Hindawi
Journal of Advanced Transportation
Volume 2018, Article ID 6235379, 12 pages
https://doi.org/10.1155/2018/6235379

WILEY | Hindawi

Research Article

Priority-Based Cloud Computing Architecture for Multimedia-Enabled Heterogeneous Vehicular Users

Amjad Ali,[1,2] **Hongwu Liu**,[1,3] **Ali Kashif Bashir**,[4] **Shaker El-Sappagh**,[1,5] **Farman Ali**,[1] **Adeel Baig**,[6,7] **Daeyoung Park**,[1] **and Kyung Sup Kwak**[1]

[1] *UWB Wireless Communications Research Center, Department of Information & Communication Engineering, Inha University, Incheon 402-751, Republic of Korea*
[2] *Department of Computer Science, COMSATS University Islamabad, Lahore Campus, Lahore, Pakistan*
[3] *Shandong Jiaotong University, Jinan 250357, China*
[4] *Faculty of Science and Technology, University of the Faroe Islands, Faroe Islands, Denmark*
[5] *Information Systems Department, Faculty of Computer and Informatics, Benha University, Benha 13518, Egypt*
[6] *College of Engineering and Architecture (CoEA), Al Yamamah University, Riyadh, Saudi Arabia*
[7] *National University of Science and Technology (NUST), Islamabad, Pakistan*

Correspondence should be addressed to Kyung Sup Kwak; kskwak@inha.ac.kr

Received 12 July 2018; Accepted 12 September 2018; Published 27 September 2018

Guest Editor: Tariq Umer

In recent days, vehicles have been equipped with smart devices that offer various multimedia-related applications and services, such as smart driving assistance, traffic congestions, weather forecasting, road safety alarms, and many entertainment and comfort applications. Thus, these smart vehicles produce a large amount of multimedia-related data that require fast and real-time processing. However, due to constrained computing and storage capacities, such huge amounts of multimedia-related data cannot be processed in on-board standalone devices. Thus, multimedia cloud computing (MCC) has emerged as an economical and scalable computing technology that can process multimedia-related data efficiently while providing improved Quality of Service (QoS) to vehicular users from anywhere, at any time and on any device, at reduced costs. However, there are certain challenges, such as fast service response time and resource cost optimization, that can severely affect the performance of the MCC. Therefore, to tackle these issues, in this paper, we propose a dynamic priority-based architecture for the MCC. In the proposed scheme, we divide multimedia processing into four different subphases, while computing resources to each computing server are assigned dynamically, according to the workload, in order to process multimedia tasks according to the multimedia user Quality of Experience (QoE) requirements. The performance of the proposed scheme is evaluated in terms of service response time and resource cost optimization using the CloudSim simulator.

1. Introduction

Recent advancements in wireless access technology and the automobile industry have made Internet access a basic need for vehicles moving on roads. These days, vehicles are equipped with several sensors, cameras, and other smart devices, and passengers enjoy a range of applications for comfort, safety, entertainment, and commercial services. In a Vehicular Ad hoc Network (VANET), vehicles communicate with each other using roadside infrastructure to share information (e.g., road map images, road safety, and traffic congestion information for safe driving). Vehicles also exchange a lot of other information, such as security alerts, distance, and collision warnings, cooperative cruise control and driving, driver assistance, global positioning syste locations, automatic parking assistance, Internet access, and dissemination of road information [1, 2]. Moreover, these days, the automobile industry and academia all around the globe are focusing on driverless or autonomous vehicles and such smart vehicles record videos and take high-resolution images, and sensors record other data for successful and smooth driving [3]. Thus, overall, vehicles are producing a huge amount

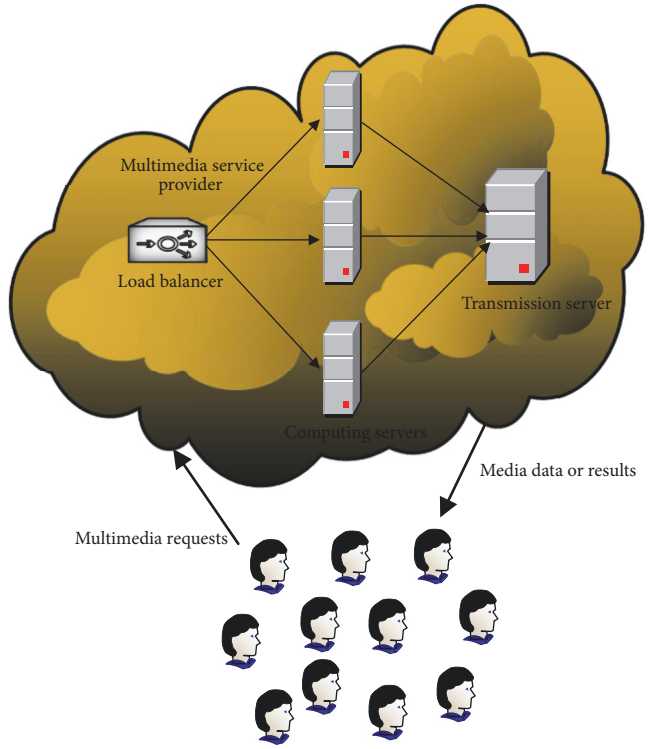

Figure 1: General architecture of multimedia cloud computing.

of critical information and delay-sensitive multimedia data for processing, such as surveillance videos, traffic-congestion images, and other entertainment data. However, due to limited computational and storage capacities, standalone on-board devices have insufficient processing power for such a huge amount of data to ensure on-time delivery of required contents to maintain the desired or guaranteed Quality of Experience (QoE) to the end user. Moreover, vehicles have some other unique features (i.e., lack of bandwidth, short-range radio communications, high mobility, and intermittent connectivity), which make multimedia content distribution in VANETs more challenging.

The emergence of cloud computing brings a revolution in the computing industry because of its unique properties, such as elimination of hardware installation at the user's end and provision of various computing functionalities as a service to end users [4–6]. Cloud computing is more attractive for multimedia-related processing because of the intensive computation and storage requirements that are not offered on standalone devices [7–10]. However, the conventional cloud computing architecture is not well suited to multimedia services and applications because of the time-critical and delay-sensitive computational nature of the applications. Moreover, heterogeneity in end users makes it even more challenging. Therefore, a novel computing paradigm known as multimedia cloud computing (MCC) is emerging to deal with such time-critical and delay-sensitive multimedia applications and services [11].

The MCC has installed high-speed Graphic Processing Units (GPUs), CPUs, and storage units to deal with multimedia-related processing, as shown in Figure 1. Thus, the one big advantage of MCC is that, while exploiting the MCC infrastructure for multimedia-related computation, multimedia end users do not need to pay for costly computing and storage resources, and instead, they only need to pay for the time for which they use the MCC resources. However, in the MCC, on-time processing of multimedia applications and services-related tasks and forwarding them to the intended end user while maintaining a guaranteed QoE at end users is a challenging issue that can severely affect the performance of delay-sensitive and time-critical multimedia processing. Moreover, another critical and challenging issue for cloud service providers is how they allocate computing resources in order to minimize the computing resources' costs while providing the desired QoE. For example, in a vehicular-based MCC scenario, the MCC may need to process and propagate the desired information about an accident that happened on the highway due to bad weather conditions (e.g., fog) and if such information is not processed and disseminated to other oncoming vehicles in time, then more accidents can happen, with the possible loss of more lives and assets. Thus, the *service response time* (that is, a measure of the total time, or latency, of a multimedia task starting when MCC receives the multimedia request at the scheduler until transmitting the processed results back to the intended end user) is a significantly important factor for measuring the QoE performance of MCC from the multimedia end users' perspective [7]. Table 1 provides the acronyms and symbols used in this paper.

Table 1: List of acronyms and symbols.

