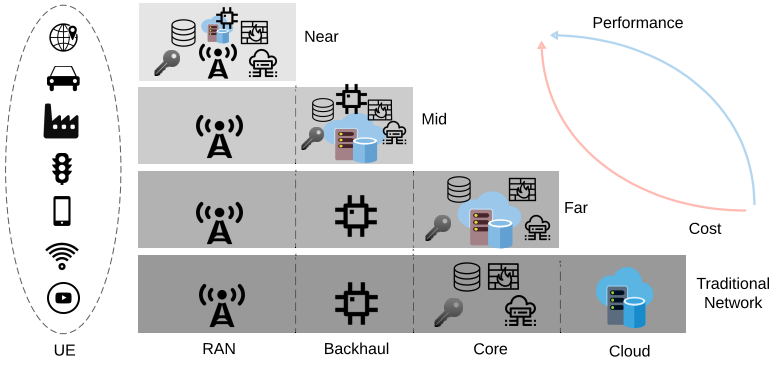


Fig. 1: Multiple edge node deployment scopes.

given virtualization technology, which denotes a certain lack of commitment to the feasibility of their propositions.

Moreover, multiple works have proposed a resource infrastructure built only by physical machines (PMs) [10], Virtual Machines (VMs) as in [11] and [14] or a mix of both as in [7]. However, although NFV has traditionally been implemented over VMs, the concept of Container-as-a-Service (CaaS) has gained momentum. In contrast to VM-based VNFs, CaaS allows VNF instances to be loaded using containers, which are known to consume less computational resources, besides having less instantiation overhead and thus being much more cost-effective [17]. Therefore, some authors consider MEC infrastructure using only containers [9].

The main problem of accepting containers as the single virtualization component for future mobile communications is that they are still not mature compared to VMs. There are multiple security risks involved in containerization since all containers in an OS share a single kernel. Hence, any breach on kernel OS can break down all containers dependent on it. Besides, isolating a fault is not easy with containers and a fault can be replicated to subsequent instances. On the other hand, containers can be used along with VMs (hybrid) in NFV environments. In our framework, we explore this property by allowing a hybrid VM-containerized infrastructure that leverages the best of both: the VM's strong isolation and the flexibility of containers.

The appropriate selection of the underlying virtual environment directly impacts the feasibility of the NFV-MEC environment. In particular, multiple works have ignored the possibility of faults related to the virtual host, which can be key for many of the afore-mentioned problem categories, e.g., if a resource dimensioning strategy does not account to the resource failure possibility, the resulting MEC dimension will likely be underestimated. We believe that an accurate model should account for virtual host failures and the repair delay, which can be found only in [11]. The works [7], [9] and [12] do not account for the repair delay, which directly impacts the resource availability and power consumption.

2.4 Performance Metrics for URLLC

From the 3GPP Release 16 onwards, potential architecture enhancements for supporting URLLC services focused on MEC-NFV have been issued. With regards to the performance metrics, in addition to the most representative metrics for URLLC, i.e., reliability and latency, the document also includes the MEC resource availability. In the following lines, we have mapped which indicators have been used in the literature. Please note that the nomenclature and definition might differ from author to author, but for classification purposes, we have grouped those that are strongly correlated, e.g., latency response time and network delay.

From the above-mentioned list, the only adopted metric in [13] was network delay. The works [15] and [12] only evaluate reliability and latency. In [14] and [7], the authors only accounted for availability and latency (network delay). Differently, in [9], besides reliability and a latency-related indicator (Network Delay), energy consumption was also analyzed. Similarly, [11] covers availability and reliability with an energy constraint per device. Lastly, in [10], both delay and energy constraints are considered, while availability and reliability are left out of scope.

Besides the performance indicators suggested by 3GPP Release 16, some authors also adopt the energy-related indicators. From the user perspective, the suggested indicators should be reasonable, however, from the service provider perspective, infrastructure cost metrics are equally relevant. The problem is that the list of location-dependent costs for building and operating edge nodes can be quite extensive, going from land acquisition to installation expenses. Besides, it is not feasible to numerically map many of these variables since they are not universal. One of the few exceptions is the computational power consumption, which does reflect part of the operational costs. Thus, together with 3GPP metrics, we have included power consumption in the evaluation so as to make our results strongly-coupled with the Service Provider reality. The resulting classification from sections 2.1-2.4 are summarized in Table 1.

