sustainability

MDPI

Article

Tradeoff between User Quality-Of-Experience and Service Provider Profit in 5G Cloud Radio Access Network

Mahbuba Afrin [1], Md. Abdur Razzaque [1], Iffat Anjum [1], Mohammad Mehedi Hassan [2,*] and Atif Alamri [2]

[1] Department of Computer Science and Engineering, University of Dhaka, Dhaka 1000, Bangladesh; m.afrin.ritu@gmail.com (M.A.); razzaque@du.ac.bd (M.A.R.); iffatanjum@cse.du.ac.bd (I.A.)
[2] College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia; atif@ksu.edu.sa
* Correspondence: mmhassan@ksu.edu.sa; Tel.: +966-114-695-202

Received: 3 October 2017; Accepted: 13 November 2017; Published: 18 November 2017

Abstract: In recent years, the Cloud Radio Access Network (CRAN) has become a promising solution for increasing network capacity in terms of high data rates and low latencies for fifth-generation (5G) cellular networks. In CRAN, the traditional base stations (BSs) are decoupled into remote radio heads (RRHs) and base band units (BBUs) that are respectively responsible for radio and baseband functionalities. The RRHs are geographically proximated whereas the the BBUs are pooled in a centralized cloud named BBU pool. This virtualized architecture facilitates the system to offer high computation and communication loads from the impetuous rise of mobile devices and applications. Heterogeneous service requests from the devices to different RRHs are now sent to the BBUs to process centrally. Meeting the baseband processing of heterogeneous requests while keeping their Quality-of-Service (QoS) requirements with the limited computational resources as well as enhancing service provider profit is a challenging multi-constraint problem. In this work, a multi-objective non-linear programming solution to the Quality-of-Experience (QoE) and Profit-aware Resource Allocation problem is developed which makes a trade-off in between the two. Two computationally viable scheduling algorithms, named First Fit Satisfaction and First Fit Profit algorithms, are developed to focus on maximization of user QoE and profit, respectively, while keeping the minimum requirement level for the other one. The simulation environment is built on a relevant simulation toolkit. The experimental results demonstrate that the proposed system outperforms state-of-the-art works well across the requests QoS, average waiting time, user QoE, and service provider profit.

Keywords: 5G; cloud radio access network; computing resource allocation; quality-of-experience; profit maximization

1. Introduction

The concept of next-generation cellular networks such as fifth-generation (5G) is becoming popular, since they can help to accommodate the indomitable increase of data traffic currently being experienced by the mobile network operators (MNOs) [1–3]. The leading MNOs are highly motivated to incorporate the virtualization concept of cloud computing in their networks, giving birth to the Cloud Radio Access Network (CRAN) [4]. In CRAN, the traditional base station (BS) is split into the radio frequency (RF) unit, referred to as the remote radio head (RRH) and the baseband processing unit (BBU) that provides the computational resources [4,5]. Although the RRHs are distributed over a wide geographic region located at each cell site, the BBUs are now pooled and moved into a centralized cloud named the BBU pool, where the RRHs and BBUs are connected via fronthaul link. The virtualization of Radio Access Networks (RANs) is acknowledged as one of the important use cases of network function virtualization (NFV). It is considered to be the best candidate solution for supporting the next generation mobile communication network (5G) [1,6], which embodies real-time cloud-computing, collaborative radio and centralized baseband processing features in cellular networks.

By virtualization, the computing resources in the BBU pool can be dynamically shared among all the cells in the network. However, due to the heterogeneity of the incoming service requests from the RRHs to BBU pool, mapping an appropriate computational resource in the BBU pool to a particular request is still an important research challenge for the next-generation CRAN systems. Since in the CRAN architecture, mobile operators are dependent on cloud service providers for computational resources, service providers' profit must also be taken into account. The maximization of user Quality-of-Experience (QoE) in the competitive market must also be addressed. In this paper, we have developed a resource allocation scheme for mapping heterogeneous requests from RRHs to BBUs so that user QoE and service providers' profit is maximized under optimal resource utilization. Figure 1 shows the CRAN service architecture and its components.

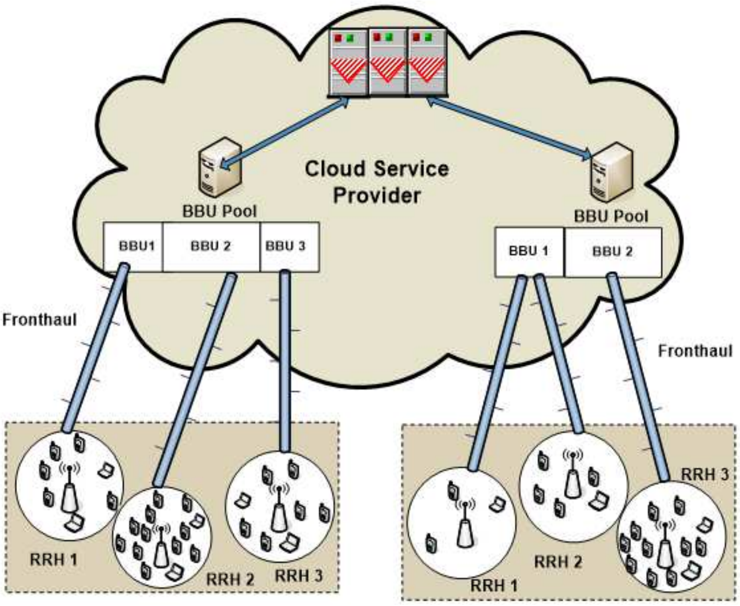


Figure 1. Fifth-generation (5G) Cloud Radio Access Network (CRAN) service architecture. BBU: baseband processing unit; RRH: remote radio head.

Resource allocation and the RRH-BBU mapping problem in 5G CRAN has been addressed in a number of research works in the literature [6–9]. However, focusing on the Quality-of-Service (QoS), Khan et al. [6] developed a dynamic RRH-BBU mapping algorithm in CRAN architecture where the service provider's profit is not a focus of attention. By employing the news-vendor game model, a resource allocation problem with a bargaining solution is investigated in [9] with the ability to reconfigure its resources with varying traffic conditions. This scheme requires additional time for resource reconfiguration, which can cause violation of QoS requirements. A renewable energy-based user association and power allocation is proposed in [10]. They have addressed the QoS requirement in terms of achievable rate, and completely disregard user QoE and service provider profit. Meeting up the QoS requirements of heterogeneous requests with the limited computational resources as well as enhancing service provider profit should be the prerequisite of a working CRAN environment.

Motivated by the aforementioned discussions, we have developed a QoE and Profit-Aware optimal Resource Allocation scheme in a 5G Cloud Radio Access Network. The novelty of this work lies in formulating a multi-constraint resource allocation problem so as to meet the heterogeneous user QoS requirements as well as to provide high profit to the operator. At first, weighted priority is calculated for each arrived task from RRHs to BBU pool so that the service requests can be scheduled according to their priorities and requirements. Here, the mapping problem of incoming service requests to the BBUs is formulated as a multi-objective non-linear programming (MONLP) optimization problem. Focusing on the QoE maximization of mobile operators as well as the subscribers and the service provider's profit, requests are selected in a scheduling slot and BBUs are allocated for the selected incoming requests. The main contributions of this paper can be summarized as:

- An integrated priority metric is developed so that the priority of an incoming request to a suitable BBU can be identified.
- Computational resource allocation problem for incoming requests is formulated as multi-objective non-linear programming optimization problem focusing on maximization of end-user QoE as well as service provider profit.
- Tradeoff between profit and customer satisfaction while selecting the BBUs for service provisioning in CRAN is made by two scheduling algorithms which are computationally viable to be deployed.
- To enhance system performance and resource utilization, the duration of the scheduling interval is determined dynamically according to the incoming requests and available resources.
- The results of our extensive simulation experiments, carried out on CloudSimSDN [11], depict that significant performance improvements in terms of user QoE, QoS satisfaction, average waiting time, and service provider profit have been achieved by the proposed system compared to the state-of-the-art works.