Abbreviation	Name
MCC	Multimedia Cloud Computing
QoE	Quality of Experience
LBM	Load Balancer Module
CC	Computing Cluster
MC	Multimedia Cloud
DC	Dynamic Conversion
DE	Dynamic Extraction
DM	Dynamic Matching
DR	Dynamic Reconstruction
TM	Transmission Module
SS	Scheduling Server
C	Total computational resources
μ	Scheduling rate
λ	Multimedia request arrival rate
N	Number of priority queues
R_i	Average result size
ϵ	Upper bound of service time
ξ	Transmission rate
F_i	Multimedia task of priority-class i

To overcome the aforementioned issues, in this paper, we propose a priority-based dynamic resource allocation scheme for MCC where the prime objective is to ensure efficient processing and on-time fulfillment of multimedia-related tasks while minimizing the computing costs. Thus, in our proposed scheme, we divided the MCC architecture into four special-purpose and dedicated computing clusters. Similarly, we divided multimedia task processing into four different subphases. The computing resources for each computing cluster are assigned based on workload estimation information. Moreover, resource allocation is periodically updated (e.g., additions and removals) to minimize the computing resource costs and to guarantee QoE for end user. In our proposed scheme, multimedia processing requests are first filed in a *request queue* that is part of Load Balancer Module (LBM). The *LBM* analyzes the nature of the received multimedia requests and assigns them to an appropriate computing cluster that processes the assigned tasks and sends them to the next computing cluster, or to a *transmission unit* based on the nature of the processing required. Finally, the transmission unit forwards the processed multimedia request to the intended end user. The proposed scheme further handles the priority or urgency of multimedia users with the help of *job queues*. In the proposed scheme, each computing cluster (including the transmission unit) contains N job or transmission queues.

The rest of the paper is organized as follows: Section 2 discusses the state of the art related to multimedia-related processing. Section 3 presents our proposed priority-based dynamic resource allocation scheme for the multimedia cloud architecture. Section 4 deals with performance analysis. Finally, Section 5 concludes the paper.

2. Related Work

This section summarizes the state of the art related to multimedia cloud computing, particularly in the context of delay-sensitive, time-critical, and QoE demands in multimedia data processing.

Liu et al. proposed a relay-selective multihop scheme, in which a vehicular network is integrated with cloud computing to provide media services, e.g., weather forecasting, traffic congestion reports, and road safety alarms [12]. In this scheme, Road Side Units (RSUs) communicate through the cloud via a road-side fixed communications sources. The proposed scheme is efficient at making transportation decisions in severe weather and under controlled traffic problems. Similarly, in [13], Joshi et al. discussed vehicular challenges, such as road safety messages, navigational data, and different conditions of vehicular mobility, and then, they proposed an integrated framework for a VANET and cloud computing services, known as "*Traffic Management as a Service (TMaaS)*." The TMaaS provides different real-time applications (e.g., tracking a vehicle's location, accident alarms, and traffic situation reports). Moreover, TMaaS also support vehicles' intracommunications mechanisms to forward, store, and process data in the centralized cloud for making decisions. Moreover, to tackle the processing of huge amounts of vehicular data, Gong et al. proposed a mobile content distribution scheme with the help of roadside parked vehicles [14]. This scheme mainly consists of two types of cloud: (i) mobile cloud and (ii) Roadside Parking Cloud (RPC), where an RPC is formed by a group of parked vehicles on the roadside and a mobile cloud is formed by a group of cooperative and moving vehicles having the same type of characteristics. Thus, in this scheme, moving vehicles get connected with the RPC and use their requested services, but if the vehicle is moving out of range from one RPC, then the first RPC forwards that vehicle's request to the next neighboring RPC.

Meneguette et al. in [15] addressed the benefits of an Intelligent Transportation System (ITS) in which vehicles provide streamlined services for managing traffic loads, and different automobile operations and to help stakeholders by providing various other useful information regarding safety. With these advantages in mind, a new vehicular cloud architecture was proposed to help manage an ITS in big city areas. Moreover, a framework was proposed that provides services that include information access and storage management to citizens, commercial vehicles, and emergency services (i.e., ambulances). It also provides the integration of VANETs with other networks for efficient monitoring and controlling of the ITS. With the evolution of VANETs, transportation systems are improving, with different enhanced features of safety, security, navigation, and road condition reports. But due to different research challenges regarding security and privacy, VANETs do not fulfill users' expectations. Thus, to address these challenges, a communications paradigm stack for VANETs based on the cloud was proposed [16]. This framework divides VANETs into three categories: (1) a vehicular cloud, (2) vehicles using the cloud, and (3) a hybrid vehicular cloud. In this scheme,

the cloud provides information to moving vehicles, and consequently, the moving vehicles provide information about roads to the cloud. The result of this mutual communication is that the cloud provides all current and near-future locations for efficient decision-making. These days, the advancements in vehicular communications have become the motivation for forming a vehicular cloud, which provides different storage, processing, and media services. But providing high-quality communications services for moving vehicles with limited resources is a challenging problem that was addressed in [17]. Those authors proposed a cloud-based Communications-as-a-Service (CaaS) scheme that guarantees continuing transmissions and manages vehicular resources while the vehicle is moving outside the range of a roadside unit. In this scheme, the vehicular cloud is further divided into three layers: (1) a vehicular cloudlet that forms a tree topology of connected vehicles in a VANETs, (2) a roadside cloudlet that is locally established by RSUs, and (3) a central cloud that is a group of servers accessed via the Internet.

In order to provide the key services for smooth driving, a priority-based RSA scheme was introduced [18]. This scheme replaces the roadside unit with a dynamic cloud for managing overhead on the network and to maintain QoS to the users. The dynamic cloud collects and provides all required information to the vehicles in its range and forwards the information to a centralized controller, so (in case of emergency) a call is routed to the nearest police station and medical emergency response team to overcome the problem. However, the integration of VANETs with cloud computing has became a new research challenge due to scalability and reliability issues, because of the huge amounts of navigational data and safety messages along with event-location information. Thus, to tackle such issues, a cloud-based vehicle monitoring system for VANETs was proposed [19]. In this scheme, moving vehicles act as surveillance nodes that gather all information, including capturing video or images of a particular location, of road conditions, and capturing accident data from events on the roads. These data are forwarded to the central cloud through vehicular cloud formed by a group of vehicles. The main goal of this proposed scheme is tracking vehicles to handle emergency situations. Moreover, the extension of vehicular services, including multimedia applications, is dependent on integration of the cloud with VANETs. Keeping this view in mind, Jelassi et al. [20] proposed a new video streaming mechanism to provide efficient services to moving and parked vehicles. The proposed streaming framework comprises a vehicular cloud, roadside cloudlets, and a central cloud. In the proposed scheme, when any vehicle wants to see a video, the request is forwarded to any cloud component based on the availability of the requested video, and a scheduling mechanism is also generated in the cloud. During this process, a roadside cloudlet is responsible for maintaining the session during movement by the vehicle.

The emergence of the ITS leads to maximum usage of vehicular data by roadside users. To deal with this, cloud computing is best suited for a vehicular network, but security for the flow of data between the cloud and roadside units is a challenge. Thus, secure vehicle traffic data dissemination and analysis for cloud-based VANETs were proposed [21]. The proposed scheme applies privacy to the identification of a vehicle and its data flow through a pseudonym mechanism. This scheme also ensures authorization through secret credentials and authentication through identification-based signatures. Hence, the verification of a user's information and identification of intruding vehicles are done through batch verification and a pseudonym block list, respectively. Moreover, the requirement for accessing information related to transportation has become a basic need for smart vehicles in VANETs. By providing complex information, such as location services, real-time functionality, and storage services, vehicular applications are considered more complex than other conventional applications [22]. Thus, Aloqaily et al. discussed on-demand functionality of a vehicular cloud and proposed a new distributed scheme [22]. In the proposed scheme, the Vehicular Trusted Third Party (VTTP) concept is exploited for efficient traffic service management. This scheme also facilitates the drivers to switch between different trusted third parties to access their desired services and to reduce latency in communications. The Vehicluar-cloud (V-cloud) is an emerging and significant computing paradigm that has great impact on managing transportation issues and other roadside safety mechanisms. The V-cloud enables VANETs to offer multiple low-cost services, including advertisements and multimedia. Garai et al. [23] presented a QoS-aware three-layer V-cloud architecture that provides a tree-based connection to vehicles in a network. They also proposed authentication and privacy mechanisms via certification for secure communications which exploits public key certificates with zone-based vehicular groups to prevent malicious attacks.