Table 1: Summary of Existing Related Works

Work	Optimization Target	Edge Position	Virtual Host Type	Virtual Host Assumptions	Performance Metrics - URLLC Availability	Reliability	Energy	Latency
[7] Yala et al.	Placement	Flexible	VMs	Failure	✓	✗	✗	✓
[8] Farhadi et al.	Placement	Near	n/a	n/a	✓	✗	✓	✗
[9] Samanta et al.	Dimensioning	Near	Container	Failure	✗	✓	✓	✓
[10] Lee et al.	Dimensioning	Near	n/a	n/a	✗	✗	✓	✓
[11] Emara et al.	Dimensioning	Flexible	VM	Failure/Repair	✓	✓	✓	✗
[12] Tong et al.	DRA	Near	n/a	Failure	✗	✓	✗	✓
[13] Sarrigiannis et al.	DRA	Near	n/a	n/a	✗	✗	✗	✓
[14] Kherraf et al.	Multipurpose	Near	VM	n/a	✓	✗	✗	✓
[15] Ma et al.	Multipurpose	Flexible	n/a	n/a	✗	✓	✗	✓
This Work	Multipurpose	Flexible	Hybrid	Failure/Repair	✓	✓	✓	✓

Contribution We propose a multipurpose analytical framework that encompasses a flexible analysis that is compatible with the MEC-NFV environment and URLLC, which enables service providers to tune multiple network and infrastructure parameters. The framework allows the evaluation of URLLC traffic behavior inside the edge node in isolation from the RAN and Core network, not being restricted by the edge node position nor a particular choice of virtual technology. Finally, taking into account the literature review and the particular URLLC requirements, the proposed framework includes formulation for the main performance metrics: Availability, Reliability, Power Consumption and Latency.

3 System Model

Thousands of Edge Nodes are expected to be distributed within large areas in upcoming URLLC scenarios. Hence analytical frameworks can be useful tools for rapidly evaluating MEC-related infrastructure projects. In this work, we evaluate a single isolated MEC node depicted in Fig. 2, where the URLLC requests originated from UEs are processed by the RAN, passed on to the MEC node and is handled by a VM-hosted (red flow) or a containerized VNF (blue flow). We have designed our framework in isolation from RAN, Core, and Central Cloud, i.e., rather than accounting for the multiple network path subparts, the only uncertainty is brought by the internal MEC components covered in further sections. This brings two advantages: the flexibility for adapting to multiple Edge node sizes and the precision due to the limited variables that can affect the QoS.

Although we did not account for a specific class of resource problems (e.g., Dimensioning), a dynamic VNF scaling strategy was embedded into our formulation to help cope with the sudden load increase caused by the intensive requests. Each VNF runs equally and independently on a single VM or container, with VMs executing uninterruptedly while containers are scaled upon demand. A centralized control unit determines if requests are admitted or blocked, only activating containers when all VMs are busy.

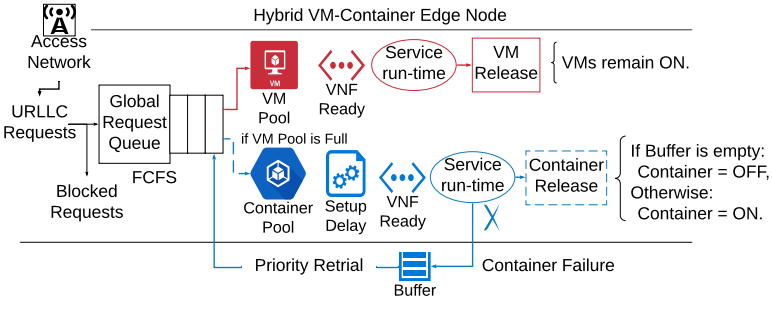


Fig. 2: Hybrid VM-Container edge node infrastructure

The containerized VNF activation comprises two phases: initializing the kernel image and launching the specified function, which is interpreted as a single transition interval (setup time), during which power and resources are consumed but no request is processed. Furthermore, active containerized VNFs may suffer failures during attendance, which implies either a service migration to an available VM/container or a reset, triggering a new setup period, with progress being lost only in the latter case. In general, recovery time depends on the failure type; for instance, a software component crash can be quickly fixed by the host in few microseconds, while others may take a few milliseconds to reboot device and VNF. Since we are dealing exclusively with URLLC flows, only the worst-case scenario is considered.

Lastly, as soon as an operative VNF concludes processing and there are no remaining requests, the VNF instance can either be powered down instantaneously 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/recovery magnitudes [3].

4 Analytical Framework

The system comprises n VMs and c containers, with a maximum limit of k simultaneous URLLC services. The service request follows a Poisson process with rate λ (requests/ms) and server capacities of one service with an exponentially distributed service rate μ for both VM-based and containerized VNFs. A Poissonian arrival process was selected for its simplicity and tractability. URLLC control applications are likely 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 well modeled as a Poisson [18, 19]. Container setup/recovery 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 due failures.

