Telecommunication Systems
https://doi.org/10.1007/s11235-019-00607-2

A multi-stage analysis of network slicing architecture for 5G mobile networks

Salman A. AlQahtani[1] · **Waseem A. Alhomiqani**[1]

Abstract
Recently, the increasing demand for low latency, the explosive growth in the volume of network traffic, the large and growing number of connected devices, and diversified multimedia applications have paved the way for a new era of mobile networks. To meet these diverse requirements of different businesses in network virtualization, network slicing has emerged as a promising paradigm of upcoming 5G mobile networks. Network slicing is a major technology, based on network function virtualization and software defined network technologies, which aims to achieve more efficient utilization of available network traffic and reduce operating costs. In this paper, we propose a network slicing architecture for 5G mobile networks involving cloud radio access network (C-RAN), mobile edge computing (MEC), and cloud data center. We model the proposed network slicing system based on queueing theory, which can be used to derive the main performance metrics such as the CPU utilization, system throughput, system drop rate, average number of message requests, average response time, and average waiting time. We provide quantitative examples to show how this proposed model could be applied to estimate the system performance and cost for a network slicing system in 5G mobile networks and the number of C-RAN and MEC cores required under diverse 5G traffic conditions. The analytical results and simulation models indicate that the proposed model has a powerful ability to assign the number of C-RAN and MEC cores required to achieve the quality of service targets of 5G slices.

Keywords 5G networks · Network slicing · SDN · NFV · QoS · Performance analysis

1 Introduction

The 5G mobile communication system, aimed to be commercialized in 2020, is actively being researched at various institutes and standardization organizations [1]. 5G systems target the simultaneous support of a wide range of application scenarios and business models (e.g., automotive, utilities, smart cities, and high-tech manufacturing) [2]. The idea of switching toward 5G mobile network is based on current drifts. It is generally assumed that 5G mobile networks must meet certain challenges that are not effectively addressed by 4G, i.e., higher data rate, higher capacity, lower latency, real-time processing, connectivity of massive numbers of devices, higher reliability, lower cost, and consistent quality of service (QoS) or quality of experience (QoE) provisioning [3].

In order to meet these challenges, paradigm shifts are required in the techniques that drive networks as well as their architecture. Currently, innovative technologies are being developed to motivate next-generation mobile communication systems. Paramount among these improvements are software defined networking (SDN) and network function virtualization (NFV) technologies [4, 5], which have now been recognized as two of the key enabling technological factors to achieve the 5G mobile networks objectives, and which represent significant changes in the way network services are operated and deployed [6]. Hence, these provide a scalable, flexible, programmable network platform over which to manage multiple services with different requirements within strict performance limits [7].

The main idea of SDN is the separation between the network control and the forwarding of data, such that it becomes directly programmable and the underlying infrastructure is abstracted for applications and network services. With this separation, significant flexibility is achieved, allowing for simple network management [4, 7, 8]. The main idea of NFV is essentially the separation of network functions from

✉ Salman A. AlQahtani
salmanq@ksu.edu.sa

[1] Computer Engineering Department, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia

Published online: 29 August 2019

Table 1 5G Network slices and their QoS requirements

5G slice	QoS requirements	Mobility	Examples
eMBB	Video cache High capacity	Yes	Real-time work in cloud environment, high-resolution video streaming, augmented reality (AR)/virtual reality (VR) immersive games, etc
mIoT	Massive number of devices Long battery life Small data volume Low device cost	NO	Sensor networks (logistics, city, home, smart metering, etc.)
URLLC	High reliability Low latency	Yes	Real-time remote surgery or coordination among vehicles, reliable remote robotic actions, factory automation, self-driving, smart grids, etc

stand-alone boxes that depend on dedicated hardware to cloud-based software. This is achieved by means of virtualization, running on stand-alone hardware, with the additional potential to be deployed anywhere.

New approaches and architectures for radio access networks have emerged with the introduction of network virtualization in the mobile communication system. One of these approaches is network slicing, which is expected to play a major role in advancing 5G. The concept of network slicing, employing network virtualization technologies with SDN and NFV in 5G mobile networks, is to divide a single physical infrastructure into multiple virtual wireless networks [5, 9]. Each network slice may have its own network architecture, protocols, and security settings. Network slicing includes slicing of the cloud radio access network (C-RAN), core network (CN), and perhaps even the end devices [10]. In this paper, we propose that each network slice consists of a separate C-RAN, mobile edge computing (MEC), and cloud data center (CDC). MEC and C-RAN are highly complementary technologies. The combination of these technologies helps make them more economically attractive. Furthermore, collocation can also help to enable MEC applications to exploit C-RAN information and enable the C-RAN to exploit MEC services. C-RAN provides improved system architecture, performance coverage, energy efficiency, and mobility while reducing the cost of deploying and operating the network. C-RAN is dependent on the fundamentals of virtualization and centralization [3, 11]. The baseband resources are grouped at the baseband unit (BBU) pool, which is located in the remote central office (not at the cell locations). In traditional cellular networks, the Ethernet, IP, and multi-protocol functionality are extended along the way to remote cell locations. Virtual BBU (vBBU) pools further facilitate scalability, reduction in time consumption, cost reduction, and integration of different services for field trials. Remote radio heads (RRHs), consisting of transceiver components, analog–digital converters, amplifiers, and duplexers enable digital processing, filtering, and power amplification. RRHs are connected to the BBU pool by high-data-rate single-mode fibers. This simplified structure paves the way for deploying dense 5G networks by making them flexible, efficient, and affordable [12].

MEC is essentially a cloud-based information technology (IT) environment at the edge of the mobile network or more generally, on the edge of any network, that offers storage and computational resources on the edge. MEC provides high bandwidth, low latency, and real-time access to radio network information, allowing mobile network operators to open their networks to a new ecosystem and value chain. MEC allows multiple types of access on the edge [13].

Technologies such as network slicing enable networks to be built in more flexible, scalable, and dynamic ways to meet different service needs by allowing the creation of multiple logical networks over a common shared physical infrastructure. Each slice provides a dedicated connection and all slices operate on the same shared infrastructure. The greater elasticity offered by network slicing will help to address the efficiency, cost, and flexibility requirements imposed by current and future demands [14].

The ability to deliver a wide range of network performance features that future services will demand is also one of the fundamental technical challenges facing service providers today. The performance requirements of the network will require communication in terms of data rate, QoS, latency, availability, security, and many other parameters, all of which vary from service to service. Therefore, network slicing provides a greater level of network resource utilization, with each dedicated network slice based on different service requirements, such as bandwidth and latency. The major service scenarios for 5G wireless communication are categorized as three types of services, namely ultra-reliable low-latency communication (URLLC), massive internet of things (mIoT), and enhanced mobile broadband (eMBB), which support QoS requirements for different types of data traffic as presented in Table 1 [1, 15, 16].

Performance modeling and analysis have been of great theoretical and practical importance in operations research in designing, developing, and improving computer applications and communication systems [17]. This involves a wide

range of research activities, from the use of more experimental methods (from experimentally changing simple existing models to building and experimenting with prototype applications) using simulations to more complex mathematical methods. These have played a crucial role in the understanding of important problems, designing, development, planning, and management of complex systems. Queuing theory has been widely studied in the area of performance modeling and QoS for different information and communications technology (ICT) systems [18]. Queuing theory deals with problems involving queuing. The queuing model is constructed such that the waiting times and queue lengths can be predicted. By the queuing model, we can usually derive the main system performance and QoS parameters, which may involve the blocking probability, throughput, number of message requests, average response time, average waiting time, and server utilization [19].

This study focuses on a performance analysis of network slicing in 5G mobile networks involving C-RAN, MEC, and CDC. Based on queuing theory, we provide an analytical model to the study network slicing performance. More precisely, we highlight how to use the proposed model to estimate QoS parameters for 5G mobile communication systems using an architecture of C-RAN, MEC, and CDC. In addition, we highlight how to use the proposed network slicing model for the dynamic scalability of C-RAN/MEC cores.

The remainder of this paper is outlined as follows: related works and contributions are discussed in Sect. 2. The proposed 5G network slicing model is introduced in Sect. 3. We present the proposed queuing model in Sect. 4. In Sect. 5, we offer a performance analysis of the proposed model. The numerical and simulation results are presented in Sect. 6. Finally, the paper is concluded in Sect. 7.

2 Related works and contributions

In recent years, there have been significant advances in the research on 5G wireless networks. Several enabling technologies for 5G mobile systems are being explored, including network slicing. Many published works about network slicing have focused on three concepts (wireless network virtualization (WNV), SDN, and NFV) [20–29]. The authors in [20] present a comprehensive vision of WNV as an enabler of network slicing, that provide the basis for designing a wireless network slicing to satisfy co-existence and isolate various WNV mapped onto the same physical network to avoid conflicts. Based on SDN to enable network slicing, many works have proposed different frameworks, designs, and tools. Some works have focused on allowing network programmability with network abstraction. One of these slicing tools based on SDN is FlowVisor, which is used in wired networks to achieve network slicing and resource isolation.