Hossain et al. [24] presented a resource allocation scheme for cloud-based surveillance video computing. The proposed scheme focuses on the optimization of virtual machine resources to fulfill the various user services provided by a media cloud. A single service is not effective for user requests but composite services can be more efficient and optimal. This scheme was a composition of two mechanisms: (1) a linear programming model and (2) a heuristic approach that can dynamically allocate resources in a media cloud. Similarly, Cicco et al. [25] introduced a resource allocation controller for streaming video over multimedia cloud computing. The aim was to provide high-quality streaming video to users and minimize distribution cost. A power saving-based scheme for multimedia streaming services for embedded systems in MCC was introduced [26]. That scheme uses digital signal processor to deliver a high-quality media service in the cloud and save power for mobile devices. Similarly, Song et al. [27] presented a two-stage scheme for task and resource management in MCC. The proposed approach focuses on resource management to define the ways to assign virtual machines to actual servers, and for task management, where virtual machines are assigned the tasks. Moreover, Wen et al. [28] highlighted the key challenges that can affect the performance of cloud computing while assigning an accurate amount of cloud resources in a short period of time to compute multimedia applications. Thus, to cope with such issues an efficient load balancing scheme called "*cloud based*

multimedia load balancing" for cloud-based processing was proposed [29]. Furthermore, the concept of a hierarchical media cloud system that consists of a resource manager, cluster heads, and server clusters was introduced [30]. In this technique, user requests arrive at a resource manager that transfers the requests to a content server and a cluster head for further processing. The main focus of this technique is how to optimally distribute the workload in the system so that it does not affect the performance of a multimedia system.

Some more work related to efficient resource allocation in MCC was presented [7, 11, 31–34]. In [7], a priority-based optimal resource allocation technique for MCC was presented. In this scheme, the users' requests are treated as multiple priority-classes and are processed according to their priority values. A media-edge cloud design architecture for MCC was presented [11]. In the proposed architecture, the authors discussed parallel and distributed cloud processing and QoS. They further divided the media cloud into three clusters: (1) CPU, (2) storage, and (3) GPU. Media graphics processes are executed on GPU clusters, while CPU clusters perform general media process computation and storage was managed by storage clusters. In [7], a queuing-based resource allocation scheme to minimize response time while allocating cloud resources was presented. In [31], a dynamic resource allocation scheme for MCC was introduced. The prime objective of that proposed scheme was to enhance system efficiency while finding the exact amount of resources to be allocated to a media cloud. Li et al. [32] proposed QoS-based resource allocation for MCC. For performance enhancement, the authors considered QoS factors, such as completion time, cost, and energy consumption. Hong et al. [33] proposed efficient resource allocation based on media-task QoS for MCC and a cost-effective and QoS-aware resource allocation scheme for MCC-based e-health systems was proposed [34].

3. Proposed Dynamic Priority-Based Efficient Resource Allocation Scheme for Multimedia Cloud

In this section, we discuss the proposed dynamic priority-based efficient resource allocation scheme for the multimedia cloud (MC). The proposed scheme is mainly divided into three parts: (1) the MC processing architecture, (2) the MC queuing model, and (3) the MC resource allocation strategy. Furthermore, the proposed scheme divides each multimedia task into four subphases: (1) conversion, (2) extraction, (3) matching, and (4) reconstruction. This method is similar to the one presented [11]. However, in our proposed architecture, these four subphases for multimedia processing are performed by four different, separate, and dedicated computing clusters, rather than a single computing cluster. In our proposed scheme, each subphase of multimedia processing can be performed individually or related to other subphases depending on user requirements. In order to efficiently compute multimedia tasks, each media computing cluster assigns computing resources dynamically.

In order to process multimedia-related tasks (e.g., images and videos), the processing architecture is mainly divided into three parts. The first is the LBM, which is designed to collect multimedia processing requests from heterogeneous multimedia users. There is a request queue in the LBM to store user requests. The objective of the LBM is to schedule user requests for processing and to estimate the received workload at any time that further helps in allocating computing resources to each media cluster. The second part is the media clusters for multimedia processing. In our proposed scheme, there are four media clusters: (1) the Dynamic Conversion (DC) cluster, (2) the Dynamic Extraction (DE) cluster, (3) the Dynamic Matching (DM) cluster, and (4) the Dynamic Reconstruction (DR) cluster, as shown in Figure 2. Each media cluster name is based on its functionality of computing a pacific multimedia subphase, such as conversion, extraction, matching, and reconstruction. There are N job queues in every media cluster to store and process multimedia tasks according to users' priorities.

In this paper, we assume that priority for each multimedia task is already assigned according to the type and nature of the task. For example, the more important and critical multimedia user's requests are assigned a higher priority and, similarly, the least important requests are assigned the lowest priority. A higher priority user's tasks are placed in a higher priority queue to meet its processing deadlines or QoE. Moreover, each subphase of multimedia processing can be performed individually or collectively, based on user processing requirements. The last module is the transmission unit (TU), which is responsible for storing and forwarding the processed multimedia users' tasks. The TU contains two components: (1) the priority-based transmissions queues that contain user processed tasks according to their priority and (2) the transmission server that forwards processed users' jobs according to display device heterogeneity. It also adjusts the quality of transmissions according to the QoS requirements of the receiver.

3.1. Priority-Based Queuing Model for Multimedia Computation. In this subsection, we describe our priority-based queuing model. As shown in Figure 3, there are three different types of queues: (1) scheduling queue, (2) computing queues, and (3) transmission queues. In this paper, we assume N priority-classes of multimedia requests. Therefore, each multimedia request corresponds to a single priority-class at a time. Smaller priority numbers represent a higher priority-class. For example, the priority 1 multimedia request class has a higher priority than priority 2 multimedia request class. Furthermore, any priority-class i multimedia requests are classified with the help of four parameters: (1) average task size, T, which specifies the total number of instructions, (2) average arrival rate, λ_i, of priority-class i multimedia requests, which denotes the number of requests per second, (3) average result size, R_i, which is measured in bits, and (4) the upper bound of service response time, ϵ_i, which is measured in seconds. Moreover, in our scenario, each multimedia task is further divided into four subtasks (the subphases discussed previously). The priority of each multimedia request is