Although the deployment details of the URLLC standard are not yet released, we can still apply queueing theory for quantitative analysis. We assume that the URLLC networks are standalone deployment and the system is modeled as an M/M/n+c/k queue with setup time and failure, $n, c, k > 0$ and $n + c \leq k$ [20]. The feasible state space (Ω) is formed by a set of states (i,j), which denotes the number of active containers (i) and online URLLC services (j). $\Omega = (i, j) \mid 0 \leq i \leq c, 0 \leq j \leq k$, provided that $i \leq j - n$. Since VMs are always active regardless of being busy or idle, the states $(0, j)$ with $0 \leq j \leq n$ indicates that the existing load is being processed in VMs, whereas states with $j \geq n$ imply that in addition to all available VMs, there are requests being processed in containers.

Table 2: Balance Equation Description

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

4.1 Performance Metrics

In this section, we consider the steady-state analysis of the CTMC under study, followed by the derivation of four performance metrics: Availability (A), Reliability (R), Mean Power Consumption (C) and Mean Response Time (T). The steady-state probabilities $\pi_{i,j}$ are extracted from the solution of a linear system formed by the normalization condition and balance equations (1-11) depicted in Table 2. Please consider $(i, j) \in \Omega$ in all equations to follow.

4.1.1 Availability (A)

It is widely accepted that the adoption of the MEC-NFV environment for Core Network and Application functions closer to the UE can reduce Latency and increase Reliability of URLLC services. However, the likely resource limitation of edge nodes restricts their service capacity and consequently its availability, i.e., if the maximum capacity is reached, the natural options are to forward the flow to a neighbor MEC node or central cloud [13], both of which incur

on a new route built of multiple intermediate hops, introducing a high degree of uncertainty towards latency and reliability. In this respect, it becomes imperative to analyze the MEC-NFV node availability for serving URLLC.

In our framework, 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 node Availability (A) (Eq.12) is obtained by the probability sum of all states except those representing full capacity, i.e., the system with k users.

$$A = 1 - \sum_{i=0}^{c} \pi_{i,k} \tag{12}$$

4.1.2 Reliability (R)

The reliability analysis of future mobile networks is of paramount importance for network operators, especially considering URLLC applications since it directly impacts the QoS and user experience. The designed framework also evaluates the Reliability (R) being given by Eq. 13, which combines the admitted flow λ^*A with the effective failure rate in the entire node, i.e., it denotes the probability that a URLLC service is served without experiencing failures while being processed by MEC VNFs.

$$R = 1 - \frac{\gamma}{\lambda * A} \sum_{i=1}^{c} \sum_{j=1}^{k} i\pi_{i,j}. \tag{13}$$

4.1.3 Power Consumption (C)

As depicted in Section 2.4, 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 tradeoff. In our framework, power consumption (C) Eq. 19 is formed from the combination of the mean number of virtual resources and energy consumption constants (P) for each virtualization technology (VM and Container) and operating states (Idle, Setup and Busy).

The mean number of VMs and containers (CT) in each state is described in Eqs. 14-18, which are detailed in the following lines. Eq. 14 captures the mean amount of VMs in idle state by iterating over each system state in which no container is active ($i = 0$) and until the total number of online services reaches the maximum amount of VMs ($j = n$). Eq. 15 has three terms: the first is similar to the one in Eq. 14, but captures only the mean amount of busy VMs within the range of states until ($j = n - 1$). The second iterates over the states where the load is equal or greater than the VM maximum amount and there are no containers ready yet. Lastly, the third term contains the mean

amount of busy VMs on states where at least one container is processing. The same idea is applied for Eq. 16 and Eq. 18, whereas Eq. 17 calculates the mean number of containerized VNFs in setup by iterating over states where the number of online services is greater than the total number of active resources (VMs and Containers).

$$\overline{VM}_{idle} = \sum_{j=0}^{n}(n-j)\pi_{0,j} \tag{14}$$

$$\overline{VM}_{busy} = \sum_{j=0}^{n-1} j\pi_{0,j} + \sum_{j=n}^{k} j\pi_{0,j} + \sum_{i=1}^{c}\sum_{j=n+i}^{k} j\pi_{i,j} \tag{15}$$

$$\begin{aligned}\overline{CT}_{idle} = \sum_{j=0}^{n} c\pi_{0,j} + \sum_{j=n+1}^{n+c}(n+c-j)\pi_{0,j} \\ + \sum_{i=1}^{c}\sum_{j=n+i}^{n+c}(n+c-j)\pi_{i,j}\end{aligned} \tag{16}$$

$$\overline{CT}_{setup} = \sum_{i=0}^{c}\sum_{j=n+i}^{k} min(j-n, c)\pi_{i,j} \tag{17}$$

$$\overline{CT}_{busy} = \sum_{i=1}^{c}\sum_{j=n+i}^{k} i\pi_{i,j} \tag{18}$$

The symbols used to denote the power consumption of each technology and state are summarized in Table 3.