The authors in [21] introduce three scheduling schemes in order to meet joint resource orchestration problem with different QoS requirements for SDN-based network slicing. The authors in [22] study a packet delay modeling for traffic flows travel through virtual network function (VNF) chains in 5G networks in order to achieve fairness of dominant-resource and high resource utilization. The authors in [23] present a technical approach based on SDN and NFV for 5G network slicing using a physical testing environment with realistic traffic conditions. The authors in [24] provide the technologies, approaches and possible solutions to meet the most critical requirements of 5G networks that focus on low-latency traffic characteristics as an RLLC service type, also they offered the MEC concept to further support low latency requirements. The authors in [25] present an SDN-based 5G network slicing approach to enable effective coexistence of IoT and eMBB slices with providing isolation between them in the same C-RAN.

From the reviewed literature, most of these studies provide frameworks, approaches, and tools for network slicing to achieve resource allocation and isolation. However, the network slicing QoS provisioning as a key QoS requirement in the coming 5G networks involving C-RAN, MEC, and CDC were not covered. In order to fill the gap in the literature, in this research, we are motivated to study the performance analysis of network slicing for 5G mobile networks. We focus on the main system performance and QoS parameters of 5G slicing involving C-RAN, MEC, and CDC, which involve the blocking probability, throughput, average number of message requests, average response time, average waiting time, and server utilization. Java Modeling Tools (JMT) is used to evaluate the performance of the proposed network slice architecture for 5G core networks. The main contributions of this study, which distinguish it from those already published on this subject, are as follows:

- A queuing model is proposed to aid in analyzing and studying the behavior of network slicing in 5G network. The proposed model consists of three sequential queuing model subsystems involving C-RAN, MEC, and CDC. More precisely, we present how to use the proposed model to estimate QoS parameters for 5G mobile networks using an architecture of C-RAN, MEC, and CDC.
- In addition, we highlight how to use the proposed network slicing model for the dynamic scalability of C-RAN/MEC cores
- An analytical model is provided and mathematical equations are derived for the main performance metrics of the entire queuing models.
- Quantitative examples are provided to show how this proposed model can be applied to estimate the performance

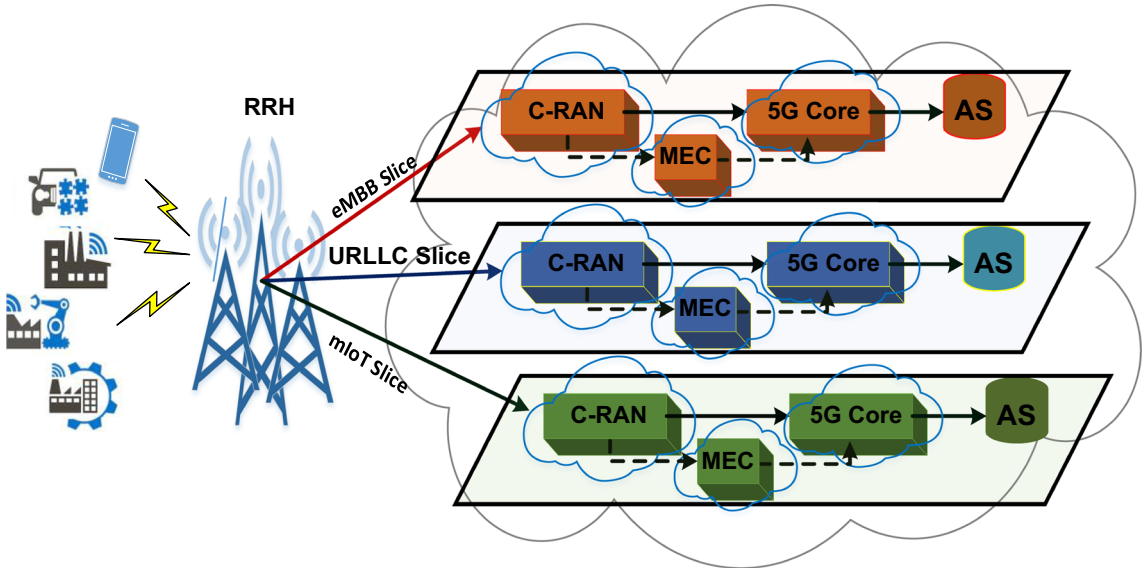

Fig. 1 Architectural model of proposed network slicing for 5G

of the network slicing system in 5G and the number of C-RAN and MEC cores required under diverse conditions of 5G traffic.

- The simulation model is developed by a discrete event simulation called the JMT simulator to validate and verify the accuracy of the proposed analytical model.

3 System modeling and assumptions

In this section, we describe the proposed system model in more detail. The proposed network slice architecture for 5G core networks, as a configuration scheme for logical networks supporting specific services, is composed from independent C-RAN, MEC, and CDC, such that each network slice is independent of the others, as shown in Fig. 1. In this study, we consider a single-cell area served by a base station through an LTE-like air interface.

For C-RAN, we consider that multiple RRHs are connected to a vBBU pool through high-bandwidth transport links, where the RRHs are distributed over certain coverage area in which they are responsible for transmission/reception. We assume that the RRH users are uniformly distributed in the coverage area. All data from the RRH are first transmitted to the vBBU pool for further processing based on their network slice types, where all data have three-network slice classes: URLLC, eMBB, and mIoT. We assume that 5G C-RAN is separated into three network slices, where each slice is independent of the others. All three network slices share the same C-RAN and each slice can be allocated a number of vBBUs based on the data type processed. We assume the number of vBBUs in each slice is n vBBUs. The arrival process of all incoming data is a Poisson arrival process with an aggregated arrival rate λ for an individual end client. We also assume that all slices are served on a first come first served (FCFS) basis, without resource reservation, and that service times are identical and independently exponentially distributed.

$$\lambda = \sum_{j=1}^{K} \lambda_i$$

where j indexes a total of K clients.

These data are then forwarded to the MEC servers for potential processing for additional services such as providing high bandwidth and low response time, particularly in URLLC and eMBB slices or forwarded to the CDC for storage and processing capabilities.

For the MEC, we consider that multiple servers are located at the network edge to compute, process, and temporarily store the data received from the C-RAN that need additional services. These servers are also connected to the CDC and are responsible for sending the remaining data to the CDC for further storage and processing. We assume that the number of MEC servers in each slice is s. In addition, we suppose that the MEC supports three types of network slices to access on the edge.

We consider the CDC to be comprised of servers, a database, and network equipment (e.g., routers, switches, and cables), power distribution, and cooling systems. Each physical server is provided as many virtual servers in real time, based on the agreement and availability of service levels. We also consider that large CDCs hosting multiple physical servers such as Microsoft, Google, Amazon, and Yahoo have

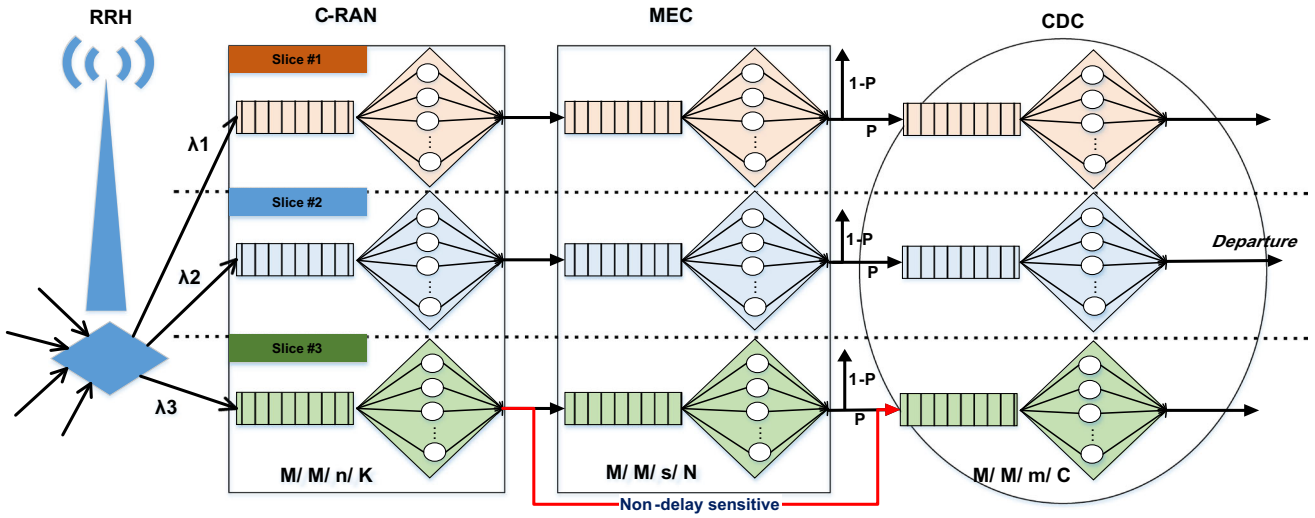

Fig. 2 Queuing model of network slicing in 5G communication

tens of thousands of physical servers. We suppose that in the CDC, there are M homogenous physical servers in each slice and each physical server can support up to m virtual servers. The proposed network slicing queuing model and other system assumptions are presented in the next section.

4 Proposed queuing model

In this section, we propose our network slicing queuing model by specifying three queuing model subsystems for each network slice, as shown in Fig. 2. The first subsystem is the network queue of C-RAN with a higher number of vBBUs with multiple cores. The C-RAN queue is modeled to have an M/M/n/K queuing model with finite capacity K. All messages of the clients are transmitted to the C-RAN queue based on their QoS requirements of the network slices, and then processed on an FCFS basis. The arrivals of incoming client messages in the C-RAN queue follow a Poisson process with an arrival rate λ_R. We consider that the processing time of the C-RAN queue is identical and independently exponentially distributed with a mean service time of $1/\mu_R$. After being processed by the C-RAN core, the client message is either forwarded to the MEC for additional services, such as high bandwidth and low latency, or forwarded to the CDC for further processing or storage.

