

Article

Efficient Multi-Player Computation Offloading for VR Edge-Cloud Computing Systems

Abdullah Alshahrani ^{1,*}, Ibrahim A. Elgendy ^{2,3,*} , Ammar Muthanna ⁴ ,
Ahmed Mohammed Alghamdi ⁵  and Adel Alshamrani ⁶

¹ Department of Computer Science and Artificial Intelligence, College of Computer Science and Engineering, University of Jeddah, Jeddah 21493, Saudi Arabia

² School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China

³ Department of Computer Science, Faculty of Computers and Information, Menoufia University, Shibin el Kom 32511, Egypt

⁴ Department of Communication Networks and Data Transmission, St. Petersburg State University of Telecommunications, St. Petersburg 193232, Russia; muthanna.asa@spbgut.ru

⁵ Department of Software Engineering, College of Computer Science and Engineering, University of Jeddah, Jeddah 21493, Saudi Arabia; amalghamdi@uj.edu.sa

⁶ Department of cybersecurity, College of Computer Science and Engineering at University of Jeddah, Jeddah 21493, Saudi Arabia; Asalshamrani@uj.edu.sa

* Correspondence: asalshahrani2@uj.edu.sa (A.A.); ibrahim.elgendy@hit.edu.cn (I.A.E.)

Received: 25 June 2020; Accepted: 3 August 2020; Published: 10 August 2020



Abstract: Virtual reality (VR) is considered to be one of the main use cases of the fifth-generation cellular system (5G). In addition, it has been categorized as one of the ultra-low latency applications in which VR applications require an end-to-end latency of 5 ms. However, the limited battery capacity and computing resources of mobile devices restrict the execution of VR applications on these devices. As a result, mobile edge-cloud computing is considered as a new paradigm to mitigate resource limitations of these devices through computation offloading process with low latency. To this end, this paper introduces an efficient multi-player with multi-task computation offloading model with guaranteed performance in network latency and energy consumption for VR applications based on mobile edge-cloud computing. In addition, this model has been formulated as an integer optimization problem whose objective is to minimize the sum cost of the entire system in terms of network latency and energy consumption. Afterwards, a low-complexity algorithm has been designed which provides comprehensive processes for deriving the optimal computation offloading decision in an efficient manner. Furthermore, we provide a prototype and real implementation for the proposed system using OpenAirInterface software. Finally, simulations have been conducted to validate our proposed model and prove that the network latency and energy consumption can be reduced by up to 26.2%, 27.2% and 10.9%, 12.2% in comparison with edge and cloud execution, respectively.

Keywords: virtual reality; edge-cloud computing; multilevel; computation offloading; latency; energy consumption; 5G

1. Introduction

Virtual reality (VR) is considered as one type of the most widely-announced applications of the fifth-generation cellular system (5G) [1–3]. In addition, many applications and market sectors are expected to be introduced in many areas of life [4]. Furthermore, augmented reality (AR) is considered as a new technology that enables the augmentation of real objects in the surrounding environments such as perceptual information extracted from multiple sensory modes containing haptic, visual and auditory sensors [5,6]. In addition to that, the recent developments and advances in sensory

manufacturing, AR devices (e.g., Epson Moverio and Google Glass) and AR toolkits (e.g., ARKit) lead to developing a wide range of AR/VR applications in various domains [7,8], which are considered to be ultra-low latency applications and required an end-to-end latency in terms of 5 ms [9–11].

However, designing a reliable AR/VR system that achieves the required latency constraints and other required quality of service (QoS) requirements is a challenge. This is due to the long distance between the application server and the end users, which leads to increase the network latency and thus, affects the user experience [12].

To address these issues, mobile edge computing (MEC), software-defined networking (SDN) and network function virtualization (NFV) have been promised as new technologies that have received much attention lately and can provide the AR/VR applications with an ultra-low latency [13–16]. MEC is a recent paradigm that brings the computing capabilities at the edge of the radio access network (RAN), near to end-users and thus, reduces the communication latency. This paradigm turns the centralized huge data centers to distributed small edge servers and thus, moves from the centralization to the decentralization of computing resources [17]. Introducing edge-cloud servers in the way between the AR end-user and the application server achieves many benefits that include the reduction of communication latency.

SDN and Network softwarization are another key paradigm and recent communication technology which provide the required flexibility and open the way for network innovation by separating data and control plans. In addition, SDN deployment for AR networks provides the required reliability and flexibility. Furthermore, SDN also can manage the distributed MEC servers and control the communication with the application server.

Based on 5G technology, data can be transferred with high speed and low latency, where society will rapidly enter the new era of smart cities and the Internet of things (IoT). Industry stakeholders have identified several potential use cases for 5G networks, and ITU-R has identified three important categories for these use cases: Enhanced Mobile Broadband (eMBB)—enhanced broadband indoor and outdoor, corporate collaboration, augmented and virtual reality; massive machine-type communications (mMTC)—IoT, smart agriculture, smart cities, energy monitoring, smart home, remote monitoring; and ultra-reliable low-latency communications (URLLC)—autonomous vehicles, smart grids, remote patient monitoring and telemedicine, production automation.

According to wireless operators, eMBB will be the primary scenario for using 5G in the early stages of network deployment. Based on eMBB, densely populated areas will be covered by high-speed mobile broadband, users will have access to high-speed streaming data on-demand on home devices, screens and mobile devices, and we will see services for corporate collaboration development. Some operators are also considering eMBB as the solution in areas where there are not enough copper cable connections or fiber-optic connections to the living quarters. 5G technology is also expected to contribute to the development of smart cities and IoT by deploying a large number of low-power sensor networks in urban and rural areas. Due to its safety and reliability, 5G technology can be used to ensure public safety, as well as to provide critical services, for example, in smart grids, the work of the police and security services, energy and water supply enterprises and healthcare.

With the development of computation offloading for MEC system, several related system architectures and approaches have been proposed in which the intensive computation tasks of the mobile devices will be offloaded to a richer resource located at the edge of the radio access network (RAN) [18,19]. However, most of these studies address only two levels of computation offloading in MEC systems [20,21], while few studies address the multi-level computation offloading [22]. In addition, most of these studies address single user or multi-user with only single computation task in which minimizing energy consumption, allocating radio and computation resources efficiently, and/or satisfying latency requirements for mobile users are the main goals. Consequently, the main motivations behind our work are based on the following observations:

- Considering a multi-level environment for multi-player computation offloading especially in the case of a large number of mobile devices and the resources of edge server are not enough is an important issue.
- In the MEC system, most of the complex mobile applications have many tasks that need to be offloaded and executed. Therefore, addressing the multi-task issue is important.
- Computing resources on the edge and cloud server and computing tasks of the mobile devices are considered to be the main factors that play an important role in the efficiency of multi-level, multi-player, multi-task edge-cloud computing system. It is, therefore, crucial to have an effective policy that is joint between them.

Motivated by such considerations, in this paper, we introduced a reliable AR system able to support ultra-low latency AR applications with the announced specifications. This system has deployed a multi-level, multi-player with multi-task computation offloading environment to provide the computing resources at the edge of the RAN which can reduce the communication latency of the VR tasks. In addition, we formulated the computation offloading as an integer optimization problem whose objective is to minimize the sum cost entire system in terms of network latency and energy consumption. The main contributions of this paper include:

- An efficient computation offloading model is formulated as an integer optimization problem with the objective of minimizing the sum cost of an entire system in terms of network latency and energy consumption for a multi-level multi-player multi-task edge-cloud computing systems. In addition, our environment has considered a single cloud computing which connected with the edge computing server via an intelligent core network that is built based on SDN technology to provide with more resources when the number of VR devices increases and the resources of edge server becomes not enough.
- An efficient algorithm has been designed which provides comprehensive processes for deriving the optimal computation offloading decision.
- Three main VR applications have been considered which are multiple player VR games, Holograms and 360-degree Rv video applications.
- Finally, simulations have been conducted to validate our proposed model and prove that the network latency and energy consumption can be reduced by up to 26.2%, 27.2% and 10.9%, 12.2% in comparison with edge and cloud execution, respectively. In addition, we provide a prototype and real implementation for the proposed system using OpenAirInterface software.