The rest of the paper is organized as follows. Section 2 describes some related works in our topic of interest. In Section 3, the system model and assumptions for execution environment of different requests are presented. The functional components of the proposed system architecture are described in detail in Section 4. In Section 5, the performance of our proposed scheduling system is analyzed, and Section 6 concludes the paper.

2. Related Works

CRAN technology is one suitable candidate for 5G systems; the advantages and challenges for various candidate architectures and their performance capacities are reviewed in [1]. Provisioning and allocation methods of virtual base stations (VBSs) in the base band unit (BBU) are proposed in [5]. The advantages, challenges, and future directions of CRAN technology are also studied in this survey paper. However, in [12], using integer linear programming (ILP) cells are optimally assigned to different BBU pools. It focuses on minimizing the capital expenditure (CAPEX) of CRAN deployment, although the operational expenditure (OPEX) is not taken into account. In [13], the Virtual Radio Access Networks (VRAN) Placement and Assignment Problem (VRAN-PAP) is formulated as a binary integer linear program (BILP) where the center of attention is only to minimize the server and front haul link setup cost rather than considering other cost issues.

Cooperative interference mitigation and handover management issues are addressed in [14], which did not analyze user QoS requirement and resource allocation schemes. The authors of [8] address the RRH to BBU assignment problem as a bin-packing problem which requires a significant amount of computational complexity, where the end-user QoE is not traced. In [6], even though QoS of requests are considered, service provider profit is not studied. In [15], the authors introduce a clustering mechanism to maximize user satisfaction and enhance network energy efficiency. Another noteworthy study [10] proposes a user association, and power allocation in mmWave-based ultra-dense bands is considered with load balancing and energy efficiency. Nevertheless, in both of the mentioned contributions, service provider profit in terms of executing a request on the cloud server is not discussed.

To exploit next-generation CRAN operations, a main challenging issue is how to properly control system resources. To address this, one article [9] employs the news-vendor game model for resource allocation. Here, the two-stage game-based resource management approach can practically adapt current system conditions. However, an optimal tradeoff between service provider profit and customer satisfaction are not concentrated here as well as in the state-of-the-art works.

In our proposed work, a QoE and Profit-Aware optimal Resource Allocation scheme in a Cloud Radio Access Network has been developed. Here, focusing on maximization of user QoE and service provider profit, the computational resource allocation problem for the incoming requests is formulated as a multi-objective non-linear programming optimization problem. While allocating resources, the amount of data to be processed, the maximum allowable service delay per request, and received signal strength of the connected device of the user are also taken into account for identifying the priority. In this work, a tradeoff is made between profit and customer satisfaction for service provisioning in 5G CRAN architecture. For efficient resource utilization and enhancement of system performance, the scheduling interval of the system is dynamically determined according to the request arrival rate and service rate. To the best of our knowledge, this work is the first to address the problem of maximizing user QoE as well as service provider profit under optimal resource utilization.

A comparative study among the state-of-the-art works focusing on different aspects of CRAN and our proposed Quality-of-Experience and Profit-aware Resource Allocation scheme or QEPRA system is given in Table 1. In the literature, several mechanisms [12,13,16] basically focus on infrastructure development of CRAN architecture. Another noteworthy paper [6] addresses QoS-aware scheduling and based on the service provider profit [9] allocates resources for incoming requests. However, our proposed system makes a tradeoff between user QoE as well as service provider profit while utilizing resources efficiently.

Table 1. Comparative study among the state-of-the-art works on CRAN. NvG: news vendor game-based resource allocation; QEPRA: Quality-of-Experience and Profit-aware Resource Allocation; QoE: Quality-of-Experience; QoS: Quality-of-Service; QoSM: QoS-aware dynamic BBU-RRH mapping; VBS: virtual base station; VRAN-PAP: Virtual Radio Access Networks Placement and Assignment Problem.

State-of-the-Art Works	QoS	User QoE	Profit	Resource Utilization
VBS Provisioning [12]			(Partially)	
VRAN-PAP [13]			(Partially)	
Fluidnet [16]				
RRH Clustering [15]	✓			
QoSM [6]	✓			
NvG [9]			✓	✓
QEPRA [Proposed]	✓	✓	✓	✓

3. System Model and Assumptions

In this section, the overall system architecture and the assumptions as well as the system components which make the allocation of computational resources for the incoming requests more efficient are well explored.

The 5G Cloud Radio Access Network (CRAN) refers to the virtualization of base station (BS) functionalities by means of cloud computing [17]. In this work, a fully centralized CRAN architecture [4] is assumed where all the baseband functions are assembled in BBUs (base band units) and the signals are centrally processed. The request management and resource allocations of BBUs of macro-cell and small-cell are co-located in the BBU pool. Remote radio heads (RRHs) integrate the radio functionalities where the RRHs and BBU pool communicate over CPRI (Common Public Radio Interface) protocol. We are considering that the overall system supports 2G, 3G, and 4G networks, and is designed for future 5G network.

In CRAN architecture, the mobile subscribers are directly connected to the RRHs of the mobile operators with heterogeneous service requests and Service Level Agreements (SLA), via supported mobile network communication protocols (e.g., GSM-Global System for Mobile communication, 3GPP-3rd Generation Partnership Project, LTE-Long Term Evolution, LTE Advanced, etc.). With the help of BBUs, residing in a BBU pool, the incoming workloads of the RRHs are processed. As a consequence, these computational resource requests are sent from RRHs to a *Request Receiver (RR)* located in the cloud service provider (CSP)'s side, as shown in Figure 2. The *RR* receives all the incoming requests with the corresponding service requirements. It then extracts the major attributes (data size, tolerable service delay, signal strength) of the requests from the requirements and sends those to the *Request Prioritizer (RP)*. The *RP* calculates the weight for each arrived request according to the requirements which denote the priority of each incoming request in a certain scheduling interval. The *RP* then sends the requests with their weights to the BBU Pool Manager which is responsible for managing the resource allocation process.

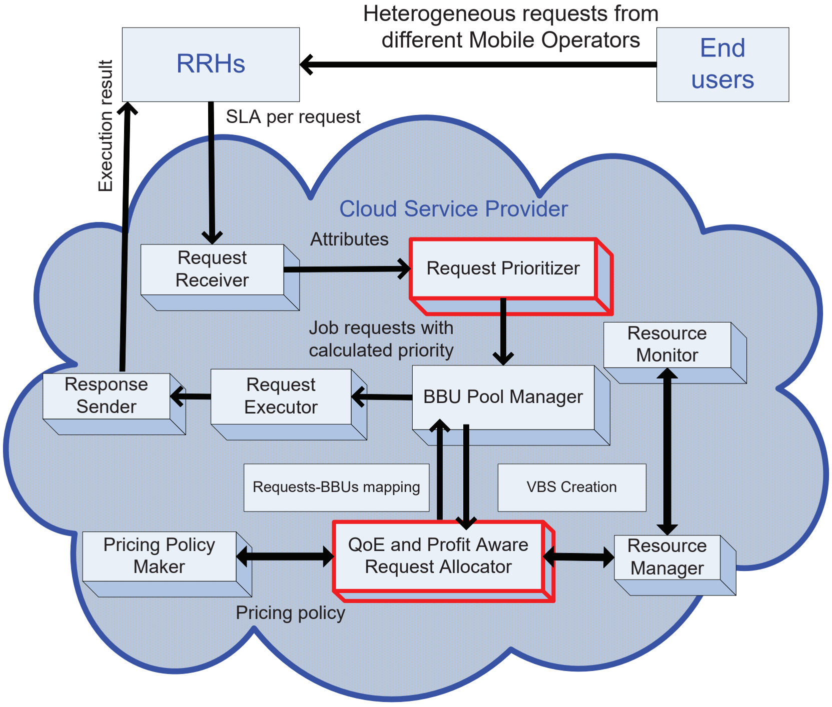


Figure 2. Proposed system model.