Table 3: Notation for Power Consumption

Virtualization	State	Status	Energy
VM-hosted	Idle	ON	P_{idle}^{VM}
VM-hosted	Busy	ON	P_{busy}^{VM}
Containerized	Idle	SLEEP	P_{idle}^{CT}
Containerized	Setup	ON	P_{setup}^{CT}
Containerized	Busy	ON	P_{busy}^{CT}

$$\begin{aligned}C = P_{idle}^{VM}\,\overline{VM}_{idle} + P_{busy}^{VM}\,\overline{VM}_{busy} \\ + P_{idle}^{CT}\,\overline{CT}_{idle} + P_{setup}^{CT}\overline{CT}_{setup} + P_{busy}^{CT}\,\overline{CT}_{busy}\end{aligned} \tag{19}$$

4.1.4 Response Time (T)

Since URLLC applications have strict communication latency requirements, analyzing the response time of URLLC services also becomes crucial for MEC node resource-related issues such as dimensions. We define the Response Time of a VNF that processes the URLLC service T 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 Response Time (Eq. 21) is obtained by calculating the mean number of online URLLC services (Eq. 20) and dividing by the mean number of accepted services.

$$\overline{U} = \sum_{j=0}^{k} j\pi_{0,j} + \sum_{i=1}^{c} \sum_{j=n+i}^{k} j\pi_{i,j} \tag{20}$$

$$T = \frac{\overline{U}}{\lambda A} \tag{21}$$

5 Validation and Numerical Results

The analytical results were validated against extensive discrete-event simulations (Figs. 3-5), where the lines denote the analytical and the markers represent simulation results. With regards to the main parameters, we have followed a subset of the 3GPP Release 16 (TR 38.824), in which the service time is 1ms (1 service/ms) while service arrivals range from 1 up to 100 requests/ms. In addition, unless otherwise stated the baseline values for failure (γ) and setup rates (α) were 0.001 and 1 unit/ms, respectively, which is in accordance with [3]. In terms of energy power consumption for VMs and containers in different operation states, we adopted the individual consumption values described in [13], summarized in Table 4 with other general parameters adopted in the evaluation. The subsections 5.1-5.3 encompass three evaluation scenarios were we show the flexibility of the proposed framework in terms of multiple edge sizes (5.1), the impact of various VM-hosted/containerized VNF arrangements (5.2) and improved setup/failure rates (5.3).

5.1 Multiple Edge Sizes (k)

This scenario shows the impacts of λ on the different MEC-enabled node sizes ($k = 25, 50, 100$) but running the same VNF scaling process with equal VM or Containerized (CT) VNF ratios ($VMs = 40\%, CTs = 60\%$). Generally, one can see that in Figs. 3a-3b the best values for Availability and Reliability are found for low workloads, respectively. However, as λ increases, the Availability tends to always decrease and it is not bounded by a particular value. In addition, although the same VM/Containerized VNF ratios were used, the number

Table 4: Simulation Parameters

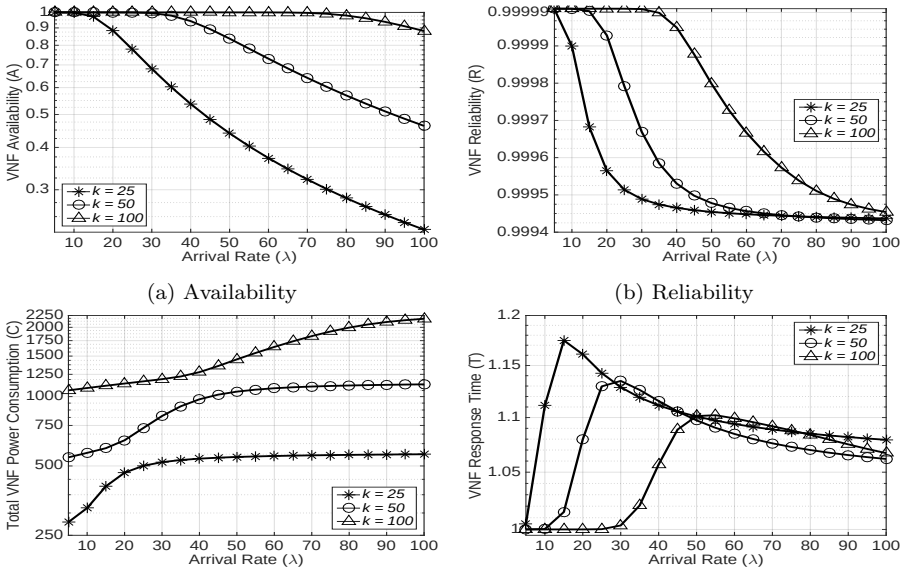
Parameter	Value
Arrival rate (λ)	[1 100] requests/ms
Service rate (μ)	1 service/ms
Failure rate (γ)	0.001 unit/ms
Setup rate (α)	1 unit/ms
Idle VM Energy Consumption (P_{idle}^{VM})	$20W$
Busy VM Energy Consumption (P_{busy}^{VM})	$25W$
Idle CT Energy Consumption (P_{idle}^{CT})	$4W$
Setup CT Energy Consumption (P_{setup}^{CT})	$8W$
Busy CT Energy Consumption (P_{busy}^{CT})	$23W$