The rest of the paper is organized as follows. Section 2 introduces three main VR applications. Section 3 reviews related work on computation offloading policies. Section 4 presents our system model for multi-level multi-player with multi-task computation offloading and the designed algorithm. Simulation experiments and prototype implementation are conducted in Section 5. Finally, Section 6 concludes the paper.

2. VR Applications

2.1. Holograms over Proposed System

To increase the application effectiveness, augmented reality and virtual reality are increasingly used in conjunction with other technologies, for example, with the Internet of things applications, Tactile Internet and holographic telepresence [23–26]. The holographic presence makes AR/VR more spectacular for the user and allows to see virtual holograms, which are volumetric color images. Modern equipment allows you to create practical holograms not different from real things. Including such an effect is achieved due to accurate tracking of the user's position in a given space and depending on his location playing him a stereoscopic image. AR/VR technologies allow us to project static or animated objects into real environments, thereby expanding the physical world. Earlier designs of holograms for AR are based on so-called displays in the air, sometimes also called displays in free

space. Projected graphic objects are displayed in the air on free projection surfaces, such as a poorly visible fog wall or fog screen created by an installed fan [27]. One of the most popular is the HoloLens equipment from Microsoft [28]. Since both Microsoft HoloLens and AR glasses are capable of tracking head movements, they create the impression of a constant presence of holographic geospatial objects in the user's environment. Even if the user walks in a certain area, usually indoors, the holograms remain and adapt to the user's location and viewing perspective. This constant and adaptable holographic projection can lead to visualization approaches that bring additional benefits for cognitive processing. Presented as the first stand-alone holographic computer, HoloLens unites the physical and digital worlds, allows users to interact with digital content and interact with holograms in mixed reality. The work [29] is devoted to a technical assessment of the use of HoloLens for multimedia applications.

2.2. Multiple Players VR Games over Proposed System

For better illustration of the operation of the cloudlet and other higher layer edge-cloud units, an example of multi-player VR games is considered. In multiple players VR games, players use their VR supported devices to play an intended game, however these games run over the remote application server. In order to reduce the latency and efficient use of energy and computing resources of users' VR devices, the proposed system is introduced. Figure 1 presents the multi-player VR game system runs over the proposed system.

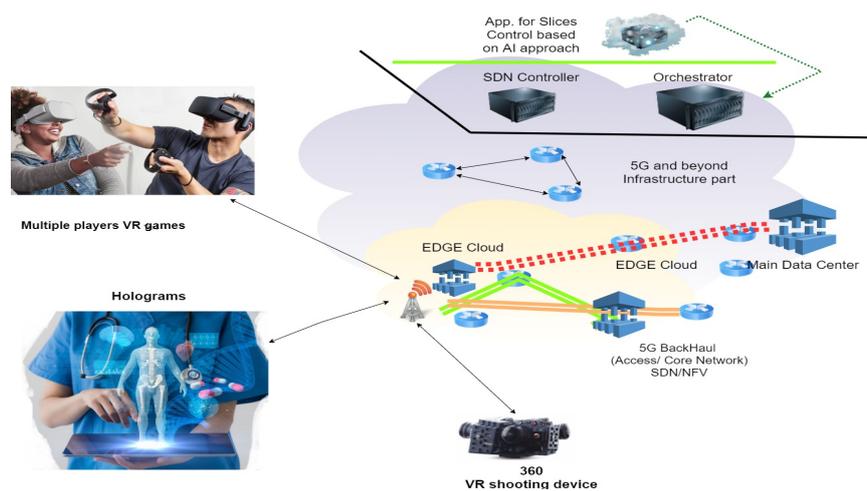


Figure 1. Intelligent core network based on mobile edge computing (MEC) servers with multiple users.

The VR user with limited computing or energy resources searches, the surrounding, for devices with available computing and energy resources capable of hosting computing tasks associated with the VR game user. These devices are referred to as cloudlet and it may be represented by a powerful smartphone, notebook or tablet. Computing tasks are offloaded from the VR user to the cloudlet over D2D communication interface. WiFi Direct represents an efficient interface and it can be deployed with the method proposed in [6]. The offloading process is held based on the developed algorithm that is introduced in Section 4.4.

If the VR game user can not find a nearby cloudlet, it turns to offload its computing tasks to the next level of edge-cloud units, which is the micro-cloud edge servers connected to cellular base stations. Micro-cloud edge servers are small edge units that have limited computing and energy resources. These servers are deployed to provide the computing and energy resources at the edge of the RAN and thus, achieve higher latency efficiency and reduce the traffic passed to the core network. Micro-cloud edge servers receive and handle computing tasks from VR users or from corresponding cloudlets that have not sufficient computing or energy resources.

Based on the available resources and the considered offloading algorithm, micro-cloud edge servers handle the received computing tasks or offload them to the higher edge-cloud units,

i.e., mini-cloud units. Each group of micro-cloud units are connected physically with a higher computing and energy capability edge-cloud unit; referred to as mini-cloud. All distributed mini-cloud edge units are connected directly to the core network cloud that represents the interface to the application server.

2.3. 360-Degree Video Streaming over Proposed System

In order to illustrate the benefits of the proposed system, another important VR application is considered. The 360-degree video streaming technology becomes a demand for many VR applications. The local execution of video processing indicates low performance for high-resolution video applications, e.g., 4 k videos and higher resolutions. To this end, MEC technology should be involved and video computing tasks should be offloaded over an appropriate communication link to the edge-cloud server. Moreover, an efficient data offloading scheme should be introduced for efficient offloading of computing tasks.

The introduction of heterogeneous distributed edge-cloud servers provides the computing and energy resources to the mobile VR devices and thus, video processing and decoding tasks can be offloaded. The considered MM-MEC system can be used for achieving high efficiency of 360-degree video applications of mobile VR devices. However, this kind of VR application requires a proper communication interface that achieves high spectral efficiency, proper for video applications with the required QoE.

To achieve high transmission QoS for VR-360-degree video applications, we consider millimeter wave (mmWave) as the communication interface. The IEEE 802.11ad standard is a multi-Gigabit wireless standard that uses the V band at a frequency of 60 GHz. The use of high wireless bandwidth is efficient for achieving a higher capacity of video-based applications. The main issue with the mmWave compared with the traditional interfaces, e.g., WiFi, is the limited communication range and thus, it is recommended for outdoor applications rather than indoor ones. Due to the recent advances in the antenna design, recent techniques have been developed for adopting mmWave to indoor applications.

3. Related Works

In recent years, numerous approaches and optimization models have been proposed for addressing the challenges of mobile devices using MEC by applying the computation offloading concept. Most of these studies handle only two levels of computation offloading in MEC systems [20,21], while few studies address the multi-level computation offloading [22]. In this section, a brief overview of the common approaches will be introduced.

Colman-Meixner et al. [30] introduced a 5G City and discussed how the advanced media services will be facilitated using 5G technology such as ultra-high-definition video, augmented and virtual reality. In addition, the opportunities provided by 5G technology and changes in the work of the telecommunications service provider are being studied. Furthermore, three different use cases are presented and their use in public networks, as well as the advantages of using this model for infrastructure owners and media service providers, is described. While in [31], Elbamby et al. studied the problem of low-latency wireless networks of virtual reality. Then, to solve this problem, the authors proposed to use information about user positions, proactive computing and caching to minimize computation latency.

Real prototypes for VR applications were implemented in which some of them used edge computing [32,33], while other prototypes did not use edge computing [34,35]. For the prototypes which used edge computing, Hou et al. discussed how to enable a portable and mobile VR device with VR glasses for wireless connecting to edge computing devices in [32]. In addition, the authors explored the main issues which are associated with the inclusion of a new approach to wireless VR with edge computing and various application scenarios. Furthermore, they provided an analysis of the delay requirements to enable wireless VR, and to study several possible solutions. Whereas, the computation

offloading for VR gaming applications has been considered in [33] in which minimizing the network latency is the main goal.