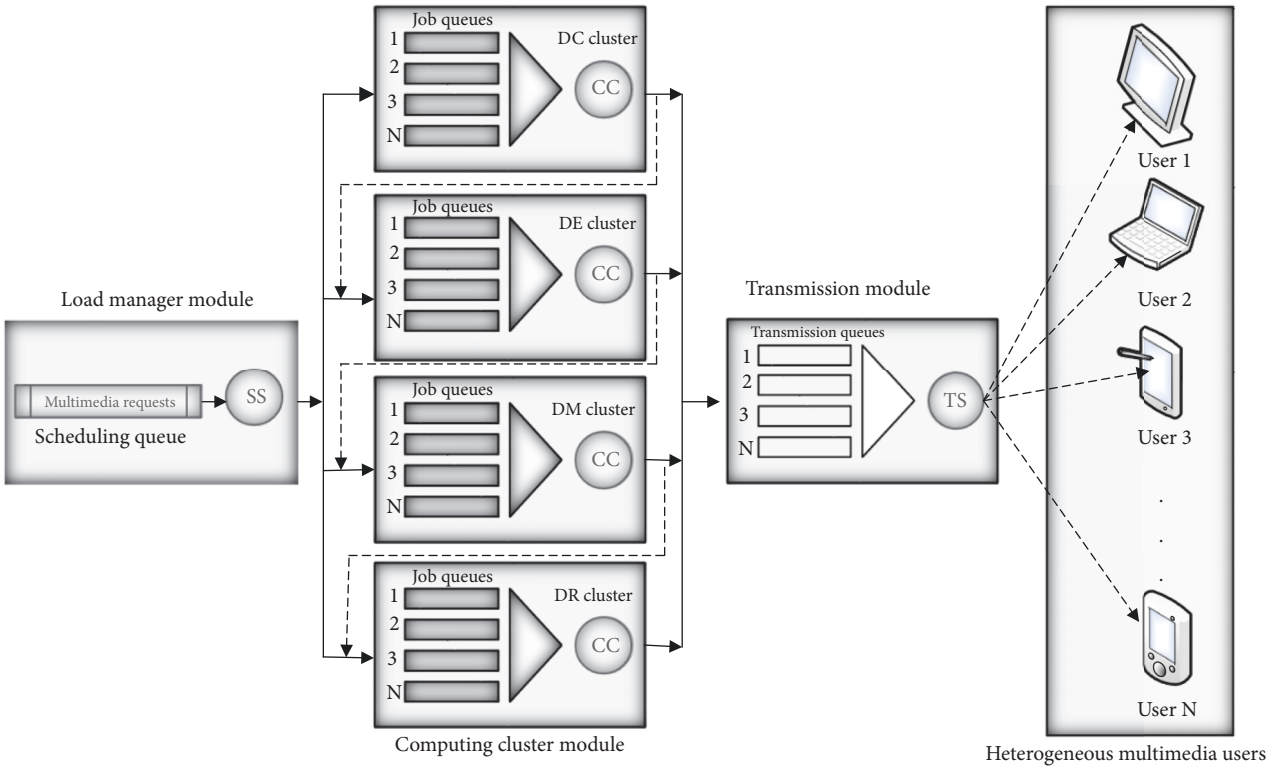

Figure 2: Multimedia cloud computing architecture.

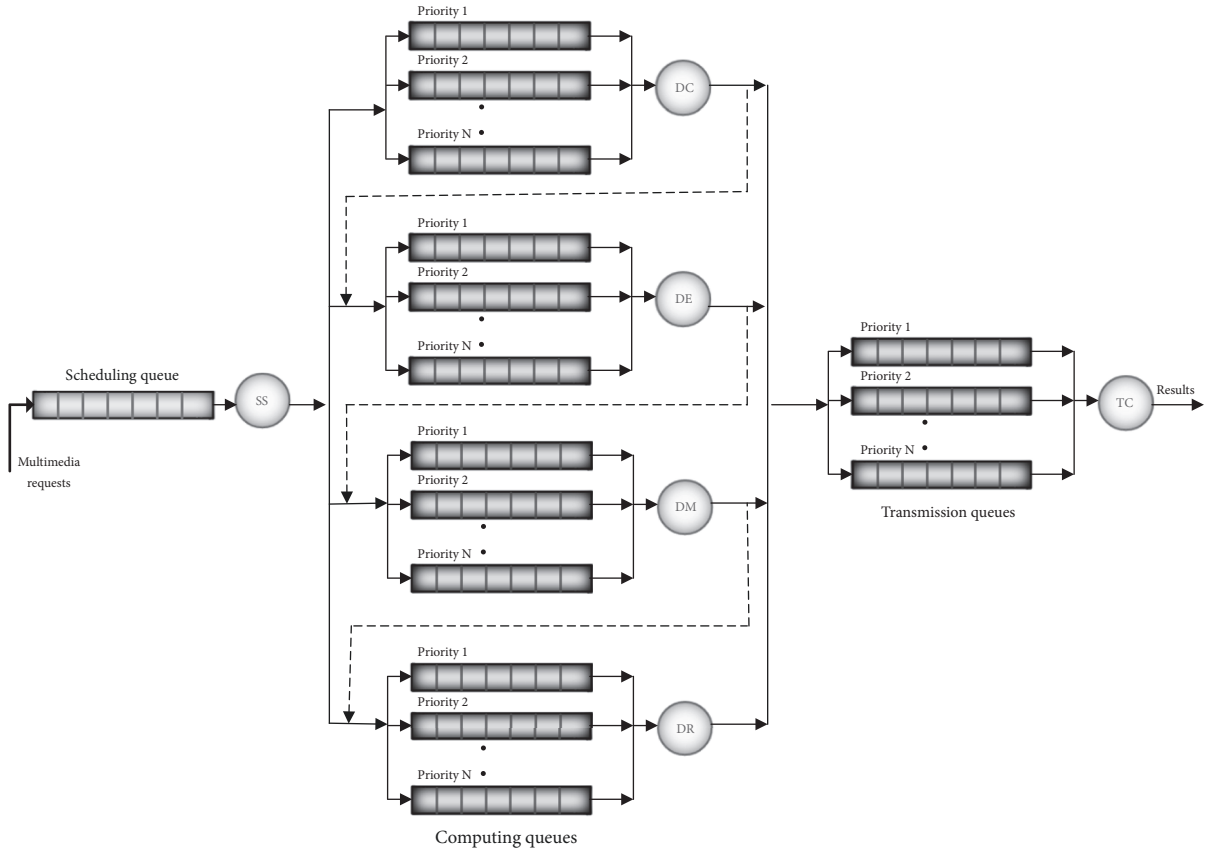

Figure 3: Priority-based queuing model for multimedia cloud computing architecture.

assigned based on its criticality or importance; however, we will not discuss this factor in detail in this work.

The Scheduling Server (SS) schedules multimedia requests at scheduling rate μ for processing according to assigned or allocated priority-classes. For this purpose, each computing cluster (CC) contains N priority queues, as shown in Figure 3. Similarly, N transmission queues store the processed data or results to forward to the intended end users at transmission rate ξ. Multimedia requests of priority-class i arrivals are modeled by using a Poisson distribution with the λ_i arrival rate. Thus, the average arrival rate of all priority-class multimedia requests computed based on the Poisson process composition property is given as follows:

$$\lambda = \sum_{i=1}^{N} \lambda_i \tag{1}$$

The SS schedules the multimedia requests with the highest priority first. Hence, the scheduling queue is modeled as preemptive priority $M/M/1$ with mean service rate η of the SS. T_i^{sch} is the mean service response time for scheduling priority-class i multimedia requests given in [35] as follows:

$$T_i^{sch} = \frac{1/\mu}{1-\sigma_{i-1}^{sch}} + \frac{\sum_{j=1}^{i}\left(\lambda_j/\mu^2\right)}{\left(1-\sigma_{i-1}^{sch}\right)\left(1-\sigma_i^{sch}\right)}, \quad \forall i = 1, 2, \ldots, N \tag{2}$$

where μ is the scheduling rate and σ_i^{sch} is given as follows:

$$\sigma_i^{sch} = \sum_{j=1}^{i} \frac{\lambda_j}{\mu}, \tag{3}$$

In order to stabilize the scheduling queue, the following condition should be satisfied:

$$\sigma_N^{sch} = \sum_{j=1}^{N} \frac{\lambda_j}{\mu} < 1 \tag{4}$$

Due to the same service rate for all priority-class multimedia requests, the total response time is the same as the $M/M/1$ queue. Hence, the mean service response time for scheduling all priority-class multimedia requests is computed as follows:

$$T^{sch} = \frac{1/\mu}{1-\lambda/\mu} \tag{5}$$

In our scenario, each multimedia task of priority-class i, F_i, is divided into four subtasks, or subphases denoted by F_i^k where $k = 1, 2, 3, 4$, and each subtask represents a unique computational phase. The highest priority multimedia tasks are computed first in a preemptive priority manner. The service time for multimedia requests of priority-class i is assumed to be exponentially distributed with mean time, F_i^k/C^k. C^k represents the total computational resources allocated to computing subphase k. Hence, the mean service response time for multimedia tasks of priority-class i in computational phase k is given in [36] as follows:

$$T_i^{com_k} = \frac{F_i/C^k}{1-\sigma_{i-1}^{com_k}} + \frac{\sum_{j=1}^{i}\left(\lambda_j\left(F_j\right)^2/\left(C^k\right)^2\right)}{\left(1-\sigma_{i-1}^{com_k}\right)\left(1-\sigma_i^{com_k}\right)}, \quad \forall i = 1, \ldots, N \tag{6}$$

where $\sigma_i^{com_k}$ is given as

$$\sigma_i^{com_k} = \sum_{j=1}^{i} \frac{\lambda_j F_j}{C^k}. \tag{7}$$

Similar to the above, the following stabilizing condition is also applied to the computing queue:

$$\sigma_N^{com_k} = \sum_{j=1}^{N} \frac{\lambda_j F_j}{C^k} < 1. \tag{8}$$

We assume very high-speed communications links within the computing clusters; therefore, the latency of moving multimedia subtasks from one computing cluster to another computing cluster is almost negligible. Thus, the mean service response time for multimedia tasks of priority-class i for all four computing subphases is given as follows:

$$T_i^{com} = \sum_{k=1}^{4} T_i^{com_k} \tag{9}$$

Finally, the mean service time for computing of all types of multimedia priority-classes under unequal service rate distribution is computed as follows:

$$T^{com} = \sum_{i=1}^{N} \frac{\lambda_i T_i^{com}}{\lambda} \tag{10}$$

Multimedia tasks processed through scheduling and computing queues are finally sent to the transmission queue at transmission rate ξ, where N priority queues store processed results of size R_i before forwarding them to the intended priority-class i end users. However, the fraction R_i/ξ represents the mean transmission time of exponentially distributed priority-class i results. Eventually, the mean service response time for transmitting/forwarding the results of priority-class i multimedia users is given as

$$T_i^{tra} = \frac{R_i/\xi}{1-\sigma_{i-1}^{tra}} + \frac{\sum_{j=1}^{i}\left(\lambda_j\left(R_i^2/\xi^2\right)\right)}{\left(1-\sigma_{i-1}^{tra}\right)\left(1-\sigma_i^{tra}\right)} \quad \forall i = 1, \ldots, N \tag{11}$$

where σ_i^{tra} is computed as

$$\sigma_i^{tra} = \sum_{j=1}^{i} \frac{\lambda_j R_i}{\xi}, \tag{12}$$

and to ensure the stability of the transmission queue, the following condition must also hold:

$$\sigma_N^{com_k} = \sum_{j=1}^{N} \frac{\lambda_j F_j}{C^k} < 1. \tag{13}$$

Finally, the service response time for transmitting all priority-class multimedia results is computed as given below:

$$T^{tra} = \sum_{i=1}^{N} \frac{\lambda_i T_i^{tra}}{\lambda} \tag{14}$$

However, the factor T_i^{tra} is computed in (11) above. Thus, based on the above queuing analysis, we find the total delay or latency of priority-class i multimedia tasks given as follows:

$$T_i^{tot} = T_i^{sch} + T_i^{com} + T_i^{tra} \tag{15}$$

Thus, by using (2), (9), and (11), T_i^{tot} can be represented as follows:

$$T_i^{tot} = \frac{1/\mu}{1-\sigma_{i-1}^{sch}} + \frac{\sum_{j=1}^{i}\left(\lambda_j/\mu^2\right)}{\left(1-\sigma_{i-1}^{sch}\right)\left(1-\sigma_i^{sch}\right)} + \sum_{k=1}^{4} T_i^{com_k} + \frac{R_i/\xi}{1-\sigma_{i-1}^{tra}} + \frac{\sum_{j=1}^{i}\left(\lambda_j\left(R_i^2/\xi^2\right)\right)}{\left(1-\sigma_{i-1}^{tra}\right)\left(1-\sigma_i^{tra}\right)} \quad \forall i = 1,\ldots,N \tag{16}$$

Thus, by substituting the values of σ_{i-1}^{sch} and σ_i^{tra} into (16), we get

$$T_i^{tot} = \frac{1/\mu}{1-\sum_{j=1}^{i-1}\left(\lambda_j/\mu\right)} + \frac{\sum_{j=1}^{i}\left(\lambda_j/\mu^2\right)}{\left(1-\sum_{j=1}^{i-1}\left(\lambda_j/\mu\right)\right)\left(1-\sum_{j=1}^{i}\left(\lambda_j/\mu\right)\right)} + \sum_{k=1}^{4} T_i^{com_k} + \frac{R_i/\xi}{1-\sum_{j=1}^{i-1}\left(\lambda_j R_i/\xi\right)} + \frac{\sum_{j=1}^{i}\left(\lambda_j\left(R_i^2/\xi^2\right)\right)}{\left(1-\sum_{j=1}^{i-1}\left(\lambda_j R_i/\xi\right)\right)\left(1-\sum_{j=1}^{i}\left(\lambda_j R_i/\xi\right)\right)} \quad \forall i = 1,\ldots,N \tag{17}$$

Finally, substituting the values of $T_i^{com_k}$ and $\sigma_i^{com_k}$ into (17), we get

$$T_i^{tot} = \frac{1/\mu}{1-\sum_{j=1}^{i-1}\left(\lambda_j/\mu\right)} + \frac{\sum_{j=1}^{i}\left(\lambda_j/\mu^2\right)}{\left(1-\sum_{j=1}^{i-1}\left(\lambda_j/\mu\right)\right)\left(1-\sum_{j=1}^{i}\left(\lambda_j/\mu\right)\right)} + \sum_{k=1}^{4}\frac{F_i/C^k}{1-\sum_{j=1}^{i-1}\left(\lambda_j F_j/C^k\right)} + \frac{\sum_{j=1}^{i}\left(\lambda_j\left(F_j\right)^2/\left(C^k\right)^2\right)}{\left(1-\sum_{j=1}^{i-1}\left(\lambda_j F_j/C^k\right)\right)\left(1-\sum_{j=1}^{i}\left(\lambda_j F_j/C^k\right)\right)} + \frac{R_i/\xi}{1-\sum_{j=1}^{i-1}\left(\lambda_j R_i/\xi\right)} + \frac{\sum_{j=1}^{i}\left(\lambda_j\left(R_i^2/\xi^2\right)\right)}{\left(1-\sum_{j=1}^{i-1}\left(\lambda_j R_i/\xi\right)\right)\left(1-\sum_{j=1}^{i}\left(\lambda_j R_i/\xi\right)\right)} \quad \forall i = 1,\ldots,N \tag{18}$$

Furthermore, the total service response time for all types of priority-classes can be computed as

$$T^{tot} = T^{sch} + T^{com} + T^{tra} \tag{19}$$

by substituting the values of T^{sch}, T^{com}, and T^{tra} into the above equation, we get

$$T^{tot} = \frac{1/\mu}{1-\lambda/\mu} + \sum_{i=1}^{N}\frac{\lambda_i T_i^{com}}{\lambda} + \sum_{i=1}^{N}\frac{\lambda_i T_i^{tra}}{\lambda} \tag{20}$$