of VM-hosted VNFs in each curve are different (n = 15, 20 and 40), which explains the discrepancy towards the results for this metric.

Contrarily, the Reliability is bounded by the number of accepted services, and since the same VM/Containerized VNF ratios were used, the curves are expected to overlap for higher λ values. The reasons are as follows. When $\lambda << n\mu$, the incoming URLLC services are usually handled by VMs, i.e., the containerized VNFs are hardly required. Moreover, they are turned on as λ approaches $n\mu$, negatively impacting the Reliability due to the increasing failures brought by the containerized VNFs. Finally, when λ approaches $(n+c)\mu$, most containerized VNFs are turned on, indicating a saturated infrastructure. The same reasoning applies for the Power Consumption in Fig. 3c. However, the curves do not overlap for higher λ since each configuration has a different absolute number of resources.

The impact of the increasing load on the mean response time (Fig. 3d) illustrates the only metric in which the trend of the curves experiences both an ascent and descent phase. In the first phase, the response time grows sharply due to the containerized VNF setup delay. Specifically, smaller response delays will happen invariably when $\lambda << n\mu$ because most services will be handled by VM-hosted VNFs. As λ approaches $n\mu$ and becomes larger, containerized VNFs are turned on. In this phase, however, the response time still increases due to the setup delay. The third phase encompasses the moment when the mean response time begins to decrease in λ = 20, 32.5, and 60 respectively for $k = 25, 50, 100$. Such behavior occurs since the containerized VNFs tend to become readily available for new service arrivals, not needing to wait for the setup delay. Lastly, a saturation phase takes place when $\lambda >> (n + c)\mu$, i.e., the system is unable to handle the load, which is only reflected in the Availability evaluation (Fig. 3a).

In order to keep the figures within the same range of λ, Fig. 3d is limited to λ = 100 which might be confusing since the best apparent result for this metric comes from the intermediate configuration (k = 50). Indeed, this



(a) Availability (b) Reliability

(c) Power Consumption (d) Mean Response Time

Fig. 3: Multiple MEC sizes (k): Far, Mid and Near Edge scope examples.

configuration shows the best result for large part of the experiment, but for $\lambda > 100$ the configuration $k = 100$ is likely to retake the advantage, similar to the comparison between $k = 25$ and $k = 50$. The fact is that at some point, the impact of the setup phase of the containerized VNFs is mitigated. This result reinforces the importance of the holistic view promoted by the evaluation of at least these four metrics concomitantly, since for a given scenario, the intermediate ($k = 50$) or even the configuration with least resources ($k = 25$) could eventually be the most cost-beneficial to handle the URLLC load.

The designed framework allows the evaluation of multiple MEC node sizes by tuning the appropriate parameters accordingly (subsection 5.1), however the next evaluations (subsections 5.2 and 5.3) focus on a single-sized MEC node ($k = 25$) positioned together within RAN equipment, i.e., similar to the Near Scope in Fig. 1. In addition, the following results display dashed lines for the Reliability and Response Time, which represent three URLLC industry vertical thresholds: Augmented Reality (AR), Smart Manufacturing (SM) and, Transport Industry (TI) that are summarized in Table 5 according to [14]. For the sake of simplicity, we have adopted specific values rather than the ranges described in the original document.

5.2 Multiple VM/Container Arrangements (n, c)

Figs. 4a-4d illustrate the impact of varying VM/Container ratios that are limited to the same amount of resources (k). The results evince that configurations with a smaller number of VMs compared to Containers tend to

Table 5: Reliability and Response Time Thresholds for URLLC

Industry Vertical	Reliability	Response Time
Augmented Reality (AR)	99.9%	1.1 ms
Smart Manufacturing (SM)	99.99%	1.15 ms
Transport Industry (TI)	99.999%	1.2 ms