Meanwhile, in [34], Hsiao et al. dealt with issues related to information security and addressed the existing security system shortcomings in augmented reality (AR) technologies, artificial intelligence, wireless, 5G, big data, massive computing and virtual stores. While in [35], Le et al. addressed the computation offloading over mmWave for mobile VR in which 360 video streaming is used as a case study. First, the authors mentioned that using 360 video streaming requires more bandwidth and faster user response. In addition, mobile virtual reality (VR) devices locally process video decoding, post-processing and rendering. However, its performance is not enough for streaming high-resolution video, such as 4 K–8 K. Therefore, the authors propose adaptive computations of a discharge scheme using a millimeter-wave (mm-wave) communication. This offloading scheme helps the mobile device to share video decoding tasks on a powerful PC. This improves the mobile device VR ability to play high-definition video. mmWave 802.11ad wireless technology promises the use of broadband wireless to improve the throughput of multimedia systems.

In 5G, network AR/VR applications will be one of the leading applications in the category of Ultra-Reliable and Low-Latency Communication [20,36]. More specifically, in [20], Liu et al. proposed a computation offloading framework for Ultra-Reliable Low Latency communications in which the computation tasks are divided into sub-tasks that will be offloaded and executed at nearby edge server nodes. In addition, the authors formulated optimization problems which jointly minimize the latency and offloading failure probability. Furthermore, three heuristic search-based algorithms are designed to solve this problem and derive the computation offloading decision. However, cloud computing does not consider environment in this work, which can leverage when the edge server resources are not enough. Similarly, in [36] Viitanen et al. described the basic functionality and demo installation for 360 degrees stereo virtual reality (VR) games remote control. They proposed a low latency approach in which the execution of the VR game is offloaded from the end-user device to the edge-cloud server. In addition, the controller feedback is transmitted over the network to the server from which the game visualized types are transmitted to the user in real-time, like encoded HEVC video frames. Finally, this approach proved that energy consumption and the computational load of end terminals are reduced through utilizing the latest advances in network connection speed.

It is observed from the above review of related work, computation offloading has been investigated for different objectives in which most of them address only two-level architecture in MEC systems. Moreover, most of these studies address a single user or multi-user with only a single computation task. This motivates the work of this paper for jointly considering the computation offloading multi-level multi-player multi-task edge-cloud computing systems. Our work aims to minimize the sum cost of the entire system in terms of network latency and energy consumption.

4. System Model

In this section, we introduce our system model which is adopted in this paper. As shown in Figure 2, we consider a set of \mathbb{N} VR game devices in which each device have a set of \mathbb{M} independent computation tasks that needed to be completed. In addition, these devices are connected with a single base station via a wireless channel which is equipped with a mobile edge computing server as well as connected with a centralized cloud computing via core network. We denote the set of VR devices and their computation tasks as $\mathbb{N} = \{1, 2, \dots, N\}$ and $\mathbb{M} = \{1, 2, \dots, M\}$ where the computation tasks can be executed locally on the device itself or will be offloaded and processed remotely on the edge server or cloud server.

In the following subsections, communication and computation models are presented with more detail, followed by the formulation of the optimization problem for our model.

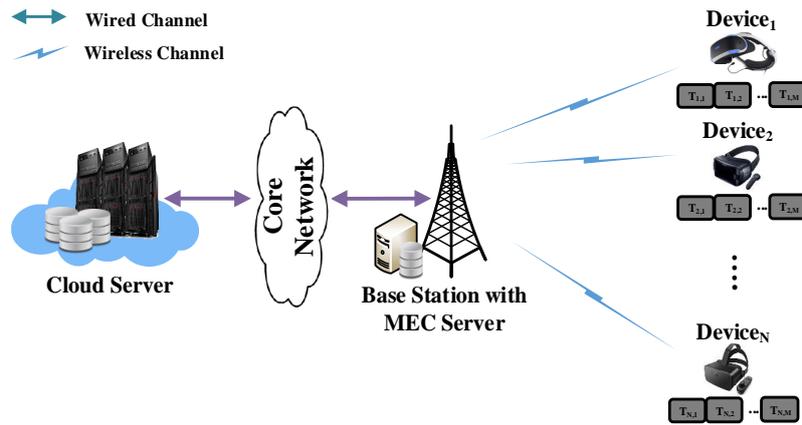


Figure 2. System model.

The notations used in this study are summarized in Table 1.

Table 1. Notations.

Notation	Description
\mathcal{N}	Set of VR devices.
\mathcal{M}	Set of Computation Tasks.
i	Refer to the i^{th} VR device.
j	Refer to the j^{th} computation task.
$a_{i,j}$	Input data size for computation task j of VR device i .
$b_{i,j}$	Output data size for computation task j of VR device i .
$c_{i,j}$	CPU cycles to accomplish the computation task j of VR device i .
$\tau_{i,j}$	Completion deadline for computation task j of VR device i .
$\alpha_{i,j,k}$	Offloading decision of computation task j of VR device i .
r_i^U	Uplink data rate.
r_i^D	Downlink data rate.
B^U	Uplink channel Bandwidth
B^D	Downlink channel Bandwidth
p_i^T	Transmission power of VR device i
p_i^R	Reception power of VR device i
G_0	Channel gain of Base Station
ω_0	Density of noise power of channel
ζ_i	Energy consumed per CPU cycle
ζ	Propagation delay between base station and Cloud
f_i^L	Computational capability of VR device i
f_i^E	CPU resource of Base Station assigned to VR device i
f_i^C	CPU resource of Cloud assigned to VR device i
β	Energy consumption for VR device being idle
$T_{i,j}^L$	Local time for executing task j of VR device i .
$E_{i,j}^L$	Local energy for executing task j of VR device i
$T_{i,j}^{off}$	Offloading time for task j of VR device i .
$T_{i,j}^{down}$	Downloading time for task j of VR device i .
$T_{i,j}^{E,exec}$	Execution time for computation task j of VR device i at Base Station.
$T_{i,j}^{C,exec}$	Execution time for computation task j of VR device i at Cloud server.
$E_{i,j}^E$	Energy consumption for executing task j of VR device i at base station.
$E_{i,j}^C$	Energy consumption for executing task j of VR device i at Cloud server.
$w_{i,j}^t, w_{i,j}^e$	Local computational time and energy weight of computation task j of VR device i .

4.1. Communication Model

We firstly introduce the communication model in edge-cloud system, in which our environment has a single base station connected with a \mathbb{N} set of VR game devices through a wireless channel as well as edge computing resources are associated with a single base station and connected with cloud computing via the core network. In addition, each device runs a VR mobile game application which has \mathbb{M} independent computation tasks that needed to be completed.

Let us denote $\alpha_{i,j,k}$ as a binary computation offloading decision for the computation task j of VR device i which defines the execution place for the computation task. More specifically, $(\alpha_{i,j,0} = 1)$ indicates that the computation task j of VR device i will be executed locally by VR device resource, while $(\alpha_{i,j,1} = 1$ and $\alpha_{i,j,2} = 1)$ indicates that the computation task j of VR device i will be offloaded and processed remotely at the base station and cloud server, respectively. Overall, each computation task j must be executed only one time whether locally ($k = 0$) or remotely ($k \in \{1,2\}$), i.e., $\sum_{k=0}^2 \alpha_{i,j,k} = 1$.

Regarding Shannon law, the maximum uplink and downlink data rate for each VR device where its computation task data are transmitted over the communications channel can be calculated as [4]:

$$r_i^U = B^U \log_2 \left(1 + \frac{p_i^T G_0^2}{\omega_0 B^U} \right) \tag{1}$$

$$r_i^D = B^D \log_2 \left(1 + \frac{p_{bs}^T G_0^2}{\omega_0 B^D} \right) \tag{2}$$

where B^U and B^D denote to the uplink and downlink channel bandwidth, p_i , and p_{bs} denote the transmission power of VR device i and base station, ω_0 and G_0 denote to the density of noise power and the corresponding channel gain between the VR device and the base station due to the path loss and shadowing attenuation.