After substituting the values of T_i^{com} and T_i^{tra} into (20), we get

$$T^{tot} = \frac{1/\mu}{1-\lambda/\mu} + \sum_{k=1}^{4}\left(\sum_{i=1}^{N}\frac{\lambda_i\sum_{j=1}^{i}\left(\lambda_j F_j^2/\left(C^k\right)^2\right)}{\lambda\left(1-\sum_{j=1}^{i-1}\left(\lambda_j F_j/C^k\right)\right)\left(1-\sum_{j=1}^{i}\left(\lambda_j F_j/C^k\right)\right)} + \sum_{i=1}^{N}\frac{\lambda_i F_i/C^k}{\lambda\left(1-\sum_{j=1}^{i-1}\left(\lambda_j F_j/C^k\right)\right)}\right) + \sum_{i=1}^{N}\frac{\lambda_i R_i/\xi}{\lambda\left(1-\sum_{j=1}^{i-1}\left(\lambda_j R_i/\xi\right)\right)} + \sum_{i=1}^{N}\frac{\lambda_i\sum_{j=1}^{i}\left(\lambda_j R_i^2/\xi^2\right)}{\lambda\left(1-\sum_{j=1}^{i-1}\left(\lambda_j R_i/\xi\right)\right)\left(1-\sum_{j=1}^{i}\left(\lambda_j R_i/\xi\right)\right)} \tag{21}$$

Thus, the end user experiences satisfactory QoE if end-to-end latency (e.g., latency from scheduling till transmission) of multimedia requests of priority-class i is less than its predefined threshold (i.e., the upper bound of service time, ϵ_i). This QoE constraint is defined as follows:

$$T_i^{tot} < \epsilon_i \tag{22}$$

Similarly, MCC is considered to be efficient if it holds to the following conditions for all multimedia users of priority-class N:

$$T^{tot} < \epsilon \tag{23}$$

However, if the MCC does not meet the above two conditions, more resources are added to meet the QoE conditions.

3.2. Step-by-Step Procedure of Multimedia Processing in MCC. The step-by-step working procedure of the proposed MCC framework shown in Figure 4 is as follows.

Step 1. The LBM receives the multimedia processing requests from different multimedia users and assigns the priority to each multimedia task based on its importance and criticality (e.g., delay) and places them into a request queue.

Step 2. The LBM periodically performs a queue analysis after every t time interval to estimate the workload.

Step 3. The LBM assigns computing resources to each computing cluster (i.e., DC, DE, DM, and DR) based on the estimated workload to meet the QoE of each multimedia user.

Step 4. The LBM assigns multimedia processing tasks to appropriate media clusters (i.e., DC, DE, DM, and DR) according to the nature of processing task. Furthermore, the

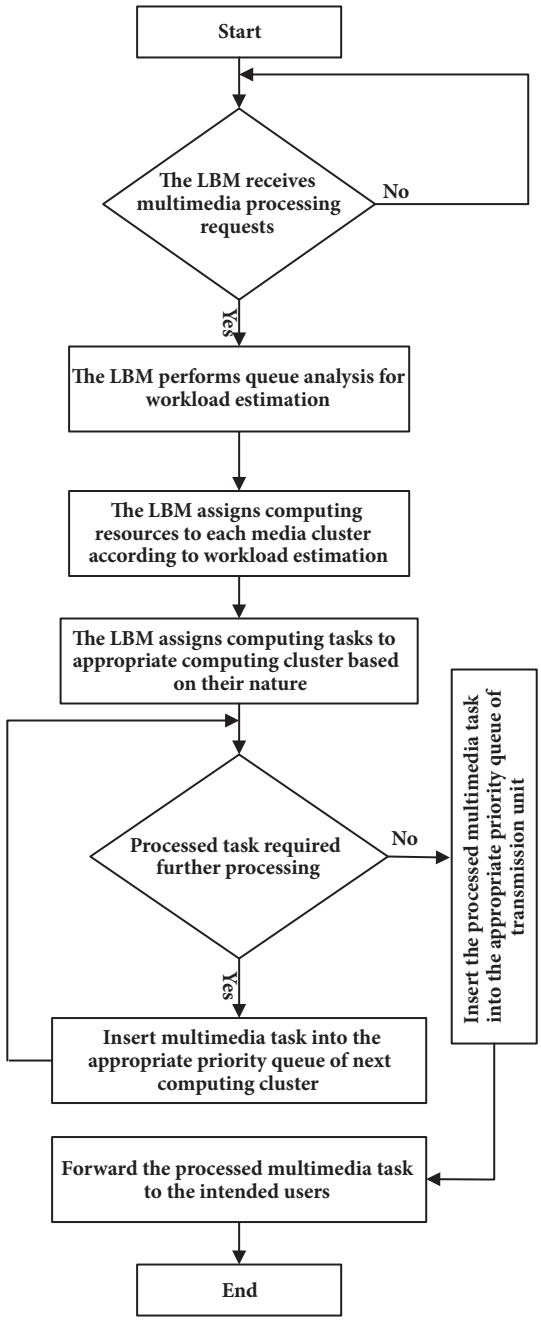

Figure 4: Working flow of MCC.

LBM places each multimedia task into the appropriate queue of each computing cluster based on their priorities.

Step 5. If the processed multimedia task requires further processing, which is decided based on the nature of multimedia task, the task is placed into the appropriate priority queues of the next computing cluster; otherwise, they are sent to respective queue of transmission units for further processing.

Step 6. The transmission unit processes the multimedia tasks (e.g., conversion and forwarding) to meet the QoE at the intended receivers.

Table 2: Simulation settings.

RAM	MIPS	CPUs
4096	4200	2

4. Performance Analysis

In this section, we first discuss the experimental setup. Secondly, we describe performance evaluation schemes and performance evaluation criteria, and finally, we present the simulation results.

4.1. Experimental Setup. In this subsection, we describe our experimental setup. The performance of the proposed scheme is evaluated by using the CloudSim [30] simulator. It is a Java-based library model that is used worldwide for cloud-based simulations [37]. The Cloudsim model cloud-based content distribution service, which sets the corresponding application complexity according to the calculation requests of the service, ensures the validity of the simulation. We designed and modeled the cloud into three main components: (I) *the LBM*, (II) the CC unit, and (III) the TU unit. We perform simulations on multimedia images and for this purpose, we changed the characteristics of the cloudlet class and modeled it according to our requirements. The arrival of multimedia requests follows a Poisson distribution with t time intervals. We evaluate the performance of the proposed scheme with other schemes and hardware (i.e., RAM, Million Instruction Per Second (MIPS), and the number of CPUs), as presented in Table 2.

4.2. Performance Evaluation Scheme and Criteria. In this subsection, we discuss the performance evaluation schemes and evaluation criteria. For simulation study, we evaluate and compare the proposed scheme's static resource allocation and baseline single cluster schemes. In the static resource-allocation scheme, computing resources for each computing cluster are allocated at initialization time and computing resources are not periodically updated. However, in the baseline single cluster scheme, all four subphases are collectively computed on a single cluster. Our performance evaluation criteria are as follows.

Computing Cost. Minimizing the computing resources costs while providing better QoE is a challenging task for any cloud service provider. In our case, computing resources costs are also directly proportional to the number of virtual machines reserved for computing multimedia tasks. The more computing resources deployed, the higher the computing costs and similarly, lesser or minimum computing resources means minimum costs.