The optimal mapping of incoming requests to BBUs in a pool is done by the *QoE and Profit-aware Resource Allocator*. With the help of *Pricing Policy Maker* and *Resource Manager*, the *QoE and Profit-aware Resource Allocator* allocates BBUs for selected requests from the incoming requests in a scheduling interval in order to maximize the total user QoE and service provider profit. As limited computational resources are available for heterogeneous incoming requests with various requirements, these resources are allocated with the help of several scheduling intervals. The scheduling interval can be defined as in [18]. Here, the *Resource Monitor* monitors the incoming load and the *Resource Manager* creates appropriate virtual base stations (VBSs) according to the requirements. In addition, the *Pricing Policy Maker* provides the per-unit prices according to the work load and priority of incoming service requests. These unit prices are dynamically adjustable, as explained in [19,20]. After that, the *Request Executor* executes a particular request on the allocated computational resource, provided by the *BBU Pool Manager*. The execution result of each request is then sent via the *Response Sender* to the RRH. The relevant notations and definitions used for modeling the system are listed in Table 2.

Table 2. Notations.

Symbol	Definition
R	Set of all incoming computational resource requests
H	Set of all remote radio heads (RRHs)
B	Set of all base band units (BBUs) in a BBU Pool
Λ_r	Set of attributes of a request, $r \in R$
d_r	Incoming data of a request, $r \in R$ to be processed
q_r	The QoS requirement of an incoming request, $r \in R$
n_r	The received signal strength of the connected device associated with a request
w_r	The priority of an incoming request
O_r	Set of objective parameters considered for executing a request, $r \in R$
$\rho_{r,b}$	Cloud service providers' profit for executing a request, $r \in R$ on a BBU, $b \in B$
$\tau_{r,b}$	Total time requires a RRH, $h \in H$ to get first response from a BBU, $b \in B$ after executing a request, $r \in R$
$\Gamma_{r,b}$	The number of scheduling intervals required for a request, $r \in R$ to be assigned to BBU
$\overline{\Delta}_{s(i)}$	Scheduling interval of the system for allocating resources
h_b	BBU rental cost for executing a request
u_b	Monetary cost for other resource usage

4. Proposed Resource Allocation Scheme

In this section the working principle and functional methodologies of our proposed system are presented in detail. As the mechanisms used in the *Request Prioritizer (RP)* and the *QoE and Profit-aware Request Scheduler* greatly control the system performance, we address these two components in the whole system to enhance the efficiency. Here, the *Request Prioritizer (RP)* helps to identify the priority of the incoming service requests so that resources can be allocated for a request with higher priority first. The multi-constraint problem of resource allocation for the incoming service requests is formulated as a multi-objective non-linear programming (MONLP) optimization problem in the *QoE and Profit-aware Resource Allocator*. As this is an NP-Hard problem, two heuristic solutions are provided here considering a single objective at a time. Two algorithms called First Fit Satisfaction (FFS) algorithm focusing on the maximization of user QoE while maintaining the unit profit threshold and the First Fit Profit (FFP) algorithm targeting the maximization of service providers' profit while satisfying the QoS requirements are introduced here. As limited computational resources are available for heterogeneous incoming requests with various requirements, these resources are allocated with the help of several scheduling intervals in the whole system. To enhance the system performance, dynamic determination of scheduling intervals for resource allocation is also addressed here.

4.1. Incoming Request Prioritization

The proposed *Request Prioritizer (RP)* prioritizes the incoming requests on a certain scheduling interval based on the attribute values, extracted by the *Request Receiver (RR)*. We assume that the *Attribute Value Extractor* residing in the *Request Receiver (RR)* is capable of extracting the requirements from the incoming baseband processing requests, as exhibited in Figure 3. The considered attributes of a request, $r \in R$, which defines the QoE are:

- Amount of data to be processed, d_r
- The tolerable service delay, q_r
- Received signal strength of the connected device, n_r.

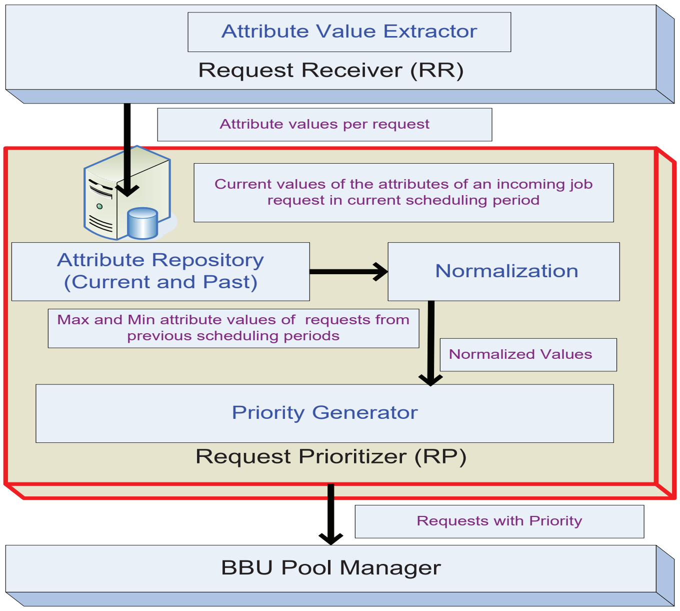

Attribute Value Extractor
Request Receiver (RR)
Attribute values per request
Current values of the attributes of an incoming job request in current scheduling period
Attribute Repository (Current and Past)
Normalization
Max and Min attribute values of requests from previous scheduling periods
Normalized Values
Priority Generator
Request Prioritizer (RP)
Requests with Priority
BBU Pool Manager

Figure 3. Request prioritization model.

The values acquired by the *Attribute Value Extractor* in the *Request Receiver (RR)* of each incoming request, $r \in R$, are sent to the *Attribute Repository*, which keeps the attribute values in reserve. As the values of the considered attributes differ in terms of norms and units, the *Normalization* function normalizes the values within a range in association with the *Attribute Repository*, so that the heterogeneity of the attributes can be alleviated. The normalized values of each attribute of a request are then fed into the *Priority Generator*, which enumerates the priority of the incoming requests in a specific scheduling interval. The resource requests are then sent along with the priority values to the *BBU Pool Manager*.

Let Λ_r denote the set of considered attributes of a request $r \in R$, where $\Lambda_r = \{d_r, q_r, n_r\}$. Each attribute $\Lambda_r^i \in \Lambda_r$ of a request $r \in R$ from the corresponding RRH, $h \in H$ can be scaled between 0 and 1 using the Min-Max normalization technique as in Equation (1):

$$\hat{\Lambda}_r^i = \frac{cur(\Lambda_r^i) - \min(\Lambda^i)}{\max(\Lambda^i) - \min(\Lambda^i)}, \tag{1}$$

where $cur(\Lambda_r^i)$ represents the current value of the corresponding attribute, $\Lambda_r^i \in \Lambda_r$ of one incoming request $r \in R$. In addition, $\max(\Lambda_r^i)$ and $\min(\Lambda_r^i)$ denote the maximum and minimum values of an attribute $\Lambda_r^i \in \Lambda_r$, which can be determined by the historical data analysis of the previous requests arrived on the previous scheduling periods stored in the *Attribute Repository*. The considered number of previous scheduling periods can be determined by the system based on the data. Using Equation (1) we get the normalized values of d_r, q_r, and n_r which are denoted as $\hat{d}_r$, $\hat{q}_r$, and $\hat{n}_r$, respectively. Now, the weight of each incoming request representing the priority can be calculated as:

$$w_r = \frac{\hat{d}_r}{\hat{n}_r + \hat{q}_r}. \tag{2}$$

The higher value of w_r denotes the higher priority of an incoming request within a specific scheduling interval. Here, we have generated the priority of each incoming service request based on the required data processing load, the connectivity of the device to the network, and the QoS requirement of the service request. This priority generation technique facilitates the faster execution of the requests with higher data rate requirement from a critical device in terms of signal strength and sending back the result within its QoS requirement. For example, if two requests arrive, one with poor connectivity and another with strong RSSI (Received signal strength indication) value, although they have same data rate requirement, our proposed system will give faster access of the BBU to the request with poor RSSI value. The same thing will happen to the request with the QoS requirement of less allowable delay to get response.