4.2. Computation Model

In this subsection, the computation offloading model is introduced. Firstly, as mentioned above, our simulation has a a single base station connected with \mathbb{N} set of VR game devices in which each device has \mathbb{M} independent computation tasks that needed to be completed. For each computation task j , we use a tuple $\{a_{i,j}, b_{i,j}, c_{i,j}, \tau_{i,j}\}$ to represent the computation task requirement, where $a_{i,j}$ and $b_{i,j}$ represent the size of input and output data that need to be transmitted and received, respectively. Whereas $c_{i,j}$ and $\tau_{i,j}$ represent the total number of CPU cycles and the completion deadline that are required for task j of VR device i . The values of $a_{i,j}$, $b_{i,j}$ and $c_{i,j}$ can be obtained through carefully profiling of the task execution [37–39].

Consequently, the computation overhead in terms of execution time and energy consumption for local, edge and cloud execution approaches will be discussed later in detail.

4.2.1. Local Execution

For the local execution approach where the computation task j will be executed locally at the VR device itself, the total execution time and energy consumption can be respectively calculated as:

$$T_{i,j}^L = \frac{c_{i,j}}{f_i^L} \tag{3}$$

$$E_{i,j}^L = \zeta_i c_{i,j} \tag{4}$$

where f_i^L denotes the computational capability (CPU cycles per seconds) of VR device, and ζ_i is a coefficient, that denotes the consumed energy per CPU cycle. we set $\zeta_i = 10^{-11} (f_i^L)^2$, where the energy consumption is a superlinear function of VR device frequency [40,41].

4.2.2. Remote Execution

For the remote execution approach where the computation task j of VR device i will be offloaded and processed at the base station or cloud server, the total execution time can be respectively calculated as:

$$T_{i,j}^E = T_{i,j}^{off} + T_{i,j}^{E_exec} + T_{i,j}^{down} \quad (5)$$

$$T_{i,j}^C = T_{i,j}^{off} + \zeta + T_{i,j}^{C_exec} + T_{i,j}^{down} \quad (6)$$

where ζ is constant denoting the propagation delay for transferring the computation task between the base station and cloud server. While $T_{i,j}^{off}$, $T_{i,j}^{down}$, $T_{i,j}^{E_exec}$ and $T_{i,j}^{C_exec}$ denote the offloading, downloading and execution time for processing the computation task j of the VR device i at the base station and cloud server, respectively which can be expressed as follows:

$$T_{i,j}^{off} = \frac{a_{i,j}}{r_i^U} \quad (7)$$

$$T_{i,j}^{down} = \frac{b_{i,j}}{r_i^D} \quad (8)$$

$$T_{i,j}^{E_exec} = \frac{c_{i,j}}{f_i^E} \quad (9)$$

$$T_{i,j}^{C_exec} = \frac{c_{i,j}}{f_i^C} \quad (10)$$

where f_i^E and f_i^C denote the computational capability of the base station and cloud server which is assigned to the VR device i .

Consequently, the energy consumption for offloading, downloading and processing the computation task j of VR device i remotely at the base station and cloud server can be expressed as follows:

$$E_{i,j}^E = p_i^T T_{i,j}^{off} + \beta T_{i,j}^{E_exec} + p_i^R T_{i,j}^{down} \quad (11)$$

$$E_{i,j}^C = p_i^T T_{i,j}^{off} + \beta T_{i,j}^{C_exec} + p_i^R T_{i,j}^{down} \quad (12)$$

where p_i^R denote the reception power of the VR device and β is constant denoting the energy consumption for being idle while processing the task at the edge and cloud.

In view of communication and computation models, the total overhead for executing the computation task j of the VR device i in terms of time and energy can be respectively expressed as:

$$Z_{i,j} = w_{i,j}^t(T_{i,j}) + w_{i,j}^e(E_{i,j}) \quad (13)$$

where w_i^e and $w_i^t \in [0, 1]$ denote the weighting parameters of execution time and energy consumption for VR device i 's decision making, respectively. While $T_{i,j}$ and $E_{i,j}$ are total time and energy which can be expressed as:

$$T_{i,j} = [\alpha_{i,j,0} T_{i,j}^L + \alpha_{i,j,1} T_{i,j}^E + \alpha_{i,j,2} T_{i,j}^C] \quad (14)$$

$$E_{i,j} = [\alpha_{i,j,0} E_{i,j}^L + \alpha_{i,j,1} E_{i,j}^E + \alpha_{i,j,2} E_{i,j}^C] \quad (15)$$

4.3. Problem Formulation

In this section, we consider the issue of achieving efficient computation offloading for multi-player VR edge-cloud computing systems. Regarding the above communication and computation models,

the computation offloading problem is formulated as the following constrained optimization formulation problem:

$$\begin{aligned}
 \min_{x,y} \quad & Z_{i,j} \\
 \text{s.t} \quad & \left[E_{i,j} - E_{i,j}^L \right] \leq 0, \quad \forall_{i,j} \quad \text{C1} \\
 & T_{i,j} \leq \tau_{i,j}, \quad \forall_{i,j} \quad \text{C2} \\
 & \sum_{k=0}^2 \alpha_{i,j,k} = 1, \quad \forall_{i,j} \quad \text{C3} \\
 & \alpha_{i,j,k} \in \{0,1\}, \quad \forall_{i,j} \quad \text{C4}
 \end{aligned} \tag{16}$$

The objective function of the optimization problem is to minimize the sum cost of the entire system in terms of time and energy through the deployment of task offloading. The constraints C1 and C2 are upper bounds of energy and time consumption, respectively. The constraint C3 guarantees that each computation task j must be executed only one time. Finally, the last constraint C4 guarantees that the computation offloading decision variable is binary.

As a result that the objective function is linear as well as all the constraints are also linear, the optimization problem (16) is an integer linear optimization problem in which the optimal solution can be obtained using the branch and bound method [42].

4.4. Multi-Player Computation Offloading Algorithm

In this subsection, we present the design of our multi-player computation offloading algorithm which provides comprehensive processes for deriving the optimal computation offloading decision of the constrained optimization problem in Equation (16) in an efficient manner.

First, all VR devices are initializing their offloading decision $\alpha_{i,j,0} = 1$, that means local execution. Then, each device uploads the computation tasks' requirements which includes $\{a_{i,j}, b_i, j, c_{i,j}, \tau_{i,j}, p_i^T, p_i^R, \zeta_i\}$ and the local computation capabilities f_i^L to the edge server. Afterwards, the edge server calculates the uplink and downlink data rate for each VR device based on the current number of VR players. In addition, the edge server finds the optimal execution place for each computation task (i.e., local, edge or cloud) through solving the optimization problem in Equation (16). Finally, each VR device receives the execution place for their computation tasks from the edge server, thereby minimizing the sum cost of the entire system in terms of time and energy.

Algorithm 1 provides the detailed process of the multi-player computation offloading algorithm in which $\mathcal{O}(NM)$ is the time complexity where N and M denote the total number of VR devices and their tasks, respectively.

4.5. Multi-Level with Multi-Edge Computing Architecture

In this section, we will describe the multi-level, multi-edge, multi-user with multi-task system architecture in which the system architecture composed of three-level as shown in Figure 3. Starting from down to up, the first level consists of a set of \mathbb{N} VR game devices in which each device has a set of \mathbb{M} independent computation tasks that needed to be completed. In the above level, we have a set of \mathbb{K} mobile edge computing servers in which the VR game devices are distributed and connected via wireless channel. In addition, we have a backbone router which can connect and control the mobile edge computing server via wired connection as well as designed using SDN technology. Finally, in the last level, we have a single cloud server which can provide more resources as well as connect with a backbone router through the core network.