Response Time. The response time, or end-to-end delay, is the time measured from when the multimedia request is received at the LBM until the requesting user receives the required response. However, in our case, QoE is directly proportional to the value of the response time. For example, if a particular multimedia user does not receive the required response in

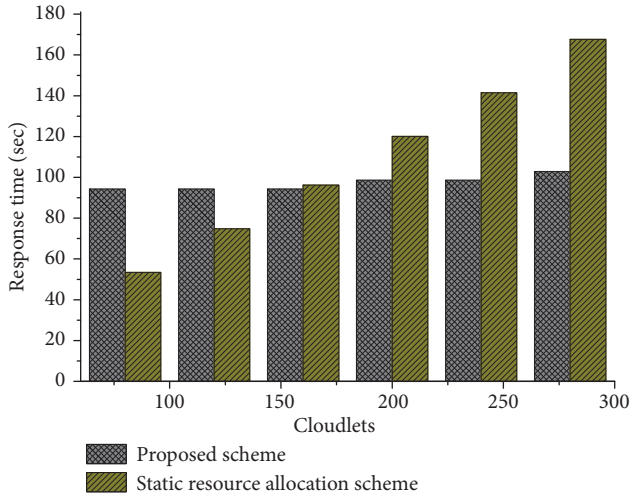

Figure 5: Proposed scheme vs. the static resource allocation scheme.

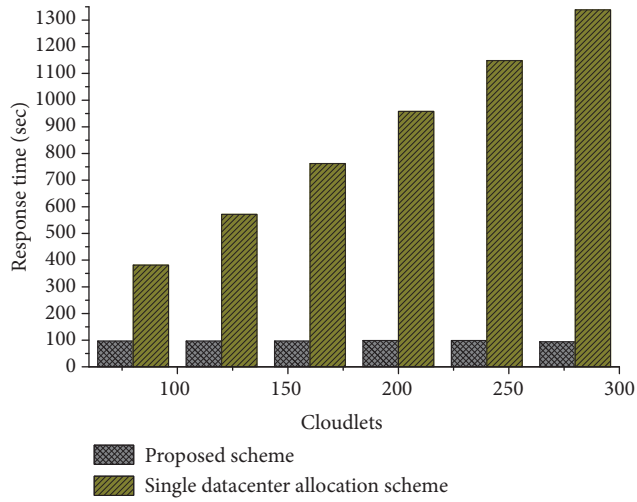

Figure 6: Proposed scheme vs. the single cluster allocation scheme.

time, then that user experiences low QoE, and similarly, if a multimedia user receives the required response in time, that user experiences better QoE.

4.3. Simulation Results. This subsection discusses the simulation results of the proposed scheme, compared with the baseline single cluster allocation and static allocation schemes. Figure 5 presents the performance evaluation results of the proposed scheme with the static resource allocation scheme in terms of response time under the configuration settings, such as a virtual machine with processing speed of 4200 MIPS, two CPUs, and 4 GB of RAM, as given in Table 2. We can see from the figure that the static scheme performs better when the number of tasks or cloudlets is fewer, but as the cloudlets increase in number, the response time also increases, so that consequently reduces the QoE experienced at the end terminal (i.e., the receiver). However, the resources in the proposed scheme are periodically updated according to total workload information. Therefore, it performs better even with more tasks and provides guaranteed QoE to multimedia users.

Figure 6 presents the simulation results of the proposed scheme compared with the baseline single cluster allocation scheme in terms of response time under the simulation settings presented in Table 2. In the single cluster computing scheme, the response time increases as the cloudlets increase. This happens because, in the baseline single cluster-based computing scheme, a single cluster is responsible for performing all four subphases of multimedia-related tasks. But, in the proposed scheme, image or multimedia-task processing is distributed over four different and dedicated computing clusters. Hence, it is clear from Figure 6 that the proposed scheme performs better than the single cluster-based computing scheme and provides the guaranteed QoE to multimedia users.

Figure 7 shows simulation results of the proposed scheme compared with the static resource allocation scheme in terms of the amount of computing resources (i.e., the number

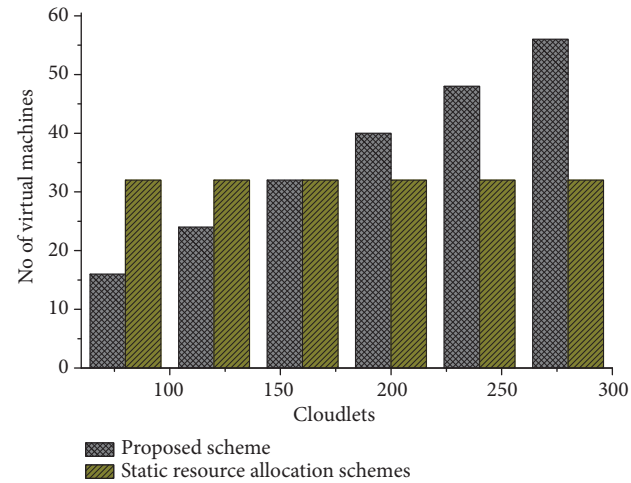

Figure 7: Proposed scheme vs. static resource allocation scheme.

of virtual machines allocated to computing the supplied multimedia tasks). Thus, Figure 7 clearly shows that, with an increase in the number of cloudlets (i.e., multimedia tasks), the number of virtual machines also increases. However, the amount of computing resources in the static resource allocation scheme is constant. This is because the proposed scheme aims at maintaining a minimum response time and providing better QoE to multimedia users.

5. Conclusion

In this paper, we proposed a dynamic priority-based efficient resource allocation scheme for multimedia-enabled vehicular users in order to address challenges like fast response times, guaranteed Quality of Experience, and minimum computing resources costs. In the proposed scheme, every multimedia-related task is divided into four subphases and each subphase is computed by a dedicated computing cluster. Priority

queues are used to ensure on-time response delivery to different multimedia end users with different priorities. Moreover, in the proposed scheme, the computing resources are dynamically updated based on workload information estimated by a load manager. The performance of the proposed scheme is evaluated in terms of response time and computing costs using the CloudSim Simulator against the static resource allocation scheme and the baseline single cluster-based computing scheme. The simulation results show that the proposed scheme outperforms the static resource allocation scheme and the baseline single cluster-based computing technique.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors have no conflicts of interest to declare.

Acknowledgments

This research is supported by the Ministry of Science and ICT (MSIT), Korea, under the Grand Information Technology Research Center support program (IITP-2018-2014-1-00729) supervised by the Institute for Information & communications Technology Promotion (IITP).

References

[1] T. Mekki, I. Jabri, A. Rachedi, and M. ben Jemaa, "Vehicular cloud networks: Challenges, architectures, and future directions," *Vehicular Communications*, vol. 9, pp. 268–280, 2017.

[2] S. Midya, A. Roy, K. Majumder, and S. Phadikar, "Multi-objective optimization technique for resource allocation and task scheduling in vehicular cloud architecture: A hybrid adaptive nature inspired approach," *Journal of Network and Computer Applications*, vol. 103, pp. 58–84, 2018.

[3] M. Gerla, E.-K. Lee, G. Pau, and U. Lee, "Internet of vehicles: from intelligent grid to autonomous cars and vehicular clouds," in *Proceedings of the IEEE World Forum on Internet of Things (WF-IoT '14)*, pp. 241–246, March 2014.

[4] H. Yuan, C.-C. J. Kuo, and I. Ahmad, "Energy efficiency in data centers and cloud-based multimedia services: an overview and future directions," in *Proceedings of the 2010 International Conference on Green Computing, Green Comp 2010*, pp. 375–382, USA, August 2010.

[5] D. Kovachev, Y. Cao, and R. Klamma, "Mobile multimedia cloud computing and the web," in *Proceedings of the 2011 Workshop on Multimedia on the Web, MMWeb 2011*, pp. 21–26, Austria, September 2011.

[6] J. Chen, S. Wuy, Y. T. Larosa, P. Yang, and Y. Li, "IMS cloud computing architecture for high-quality multimedia applications," in *Proceedings of the 2011 7th International Wireless Communications and Mobile Computing Conference (IWCMC 2011)*, pp. 1463–1468, Istanbul, Turkey, July 2011.