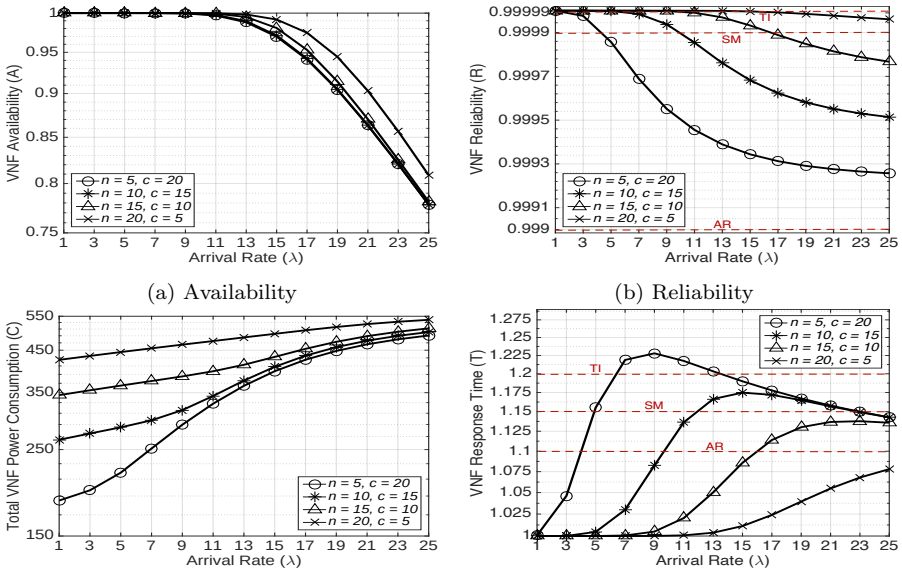
have lower Availability, faster Response Times and higher Reliability, which is expected since the VM-hosted VNFs are stable compared to containerized VNFs, i.e., not prone to failures. For these metrics, we expected similar results between curves at least until $\lambda = 5$, since this is the load in which all analyzed configurations have enough VMs $n = (5, 10, 15, 20)$. However, for the Availability (Fig. 4a) the curves start to differentiate only at $\lambda = 11$ whereas the Reliability and Response Times responded from $\lambda = 3$.

With regards to Reliability in Fig. 4b, not even the arrangement with most VMs ($n = 20, c = 5$) was able to keep itself above all dashed lines, breaking the TI threshold at $\lambda = 15$, i.e., with a significant distance between its theoretical VM capacity at $\lambda = 20$. The fact is that $\lambda <= c$ does not guarantee that all requests will be processed by VMs. Considering URLLC applications, if even relatively few requests are experiencing setup delays, one or multiple failures can cause serious capacity issues, leading to lower Reliability as more containerized VNFs are needed. On the other hand, we expected that a platform formed only by VMs would become costly in terms of power consumption since VMs are not fast enough to be scaled for URLLC applications, and therefore must be continuously active, regardless of being idle or busy.

In Fig. 4c we observed that although there is a large difference in terms of power consumption until $\lambda = 7$, from this point on, the curves rapidly converge despite the scaling feature and lower consumption of containerized VNFs, possibly because the absolute difference between the curves is largely correlated with the difference between the adopted constants for the consumption of VMs and Containers in Busy state, which is only of 2W. Thus the scaling strategy is not as effective for higher loads as it is in lower loads.

A possible solution is to adjust the VM/container ratio according to the demand, i.e., an operator can enable arrangements with more containers for low demands or with more VMs for higher loads, similarly to the approach used in [11]. For instance, considering only the Reliability and the Smart Manufacturing applications (Fig. 4b), the configuration with 5 VMs and 20 containers could be used if $\lambda < 3$, whereas if $3 < \lambda < 15$ a more balanced set with 15 VMs and 10 containers could take place and finally, the arrangement with 20 VMs and 5 containers would only become available if $\lambda > 15$. Please note that this example would not be applicable considering the Mean Response Time (Fig. 4d), i.e., the intervals for swapping between arrangements would necessarily differ. In [11], although multiple indicators were suggested, the authors proposed a resource optimization solution based only in one argument. On

the other hand, besides considering the virtual host pitfalls, our framework allows using multiple performance metrics and URLLC application thresholds to formulate specific and practical solutions.



(a) Availability (b) Reliability

(c) Power Consumption (d) Mean Response Time

Fig. 4: Impact of multiple VM/Containerized VNF arrangements (n, c).