Algorithm 1 Multi-Player Computation Offloading Algorithm

- 1: **Initialization:** Each VR device i initializes the offloading decision for their computation tasks with $\alpha_{i,j,0} = 1, \forall i, j$
 - 2: **for all** each VR device i and at given time slot t **do**
 - 3: **for all** each computation task j **do**
 - 4: **Uploads** the computation tasks' requirements $\{a_{i,j}, b_{i,j}, c_{i,j}, \tau_{i,j}, p_i^T, p_i^R, \zeta_i\}$ and the local computation capabilities f_i^L to the edge server.
 - 5: **Calculate** the uplink and downlink data rate r_i^U, r_i^D for each VR device based on Equations (1) and (2).
 - 6: **Solve** the optimization problem in Equation (16) and obtain the optimal computation offloading decision values $\alpha_{i,j,k}$ for each computation task at VR devices in which the sum cost of the entire system is minimized.
 - 7: **Send** the offloading decision values $\alpha_{i,j,k}$ to each VR device.
 - 8: **end for**
 - 9: **end for**
-

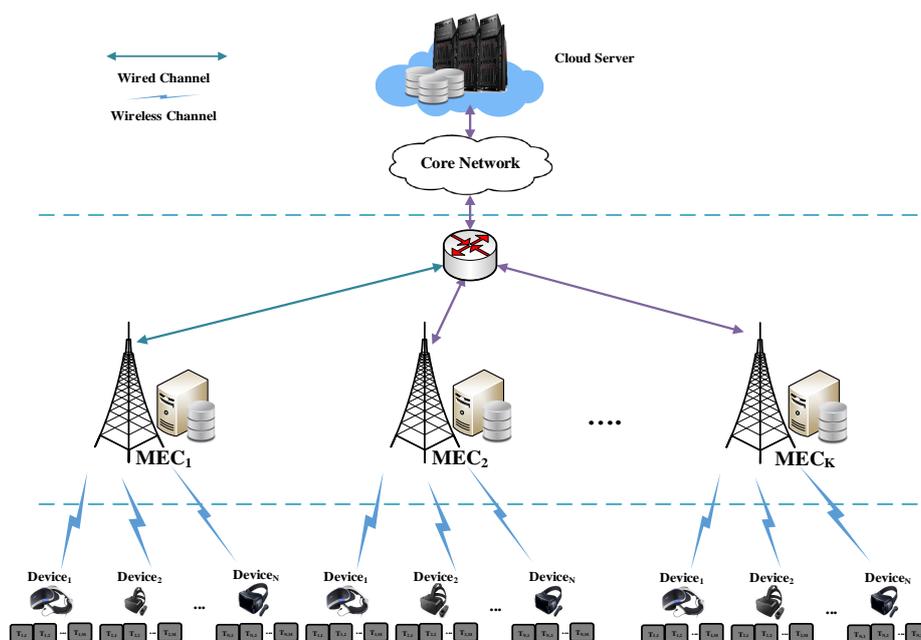


Figure 3. Multi-level, multi-edge, multi-user, with multi-task architecture.

Regarding the simulation and results for this architecture, there are some issues that should be handled for the computation offloading which are:

- **Scalability Issue:** In the multi-edge environment, the number on VR game devices becomes large, which has more computation tasks that need to be offloaded and executed remotely. Simultaneously, the mobile edge computing servers should provide with the resources' scalability. Thus, an intelligent algorithm should be designed to scale the computation resources of the overloaded edge server using the computation resources of the neighbor edge server and cloud server which are underloaded.
- **Load Balancing Issue:** As mentioned above, the VR game devices are distributed across the edge computing servers. In addition, due to the randomization distribution of the users, some of the edge servers will be overloaded whereas other servers will be underloaded. Consequently,

this will affect the computation offloading process and may lead to poor service quality and long delays due to network congestion. Therefore, it is important to propose an efficient algorithm to balance the load between the edge server and improve the quality of service for the VR game device user.

- **Mobility Issue:** In the multi-edge environment, each VR game device has the ability to depart and leave dynamically between the edge servers within a computation offloading period which will be interesting and technically challenging, where the offloading decision and the execution location will be affected. Thus, an intelligent approach should be developed to determine the best execution place for the computation task in which the overall consumption in terms of time and energy will be minimized.
- **Possible Output:** Finally, if the scalability, load balancing and mobility issues are handled as mentioned above, the proposed model can operate and derive the computation offloading decision in an efficient manner. Further, the weighted sum cost of the entire system in terms of energy and time will be optimized.

5. Simulation Results and Prototype Implementation

5.1. Simulation Results and Discussion

In our simulation settings, a Python-based simulator is used, in which the computer is equipped with Intel[®] Core(TM) i7-4770 CPU with 3.4 GHz frequency and 8 GB RAM capacity running Windows 10 Professional 64-bit platform. We consider a multi-player multi-task edge-cloud computing system with single cloud server, single small base station, $N = 20$ VR devices and each device has $M = 5$ independent computation tasks that can be executed locally or offloaded and processed remotely at the available base station or cloud server. Each computation task has an input data size which is uniformly distributed within the range (5, 15) MB while the output data size is assumed to be 20% of the input data size. In addition, the total number of CPU cycles required to complete each task is assigned to 1500 cycles/bit. The CPU computational capability of each VR device is uniformly distributed within range $\{0.5, 0.6, \dots, 1.0\}$ GHz, while the CPU computational capability of edge server and cloud are set to 20 and 50 GHz, respectively. The local computing energy consumption per cycle follows a uniform distribution in the range (0.20×10^{-11}) J/cycle. The other simulation settings employed in the simulations are summarized in Table 2.

Table 2. Simulation parameters.

Parameter	Value
Number of VR Devices N	20
Number of Computation tasks M	5
Uplink Bandwidth B^U	20 MHz
Downlink Bandwidth B^D	40 MHz
Transmission and Reception Power p_i^T and p_i^R	100 mWatts
Background Noise ω_0	-100 dBm
Input Data Size $a_{i,j}$	(5, 15) MB Unif-Dist
Output Data Size $b_{i,j}$	$0.2a_{i,j}$
Number of CPU cycles per bit $c_{i,j}$	1500 Cycles/bit
Computation Capability of each VR device f_i^L	$\{0.5, 0.6, \dots, 1.0\}$ GHz
Energy Consumption per CPU Cycle ζ_i	(0.20×10^{-11}) J/cycle
Weights $w_{i,j}^t, w_{i,j}^e$	0.5
Computation Capability of Edge Server F	20 GHz
Computation Capability of Cloud Server F	50 GHz
Cloud propagation delay	15 ms

Figure 4 shows the energy consumption of executing the computation tasks for the three different scenarios versus different values of edge server computation capabilities. It is observed from the figure that the energy consumption for our proposed system can achieve the lowest result. Specifically, the edge server scenario is decreasing as the edge server computation capability is increasing and becoming lesser than the cloud execution scenario. This is due to the energy consumption becoming shorter as the VR device is allocated more resources, whereas the cloud execution and local execution scenarios do not affect because they do not depend on the edge server resources. In addition, our proposed system gets the best execution place (i.e., local, edge and cloud).

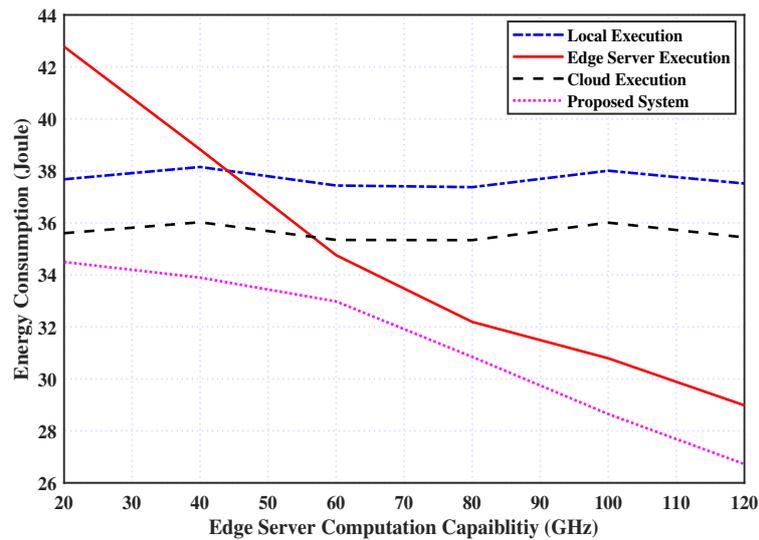


Figure 4. Energy consumption comparison under different values of edge server capability.