The middle subsystem is the MEC network queue, which includes enough intelligent edge cores to calculate, process, and store the received messages temporarily and redirect the other remaining messages to the CDC for further storage or processing. After being processed by the MEC subsystem, the client messages are either redirected to the CDC for further processing or storage with probability (P_{ec}) or depart from the system because the service has been completed, with probability ($1 - P_{ec}$). We model each MEC as an M/M/s/N queueing model. We also consider that the service times of virtual servers are identical and independently exponentially distributed with mean service time $1/\mu_E$. The M/M/s/N queue is employed to distinguish each physical server (MEC core) with finite capacity N.

The last subsystem is the CDC network queue. This subsystem is higher number of data centers that contain a set of multiple cores, where vCPUs run on top of CPU cores that are capable of storing and processing a huge amount of data. The service times of the vCPU are assumed to be identical and independently exponentially distributed with rate $1/\mu_C$. The M/M/m/C queue is employed to distinguish each CPU core with finite capacity C.

The main procedures of the proposed model are described in the following steps:

- The clients send their messages to a vBBU in the C-RAN via multiple RRH.
- The client messages are inserted into their proper queue based on their network slice (URLLC, mIoT, or eMBB).
- In each slice, vBBUs process client messages and send these messages either to the MEC server queue for additional services, such as real-time processing and low latency, or to the CDC queue for further processing or storage.
- In the MEC subsystem, the client messages are processed and either redirected to the CDC queue for further processing or depart from the system because the service has been completed.
- In the CDC subsystem, the huge number of client messages are inserted to the CDC queues for processing and storage.

Table 2 Frequently used key parameters in C-RAN queueing system

Parameters	Description
λ_R	Message arrival rate to C-RAN
$1/\mu_R$	Mean C-RAN service time
U_R	Utilization of each vBBU core in C-RAN
H	Number of vBBU cores in C-RAN subsystem
n	Number of vBBU in each core
k	Limiting number of messages in each vBBU cores
P_i	Equilibrium probability
σ	$\lambda_R/n\mu_R$ offered load in each vBBU cores in C-RAN
P_B^i	Blocking probability
$\bar{\gamma}_R^i$	Throughput service for each vBBU cores in C-RAN
$\bar{L}_R^i$	Average number of message requests for each vBBU core in C-RAN
$\bar{Q}_R^i$	Average number of message requests waiting for each vBBU core in C-RAN
$\bar{R}_R^i$	Average response time for each vBBU core in C-RAN
$\bar{W}_R^i$	Average waiting time for each vBBU core in C-RAN
$\bar{\gamma}_R$	Throughput service of C-RAN
$\bar{L}_R$	Average number of message requests in C-RAN
$\bar{Q}_R$	Average number of message requests waiting in C-RAN
$\bar{R}_R$	Average response time in C-RAN
$\bar{W}_R$	Average waiting time in C-RAN

5 Performance analysis

In this section, an analysis of the three concatenated queuing subsystems is presented. We derive the main performance formulas for the proposed queuing models, including the throughput, CPU utilization, number of message requests, average response time, average waiting time, and system loss rate. Finally, we calculate the cost of the overall system by computing the expected service costs of the system and the expected request waiting time costs in the proposed model. There are three queuing model subsystems for each network slice. In this section, we study the analysis of one network slice, because the analyses of the remaining slices are similar to that of the selected slice. Now, the main performance formulas for the proposed queuing models can be derived as follows.

5.1 Cloud radio access network model (C-RAN)

This model describes each vBBU core in the C-RAN, corresponding to the M/M/n/k queuing model. Table 2 presents the main parameters frequently used in this section and their descriptions.

The M/M/n/k queuing system is a variation of a multicore system and only a maximum of K message requests in each vBBU core is allowed to stay in the system. As mentioned earlier, the number of message requests in the system is a birth–death process with appropriate rates for the steady-state distribution [18, 30]. The equilibrium state distribution is given below:

$$P_i = \begin{cases} P_0 \frac{\lambda_R^i}{i! \mu_R^i}, & for\ 0 \le i \le n \\ P_0 \frac{\lambda_R^i}{n! n^{i-n} \mu_R^i}, & for\ n \le i \le k \end{cases} \tag{1}$$

$$P_k = \begin{cases} P_0 \frac{\lambda_R^k}{k! \mu_R^k}, & for\ k < n \\ P_0 \frac{\lambda_R^k}{n! n^{k-n} \mu_R^k}, & for\ k \ge n \end{cases} \tag{2}$$

where P_0 can be derived from a normalizing condition as:

$$P_0 = \left[\sum_{i=0}^{n-1} \frac{\lambda_R^i}{i!\, \mu_R^i} + \sum_{i=n}^{k} \frac{\lambda_R^i}{n^{i-n} n!\, \mu_R^i} \right]^{-1} \tag{3}$$

To simplify this expression, let

$$\rho = \frac{\lambda_R}{\mu_R}\ and\ \sigma = \frac{\rho}{n}$$

Then,

$$\sum_{i=n}^{k} \frac{\lambda_R^i}{n^{i-n} n!\, \mu_R^i} = \sum_{i=n}^{k} \frac{\rho^i}{n^{i-n} n!} = \frac{\rho^i}{n!} \sum_{i=n}^{k} \sigma^{i-n} = \begin{cases} \frac{\rho^n}{n!} \frac{1-\sigma^{k-n+1}}{1-\sigma}, \sigma \ne 1 \\ \frac{\rho^n}{n!} (1 - n + 1), \sigma = 1 \end{cases}$$

Thus,

$$P_0 = \begin{cases} \left[\frac{\rho^n}{n!} \frac{1-\sigma^{k-n+1}}{1-\sigma} + \sum_{i=0}^{n-1} \frac{\rho^i}{i!} \right]^{-1}, & \sigma \ne 1 \\ \left[\frac{\rho^n}{n!} (1 - n + 1) + \sum_{i=0}^{n-1} \frac{\rho^i}{i!} \right]^{-1}, & \sigma = 1 \end{cases} \tag{4}$$

Now, we can derive the main performance metrics. First, the blocking probability can be expressed as follows:

$$P_B^i = P_k^i \tag{5}$$

P_k represents the probability for the number of message requests of not joining the system. Hence, $\lambda_R P_k$ represents the average number of message requests lost owing to the finite queue. Therefore, the effective arrival rate ($\boldsymbol{\lambda_e}$) in such queues is represented as follows:

$$\lambda_e = \lambda_R \left(1 - P_k^i\right) \tag{6}$$

The mean throughput service $\bar{\gamma}_R^i$ of each vBBU core is given by:

$$\bar{\gamma}_R^i = \lambda_R\left(1 - P_B^i\right) \tag{7}$$

The core utilization of each vBBU core instant can be written as follows:

$$U_R^i = \frac{\bar{\gamma}_R^i}{n\mu_R} = \sigma\left(1 - P_B^i\right) \tag{8}$$

Now, we derive the average number of message requests for each vBBU core, and obtain:

$$\bar{L}_R^i = \sum_{j=1}^{k} j\, P_j^i \tag{9}$$

The average number of message requests waiting for each vBBU core can be expressed as follows:

$$\bar{Q}_R^i = \sum_{j=n+1}^{k} (j-n) P_j^i \tag{10}$$

Finally, using Little's formula [31], the average response time and average waiting time for each vBBU core are written as follows:

$$\bar{R}_R^i = \frac{\bar{L}_R^i}{\lambda_e} = \frac{\bar{L}_R^i}{\lambda_R\left(1 - P_k^i\right)} \tag{11}$$

$$\bar{W}_R^i = \frac{\bar{Q}_R^i}{\lambda_e} = \frac{\bar{Q}_R^i}{\lambda_R\left(1 - P_k^i\right)} \tag{12}$$

Now, we analyze the performance of the C-RAN subsystem by deducing the equations for the main performance parameters of the queue model, beginning with the blocking probability (loss probability) in all vBBU core queues in the C-RAN:

$$P_B = \sum_{i=1}^{H} P_k^i \tag{13}$$

The average number of message requests in the C-RAN subsystem can expressed as follows:

$$\bar{L}_R = \sum_{i=1}^{H} \bar{L}_R^i \tag{14}$$

The average number of message requests for all vBBU core queues in the C-RAN subsystem can be obtained as follows:

$$\bar{Q}_R = \sum_{i=1}^{H} \bar{Q}_R^i \tag{15}$$

Table 3 Main parameters used frequently in MEC Queueing system

Parameters	Description
λ_E	Arrival rate to MEC
$1/\mu_E$	Mean vServers service time in MEC
U_E	Utilization of each MEC server
Z	Number of MEC cores
s	Number of vServers in each MEC core
N	Limiting number of messages in each MEC core
P_i	Equilibrium probability
$\boldsymbol{\omega}$	$(\lambda_E)/(s\mu_E)$ offered load in each MEC core
P_l^i	Blocking probability in MEC subsystem
$\bar{\gamma}_E^i$	Throughput service for each MEC core
$\bar{L}_E^i$	Average number of message requests for each MEC core
$\bar{Q}_E^i$	Average number of message requests waiting for each MEC core
$\bar{R}_E^i$	Average response time for each MEC core
$\bar{W}_E^i$	Average waiting time for each MEC core
$\bar{\gamma}_E$	Throughput service of MEC subsystem
$\bar{L}_E$	Average number of message requests in MEC subsystem
$\bar{Q}_E$	Average number of message requests waiting in MEC subsystem
$\bar{R}_E$	Average response time in MEC subsystem
$\bar{W}_E$	Average waiting time in MEC subsystem

The mean throughput of the C-RAN subsystem can be formulated as:

$$\bar{\gamma}_R = \sum_{i=1}^{H} \bar{\gamma}_R^i \tag{16}$$

Finally, the average response time and the average waiting time in all vBBU core queues for the C-RAN can be expressed as:

$$\bar{R}_R = \frac{\bar{L}_R}{\bar{\gamma}_R} \tag{17}$$

$$\bar{W}_R = \frac{\bar{Q}_R}{\bar{\gamma}_R}. \tag{18}$$

5.2 Multi-access edge computing model (MEC)

The MEC subsystem in our proposed network slicing offers storage and computational resources at the edge. The subsystem contains (s) MEC cores and each can be modeled as an M/M/s/N queuing system. Table 3 presents most of the parameters that we need in this section and their descriptions.