4.2. Optimal Problem Formulation

The computational resource allocation problem for the incoming requests from RRHs to BBUs is formulated as a multi-objective non-linear programming (MONLP) optimization problem, focusing on maximization of end-user Quality-of-Experience (QoE) as well as service provider profit.

In order to initiate a radio communication service, soon after receiving the service requests from RRHs, a BBU must perform a specific task for each request to commence the corresponding radio access communication. The outcome of such tasks appear to the end users as the respective initial responses to the service requests. In this phase of this work, we target this sort of computation at the BBU pool. After the initialization of the service, radio communication will continue spontaneously. Here, the transmission delay is considered as the time required to send the request and receive the result, which depends on the connected network. Now, let the total time to get first response or the amount of time from when a request is submitted until the first response is produced from a BBU, $b \in B$ to a RRH, $h \in H$ after executing a request, $r \in R$ can be defined as the summation of request processing time and waiting time, as shown in Equation (3):

$$\tau_{r,b} = \zeta_{r,b} + \varpi_r, \tag{3}$$

where $\zeta_{r,b}$ denotes the time required to process the corresponding submitted request and ϖ_r represents the waiting time or the time count when a request arrives to the request queue until it goes for service. The *QoE and Profit-aware Resource Allocator* always tries to allocate a specific BBU for a particular request, which can reduce the total response time to enhance the user's Quality-of-Experience (QoE).

Let the total profit achieved by the service provider for executing a request, $r \in R$, on a BBU, $b \in B$, $\rho_{r,b}$ be defined as follows,

$$\rho_{r,b} = (d_r \times p_{d_r} + w_r \times p_{w_r}) - (h_b + u_b), \tag{4}$$

where h_b and u_b denote the monetary cost for renting BBU servers and other resource usage for processing a request on the BBU. Let p_{d_r} represent price per request with data to be processed d_r for executing the corresponding resource request, $r \in R$, and p_{w_r} represents the additional unit price for executing a request with priority w_r, provided by the *Pricing Policy Maker*. Theses values can be reactively determined based on the resource requirements as addressed in [19,20]. Thus, the total profit gained by the service provider to allocate resource for requests can be calculated as in Equation (4). This value always ensures higher profit for the request with higher priority, as additional charge is required for executing a request with higher priority.

The objective of our proposed methodology is to allocate a BBU, $b \in B$ for each selected request, $r \in R$ of an RRH, $r \in R$ in such a way so that the total user QoE and the service provider's profit is maximized for executing all the selected service requests in a certain scheduling interval. Let O_r= $\{\rho_{r,b}, \tau_{r,b}\}$ be the set of objective parameters for executing a request r, which consists of service provider's profit and request response time. As these two values belong to different units, the profit and response time are scaled within a range. Each element $O^i \in O_r$ can be normalized as Equation (5), from which we get the normalized value of the service provider's profit, $\hat{\rho}_{r,b}$ and response time, $\hat{\tau}_{r,b}$ for executing an incoming request on a BBU:

$$\hat{O^i_r} = \frac{cur(O^i_r) - \min(O^i)}{\max(O^i) - \min(O^i)} \tag{5}$$

Here, $cur(O_r^i)$ represents the current value and the $\max(O^i)$ and $\min(O^i)$ are the maximum and minimum values of the corresponding objective parameters. Assuming that the execution process of any request, $r \in R$ is non-preemptive and all the virtual base stations (VBSs), $v \in V$, in a BBU will be created on demand of the incoming request to the BBU Pool. Computational resources are allocated for the incoming requests with the help of a certain number of scheduling intervals according to the available BBUs and resource requirements. Let $\Gamma_{r,b}$ denote the number of scheduling intervals required for a request, $r \in R$, to be assigned to a BBU, $b \in B$. This value is incremented by one as the scheduling interval number increases until a BBU is allocated for a request. The objective function of this resource allocation problem is formulated as:

$$\max_{r,b} \sum_{r \in R} \sum_{b \in B} (\alpha \times \hat{\rho}_{r,b} - \beta \times \hat{\tau}_{r,b} - \frac{\gamma}{\Gamma_{r,b}}), \tag{6}$$

which is subject to some constraints that are discussed elaborately in Equations (8)–(13). Equation (6) reveals that the total QoE offered by the system to the end user for all the requests will be maximized through minimization of the total time to get a response. It also ensures the earlier execution of the requests that promise higher profit to the service provider. Furthermore, we also minimize the starvation of any specific request by ensuring a higher chance of execution of requests with a higher number of waiting slots. Here the greater the difference of these considered values, the greater the maximization of Quality-of-Experience (QoE) and profit is achieved without having long starvation on the request queue. Here, α, β, and γ are the system parameters which can be dynamically changed according to the system environment satisfying Equation (7):

$$\alpha + \beta + \gamma = 1 \tag{7}$$

4.2.1. Constraints

The constraints of the aforementioned objective function can be listed as follows:

- **BBU Constraint:** The total number of BBUs in a pool must be constrained as

$$|B| \leq B_{max}, \tag{8}$$

  where $|B|$ denotes the total number of BBUs in a pool and B_{max} represents the threshold value of the maximum number of BBUs that can be pooled.

- **Capacity Constraint:** The capacity constraint represents that the sum of the processing capacities of the BBUs in a pool must be constrained by the total capacity of a BBU pool. This can be represented as

$$\sum_{b \in B} \eta_b \leq \eta, \tag{9}$$

  where η_b is the size of each BBU and η is the total capacity of a BBU pool.

- **Request Assignment Constraint:** It ensures that at a time, each request b $r \in R$ of one RRH $h \in H$ is always assigned to one BBU $b \in B$ of a BBU pool,

$$\sum_{r \in R} \xi_{r,b} \leq 1, \quad \forall r \in R, \forall b \in B, \tag{10}$$

  where $\xi_{r,b}$ is a Boolean variable, equal to 1 if a request $r \in R$ is assigned to a BBU $b \in B$ of a pool; otherwise, it is 0.

- **Virtual BBU Allocation Constraint:** The BBU allocation constraint defines that at a given time, one BBU will be allocated for one request, which is represented as

$$\sum_{b \in B} \Psi_{r,b} \leq 1, \quad \forall r \in R, \forall b \in B, \tag{11}$$

where $\Psi_{r,b}$ is a Boolean variable, equal to 1 if a BBU $b \in B$ of a BBU pool is allocated for a requested request $r \in R$; otherwise, it is 0.

- **Profit Constraint:** The profit constraint can be represented as

$$\rho_{r,b} \geq \rho_{min}, \tag{12}$$

where $\rho_{r,b}$ is the profit that is gained by the cloud service provider for executing a request, $r \in R$ on a BBU $b \in B$ of a BBU pool and ρ_{min} is the minimum profit that must be gained by the service provider for executing that request.