Figure 5 presents the processing time of executing the computation tasks for different edge server’s capability. It is seen in this figure that the cloud execution and local execution policies are not affected by the edge server capability, whereas the time execution of our proposed system and edge server policies gradually decreases as edge server capability increasing. This is because of the shorter latency as the VR devices are allocated more resources.

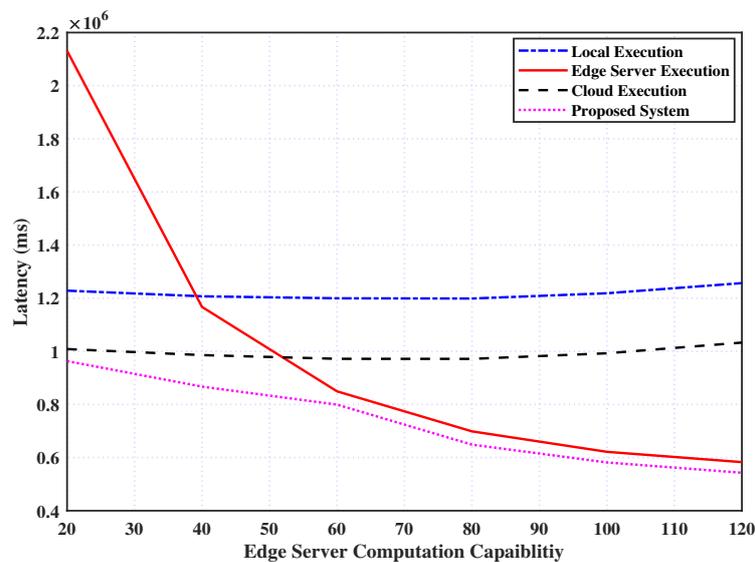


Figure 5. Latency comparison under different values of edge server capability.

The processing time and energy consumption of executing the computation tasks over different values of input data size (Input data size is uniformly distributed within the range (0,i) MB where i is the value of x-axis) are shown in Figures 6 and 7, respectively. It can be deduced from the figures that the cost wise (time and energy) of our proposed system can achieve better performance and able to maintain a lower overhead in comparison with the other policies. In addition, the edge execution policy exceeds the local execution as the number of data size increases (i.e., data size > 8 MB). This is due to our proposed model can select some appropriate tasks to be executed remotely (i.e., at edge or cloud server) while rejecting others in an optimal way, which minimizes the sum cost of the entire system.

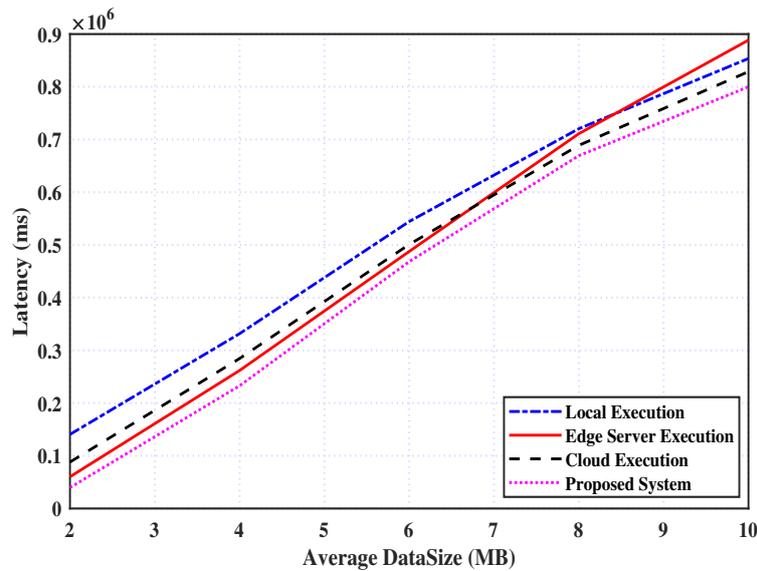


Figure 6. Latency comparison under different values of input data size.

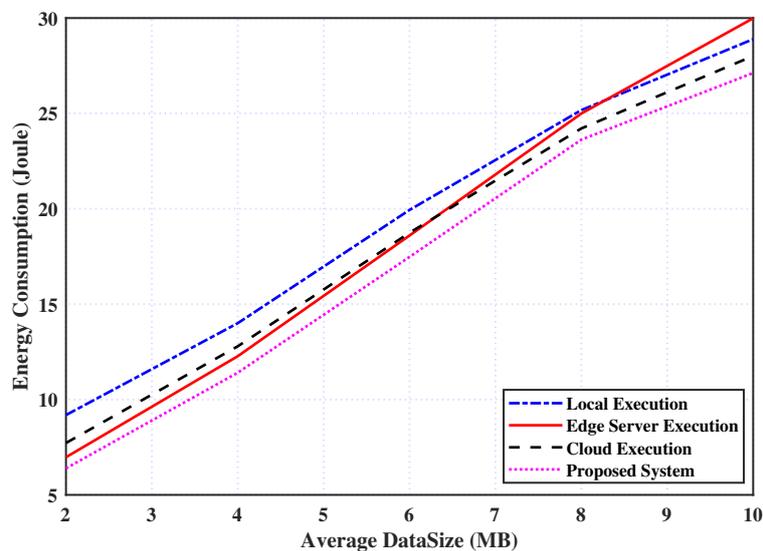


Figure 7. Latency comparison under different values of input data size.

5.2. Prototype Implementation and Measurements

Network segment includes hardware and software such as: NI USRP boards, which provide the ability to efficiently study, analyze/emulate LTE/LTE-A networks, 5G New Radio and other wireless technologies; GNU Radio, Amarisoft, srsLTE software packages provide an opportunity to study/test network protocols, signaling technologies and access to the radio channel. This experiment

is conducted with the OpenAirInterface software package for virtualization of mobile communication components. Additionally deployed is a virtual environment in which vEPC, Amarisoft are installed in the form of containers and virtual machines for convenient infrastructure management, which makes it possible to emulate virtualized components of a wireless radio access network (HSS, vSGW, vPGW, vMME).

The developed prototype includes transport network, logical transport network that is the link between the network core and the radio access network (RAN). It is based on SDN technology and has the ability to flexibly and quickly manage all nodes. Using API, it can automatically change the configuration depending on the requirements, the second part is the core network, which consists of a virtualized segment. In this part, OpenAirInterFace, srsLTE, AMAISOFRT, openEPC can be used. Each element of this zone can be represented as a docker container and a virtual machine, and the last part is radio access network—an area consisting of NI-USRP 2954R software-defined radio systems (SDR), on which an LTE access network, New Radio sub6, LoRa, NB-IoT, etc. can be deployed. For the organization of radio, interfaces used are OpenAirInterface, GNU Radio, srsENB solutions. A deployed 5G NSA network model includes a set of docker containers in which network elements are packed, and SDR hardware modules. A Docker image is a service with its required dependencies and libraries that are directly related to running the application. For example, for an HSS element docker, the container will include hss.h, security.h, etc. Each container is installed either in separate VMs or multiple containers in one VM. It is important to note that the N26 interface is a key interface between MME (EPC) and AMF as a signaling exchange point for UE radio control. Therefore, these two elements must be at a sufficient distance to satisfy synchronization tasks. In our case, for network hardware devices we used the software-defined radio (SDR.). The program control has a wide range of tools that are used in our laboratory, for example, real-time or offline/post-processing, C++ and the USRP Hardware Driver (UHD) API (open-source), GNU Radio (Python, NumPy, SciPy, Matplotlib, etc.) (open-source), LabVIEWTM (National Instruments), MATLABTM (The MathWorks), Application-Specific (Cellular: OpenBTS, OpenBTS-UMTS, OpenLTE, srsLTE, Amarisoft).

As shown in Figure 8, to realize the proposed system, we used in our prototype the following devices:

- Edge hosting:
 - CPU Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz,
 - RAM 64GB,
 - GPU [GeForce GT 1030] per node.
- Network node:
 - LaNNER x86,
 - CPU Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz,
 - RAM 32GB,
 - Interface 10xSFP+,
 - Virtual interface VLAN,QinQ.

To create a cluster rendering of VR applications, we used several EDGE hosts with an interface between 10GBE hosts. Using a medium-quality graphics card, the GeForce GT 1030 was used as the rendering core.