The M/M/s/N queuing system is a variation of a multi-core system and only a maximum of N message requests in

each MEC server is allowed to stay in the system. We obtain the steady-state probability of N messages requests for each MEC core based on the global balance equation and normalization condition [17, 19] as follows:

$$P_i = \begin{cases} P_0 \frac{\lambda_E^i}{i! \mu_E^i}, & for\ 0 \le i \le s \\ P_0 \frac{\lambda_E^i}{s! s^{i-s} \mu_E^i}, & for\ s \le i \le N \end{cases} \tag{19}$$

$$P_N = \begin{cases} P_0 \frac{\lambda_E^i}{N! \mu_E^N}, & for\ N < s \\ P_0 \frac{\lambda_E^i}{s! s^{N-s} \mu_E^N}, & for\ N \ge s \end{cases} \tag{20}$$

where P_0 can be derived from the normalizing condition as:

$$P_0 = \left[\sum_{i=0}^{s-1} \frac{\lambda_E^i}{i!\, \mu_E^i} + \sum_{i=s}^{N} \frac{\lambda_E^i}{s^{i-s} s!\, \mu_E^i} \right]^{-1} \tag{21}$$

To simplify this expression, let

$$\rho = \frac{\lambda_E}{\mu_E} \ and \ \omega = \frac{\rho}{s}$$

Then,

$$\sum_{i=s}^{N} \frac{\lambda_E^i}{s^{i-s} s!\, \mu_E^i} = \sum_{i=s}^{N} \frac{\rho^i}{s^{i-s} s!} = \frac{\rho^i}{s!} \sum_{i=s}^{N} \omega^{i-s} = \begin{cases} \frac{\rho^s}{s!} \frac{1-\sigma^{N-s+1}}{1-\omega}, & \omega \ne 1 \\ \frac{\rho^s}{s!} (1 - s + 1), & \omega = 1 \end{cases}$$

Thus,

$$P_0 = \begin{cases} \left[\frac{\rho^s}{s!} \frac{1-\omega^{N-s+1}}{1-\omega} + \sum_{i=0}^{s-1} \frac{\rho^i}{i!} \right]^{-1}, & \omega \ne 1 \\ \left[\frac{\rho^s}{s!} (1 - s + 1) + \sum_{i=0}^{s-1} \frac{\rho^i}{i!} \right]^{-1}, & \omega = 1 \end{cases} \tag{22}$$

Now, we can derive the main performance metrics as follows. First, the blocking probability in the MEC subsystem can be obtained as given below:

$$P_l^i = P_N^i \tag{23}$$

P_N represents the probability for the number of message requests of not joining the system. Hence, $\lambda_E P_N$ represents the average number of message requests lost owing to the finite queue. Therefore, the effective arrival rate ($\boldsymbol{\lambda_e}$) in such queues is represented as follows:

$$\lambda_e = \lambda_E \left(1 - P_N^i\right) \tag{24}$$

The mean throughput service $\bar{\gamma}_R^i$ of each MEC core is given by:

$$\bar{\gamma}_E^i = \lambda_E \left(1 - P_l^i\right) \tag{25}$$

The core utilization of each MEC core instant can be written as follows:

$$U_E^i = \frac{\bar{\gamma}_E^i}{s \mu_E} = \omega \left(1 - P_l^i\right) \tag{26}$$

Now, we can derive the average number of message requests for each MEC core, and obtain:

$$\bar{L}_E^i = \sum_{j=1}^{N} j P_j^i \tag{27}$$

The average number of message requests waiting in each MEC core can be expressed as:

$$\bar{Q}_E^i = \sum_{j=s+1}^{N} (j - s) P_j^i \tag{28}$$

Finally, using Little's formula [31], the average response time and average waiting time for each MEC core are written as follows:

$$\bar{R}_E^i = \frac{\bar{L}_E^i}{\lambda_e} = \frac{\bar{L}_E^i}{\lambda_E \left(1 - P_N^i\right)} \tag{29}$$

$$\bar{W}_E^i = \frac{\bar{Q}_E^i}{\lambda_e} = \frac{\bar{Q}_E^i}{\lambda_E \left(1 - P_N^i\right)} \tag{30}$$

Now, we analyze the performance of the MEC subsystem by deducing the equations for the main performance parameters of the queue model. First, the blocking probability (loss probability) in all MEC core queues is given by:

$$P_l = \sum_{i=1}^{Z} P_N^i \tag{31}$$

The average number of message requests in the MEC subsystem can be expressed as follows:

$$\bar{L}_E = \sum_{i=1}^{Z} \bar{L}_E^i \tag{32}$$

The average number of message requests waiting in all MEC core queues can be obtained as follows:

$$\bar{Q}_E = \sum_{i=1}^{Z} \bar{Q}_E^i \tag{33}$$

The mean throughput of the MEC subsystem can be formulated as:

$$\bar{\gamma}_E = \sum_{i=1}^{Z} \bar{\gamma}_E^i \tag{34}$$

Finally, the average response time and the average waiting time in all MEC core queues can be expressed as:

$$\bar{R}_E = \frac{\bar{L}_E}{\bar{\gamma}_E} \tag{35}$$

$$\bar{W}_E = \frac{\bar{Q}_E}{\bar{\gamma}_E}. \tag{36}$$

5.3 5G cloud data center model (CDC)

This model can describe each vCPU core in the CDC subsystem that can be modeled as the M/M/m/C queuing model. As mentioned earlier, the client message after being processed by the MEC subsystem is either redirected to the CDC for further processing or storage with probability (P_{ec}) or departs from the system because the service has been completed with probability ($1 - P_{ec}$). Table 4 presents the main parameters used frequently in this section and their descriptions.

The M/M/m/C queuing system is a variation of a multicore system and only a maximum of C message requests in each vCPU core is allowed to stay in the system. Again, the number of message requests in the system is a birth–death process with appropriate rates and for the steady-state distribution [18, 30], we have:

$$P_i = \begin{cases} P_0 \frac{(P_{ec}\lambda_C)^i}{i!\mu_C^i}, & for\ 0 \le i \le m \\ P_0 \frac{(P_{ec}\lambda_C)^i}{m!m^{i-m}\mu_C^i}, & for\ m \le i \le C \end{cases} \tag{37}$$

$$P_C = \begin{cases} P_0 \frac{(P_{ec}\lambda_C)^C}{C!\mu_C^C}, & for\ C < m \\ P_0 \frac{(P_{ec}\lambda_C)^C}{m!m^{C-m}\mu_C^C}, & for\ C \ge m \end{cases} \tag{38}$$

where P_0 can be derived from the normalizing condition as:

$$P_0 = \left[\sum_{i=0}^{m-1} \frac{(P_{ec}\lambda_C)^i}{i!\,\mu_C^i} + \sum_{i=m}^{C} \frac{(P_{ec}\lambda_C)^i}{m^{i-m}m!\,\mu_C^i} \right]^{-1} \tag{39}$$

To simplify this expression, let

$$\rho = \frac{P_{ec}\lambda_C}{\mu_C} \; and \; \eta = \frac{\rho}{m}$$

Table 4 Key parameters used frequently in CDC queueing system

Parameters	Description
$P_{ec}\lambda_C$	Message arrival rate to CDC
$1/\mu_C$	Mean vCPU service time in CDC subsystem
P_{ec}	Probability of redirecting a message request from MEC to CDC subsystem
$1 - P_{ec}$	Possibility of leaving a message request from the MEC owing to completion of their service
U_R	Utilization of each core of CPUs in CDC subsystem
m	Number of vCPUs in each core
C	Limiting number of messages in each CPU core
M	Number of CPU cores in CDC subsystem
P_i	Equilibrium probability
η	$(P_{ec}\lambda_C)/(m\ \mu_C)$ offered load in each vCPU core
P_s^i	Loss probability CDC subsystem
$\bar{\gamma}_C^i$	Throughput service for each CPU core
$\bar{L}_C^i$	Average number of message requests for each CPU core
$\bar{Q}_C^i$	Average number of message requests waiting for each CPU core
$\bar{R}_C^i$	Average response time for each CPU core
$\bar{W}_C^i$	Average waiting time for each CPU core
$\bar{\gamma}_C$	Throughput service of CDC subsystem
$\bar{L}_C$	Average number of message requests in CDC subsystem
$\bar{Q}_C$	Average number of message requests waiting in CDC subsystem
$\bar{R}_C$	Average response time in the CDC subsystem
$\bar{W}_C$	Average waiting time in the CDC subsystem