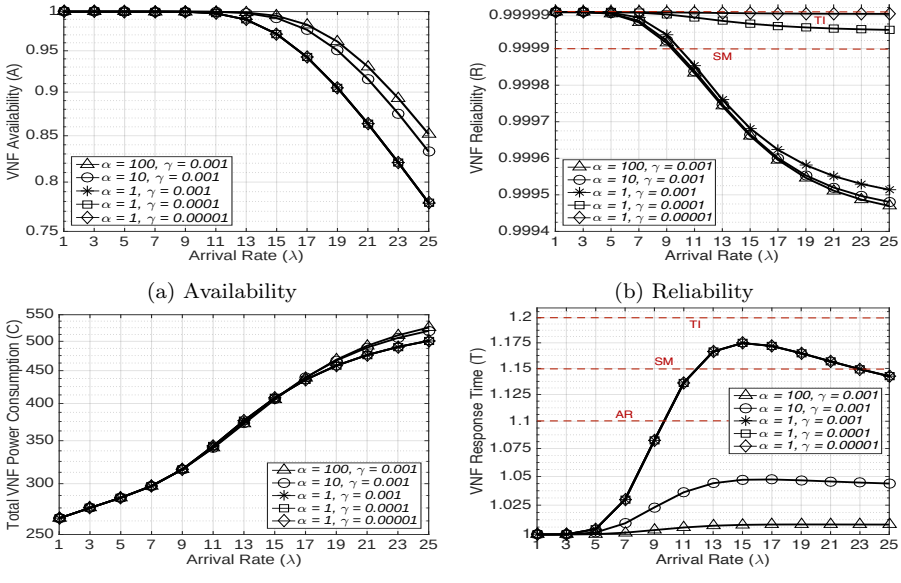
5.3 Multiple Setup and Failure Rates (α, γ)

In this evaluation, a single configuration with $k = 25$ was adopted, but with a fixed arrangement of 10 VMs and 15 containers and varying setup (α) and failure (γ) rates. This configuration was analyzed in section 5.1 with fixed $\alpha = 1$ and $\gamma = 0.001$ and becomes interesting for the current experiment since it is balanced in terms of both resource types and yet it is prone also to be impacted by α and γ.

A larger α means smaller container setup delays, i.e., more VNF instances become available per unit time. As expected, higher α rates resulted in a blocking probability reduction (Fig. 5a), but interestingly, Fig. 5b reveals the opposite: higher α rates actually increased the system's failure probability. This unexpected behavior indicates that isolated improvements in the setup rate may help the admission process but become a burden to the admitted URLLC flows. In other words, a higher setup rate increases the flow served by the failure-prone component (containerized VNFs) per unit time, therefore also increasing the chances of failures, which suggests that future enhancements in this parameter might be insufficient for URLLC.

In Fig. 5c, the Power Consumption is expected to differ since α variations necessarily impact the amount of powered on container per unit time. However, this experiment has shown that the baseline configuration ($\alpha = 1$) presented a similar power consumption pattern compared to those with higher α values throughout the entire evaluation. A performance difference is also observed in Fig. 5d, where the Mean Response Times from the baseline curve is significantly greater than the curves with enhanced α rates. In $\lambda = 15$ the difference between baseline and the curve with $\alpha = 100$ reaches a maximum of 0.150 ms.

A larger γ means smaller intervals between successive containerized VNF failures. In contrast to α, this parameter improves as it gets smaller. Thus, we have enhanced γ by dividing its default value by 10 e and 100 times. However, if, on the one hand, our expectations were met regarding the overall Reliability (Fig. 5b), i.e., lower failure rates allowed the two curves to surpass the Smart Manufacturing Reliability Threshold, to our surprise, the Availability (Fig. 5a), Power Consumption (Fig. 5c) and Response Time (Fig. 5d) remained almost unchanged, despite the large difference between the adopted γ values. These findings evince what level of improvement containerized VNFs must achieve to meet specific requirements, recalling that both software and hardware share relevance towards this aspect, as investigated in [21]. In brief, it becomes clear that containerized VNF setup delays critically impact the admission (Availability), whereas the Reliability reacts severely to failures.



(a) Availability

(b) Reliability

(c) Power Consumption

(d) Mean Response Time

Fig. 5: Impact of multiple setup and failure rates (α, γ).

6 Conclusion and Future Directions

In this work, we thoroughly analyzed the existing literature to investigate analytical models for URLLC evaluation considering a MEC node in isolation. Hence, we provide a new flexible analytical framework for mobile service providers to design optimization strategies for multiple problems such as MEC dimensioning, placement, and dynamic resource allocation. In addition, the most relevant performance metrics were formulated and analyzed together to provide means for building optimal provisioning strategies and better understand the relationship between several parameters under various scenarios, including multiple MEC node sizes, virtual resource types, setup/failure rates, and traffic loads. Among other insights, the joint analysis revealed that individual containerized VNF setup/failure rate improvements may not positively impact the overall performance. We plan to extend this work to support other random process models and service categories such as enhanced Mobile Broadband (eMBB) and massive Machine Type Communications (mMTC). Thus, multiple optimization problems can be formulated by adapting the current model to decide upon processing eMBB, mMTC, and URLLC flows considering penalties for when SLA agreements are violated in each case.