Using a cluster with a different number of nodes, we measured the average energy consumption of a node per Mbps of information transfer. Then, after rendering, the image is broadcast to the end device. Such a construction of the architecture will allow us to study the average content delivery delay depending on the number of EDGE hosts in the cluster.

During experiments, the following results were obtained from edge computing clusters:

- Average power consumption of one edge computing per one Mbps of information transfer.

- The average delay between the end device—the VR application server, depending on the number of hosts in the rendering cluster.



Figure 8. Architecture of the prototype system.

In general, the use of edge computing at base stations or near network nodes increases power consumption. The more users accessing this unit increases at times (Figure 9). However, the price of electricity annually falls, and at the same time, the power of elemental bases increases.

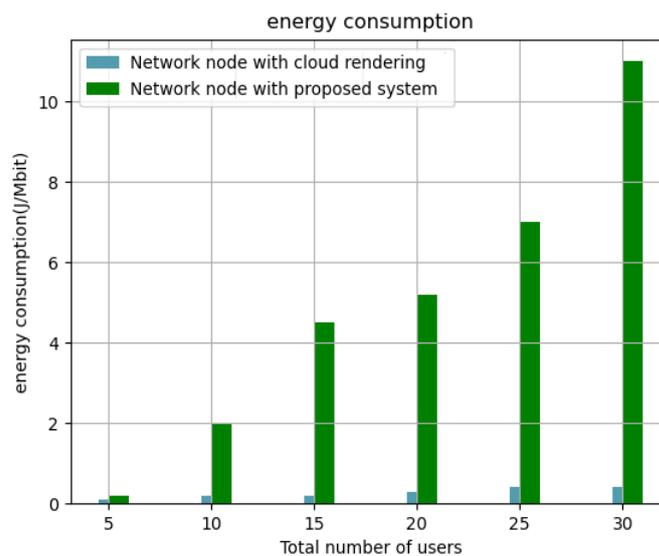


Figure 9. Total number of users.

To improve the QoS of the VR application, we have to solve the problem with energy consumption. In Figure 10, the dependence of the response delay on the number of devices in the VR rendering cluster

was shown. In case of the larger cluster, the VR application will be generated faster. This happens due to computing parallelism.

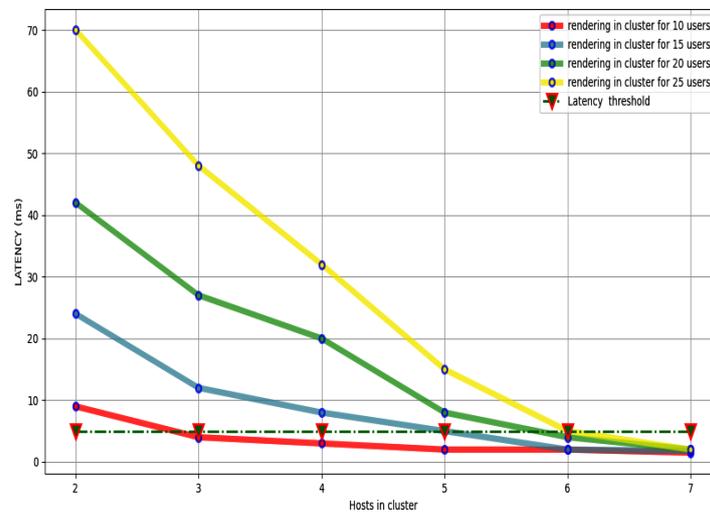


Figure 10. Host in cluster.

6. Conclusions

In this study, we proposed an efficient multi-player computation offloading approach for VR edge-cloud computing systems. Firstly, the computation offloading is formulated as an integer optimization problem whose objective is to minimize the weighted cost of the entire system in terms of time and energy. This model is considered as latency and energy-aware in which it can select the execution place for the VR computing tasks in away to achieve the best energy and latency efficiency. The proposed system is integrated to achieve the best VR user experience. In addition, a low complexity multi-player computation offloading algorithm is designed to derive the optimal computation offloading decision. Finally, this system is simulated over a reliable environment for various simulation scenarios. Finally, simulations have been conducted to validate our proposed model and prove that the network latency and energy consumption can be reduced by up to 26.2%, 27.2% and 10.9%, 12.2% in comparison with edge and cloud execution, respectively.

In ongoing and future work, a new effective compression layer will be introduced in which the offloading data will be compressed in the low bandwidth state using an efficient algorithm. Hence, the communication time and energy will be reduced and the performance of the entire system will be enhanced. In addition, a more general case will be considered where there are multi-edge server and the mobility issue for the mobile device will be handle in which each mobile device may depart and leave dynamically within a computation offloading period, which will be interesting and technically challenging.

Author Contributions: Conceptualization, A.A. (Abdullah Alshahrani) and A.M.; Formal analysis, A.A. (Adel Alshamrani); investigation, I.A.E.; methodology, I.A.E.; software, A.M.A.; supervision, A.M.; validation, A.M. and A.A. (Abdullah Alshahrani); writing—original draft, I.A.E. and A.M.; writing—review and editing, A.M.A. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Deanship of Scientific Research (DSR), University of Jeddah, Jeddah, under grant No. (UJ-02-037-DR). The authors, therefore, acknowledge with thanks DSR technical and financial support.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Jiang, D.; Liu, G. An overview of 5G requirements. In *5G Mobile Communications*; Springer: Berlin, Germany, 2017; pp. 3–26.
2. Ateya, A.A.; Muthanna, A.; Makolkina, M.; Koucheryavy, A. Study of 5G services standardization: Specifications and requirements. In Proceedings of the 2018 10th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), Moscow, Russia, 5–9 November 2018; pp. 1–6.
3. Vega, M.T.; Mehmlı, T.; van der Hooft, J.; Wauters, T.; De Turck, F. Enabling Virtual Reality for the Tactile Internet: Hurdles and Opportunities. In Proceedings of the 2018 14th International Conference on Network and Service Management (CNSM), Rome, Italy, 5–9 November 2018; pp. 378–383.
4. Jetter, J.; Eimecke, J.; Rese, A. Augmented reality tools for industrial applications: What are potential key performance indicators and who benefits? *Comput. Hum. Behav.* **2018**, *87*, 18–33. [[CrossRef](#)]
5. Qiao, X.; Ren, P.; Dustdar, S.; Liu, L.; Ma, H.; Chen, J. Web AR: A Promising Future for Mobile Augmented Reality—State of the Art, Challenges, and Insights. *Proc. IEEE* **2019**, *107*, 651–666. [[CrossRef](#)]
6. Wang, X.; Ong, S.K.; Nee, A.Y. A comprehensive survey of augmented reality assembly research. *Adv. Manuf.* **2016**, *4*, 1–22. [[CrossRef](#)]
7. Fraga-Lamas, P.; Fernández-Caramés, T.M.; Blanco-Novoa, Ó.; Vilar-Montesinos, M.A. A review on industrial augmented reality systems for the industry 4.0 shipyard. *IEEE Access* **2018**, *6*, 13358–13375. [[CrossRef](#)]
8. Dilek, U.; Erol, M. Detecting position using ARKit. *Phys. Educ.* **2018**, *53*, 025011. [[CrossRef](#)]
9. International Telecommunication Union (ITU). *IMT Vision—Framework and Overall Objectives of the Future Development of IMT for 2020 and beyond*; ITU, Radiocommunication Study Groups: Geneva, Switzerland, 2015.
10. 3GPP. *Study on Scenarios and Requirements for Next Generation Access Technologies*; 3GPP TR 38.913; ESTI: Sophia Antipolis, France, 2016.
11. Khayyat, M.; Alshahrani, A.; Alharbi, S.; Elgendy, I.; Paramonov, A.; Koucheryavy, A. Multilevel Service-Provisioning-Based Autonomous Vehicle Applications. *Sustainability* **2020**, *12*, 2497. [[CrossRef](#)]
12. Zhang, W.; Han, B.; Hui, P. On the networking challenges of mobile augmented reality. In Proceedings of the Workshop on Virtual Reality and Augmented Reality Network, Los Angeles, CA, USA, 21–25 August 2017; pp. 24–29.
13. Bastug, E.; Bennis, M.; Médard, M.; Debbah, M. Toward interconnected virtual reality: Opportunities, challenges, and enablers. *IEEE Commun. Mag.* **2017**, *55*, 110–117. [[CrossRef](#)]
14. Liu, Y.; Ling, J.; Shou, G.; Seah, H.S.; Hu, Y. Augmented reality based on the integration of mobile edge computing and fiber-wireless access networks. In Proceedings of the International Workshop on Advanced Image Technology (IWAIT), Singapore, 6–9 January 2019; International Society for Optics and Photonics: Bellingham, WA, USA, 2019; Volume 11049, p. 110490M.
15. Erol-Kantarci, M.; Sukhmani, S. Caching and computing at the edge for mobile augmented reality and virtual reality (AR/VR) in 5G. In *Ad Hoc Networks*; Springer: Berlin, Germany, 2018; pp. 169–177.
16. Khayyat, M.; Elgendy, I.A.; Muthanna, A.; Alshahrani, A.; Alharbi, S.; Koucheryavy, A. Advanced Deep Learning-based Computational Offloading for Multilevel Vehicular Edge-Cloud Computing Networks. *IEEE Access* **2020**; doi: [10.1109/ACCESS.2020.3011705](#). [[CrossRef](#)]
17. Hu, Y.C.; Patel, M.; Sabella, D.; Sprecher, N.; Young, V. Mobile edge computing—A key technology towards 5G. *ETSI White Pap.* **2015**, *11*, 1–16.
18. Mach, P.; Becvar, Z. Mobile edge computing: A survey on architecture and computation offloading. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 1628–1656. [[CrossRef](#)]
19. Mao, Y.; You, C.; Zhang, J.; Huang, K.; Letaief, K.B. A survey on mobile edge computing: The communication perspective. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 2322–2358. [[CrossRef](#)]
20. Liu, J.; Zhang, Q. Offloading schemes in mobile edge computing for ultra-reliable low latency communications. *IEEE Access* **2018**, *6*, 12825–12837. [[CrossRef](#)]
21. Yang, X.; Chen, Z.; Li, K.; Sun, Y.; Liu, N.; Xie, W.; Zhao, Y. Communication-constrained mobile edge computing systems for wireless virtual reality: Scheduling and tradeoff. *IEEE Access* **2018**, *6*, 16665–16677.
22. Huynh, L.N.; Pham, Q.V.; Pham, X.Q.; Nguyen, T.D.; Hossain, M.D.; Huh, E.N. Efficient Computation Offloading in Multi-Tier Multi-Access Edge Computing Systems: A Particle Swarm Optimization Approach. *Appl. Sci.* **2020**, *10*, 203. [[CrossRef](#)]