- **QoS Constraint:** The QoS constraint can be represented as

$$\tau_{r,b} \leq q_r, \tag{13}$$

where $\tau_{r,b}$ is the total time to get any response after executing a request $r \in R$ on a BBU $b \in B$ of a BBU pool, and q_r is the required QoS of a request or the maximum allowable time to obtain a result.

4.2.2. Computational Complexity of Resource Allocation Scheme

The Quality-of-Experience (QoE) and Profit-aware Resource Allocation or assignment of requests to the BBUs as formulated in Equation (6) is a multi-objective non-linear programming (MONLP) problem. Here the size of service request or data to be processed, d_r, as well as the processing capacity of BBU, η_b, vary from one to the other. Any BBU can be allocated to perform any request and processing time, and profit gain may vary depending on the request-BBU assignment. It is required to process all the selected requests by assigning exactly one request to each available BBU in such a way that the total profit of the assignment is maximized and the time delay to get first response is minimized. There exists a nonlinear relationship among the considered variables in Equation (6) and its constraints. Therefore, the resource allocation problem can be identified as a combinatorial optimization problem [21] and modeled as a generalized assignment problem (GAP) [22], which is a proved NP-Hard problem [23]. Thus, our optimal solution is proven to be an NP-Hard one.

4.3. Tradeoff between Customer Satisfaction and Service Provider Profit

Two first-fit greedy scheduling algorithms are invoked here for the system, when requests arrive in a scheduling interval; and the cloud service provider wants to maximize one objective parameter while maintaining the other one within a bounded value. In this work, the First Fit Satisfaction (FFS) algorithm is used to expand customer satisfaction while keeping a bound of unit profit, while the First Fit Profit (FFP) algorithm is proposed to escalate profit by affirming a target of customer satisfaction.

4.3.1. Satisfaction Optimization with a Profit Bound

This strategy is for service providers who aim to sustain a minimal unit profit ρ_{min} for each request. This value is predetermined by some market analysis by the service provider for each type of requested task [19]. Algorithm 1 summarizes the steps of the First Fit Satisfaction algorithm for user Quality-of-Experience (QoE) maximization. At first, the incoming requests are sorted in decreasing order according to priority, w_r, so that the request with the highest priority gets quicker access to the BBU. The target of this algorithm is to elect that BBU for each request, $r \in R$, which offers the fastest processing of one particular request. The faster execution or processing of a request bestowed by the BBU ensures a higher value of user satisfaction received by end user from a particular BBU. The BBU, Υ_b, providing the highest value of satisfaction, κ_r, is picked out to be provisioned for one request, $r \in R$. Therefore, the total time required to process a request, $r \in R$, on BBU, $b \in B$, is enumerated by the ratio of total data to be processed of a request, d_r, to the processing capacity of the BBU η_b as:

$$\zeta_{b,r} = \frac{d_r}{\eta_b} \tag{14}$$

Algorithm 1 First Fit Satisfaction Algorithm for Maximizing User Satisfaction

INPUT: Processing Capacity of all BBUs, $\eta_b, \forall b \in B$, priority of each incoming request, w_r, on a scheduling interval and QoS of the incoming requests.

```
Sort the incoming requests in decreasing order according to the value of $w_r$
for each incoming request $r \in R$ in the sorted array do
    $\kappa_r = 0$ and $\Upsilon_b = 0$
    for each available BBU $b \in B$ in the BBU pool do
        Calculate $\zeta_{b,r}$ using Equation (14)
        Calculate satisfaction of running a request on BBU b
        $\kappa_{b,r} = q_r - (\zeta_{b,r} + \varpi_r)$
        if $\kappa_r \leq \kappa_{b,r}$ then
            $\kappa_r = \kappa_{b,r}$
            $\Upsilon_b = b$
        end if
    end for
    Select BBU $\Upsilon_b$ for processing the request $r \in R$
end for
```

In this FFS algorithm, the BBU is selected focusing on maximizing the value of user satisfaction or user Quality-of-Experience (QoE). It searches for that BBU which can process earlier than the QoS requirement of a particular request. The BBU which provides the maximum distance from the value of QoS of a request, q_r to the value of processing time, ζ_b^r, is selected for a request. A higher value of this distance assures higher satisfaction.

4.3.2. Profit Optimization Under a Satisfaction Target

This strategy is for service providers who aim to maximize profit while maintaining a minimal satisfaction level that is the QoS of the request, q_r. Among all incoming requests from the RRHS to a BBU pool, that request will be scheduled or executed on the BBU $b \in B$ that ensures the maximum profit by using the computational resources available to the service provider by maintaining a minimum user satisfaction level.

In Algorithm 2, the incoming requests are sorted according to their priorities. After that, each request is assigned to the BBU, focusing on minimizing the cost and maximizing the profit. For that, to reduce the BBU rental cost in the current scheduling interval, this algorithm calculates the remaining time of a BBU to become free after processing the request which was scheduled on the most recent scheduling interval. So, the total time to get a response from a BBU $b \in B$ after executing a currently scheduled request, $r \in R$, on that BBU is enumerated as the time required by BBU to become free plus the execution time to execute that request. The time required a BBU to become free after finishing the execution of a running request is calculated as:

$$t_f^b = t_{r-1}^a + \zeta_{(r-1),b} - t^c, \tag{15}$$

where t^a_{r-1} denotes the time when the currently running request arrives, $\zeta_{(r-1),b}$ is the required time to process the request, and t^c represents the current time. If a BBU, $b \in B$, currently does not process any request, then the value of t^b_f is considered to be 0. As a consequence, the total time to get a response from a BBU is represented as:

$$\hat{\zeta}_{r,b} = t^b_f + \frac{d_r}{\eta_b} \tag{16}$$

where d_r is the request data to be processed, and η_b is the processing capacity of the BBU $b \in B$. If the value of $\hat{\zeta}_{b,r}$ is less than or equal to the corresponding requests QoS requirement, qos_r, then the request $r \in R$ is assigned to BBU $b \in B$ rather than renting another BBU. As a result, the BBU renting cost of BBU for this request $r \in R$, $\mu_{r,b}$ becomes 0. After that, using Equation (4), the profit gain from a particular request for executing on a BBU $b \in B$ is calculated. Therefore, the request providing highest profit to be executed on BBU $b \in B$ is selected to be provisioned.

Algorithm 2 First Fit Profit Algorithm for Maximizing Service Provider Profit

INPUT: Weighted priority of each incoming request, $w_r \in B$, processing capacity of all BBUs, $\eta_b, \forall b \in B$.

```
1:  Sort the incoming requests in decreasing order according to the value of $w_r$
2:  for each request $r \in R$ in the sorted array of requests do
3:      $\rho_r = 0$ and $\Upsilon_b = 0$
4:      for each BBU $b \in B$ in the sorted array do
5:          $h_b$ is the corresponding rental cost of the BBU, $b$
6:          Calculate $t^b_f$ and $\hat{\zeta}_{r,b}$ using Equation (15) and Equation (16), respectively
7:          if $\hat{\zeta}_{r,b} \leq q_r$ then
8:              $\zeta_r = \hat{\zeta}_{r,b}$ and $h_b = 0$
9:          else
10:             $\zeta_r = \frac{d_r}{\eta_b}$
11:         end if
12:         Calculate the profit $\rho_{r,b}$ using Equation (4)
13:         if $\rho_{r,b} \geq \rho_r$ then
14:             $\rho_r = \rho_{r,b}$
15:             $\Upsilon_b = b$
16:         end if
17:     end for
18:     Select request $r \in R$ to be executed on the BBU, $\Upsilon_b$
19: end for
```

The complexities of the proposed algorithms are quite straightforward. Firstly, in order to sort the set of incoming resource requests R in descending order in line 2 of both Algorithms 1 and 2, we use a merge sort algorithm which has the worst-case complexity of $O(|R| \log |R|)$. The statements in lines 5–12 in Algorithm 1 and lines 5–14 in Algorithm 2 are enclosed in a loop that iterates $|B|$ times. The rest of the statements have constant unit time complexities. Therefore, the worst-case computational complexity of the algorithms is $O(|R|^B)$.