Then,

$$\sum_{i=m}^{C} \frac{(P_{re}\lambda_C)^i}{m^{i-m}m!\,\mu_C^i} = \sum_{i=m}^{C} \frac{\rho^i}{m^{i-m}m!} = \frac{\rho^i}{m!} \sum_{i=m}^{C} \eta^{i-m} = \begin{cases} \frac{\rho^m}{m!} \frac{1-\eta^{C-m+1}}{1-\eta}, & \eta \ne 1 \\ \frac{\rho^m}{m!}(1-m+1), & \eta = 1 \end{cases}$$

Thus,

$$P_0 = \begin{cases} \left[\frac{\rho^m}{m!} \frac{1-\eta^{C-m+1}}{1-\eta} + \sum_{i=0}^{m-1} \frac{\rho^i}{i!} \right]^{-1}, & \eta \ne 1 \\ \left[\frac{\rho^m}{m!}(1-m+1) + \sum_{i=0}^{m-1} \frac{\rho^i}{i!} \right]^{-1}, & \eta = 1 \end{cases} \tag{40}$$

Now, we can derive the main performance metrics. First, the loss probability can be calculated as follows:

$$P_s^i = P_C^i \tag{41}$$

P_C represents the probability for the number of message requests of not joining the system. Hence, $P_{ec}\lambda_C P_C$ represents the average number of message requests lost owing to the finite queue. Therefore, the effective arrival rate (λ_e) in such queues is represented as follows:

$$\lambda_e = P_{ec}\lambda_C\left(1 - P_C^i\right) \tag{42}$$

The mean throughput service $\bar{\gamma}_C^i$ of each vCPU core is given by:

$$\bar{\gamma}_C^i = P_{ec}\lambda_C\left(1 - P_s^i\right) \tag{43}$$

The core utilization of each vCPU core instant can be written as follows:

$$U_C^i = \frac{\bar{\gamma}_C^i}{m\mu_C} = \eta\left(1 - P_s^i\right) \tag{44}$$

Now, we can derive the average number of message requests for each vCPU core, and obtain:

$$\bar{L}_C^i = \sum_{j=1}^{C} j P_j^i \tag{45}$$

The average number of message requests waiting for each vCPU core can be expressed as:

$$\bar{Q}_C^i = \sum_{j=m+1}^{C} (j-m) P_j^i \tag{46}$$

Finally, using Little's formula [31], the average response time and average waiting time for each vCPU core are written as follows:

$$\bar{R}_C^i = \frac{\bar{L}_C^i}{\lambda_e} = \frac{\bar{L}_C^i}{P_{ec}\lambda_C\left(1 - P_C^i\right)} \tag{47}$$

$$\bar{W}_C^i = \frac{\bar{Q}_C^i}{\lambda_e} = \frac{\bar{Q}_C^i}{P_{ec}\lambda_C\left(1 - P_C^i\right)} \tag{48}$$

Now, we analyze the performance of the CDC subsystem by deducing the equations for the main performance parameters of the queue model. First, the loss probability in all vCPU core queues in CDC subsystem is written as:

$$P_s = \sum_{i=1}^{M} P_C^i \tag{49}$$

The average number of messages requests for the CDC subsystem can be expressed as follows:

$$\bar{L}_C = \sum_{i=1}^{M} \bar{L}_C^i \tag{50}$$

The average number of message requests waiting for vCPU core queues in CDC subsystem can be calculated as follows:

$$\bar{Q}_C = \sum_{i=1}^{M} \bar{Q}_C^i \tag{51}$$

The mean throughput of the CDC subsystem can be formulated as:

$$\bar{\gamma}_C = \sum_{i=1}^{M} \bar{\gamma}_C^i \tag{52}$$

Finally, the average response time and the average waiting time for vCPU core queues in the CDC subsystem can be given as:

$$\bar{R}_C = \frac{\bar{L}_C}{\bar{\gamma}_C} \tag{53}$$

$$\bar{W}_C = \frac{\bar{Q}_C}{\bar{\gamma}_C}. \tag{54}$$

5.4 Overall system performance metrics

Based on the analysis of the queuing systems, we can now derive the main performance measures for the case where the system is used with all three queueing models (C-RAN, MEC, and CDC) as follows. First, the system throughput is the total throughput of entire queuing model in the proposed architecture, which can be expressed as follows:

$$\bar{\boldsymbol{\gamma}} = \bar{\boldsymbol{\gamma}}_{\boldsymbol{R}} + \bar{\boldsymbol{\gamma}}_{\boldsymbol{M}} + \bar{\boldsymbol{\gamma}}_{\boldsymbol{C}} \tag{55}$$

Next, the average response time in the proposed model is the sum of the average response times of the three queuing forms. The average response time equation can be written as:

$$\bar{R} = \bar{R}_R + \bar{R}_M + \bar{R}_C \tag{56}$$

Then, the average waiting time in proposed model is the sum of the average waiting times of all the queuing models in our proposed model. The average system waiting time can be expressed as:

$$\bar{W} = \bar{W}_R + \bar{W}_M + \bar{W}_C \tag{57}$$

The average number of message requests in the system is the total number of messages of all the queuing models, C-RAN, MEC, and CDC, which can be obtained as follows:

$$\bar{L} = \bar{L}_R + \bar{L}_M + \bar{L}_C \tag{58}$$

The average number of message requests waiting in the system is the average of the all the client messages in all the queuing models, which can be expressed as follows:

$$\bar{Q} = \bar{Q}_R + \bar{Q}_M + \bar{Q}_C \tag{59}$$

The loss probability of the system according to the blocking rate for the C-RAN, MEC, and CDC queuing models is written as:

$$P_{LOSS} = P_B + P_l + P_s \tag{60}$$

Finally, the expected system costs can be obtained as follows:

$$Cost_S = N_{vBBU} \cdot C_{vBBU} + N_{vMEC} \cdot C_{vMEC} + N_{vCPU} \cdot C_{vCPU} \tag{61}$$

where N_{vBBU} is the total number of vBBU cores in the C-RAN, N_{vMEC} is the total number of virtual servers in the MEC, N_{vCPU} is the total number of vCPU in the CDC, $\boldsymbol{C_{vBBU}}$ is the service cost for each vBBU core in the C-RAN, C_{vMEC} is the service cost for each virtual server in the MEC, and C_{vCPU} is the service cost for each vCPU core in the CDC.

The expected message request waiting time cost in the overall system can be expressed as follows:

$$Cost_W = \lambda \cdot \bar{W} \cdot C_W \tag{62}$$

where λ refers to the total arrival rate of all message requests, $\bar{W}$ indicates to the average waiting time in the system, and C_W refers the cost of message request waiting times. Now, we conclude that the overall system cost can be calculated as follows:

$$Cost = Cost_S + Cost_W \tag{63}$$

6 Results discussions

In this section, we present the numerical results of the proposed analytical model and compare them with the results obtained using the JMT simulator assuming the use of a single network slice, because the study of the remaining slices is similar to that of this network slice.

6.1 Simulation setup

Based on JMT [32], we study the results obtained by the proposed queuing model. The JMT suite is a collection of free, open-source discrete event simulation (DES) tools for the performance evaluation and modeling of computer systems and networks based on queuing system models. JMT contains six tools supporting various analyses: simulation of queuing network models (JSIMwiz), a graphical user-friendly interface for the simulator engine JSIM used by JSIMwiz (JSIMgraph), a workload characterization phase (JWAT), the identification of bottlenecks (JABA), the animation of Markov chain models underlying the queuing system models (JMCH), and a solution for the analysis of single or multiclass queuing networks with algorithms (JMVA) [33].

We propose various values of the simulation parameters to obtain a stable system. The main simulation parameters are presented in Table 5.

6.2 Results and discussion

In this section, we demonstrate through simulations and obtained performance measures the following: (i) the validation of our analytical model and (ii) the enhancement on the QoS performance when using the proposed model. We consider three different scenarios in order to study the performance of the proposed system in different environments as follows:

6.2.1 Scenario 1: impact of vBBU cores on system performance

This scenario studies the effect of the number of vBBU cores on the system performance parameter to validate the key performance formulas obtained from the analytical model by comparing them with the results obtained by the JMT simulator. We assume the different numbers of vBBU cores in our simulation are 26, 32, 38, and 44 cores. The following figures illustrate the performance of the system by the mean throughput, CPU utilization, number of message requests, average response time, average waiting time, and system loss rate.

In Fig. 3, we demonstrate the CPU utilization when the number of vBBU cores in the C-RAN changes [Eq. (8)] with respect to the message arrival rate. We noticed that the CPU utilization increases with the message arrival rate. Moreover, with the increase in the number of vBBU cores, the system yields higher performance. Therefore, we conclude that having a higher number of vBBU cores will increase the resource sharing scheme to gain more opportunities in obtaining the inactive resources; hence, enhancing the CPU utilization in the network.