23. Keil, J.; Edler, D.; Dickmann, F. Preparing the HoloLens for user studies: an augmented reality interface for the spatial adjustment of holographic objects in 3D indoor environments. *KN-J. Cartogr. Geogr. Inf.* **2019**, *69*, 205–215. [CrossRef]
24. The Future of Reality—Augmented, Virtual or Mixed? Available online: <https://medium.com/@Unfoldlabs/the-future-of-reality-augmentedvirtual-or-mixed-8687cc25ee48> (accessed on 6 June 2020).
25. Çöltekin, A.; Oprean, D.; Wallgrün, J.O.; Klippel, A. *Where Are We Now? Re-Visiting the Digital Earth through Human-Centered Virtual and Augmented Reality Geovisualization Environments*; Taylor & Francis: Abingdon, UK, 2019; doi:10.1080/17538947.2018.1560986. [CrossRef]
26. Clemm, A.; Vega, M.T.; Ravuri, H.K.; Wauters, T.; De Turck, F. Toward truly immersive holographic-type communication: Challenges and solutions. *IEEE Commun. Mag.* **2020**, *58*, 93–99. [CrossRef]
27. DiVerdi, S.; Olwal, A.; Rakkolainen, I.; Höllerer, T. An immaterial pseudo-3D display with 3D Interaction. In *Three-Dimensional Television*; Springer: Berlin, Germany, 2008; pp. 505–528.
28. Mixed Reality with HoloLens. Available online: <https://www.microsoft.com/en-us/hololens/why-hololens> (accessed on 6 June 2020).
29. Liu, Y.; Dong, H.; Zhang, L.; El Saddik, A. Technical evaluation of HoloLens for multimedia: A first look. *IEEE MultiMedia* **2018**, *25*, 8–18. [CrossRef]
30. Colman-Meixner, C.; Diogo, P.; Siddiqui, M.S.; Albanese, A.; Khalili, H.; Mavromatis, A.; Luca, L.; Ulisses, A.; Colom, J.; Nejabati, R.; et al. 5G city: A novel 5G-enabled architecture for ultra-high definition and immersive media on city infrastructure. In Proceedings of the 2018 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), Valencia, Spain, 6–8 June 2018; pp. 1–5.
31. Elbamby, M.S.; Perfecto, C.; Bennis, M.; Doppler, K. Edge computing meets millimeter-wave enabled VR: Paving the way to cutting the cord. In Proceedings of the 2018 IEEE Wireless Communications and Networking Conference (WCNC), Barcelona, Spain, 15–18 April 2018; pp. 1–6.
32. Hou, X.; Lu, Y.; Dey, S. Wireless VR/AR with edge/cloud computing. In Proceedings of the 2017 26th International Conference on Computer Communication and Networks (ICCCN), Vancouver, BC, Canada, 31 July–3 August 2017; pp. 1–8.
33. Schmoll, R.S.; Pandi, S.; Braun, P.J.; Fitzek, F.H. Demonstration of VR/AR offloading to mobile edge cloud for low latency 5G gaming application. In Proceedings of the 2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 12–15 January 2018; pp. 1–3.
34. Hsiao, T.C.; Tai, K.Y.; Huang, Y.M.; Chung, Y.F.; Wu, Y.C.; Kurniati, T.; Chen, T.S. An Implementation of Efficient Hierarchical Access Control Method for VR/AR Platform. In Proceedings of the 2018 16th International Conference on Emerging eLearning Technologies and Applications (ICETA), Stary Smokovec, Slovakia, 15–16 November 2018; pp. 205–208.
35. Le, T.T.; Van Nguyen, D.; Ryu, E.S. Computing offloading over mmWave for mobile VR: Make 360 video streaming alive. *IEEE Access* **2018**, *6*, 66576–66589. [CrossRef]
36. Viitanen, M.; Vanne, J.; Hämmäläinen, T.D.; Kulmala, A. Low latency edge rendering scheme for interactive 360 degree virtual reality gaming. In Proceedings of the 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS), Vienna, Austria, 2–6 July 2018; pp. 1557–1560.
37. Elgandy, I.A.; Zhang, W.; Tian, Y.C.; Li, K. Resource allocation and computation offloading with data security for mobile edge computing. *Future Gener. Comput. Syst.* **2019**, *100*, 531–541. [CrossRef]
38. Lyu, X.; Tian, H. Adaptive receding horizon offloading strategy under dynamic environment. *IEEE Commun. Lett.* **2016**, *20*, 878–881. [CrossRef]
39. Elgandy, I.A.; El-kawkagy, M.; Keshk, A. An efficient framework to improve the performance of mobile applications. *Int. J. Digit. Content Technol. Appl. (JDCTA)* **2015**, *9*, 43–54.
40. Chen, X. Decentralized Computation Offloading Game for Mobile Cloud Computing. *IEEE Trans. Parallel Distrib. Syst.* **2015**, *26*, 974–983. doi:10.1109/TPDS.2014.2316834. [CrossRef]
41. Elgandy, I.; Zhang, W.; Liu, C.; Hsu, C.H. An efficient and secured framework for mobile cloud computing. *IEEE Trans. Cloud Comput.* **2018**. [CrossRef]
42. Stidsen, T.; Andersen, K.A.; Dammann, B. A branch and bound algorithm for a class of biobjective mixed integer programs. *Manag. Sci.* **2014**, *60*, 1009–1032. [CrossRef]