5. Performance Evaluation

In this section, the efficacy of our proposed scheme is validated through simulation. The performance of the proposed QEPRA and First Fit Satisfaction (FFS) and First Fit Profit (FFP) algorithms are compared with some of the existing schemes in the literature. Here, the proposed system is assimilated with the NvG (news vendor game-based resource allocation) scheme [9] and QoSM (Quality-of-Service-aware dynamic BBU-RRH mapping) methodology [6].

5.1. Simulation Environment

The simulation environment of the Quality-of-Experience (QoE) and Profit-aware Resource Allocation framework in CRAN was designed using the CloudSimSDN [11] simulation toolkit. The simulated CRAN environment consists of one BBU pool with five BBUs. The computation speeds of the BBUs vary from 20 to 50 MHz. Through the simulation environment, it has also been imaged that 10 RRHs are geographically distributed to interact with end users directly and that they are supported by the BBUs of the BBU pool. The arrival pattern of the computational resource requests from the RRHs to the virtualized BBU pool is Poisson distributed, and the size of each request ranges from 20 to 600 KB. Heterogeneous attributes (d_r, q_r and n_r) associated with each request may have random values. Here, a random waypoint mobility model [24] is envisaged in which users with mobile wireless devices move independently to a randomly chosen destination with a random speed. The received signal strength or the RSSI values of the user devices vary due to their mobility pattern. The simulation is run for 500 s. Each data point in the graph corresponds to the mathematical average of the results from 50 simulation runs. In order to emulate a real CRAN system environment and for a fair comparison, application types, attributes, and system parameters are carefully selected for a realistic simulation scenario [25–27]. Table 3 shows the system parametric values used in our simulation.

Table 3. Simulation parameters.

Parameter	Value
Number of BBU	5
BBU processing speed	20∼50 MHz
Number of RRH	10
Incoming data per request to be processed	20∼600 KB
Maximum allowable delay (QoS)	20∼200 ms
RSSI value	−15∼−75 dB
Simulation Duration	500 s

5.2. Performance Matrices

The following performance matrices are evaluated for comparing our proposed QEPRA, FFS, and FFP algorithms along with two existing scheduling approaches (NvG, QoSM) in the literature.

5.2.1. Quality-of-Experience

Quality-of-Experience (QoE) is a measure of the overall level of user satisfaction, which can be increased by enhancing the gap between maximum allowable time to get a response (i.e., QoS) and the response time for a request [18]. We measured the average QoE for all requests, R, arriving during the simulation period as follows:

$$\overline{\kappa} = \frac{1}{|R|} \sum_{r \in R} \frac{q_r - (\varpi_r + \zeta_{r,b})}{q_r}, \tag{17}$$

where ϖ_r is the waiting time in the queue and $\zeta_{r,b}$ represents the required time to process a request, r, on a BBU, b. The greater the value of $\overline{\kappa}$, the better is the capability of the system to maximize user QoE.

5.2.2. Percentage of Requests Satisfying QoS

The percentage of requests satisfying QoS is calculated as the ratio of the successfully completed services to the total number of arrived requests. A higher value indicates that a higher number of service requests have been processed by the system satisfying the QoS requirements, hence higher user satisfaction.

$$q\hat{o}s_r = \frac{R^e}{R^a} \times 100\%, \tag{18}$$

where R^e denotes the number of QoS satisfied requests and R^a is the total number of requests.

5.2.3. Average Waiting Time

The waiting time of a request is the time count from its arrival to the request queue until it is assigned to a BBU in the BBU pool. The smaller the average waiting time of the incoming requests, the more quickly users will get a response from the system, which indicates greater user satisfaction. Let ϖ_r^a and ϖ_r^l denote the arrival time to the queue and leaving time from the queue of a request, respectively. The average waiting time of each request is defined as:

$$\overline{\varpi_r} = \frac{1}{|R|}\sum_{r\in R}\left(\varpi_r^a - \varpi_r^l\right), \tag{19}$$

where R is the set of all requests arrived during the whole simulation period. A lower $\overline{\varpi_r}$ value corresponds to better system performance.

5.2.4. Service Provider Profit

The profit gained by the service provider is calculated by the difference between the revenue gained and cost required to execute a request as in Equation (4), and then the average is taken for all requests arrived during the simulation period.

The average profit gained by the service provider for each request can be calculated as:

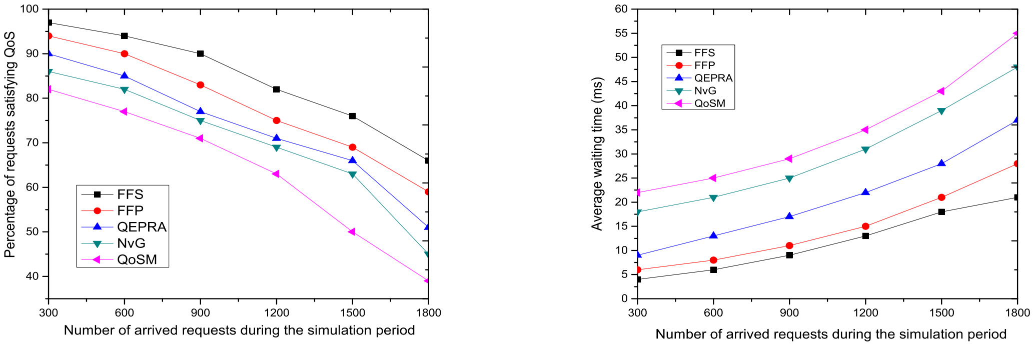
$$\overline{\rho}_{r,b} = \frac{1}{|R|}\sum_{r\in R}(\rho_{r,b}), \tag{20}$$

where $\rho_{r,b}$ represents the profit of processing a request gained by the service provider, and R is the total number of assigned requests. The higher the value of $\overline{\rho}_{r,b}$, the higher the profit obtained by the service provider for each request.

5.3. Results and Discussion

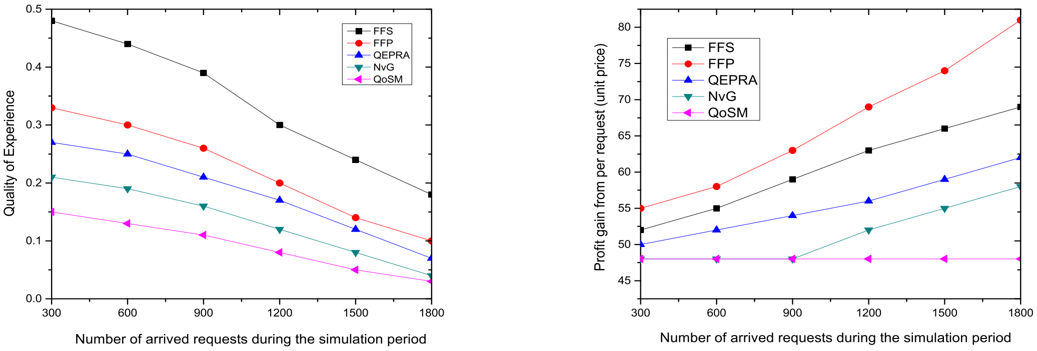
Assessment results attained by implementing the aforementioned simulation framework are described in this subsection. Here, the results and discussion are presented according to the impact of varying the number of requests, as well as the QoS requirements of requests.