We can clearly observe that when the 5G network has 300 k messages per second, with the usage of 26 vBBUs, the CPU

Table 5 Key parameter values used in our simulation

Parameters	Description	Value
λ	Message request arrival rate	[100,000 to 1,000,000] (Mesg/s)
$1/\mu_R$	Mean message request vBBU core service time in C-RAN subsystem	0.0001 (s)
$1/\mu_E$	Mean message request vServers service time in MEC subsystem	0.00001(s)
$1/\mu_C$	Mean message request vCPU service time in CDC subsystem	0.0002(s)
k	Limiting number of messages in each vBBU core	300
Z	Number of MEC nodes in MEC subsystem	5
N	Limiting number of messages in each MEC core	200
P_r	Possibility of leaving a message request from the MEC owing to completion of their service	0.4
M	Number of CPU cores in the CDC subsystem	20
m	Number of vCPUs in each core	10
C	Limiting number of messages in each CPU core	500

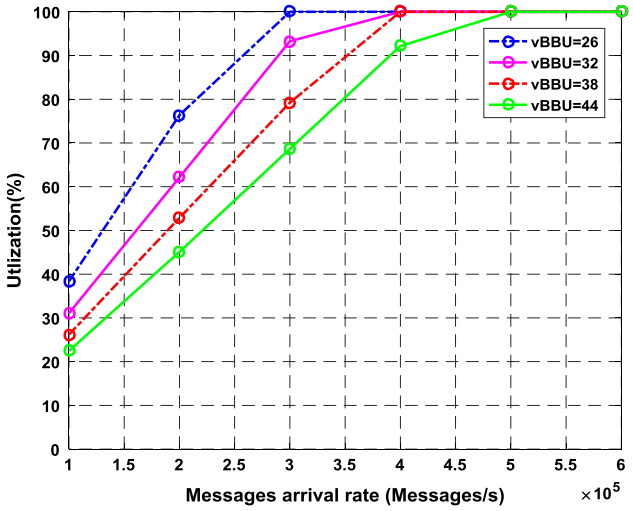

Fig. 3 Impact of vBBU cores on CPU utilization

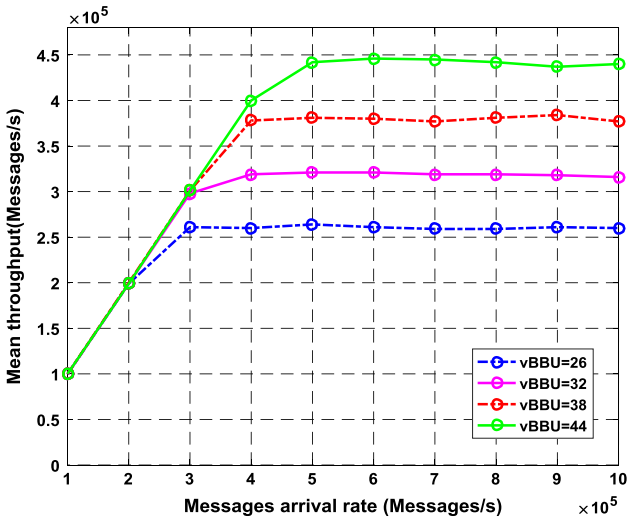

Fig. 4 Impact of vBBU cores on throughput

utilization reaches 100%. However, with 44 vBBUs, the CPU utilization is 70%. Clearly, it can be concluded that increasing the number of vBBUs reduces the CPU utilization, therefore, it is highly recommended to increase the number of vBBUs in 5G network systems. Based on what we have previously learned, we need to know that the studied 5G performance measures, in terms of mean response time, throughput, and average number in queue, will mainly depend on the CPU utilization level. Therefore, when the CPU reaches its maximum utilization, the vBBUs become saturated. At this point, there will be no further improvements in terms of all those performance measures and their values become constant as we can observe and explain in the next coming figures.

Figure 4 shows the system throughput with different numbers of vBBU cores at different arrival rates [Eq. (55)]. The system throughput evaluation is performed for messages arrival rates varying from 100,000 to 1,000,000 messages/s. We can see that as the number of vBBU cores increases, the throughput of the system increases. However, in the data performance case, when the arrival rate reaches 500,000 messages/s, we see that as the message arrival rate and the number of vBBU cores increases, the throughput increases. It can be concluded that the higher performance level is achieved with the higher number of vBBU cores.

As we explained in Fig. 3 in which the CPU reaches its maximum utilization at different vBBU numbers and message arrival rates, the vBBU will reach its maximum throughput for all the times when the CPU is fully utilized. For example, when we have 26 vBBU, the CPU reaches its maximum utilization when the messages arrival rate is 300 k messages/s, and therefore, the throughput in Fig. 4 reaches its maximum value and stays constant.

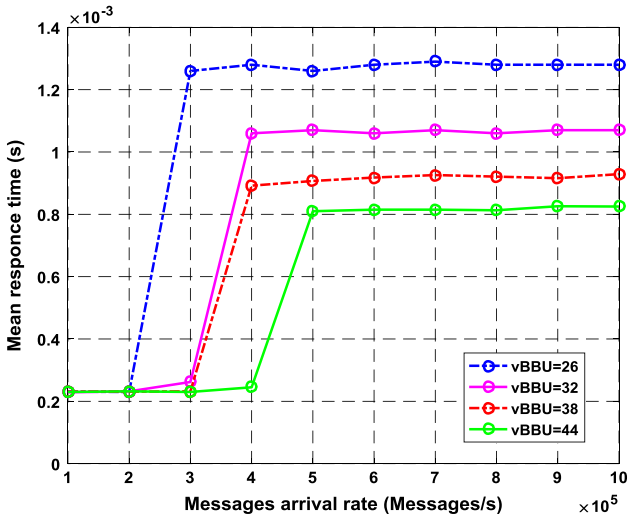

Fig. 5 Impact of vBBU cores on average response time

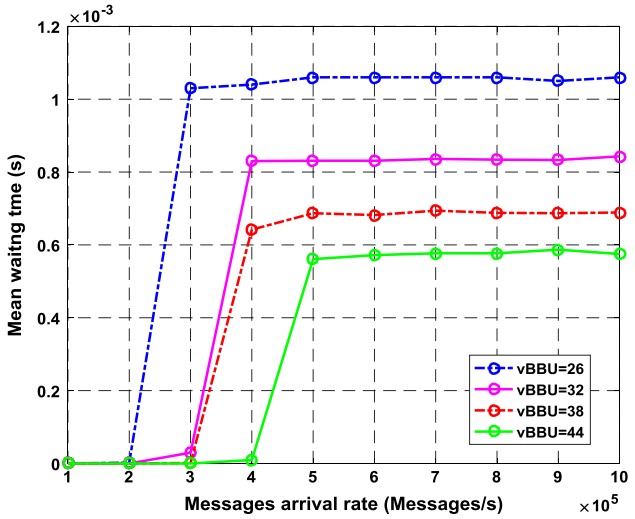

Fig. 6 Impact of vBBU cores on average waiting time

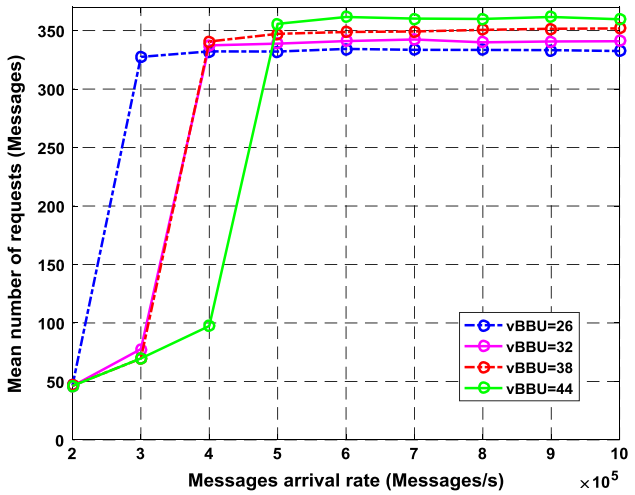

Fig. 7 Impact of vBBU cores on average number of requests

Figure 5 shows the average response time for different arrival rates [Eq. (56)] with four different vBBU core numbers of 26, 32, 38 and 44. It can be seen that the average response time decreases as the message arrival rate and the vBBU core count increase. For example, when we use the vBBU with 26 cores and the message arrival rate reaches 1,000,000 messages/s in our system, the average response time is 1.28 ms, whereas with 44 cores, the average response time is 0.825 ms. Clearly, we conclude that when the number of vBBU cores increases, the average response time parameter improves; thus, enhancing the QoS performance.

From Fig. 5, we can clearly observe that when the messaging rate exceeds 500 k messages/s, the mean response time rapidly increases to 0.8 ms when the number of vBBU is 44. The cause of this rapid change is already explained in Fig. 3, in which the utilization of the CPU approaches 100% when vBUU is 44, this saturates the vBBU and the mean response time will consequently increase rapidly. This phenomenon will be obvious in all response time figures where the response time increases rapidly when the CPU reaches its maximum utilization. In order to reduce the average response time, we need to increase the number of vBBUs. Therefore, for example when we increase the number of vBBUs from 26 to 44, the average response time greatly decreases from around 1.2 ms to almost 0.25 ms.

Figure 6 shows the effect of the number of vBBU cores on the average waiting time at different arrival rates [Eq. (57)]. It is evident that when the message arrival rate is low, the average waiting time for the queues are low. However, as the arrival rate increases, the average waiting time starts to increase. For higher arrival rates, the average waiting time remains almost unchanged. We also clearly see that the average waiting time depends on the number of vBBU cores; when the number of vBBU cores increases, the average waiting time decreases. In other words, we get better performance with a larger number of vBBU cores.