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

References

1. D. Feng et al.,"Toward Ultrareliable Low-Latency Communications: Typical Scenarios, Possible Solutions, and Open Issues," IEEE Vehicular Technology Magazine, vol. 14, no. 2, pp. 94-102, 2019, doi: 10.1109/MVT.2019.2903657
2. K. Antevski, C. Bernardos, L. Cominardi, A. Oliva, A. Mourad., "On the integration of NFV and MEC technologies: architecture analysis and benefits for edge robotics," Computer Networks, vol. 175, pp. 1389-1286, 2020, doi.org/10.1016/j.comnet.2020.107274.
3. 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 , doi: 10.1109/MWC.2017.1600427.
4. S. Sultan, I. Ahmad and T. Dimitriou,"Container Security: Issues, Challenges, and the Road Ahead," in IEEE Access, vol. 7, pp. 52976-52996, 2019, doi: 10.1109/ACCESS.2019.2911732.
5. A. Santoyo-González and C. Cervelló-Pastor,"Edge Nodes Infrastructure Placement Parameters for 5G Networks," IEEE Conference on Standards for Communications and Networking (CSCN), pp. 1-6, 2018, doi=10.1109/CSCN.2018.8581749

6. Li, C., Cai, Q., Zhang, C. et al."Computation offloading and service allocation in mobile edge computing," in Journal of Supercomputing, 2021, doi: 10.1007/s11227-021-03749-w.
7. L. Yala, P. A. Frangoudis and A. Ksentini,"Latency and Availability Driven VNF Placement in a MEC-NFV Environment," IEEE Global Communications Conference (GLOBECOM), pp. 1-7, 2018, doi: 10.1109/GLOCOM.2018.8647858.
8. V. Farhadi et al.,"Service Placement and Request Scheduling for Data-intensive Applications in Edge Clouds," IEEE Conference on Computer Communications, pp. 1279-1287, 2019, doi: 10.1109/INFOCOM.2019.8737368.
9. A. Samanta and J. Tang,"Dyme: Dynamic Microservice Scheduling in Edge Computing Enabled IoT," IEEE Internet of Things Journal, vol. 7, no. 7, pp. 6164-6174, 2020, doi: 10.1109/JIOT.2020.2981958.
10. 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 (ICTC), pp. 879-882, 2019, doi: 10.1109/ICTC46691.2019.8939566.
11. 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.
12. Z. Tong, T. Zhang, Y. Zhu and R. Huang,"Communication and Computation Resource Allocation for End-to-End Slicing in Mobile Networks," IEEE/CIC International Conference on Communications in China (ICCC), pp. 1286-1291, 2020, doi: 10.1109/ICCC49849.2020.9238794.
13. I. Sarrigiannis, K. Ramantas, E. Kartsakli, P. Mekikis, A. Antonopoulos and C. Verikoukis,"Online VNF Lifecycle Management in an MEC-Enabled 5G IoT Architecture," IEEE Internet of Things Journal, vol. 7, no. 5, pp. 4183-4194, 2020, doi:10.1109/JIOT.2019.2944695.
14. 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," IEEE Transactions on Network and Service Management, vol. 16, no. 2, pp. 459-474, 2019, doi: 10.1109/TNSM.2019.2894955.
15. Ma, S., Chen, X., Li, Z. et al. Performance Evaluation of URLLC in 5G Based on Stochastic Network Calculus. Mobile Netw Appl (2019). https://doi.org/10.1007/s11036-019-01344-1
16. Y. Ren, T. Phung-Duc, J. Chen and Z. Yu,"Dynamic Auto Scaling Algorithm (DASA) for 5G Mobile Networks," IEEE Global Communications Conference (GLOBECOM), pp. 1-6, 2016, doi: 10.1109/GLOCOM.2016.7841759.
17. R. Morabito,"Power Consumption of Virtualization Technologies: An Empirical Investigation," 2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC), pp. 522-527, 2015, doi: 10.1109/UCC.2015.93.

18. 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, 2018, doi: 10.1109/JSAC.2018.2874122.
19. Li, W., Jin, S.,"Performance evaluation and optimization of a task offloading strategy on the mobile edge computing with edge heterogeneity," in Journal of Supercomputing, 2021, doi: 10.1007/s11227-021-03781-w.
20. L. Kleinrock, "Queueing Systems", Volume 1, Theory, John Wiley and Sons, New York, 1975.
21. S. Lal, S. Ravidas, I. Oliver and T. Taleb, "Assuring virtual network function image integrity and host sealing in Telco cloud", IEEE International Conference on Communications (ICC), pp. 1-6, 2017, doi: 10.1109/ICC.2017.7997299.