The percentage of successfully executed service requests in terms of maintaining QoS level is enumerated here for varying number of arrived requests, and is delineated in Figure 4a. However, the graphs of Figure 4a illustrate that the percentage of successfully executed requests satisfying QoS level sharply declines in all of the studied resource allocation strategies with increasing arrived requests. The growing number of requests augments the waiting time for the requests, which imposes many of the service requests to overstep the maximum allowable delay.



(**a**) (**b**)

Figure 4. *Cont.*



(**c**) (**d**)

Figure 4. Impacts of varying number of arrived requests. (**a**) Percentage of requests satisfying QoS; (**b**) Average waiting time; (**c**) Quality-of-Experience of users; (**d**) Service provider profit. FFP: First Fit Profit; FFS: First Fit Satisfaction.

However, the rate of decrease is significantly less in the FFS algorithm, as this is especially designed for maximization of user Quality-of-Experience (QoE). In FFS, one BBU which can execute a request with the fastest processing time is allocated for a request. On the other hand, FFP targets on maximization of profit while maintaining the minimum QoS level. As in QEPRA, a multi-objective optimization problem is formulated, the proposed FFS algorithm outperforms QEPRA and FFP. Yet, the QEPRA provides a better result compared to the NvG and QoSM protocols because the proposed QEPRA technique prioritizes the scheduling of requests following their QoS constraints. The rationality for better performance of QEPRA compared to the other two reviewed schemes from the literature can be described by a scenario as, suppose that in the system three requests are being executed using the total system resources. Then, two other requests come one after another. The first request contains flexible QoS requirements, while the second is stringent in terms of QoS. As there is no priority based on the QoS requirements in the NvG approach, generally at first the system will adjust the total system resources for the first application request. After adjusting the request, it may find that the later request cannot be sent for execution since the required adjustment to allocate resources for that request cannot be made in the present system condition. In this circumstance, the QoS of the later service request will be violated. As the number of requests increases in the system, the rate of this type of QoS violation will increase. In this case, our proposed execution scheduler works significantly well, since it prioritizes the newly arrived requests based on their QoS requirements. Moreover, as the number of requests increases in the system, the genetic solution in QoSM consumes much more time than the proposed QEPRA and affects the QoS requirement of the service request. In addition, QEPRA makes a tradeoff between profit and QoE and FFP maintains minimum QoS level. On the other hand, FFS is particularly designed for QoE maximization and thus FFS provides the best result here.

5.3.1. Impacts of a Varying Number of Incoming Requests

The impacts of varying the number of incoming requests of the incoming service requests on the performance of our proposed algorithms are manifested in Figure 4.

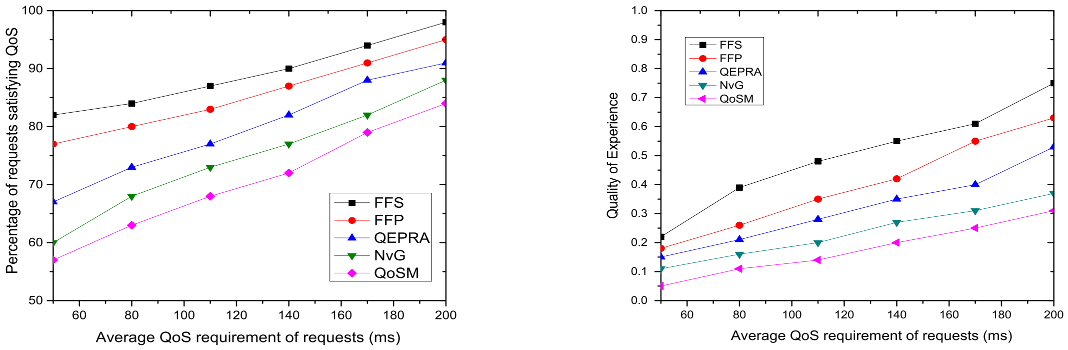
The graphs of Figure 4b illustrate that for all the approaches, the average waiting time of the requests will increase as the number of service requests increases. However, for the genetic-based approach (QoSM), it is quite high since the algorithm itself contributes to the waiting time of the requests. On the other hand, to adjust the resources for incoming requests in the news vendor game theory-based approach (NvG), some additional waiting time is observed by each request. Although our proposed approaches schedule the request in priority basis, it also ensures higher probability of the deferred requests to be scheduled in the subsequent scheduling intervals. Due to this consideration, average waiting of the requests in this approach is less compared to the existing works. Moreover, as in our proposed FFP algorithm, minimum QoS requirement is maintained, and waiting time is also reduced. As the FFS algorithm focuses on maximizing the time gap between the maximum allowable delay to get a response and summation of waiting time and processing time, among our proposed QEPRA, FFS, and FFP algorithms, FFS provides the best result in terms of reducing the average waiting time, as shown in Figure 4b.

The performance results of the proposed system in terms of Quality-of-Experience (QoE) are depicted in the graphs of Figure 4c. Our approaches execute the requests according to the priority of the requests which is based on their requirements. As a result, most of the requests get their results much before their deadline, which is eventually reflected in their QoE. As the FFS algorithm is especially designed for QoE maximization, it outperforms the other proposed algorithms as well as the existing works, as illustrated in Figure 4c.

A comparative study of the service provider profit for a varying number of arrived requests is provided in Figure 4d. In our approaches, on a scheduling interval, when an arrived request with higher scheduling priority executes in the system, the corresponding price of the service is also maximized. Our proposed FFP algorithm targets the maximization of service provider profit by reducing the resource rental cost. The per-unit profit gain also remains increasing as the number of incoming requests increases. QEPRA makes a tradeoff between profit and QoE; on the other hand, in the FFS algorithm, minimum per-unit profit is maintained and so this algorithm provides a better result than the others, except FFP. On the other hand, in the news vendor game-based approach, the scope of maximizing profit will be in its bargaining phase when the total service request is larger than the system capacity. As a result, the profit gain remains static for a certain number of requests in NvG. As QoSM does not address the service provider profit, the average per-unit profit gain always remains constant for the execution of an increasing number of requests.

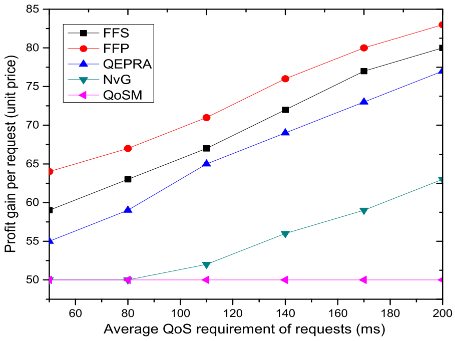
5.3.2. Impacts of Varying Average QoS Requirement per Request

The impacts of average QoS requirement of the incoming service requests on the performance of our proposed algorithms are manifested in Figure 5. As the average QoS requirements of the incoming requests increases, the maximum allowable delay to get the first response time becomes flexible. Hence, the QoS satisfaction percentage is increased in all approaches, as depicted in Figure 5a. However, as the FFS algorithm is specially designed for QoE maximization, it implicitly provides the best result among the studied approaches.



(**a**) (**b**)

Figure 5. *Cont.*



(**c**)

Figure 5. Impacts of varying average QoS requirements of requests. (**a**) Percentage of requests satisfying QoS; (**b**) Quality-of-Experience of users; (**c**) Service provider profit.

Moreover, as it relaxes the time boundary of performing baseband processing, there is a large possibility to execute within a minimal time regarding its QoS requirement. As a result, the QoE also keeps rising in all graphs in Figure 5b for our proposed algorithms as well as to the studied works. In addition, as the increase of average timeline QoS requirement offers a higher number of requests processing successfully, the service provider's profit gain also increases, as in Figure 5c.

Our in-depth look into the simulation trace file reveals that the considered simulation matrices give better performance value for our proposed system compared to the other two reviewed schemes. However, as the BBU pool has limited computational resources, the performance of the QEPRA as well as FFS and FFP decline with the increasing number of requests, which opens a new window for further developments.