Figure 7 shows the average number of messages when the number of vBBU cores in C-RAN and the message arrival rate change [Eq. (58)]. The figure shows that when the message arrival rate increases with more vBBU cores, the system can process more messages; this also allows for a reduction in the messages in the system, which is natural and expected.

As we explained in Fig. 3 in which the CPU reaches its maximum utilization at different vBBU numbers and message arrival rates, the queue size will fill-up, and the average number of messages in the queue will reach its maximum value for all the times when the CPU is fully utilized. For example, when we have 44 vBBU, the CPU reaches its maximum utilization when the messages arrival rate is 500 k messages/s, and therefore, the queue is full and stays constant.

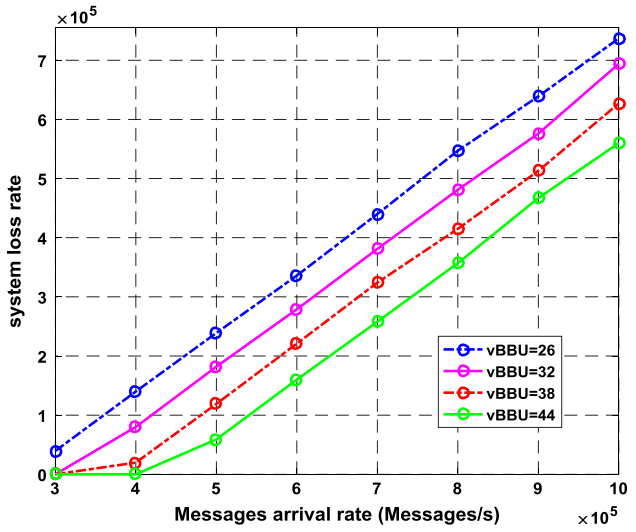

Fig. 8 Impact of vBBU cores on system loss rate

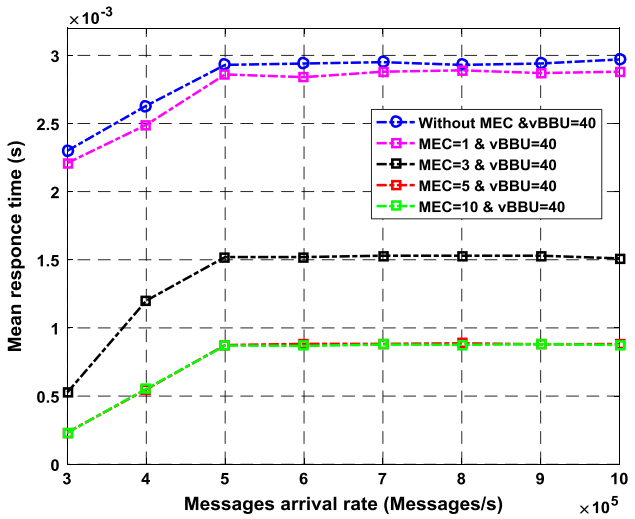

Fig. 9 Impact of MEC nodes on average response time

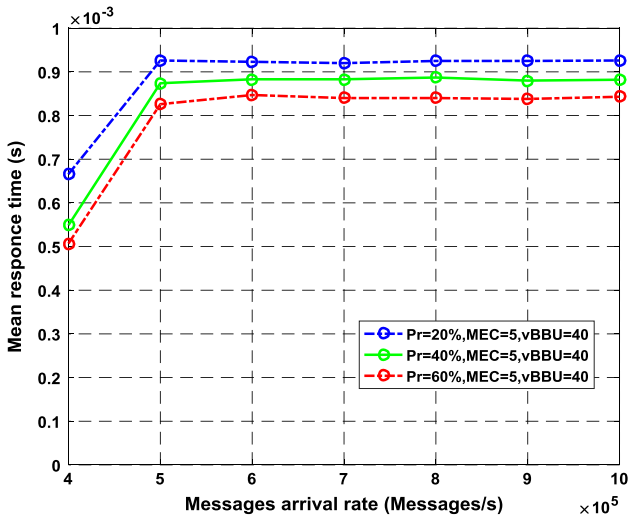

Fig. 10 Impact of MEC nodes on average response time according to probability P_{ec}

Figure 8, we depict the system loss rate against the message arrival rate with respect to the number of vBBU cores (Eq. (60). The system loss rate evaluation is performed for message arrival rates varying from 100,000 to 1,000,000 messages/s. From Fig. 8, we can see that the blocking probability of the system with a higher number of vBBU cores is better. For example, when we use a vBBU with 32 cores and the message arrival rate reaches 1,000,000 messages/s in our system, the system loss rate indicates to 694,000 messages/s, whereas with 44 cores, the system loss rate indicates 560,000 messages/s. Consequently, it can be concluded that the system with more vBBU cores produces the best performance in terms of the blocking probability compared to the system with fewer vBBU cores; thus, improving the QoS performance for mobile networks.

From above figures in this scenario, it can be observed there is a close consistency between the analytical model and the results obtained from the JMT simulation.

6.2.2 Scenario 2: impact of MEC nodes on system delay

This scenario studies the impact of the MEC nodes on the system delay according to different message arrival rates, as shown in Figs. 9 and 10. We compare the simulation results for the average response time with and without MEC nodes. Figure 9 shows the average response time at different message arrival rates either using the model without MEC nodes or with 1, 3, 5, and 10 MEC nodes. The average response time without MEC node is higher than with MEC nodes when the messages arrival rate is high. However, when the number of MEC nodes reaches a certain limit (for example, five MEC nodes in Fig. 9), there is no change in the average response time as the number of MEC node increases further. Consequently, the system with more MEC nodes results in higher performance in the system delay compared to the system with fewer MEC nodes. Furthermore, Fig. 10 shows that improving the average response time also depends on the ability of the MEC servers to complete the service for many messages in the system. Moreover, we observe that when P_{ec} is higher, the average response time is low. Thus, we conclude that MEC servers are important to provide low latency to radio network information on the edge of the mobile network.

6.2.3 Scenario 3: effects of number of vBBU and MEC on overall system cost

This scenario studies the effects of the number of vBBU cores and MEC nodes on the overall system cost according to different message arrival rates. As mentioned earlier, the

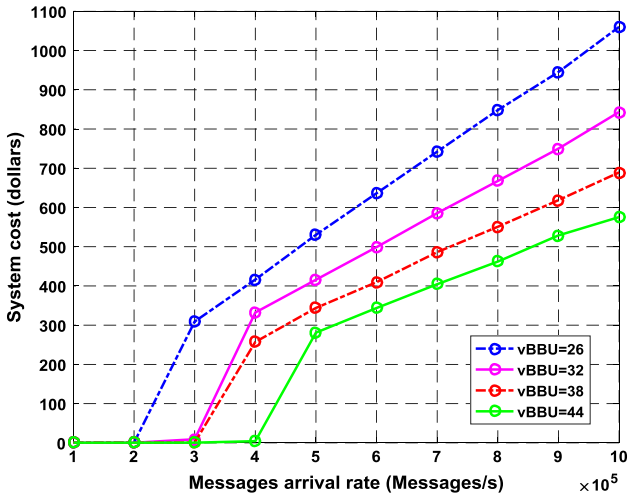

Fig. 11 Impact of vBBU cores on overall system cost

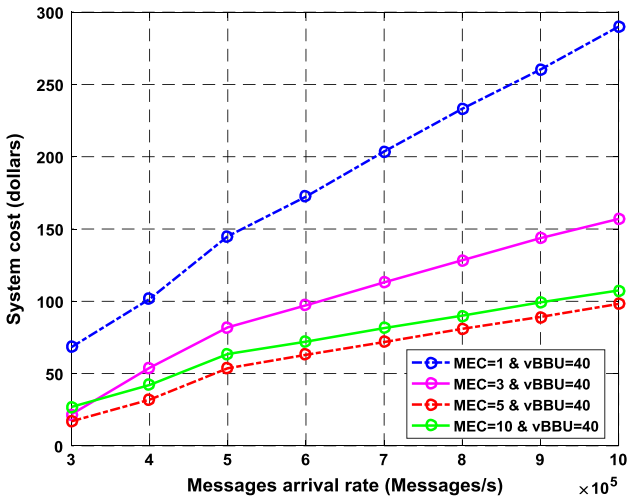

Fig. 12 Impact of MEC nodes on overall system cost

expected system cost is calculated by [Eq. (63)], based on expected waiting time cost and expected service cost of the system. To study the impact of the number of vBBU cores on the total system cost, we assume that waiting time cost is \$1/s (1 dollar per second) and the service cost of each vBBU core is \$0.01/s (0.01 dollar per second). We ignore the cost of service for the MEC and CDC cores and assume they are fixed. Figure 11 shows the overall system cost, ignoring the costs of the MEC and CDC cores, when the number of vBBU cores in the C-RAN changes. It can be observed that the overall system cost increases as the message arrival rate increases. In addition, when the system runs a vBBU with more cores, the system cost is lower compared to the system with fewer vBBU cores. For example, the system cost with 26 vBBU cores reaches \$1050, whereas with 44 vBBU cores, the cost is decreased and shows a value of \$590 at 1,000,000 messages/s. Therefore, we conclude that the system with more vBBU cores shows better performance in the cost, overcoming the losses resulting from the use of vBBU cores.