[7] X. Nan, Y. He, and L. Guan, "Optimal resource allocation for multimedia cloud based on queuing model," in *Proceedings of the 3rd IEEE International Workshop on Multimedia Signal Processing (MMSP '11)*, pp. 1–6, November 2011.

[8] Y. Wu, C. Wu, B. Li, X. Qiu, and F. C. M. Lau, "CloudMedia: When cloud on demand meets video on demand," in *Proceedings of the 31st International Conference on Distributed Computing Systems*, pp. 268–277, July 2011.

[9] R. H. Glitho, "Cloud-based multimedia conferencing: Business model, research agenda, state-of-the-art," in *Proceedings of the 13th IEEE International Conference on Commerce and Enterprise Computing, CEC 2011*, pp. 226–230, Luxembourg, September 2011.

[10] C. Gadea, B. Solomon, B. Ionescu, and D. Ionescu, "A collaborative cloud-based multimedia sharing platform for social networking environments," in *Proceedings of the 2011 20th International Conference on Computer Communications and Networks, ICCCN 2011*, USA, August 2011.

[11] W. Zhu, C. Luo, J. Wang, and S. Li, "Multimedia cloud computing," *IEEE Signal Processing Magazine*, vol. 28, no. 3, pp. 59–69, 2011.

[12] X. Liu, Z. Chen, K. Hua, M. Liu, and J. Zhang, "An adaptive multimedia signal transmission strategy in cloud-assisted vehicular networks," in *Proceedings of the 2017 IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud)*, pp. 220–226, Prague, Czech Republic, August 2017.

[13] J. Joshi, K. Jain, Y. Agarwal, M. J. Deka, and P. Tuteja, "TMaaS: Traffic management as a service using cloud in VANETs," in *Proceedings of the 3rd IEEE International Conference on Smart Instrumentation, Measurement and Applications, ICSIMA 2015*, Malaysia, November 2015.

[14] H. Gong, L. Yu, N. Liu, and X. Zhang, "Mobile content distribution with vehicular cloud in urban VANETs," *China Communications*, vol. 13, no. 8, Article ID 7563691, pp. 84–96, 2016.

[15] R. I. Meneguette, "A vehicular cloud-based framework for the intelligent transport management of big cities," *International Journal of Distributed Sensor Networks*, vol. 12, no. 5, p. 8198597, 2016.

[16] R. Hussain, Z. Rezaeifar, and H. Oh, "A paradigm shift from vehicular ad hoc networks to VANET-based clouds," *Wireless Personal Communications*, vol. 83, no. 2, pp. 1131–1158, 2015.

[17] M. Garai, S. Rekhis, and N. Boudriga, "Communication as a service for cloud VANETs," in *Proceedings of the 20th IEEE Symposium on Computers and Communication, ISCC 2015*, pp. 371–377, Cyprus, July 2015.

[18] S. Singh, S. Negi, and S. K. Verma, "VANET based p-RSA scheduling algorithm using dynamic cloud storage," *Wireless Personal Communications*, vol. 98, no. 4, pp. 3527–3547, 2018.

[19] J. Joshi, K. Jain, and Y. Agarwal, "CVMS: Cloud based vehicle monitoring system in VANETs," in *Proceedings of the International Conference on Connected Vehicles and Expo, ICCVE 2015*, pp. 106–111, China, October 2015.

[20] S. Jelassi, A. Bouzid, and H. Youssef, "QoE-driven video streaming system over cloud-based VANET," in *Communication Technologies for Vehicles*, vol. 9066 of *Lecture Notes in Computer Science*, pp. 84–93, Springer International Publishing, Cham, 2015.

[21] L. Nkenyereye, Y. Park, and K.-H. Rhee, "Secure vehicle traffic data dissemination and analysis protocol in vehicular cloud computing," *The Journal of Supercomputing*, vol. 74, no. 3, pp. 1024–1044, 2018.

[22] M. Aloqaily, B. Kantarci, and H. T. Mouftah, "Trusted Third Party for service management in vehicular clouds," in *Proceedings of the 13th IEEE International Wireless Communications and Mobile Computing Conference, IWCMC 2017*, pp. 928–933, Spain, June 2017.

[23] M. Garai, S. Rekhis, and N. Boudriga, "A vehicular cloud for secure and QoS aware service provision," in *Advances in Ubiquitous Networking 2*, vol. 397 of *Lecture Notes in Electrical Engineering*, pp. 219–233, Springer Singapore, Singapore, 2017.

[24] M. S. Hossain, M. M. Hassan, M. A. Qurishi, and A. Alghamdi, "Resource allocation for service composition in cloud-based video surveillance platform," in *Proceedings of the IEEE International Conference on Multimedia and Expo Workshops (ICMEW '12)*, pp. 408–412, Melbourne, Australia, July 2012.

[25] L. De Cicco, S. Mascolo, and D. Calamita, "A resource allocation controller for cloud-based adaptive video streaming," in *Proceedings of the 2013 IEEE International Conference on Communications Workshops, ICC 2013*, pp. 723–727, Hungary, June 2013.

[26] Y.-W. Ma, J.-L. Chen, C.-H. Chou, and S.-K. Lu, "A power saving mechanism for multimedia streaming services in cloud computing," *IEEE Systems Journal*, vol. 8, no. 1, pp. 219–224, 2014.

[27] B. Song, M. M. Hassan, A. Alamri et al., "A two-stage approach for task and resource management in multimedia cloud environment," *Computing: Archives for Scientific Computing*, vol. 98, no. 1-2, pp. 119–145, 2016.

[28] W. Hui, C. Lin, H.-Y. Zhao, and Y. Yang, "Effective load balancing for cloud-based multimedia system," in *Proceedings of the 2011 International Conference on Electronic and Mechanical Engineering and Information Technology, EMEIT 2011*, pp. 165–168, China, August 2011.

[29] C.-C. Lin, H.-H. Chin, and D.-J. Deng, "Dynamic multiservice load balancing in cloud-based multimedia system," *IEEE Systems Journal*, vol. 8, no. 1, pp. 225–234, 2014.

[30] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. de Rose, and R. Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and Experience*, vol. 41, no. 1, pp. 23–50, 2011.

[31] K. Sembiring and A. Beyer, "Dynamic resource allocation for cloud-based media processing," in *Proceedings of the 23rd ACM Workshop on Network and Operating Systems Support for Digital Audio and Video, NOSSDAV 2013*, pp. 49–54, Norway, February 2013.

[32] Y. Li, L. Zhuo, and H. Shen, "An efficient resource allocation method for multimedia cloud computing," in *Intelligence Science and Big Data Engineering*, vol. 8261 of *Lecture Notes in Computer Science*, pp. 246–254, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[33] B. Hong, Y. Zhai, R. Tang, and Y. Feng, "A resources allocation algorithm based on media task QoS in cloud computing," in *Proceedings of the 2013 4th IEEE International Conference on Software Engineering and Service Science, ICSESS 2013*, pp. 841–844, China, May 2013.

[34] M. M. Hassan, "Cost-effective resource provisioning for multimedia cloud-based e-health systems," *Multimedia Tools and Applications*, vol. 74, no. 14, pp. 5225–5241, 2015.

[35] X. Nan, Y. He, and L. Guan, "Optimal resource allocation for multimedia cloud in priority service scheme," in *Proceedings of the 2012 IEEE International Symposium on Circuits and Systems, ISCAS 2012*, pp. 1111–1114, Republic of Korea, May 2012.

[36] N. K. Jaiswal, *Priority queues*, Mathematics in Science and Engineering, Vol. 50, Academic Press, New York-London, 1968.

[37] T. Goyal, A. Singh, and A. Agrawa, "Cloudsim: simulator for cloud computing infrastructure and modeling," in *Proceedings of the International Conference on Modelling Optimization and Computing*, pp. 3566–3572, April 2012.