6. Conclusions

In this work, a multi-objective non linear programming solution to the Quality-of-Experience (QoE) and Profit-aware Resource Allocation problem has been developed which makes a trade off between user QoE and service provider profit. The incoming requests have been prioritized based on their required data rates, QoS, and connectivity. Then, an optimal resource scheduling is developed. The dynamic determination of scheduling interval has helped to increase the performance significantly. The greedy FFP and FFS algorithms are proven to be computationally viable and has offered results near to optimal solution. The simulation results depict the effectiveness of our proposed system in terms of QoE, QoS, and profit margin. In the future, we would like to extend our simulation environment with an extensive performance analysis in contrast to the state-of-the-art works.

Acknowledgments: The authors extend their appreciation to the Deanship of Scientific Research at King Saud University for funding this work through research group no (RGP-281).

Author Contributions: Mahbuba Afrin and Md. Abdur Razzaque designed and developed the proposed idea. Iffat Anjum and Mohammad Mehedi Hassan designed and performed the experiments and finally Atif Alamri provided critical comments and help in writing the paper.

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

References

1. Miyanabe, K.; Suto, K.; Fadlullah, Z.M.; Nishiyama, H.; Kato, N.; Ujikawa, H.; Suzuki, K.I. A cloud radio access network with power over fiber toward 5G networks: QoE-guaranteed design and operation. *IEEE Wirel. Commun.* **2015**, *22*, 58–64.

2. Bassoli, R.; Di Renzo, M.; Granelli, F. Analytical energy-efficient planning of 5G cloud radio access network. In Proceedings of the 2017 IEEE International Conference on Communications (ICC), Paris, France, 21–25 May 2017; pp. 1–4.
3. Sigwele, T.; Alam, A.S.; Pillai, P.; Hu, Y.F. Energy-efficient cloud radio access networks by cloud based workload consolidation for 5G. *J. Netw. Comput. Appl.* **2017**, *78*, 1–8.
4. China Mobile Research Institute. *C-RAN The Road towards Green RAN, White Paper, Version 3.0*; Technical Report, China Mobile Research Institute: Beijing, China, 2013.
5. Pompili, D.; Hajisami, A.; Viswanathan, H. Dynamic Provisioning and Allocation in Cloud Radio Access Networks. *Ad Hoc Netw.* **2015**, *30*, 128–143.
6. Khan, M.; Alhumaima, R.S.; Al-Raweshidy, H.S. Quality of Service aware dynamic BBU-RRH mapping in Cloud Radio Access Network. In Proceedings of the 2015 International Conference on Emerging Technologies (ICET), Peshawar, Pakistan, 19–20 December 2015; pp. 1–5.
7. Karneyenka, U.; Mohta, K.; Moh, M. Location and Mobility Aware Resource Management for 5G Cloud Radio Access Networks. In Proceedings of the 2017 International Conference on High Performance Computing & Simulation (HPCS), Genoa, Italy, 17–21 July 2017; pp. 168–175.
8. Qian, M.; Hardjawana, W.; Shi, J.; Vucetic, B. Baseband Processing Units Virtualization for Cloud Radio Access Networks. *IEEE Wirel. Commun. Lett.* **2015**, *4*, 189–192.
9. Kim, S. News-vendor game-based resource allocation scheme for next-generation C-RAN systems. *EURASIP J. Wirel. Commun. Netw.* **2016**, *2016*, 158.
10. Zhang, H.; Huang, S.; Jiang, C.; Long, K.; Leung, V.C.M.; Poor, H.V. Energy Efficient User Association and Power Allocation in Millimeter-Wave-Based Ultra Dense Networks with Energy Harvesting Base Stations. *IEEE J. Sel. Areas Commun.* **2017**, *35*, 1936–1947.
11. Son, J.; Dastjerdi, A.V.; Calheiros, R.N.; Ji, X.; Yoon, Y.; Buyya, R. CloudSimSDN: Modeling and Simulation of Software-Defined Cloud Data Centers. In Proceedings of the 2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), Shenzhen, China, 4–7 May 2015; pp. 475–484.
12. Holm, H.; Checko, A.; Al-obaidi, R.; Christiansen, H. Optimal assignment of cells in C-RAN deployments with multiple BBU pools. In Proceedings of the 2015 European Conference on Networks and Communications (EuCNC), Paris, France, 29 June–2 July 2015; pp. 205–209.
13. Mijumbi, R. Placement and Assignment of Servers in Virtualized Radio Access Networks. *CoRR* **2015**, doi: 10.1109/CNSM.2015.7367390.
14. Zhang, H.; Jiang, C.; Cheng, J.; Leung, V.C.M. Cooperative interference mitigation and handover management for heterogeneous cloud small cell networks. *IEEE Wirel. Commun.* **2015**, *22*, 92–99.
15. Boulos, K.; Helou, M.E.; Lahoud, S. RRH clustering in cloud radio access networks. In Proceedings of the 2015 International Conference on Applied Research in Computer Science and Engineering (ICAR), Beirut, Lebanon, 8–9 October 2015; pp. 1–6.
16. Sundaresan, K.; Arslan, M.Y.; Singh, S.; Rangarajan, S.; Krishnamurthy, S.V. FluidNet: A Flexible Cloud-Based Radio Access Network for Small Cells. *IEEE/ACM Trans. Netw.* **2016**, *24*, 915–928.
17. Simeone, O.; Maeder, A.; Peng, M.; Sahin, O.; Yu, W. Cloud radio access network: Virtualizing wireless access for dense heterogeneous systems. *J. Commun. Netw.* **2016**, *18*, 135–149.
18. Mahmud, M.R.; Afrin, M.; Razzaque, M.A. Maximizing quality of experience through context-aware mobile application scheduling in cloudlet infrastructure. *Softw. Pract. Exp.* **2016**, *46*, 1525–1545.
19. Li, C.; Li, L.Y. Optimal resource provisioning for cloud computing environment. *J. Supercomput.* **2012**, *62*, 989–1022.
20. Wan, J.; Zhang, R.; Gui, X.; Xu, B. Reactive Pricing: An Adaptive Pricing Policy for Cloud Providers to Maximize Profit. *IEEE Trans. Netw. Serv. Manag.* **2016**, *13*, 941–953.
21. Wang, H.; Wang, J. An Effective Image Representation Method Using Kernel Classification. In Proceedings of the 2014 IEEE 26th International Conference on Tools with Artificial Intelligence, Limassol, Cyprus, 10–12 November 2014; pp. 853–858.
22. Öncan, T. A Survey of the Generalized Assignment Problem and Its Applications. *Inf. Syst. Oper. Res.* **2007**, *45*, 123–141.
23. Marshall, L.; Fisher, R.; Jaikumar, L.N.V.W. A Multiplier Adjustment Method for the Generalized Assignment Problem. *Manag. Sci.* **1986**, *32*, 1095–1103.

24. Broch, J.; Maltz, D.A.; Johnson, D.B.; Hu, Y.C.; Jetcheva, J. A Performance Comparison of Multi-hop Wireless Ad Hoc Network Routing Protocols. In Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '98), Dallas, TX, USA, 25–30 October 1998; pp. 85–97.
25. How Much Data (MB) Does Skype Consume in a 1 Minute Call? Available online: http://superuser.com/questions/703399/how-much-data-mb-does-skype-consume-in-a-1-minute-call (accessed on 27 May 2015).
26. Maximum Audio/Video Delay for Peer to Peer Communication. Available online: http://stackoverflow.com/questions/32879928/ (accessed on 11 November 2017).
27. What Is the Minimum RSSI Needed for 3G or LTE? Available online: https://www.linkedin.com/pulse/what-minimum-rssi-needed-3g-lte-andre-fourie (accessed on 14 December 2015).