To study the impact of the number of MEC nodes on the total system cost, we assume that the waiting time cost is \$0.1/s (0.1 dollar per second) and the service cost of each MEC node is \$2/s (2 dollars per second). We ignore the cost of service for the C-RAN and CDC cores and assume they are fixed. Figure 12 shows the total cost of the system, ignoring the costs of the C-RAN and CDC cores when the number of MEC nodes changes. We can see that the total cost of the system decreases with the increase in the number of MECs. However, when the number of MEC nodes reaches a certain limit (for example, five MEC nodes in Fig. 12), we can see that the total cost increases as the number of MEC nodes increases, because the average waiting time becomes constant or zero. Thus, further increases in MEC nodes in the mobile network become useless. Therefore, we conclude that the system with more MEC to certain limit commensurate with network resources produces better performance in the cost, overcoming the losses resulting from the use of MEC nodes.

7 Conclusion

In this paper, analytical and simulation models for a proposed network slicing system within 5G mobile networks have been presented. We proposed that each network slice consists of a separate C-RAN, MEC and CDC, where MEC and C-RAN are highly complementary technologies to meet the key QoS parameters. We derived closed-form formulas for the main performance measures, such as CPU utilization, system throughput, system drop rate, average number of message requests, average response time, and average waiting time. The proposed system model was cross-validated based on the JMT simulator.

We considered three different scenarios in order to study the performance of the proposed system in different environments: the impact of the number of vBBU cores on the performance of system, the impact of the number of MEC nodes on the system delay, and the effects of the number of vBBU and MEC on the overall system cost. We have provided many quantitative examples, including C-RAN with 26, 32, 38, and 44 vBBU cores. From the simulation results, it can be observed there is a close consistency between the analytical model and the results obtained from the JMT simulation. Furthermore, we conclude that MEC servers are important to provide low latency to radio network information on the edge of the mobile network. In addition, a system with more vBBU cores gives better performance in the cost, overcoming the losses resulting from the use of vBBU cores.

Acknowledgements This work was supported by the Research Center of College of Computer and Information Sciences, King Saud University. The authors are grateful for this support.

Compliance with ethical standards

Conflict of interest All authors declare that they have no conflict of interest.

References

1. Choi, Y.-I., & Park, N. (2017). Slice architecture for 5G core network. In *9th International conference on ubiquitous and future networks (ICUFN), 2017* (pp. 1–7).
2. Pérez-Romero, J. et al. (2018). On the configuration of radio resource management in a sliced RAN. In *IEEE/IFIP network operations and management symposium, 2018* (pp. 1–9).
3. Agiwal, M., Roy, A., & Saxena, N. (2016). Next generation 5G wireless networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials, 18*(3), 1617–1655.
4. Yousaf, F. Z., et al. (2017). NFV and SDN-key technology enablers for 5G networks. *IEEE Journal on Selected Areas in Communications, 35*(11), 2468–2478.
5. Velasco, L., et al. (2018). An architecture to support autonomic slice networking. *Journal of Lightwave Technology, 36*(1), 135–141.
6. Wei, H., Zhang, Z., & Fan, B. (2017). Network slice access selection scheme in 5G. In *IEEE 2nd information technology, networking, electronic and automation control conference (ITNEC), 2017* (pp. 1–8).
7. Richart, M., et al. (2016). Resource slicing in virtual wireless networks: A survey. *IEEE Transactions on Network and Service Management, 13*(3), 462–476.
8. Nguyen, V.-G., et al. (2017). SDN/NFV-based mobile packet core network architectures: A survey. *IEEE Communications Surveys and Tutorials, 19*(3), 1567–1602.
9. Kalyoncu, F., Zeydan, E., & Yigit, I. O. (2018) A data analysis methodology for obtaining network slices towards 5G cellular networks. In *IEEE 87th Vehicular Technology Conference (VTC Spring)* (pp. 1–7).
10. Afolabi, I., et al. (2017). End-to-end network slicing enabled through network function virtualization. In *IEEE Conference on Standards for Communications and Networking (CSCN)* (pp. 1–8).
11. Olwal, T. O., Djouani, K., & Kurien, A. M. (2016). A survey of resource management toward 5G radio access networks. *IEEE Communications Surveys & Tutorials, 18*(3), 1656–1686.
12. Checko, A., et al. (2015). Cloud RAN for mobile networks—A technology overview. *IEEE Communications surveys & tutorials, 17*(1), 405–426.
13. Taleb, T., et al. (2017). On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration. *IEEE Communications Surveys & Tutorials, 19*(3), 1657–1681.
14. Foukas, X., et al. (2017). Network slicing in 5G: Survey and challenges. *IEEE Communications Magazine, 55*(5), 94–100.
15. Zhang, L., et al. (2018). Filtered OFDM systems, algorithms, and performance analysis for 5G and beyond. *IEEE Transactions on Communications, 66*(3), 1205–1218.
16. Akihiro, N., et al. (2017). End-to-end Network Slicing for 5G Mobile Networks. *Journal of Information Processing, 25,* 153–163.
17. Chen, H., & Yao, D. D. (2013). *Fundamentals of queuing networks: Performance, asymptotic, and optimization* (Vol. 46). Berlin: Springer.
18. Bolch, G., et al. (2006). *Queuing networks and Markov chains: Modeling and performance evaluation with computer science applications*. Hoboken: Wiley.
19. Bhat, U. N. (2015). *An introduction to queuing theory: Modeling and analysis in applications*. Basel: Birkhäuser.
20. Liang, C., & Yu, F. R. (2015). Wireless network virtualization: A survey, some research issues and challenges. *IEEE Communications Surveys & Tutorials, 17*(1), 358–380.
21. Han, B., et al. (2018). Admission and congestion control for 5G network slicing. In *IEEE Conference on Standards for Communications and Networking (CSCN)* (pp. 1–9).
22. Narmanlioglu, O., Zeydan, E., & Arslan, S. S. (2018). Service-aware multi-resource allocation in software-defined next generation cellular networks. *IEEE Access, 6,* 20348–20363.
23. Ye, Q., et al. (2019). End-to-end delay modeling for embedded VNF chains in 5G core networks. *IEEE Internet of Things Journal, 6*(1), 692–704.
24. Kurtz, F., et al. (2018). Network slicing for critical communications in shared 5G infrastructures-an empirical evaluation. In *4th IEEE Conference on Network Softwarization and Workshops (NetSoft)* (pp. 1–9).
25. Zanzi, L., & Sciancalepore, V. (2018) On guaranteeing end-to-end network slice latency constraints in 5G networks. In *15th International Symposium on Wireless Communication Systems (ISWCS)* (pp. 1–8).
26. Costanzo, S., et al. (2018). Dynamic network slicing for 5G IoT and eMBB services: A new design with prototype and implementation results. In *3rd Cloudification of the Internet of Things (CIoT)* (pp. 1–9).
27. Matthiesen, B., Aydin, O., & Jorswieck, E. A. (2018). Throughput and energy-efficient network slicing. In *22nd International ITG Workshop on Smart Antennas, 2018* (pp. 1–9).
28. Afolabi, I., et al. (2018). Network slicing and softwarization: A survey on principles, enabling technologies, and solutions. *IEEE Communications Surveys & Tutorials, 20*(3), 2429–2453.
29. Toosi, A. N., et al. (2019). *Management and orchestration of network slices in 5G, fog, edge and clouds* (pp. 79–101). Fog and Edge Computing: Principles and Paradigms.
30. Sahner, R. A., Trivedi, K., & Puliafito, A. (2012). *Performance and reliability analysis of computer systems: An example-based approach using the SHARPE software package*. Berlin: Springer.
31. Nelson, R. (2013). *Probability, stochastic processes, and queueing theory: The mathematics of computer performance modeling*. Berlin: Springer.
32. Bertoli, M., Casale, G., & Serazzi, G. (2009). JMT: Performance engineering tools for system modeling. *ACM SIGMETRICS Performance Evaluation Review, 36*(4), 10–15.
33. Fishman, G. S. (2013). *Discrete-event simulation: Modeling, programming, and analysis*. Berlin: Springer.

Salman A. AlQahtani is currently a Full professor at the department of computer engineering, college of computer and information sciences, King Saud University, Riyadh, Saudi Arabia. He serves also as a senior consultant in computer communications, integrated solutions and digital forensics for few development companies and government sectors in Saudi Arabia. Dr AlQahtani's main research activities are in Radio Resource Management (RRM) for wireless and cellular networks (4G, 5G, IoT, Industry 4.0, LTE, LTE-Advanced, Femtocell, Cognitive radio, Cyber Sovereignty...) with focus on Call Admission Control (CAC), Packet Scheduling, radio resource sharing and Quality-of-Service (QoS) guarantees for data services. In addition, his interests also include performance evaluation of packet switched network, system model and simulations and integration of heterogeneous wireless networks. Finally, my interests also extend to the area of digital forensics .

Waseem A. Alhomiqani received the B.Sc. degree in Computer Science and Engineering from Aden University, Yemen in 2006 and the M.Sc. degree in Computer Engineering from King Saud University, Saudi Arabia in 2016. Currently, he is a PhD student in the Department of Computer Engineering at King Saud University, Riyadh, Saudi Arabia. He has work as lecturer at Seiyun University, Seiyun, Yemen since 2007 . His current research interests include Wi-Fi, WiMAX, LTE and Cellular Access, 5G wireless networks, Network Slicing, Internet of Things, Packet Scheduling, Quality of Service (QoS) and Network Computer Modelling.