Adaptive Multi-Resource Allocation for Cloudlet-Based Mobile Cloud Computing System

Yanchen Liu, Myung J. Lee, and Yanyan Zheng

Abstract—Mobile cloud computing utilizing cloudlet is an emerging technology to improve the quality of mobile services. In this paper, to better overcome the main bottlenecks of the computation capability of cloudlet and the wireless bandwidth between mobile devices and cloudlet, we consider the multi-resource allocation problem for the cloudlet environment with resource-intensive and latency-sensitive mobile applications. The proposed multi-resource allocation strategy enhances the quality of mobile cloud service, in terms of the system throughput (the number of admitted mobile applications) and the service latency. We formulate the resource allocation model as a semi-Markov decision process under the average cost criterion, and solve the optimization problem using linear programming technology. Through maximizing the long-term reward while meeting the system requirements of the request blocking probability and service time latency, an optimal resource allocation policy is calculated. From simulation result, it is indicated that the system adaptively adjusts the allocation policy about how much resource to allocate and whether to utilize the distant cloud according to the traffic of mobile service requests and the availability of the resource in the system. Our algorithm outperforms greedy admission control over a broad range of environments.

Index Terms—Mobile cloud computing, semi-Markov decision processing, joint resource allocation, admission control

1 Introduction

Mobile Cloud Computing (MCC) is a promising system that introduces powerful cloud computing into a mobile computing environment, where mobile devices connect to the Internet through wireless network and then communicate with the remote cloud [1]. Compared to mobile devices, the cloud server in MCC can provide huge storage, high computation power, as well as reliable security [2]. By offloading subcomponents of mobile application to the cloud server for execution [3], the performance of mobile applications can be greatly improved [4] and the energy consumption of mobile devices can be significantly reduced [5], [6]. Furthermore, MCC extends the variety of mobile applications that are too resource-intensive to execute solely on mobile devices, such as virus scanning [7].

However, for latency-sensitive mobile applications, such as augmented reality with real-time constraints, offloading to the remote cloud is insufficient, because of the high latency of Wide Area Networks (WAN). Cloudlet is proposed as an emerging paradigm to better support both latency-sensitive and resource-intensive mobile applications [8]. It is located at the middle entity of the three-tier hierarchy: mobile device, cloudlet, and cloud with functionalities of data routing and security guard, similar to a proxy in cloud computing system [9]. Additionally, cloudlet can speed up mobile application executions by providing powerful computing capabilities. Through providing a resource-rich server/cluster within the near vicinity of the mobile users, the requirement of real-time interactive response can be met by one-hop high-bandwidth wireless access to the cloudlet. Based on the mobile users request, one or more custom virtual machines (VMs) can be instantiated immediately on the cloudlet for remote execution of applications in a thin client fashion [10]. A cloudlet is usually set up at the public place, like shopping center, theater, office building, and assembly room to enable convenient access for mobile devices.

Compared to the conventional MCC, the efficiency of the cloudlet system is greatly improved by providing the powerful computing resource closer through one-hop wireless network. Whereas in reality, the computing resource of the cloudlet server cannot be treated as sufficient as the remote cloud cluster, and the wireless bandwidth that connects mobile devices and the cloudlet server is limited and raced. There is a high probability that the cloudlet runs out of resources so that no new mobile request can be admitted, when an excessive number of mobile users offloading their applications for execution at the cloudlet [11]. Especially, when the offloaded applications are mainly resource-intensive and latency-sensitive, such as interactive high-definition video gaming, the resource of computing capability and wireless bandwidth will become exhausted rapidly and the quality of service (QoS) will be seriously degraded. Therefore, the coordinated allocation of computing resource and wireless bandwidth is a critical issue, in order to improve the quality of experience from mobile users' point of view. In contrast to traditional network resource allocation problems, new challenges are brought in by the cloudlet-based MCC system. One is how to utilize the computing resource at both the cloudlet server and the distant cloud coordinately to fully exploit the system capabilities. The other is how to perform

- *Y. Liu and M.J. Lee are with the City College, The City University of New York, 160 Convent Avenue, New York, NY 10031.
E-mail: {yliu2, mlee}@ccny.cuny.edu.*
- *Y. Zheng is with Google, Inc., 1600 Amphitheatre Pkwy, Mountain View, CA 94043. E-mail: yanyanz@google.com.*

*Manuscript received 9 June 2014; revised 9 Dec. 2014; accepted 17 Nov. 2015. Date of publication 26 Nov. 2015; date of current version 29 Aug. 2016.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TMC.2015.2504091*

multi-resource allocation jointly considering both the wireless bandwidth and the computing resource.

In this paper, we propose a joint multi-resource allocation framework in the cloudlet system based on semi-Markov Decision Processing (SMDP) [12]. The objective of the framework is to obtain the optimal decision of computing and wireless bandwidth resource allocation among multiple mobile users in cloudlet-based mobile cloud computing environments, by maximizing the overall benefits of the whole system to consequently enhance the quality of service for mobile users (i.e., low service rejection probability and short service time).

The work of this paper is summarized as follows:

- We develop the system reward model for resource allocation with wireless bandwidth, and computing resource of both cloudlet and distant cloud. The reward model considers the system benefits or impacts in accepting or rejecting the new request of using resource according to the current system traffic, the availability of the system resources, and the QoS guarantee of mobile users.
- Based on the reward model, we propose a multi-resource allocation strategy, which can determine whether to accept a new mobile service request for execution at the cloudlet or the distant cloud. Furthermore, the strategy can adaptively determine the optimal amount of wireless bandwidth and computing resource to allocate to the accepted request, and thus achieve the optimal system benefits.
- We formulate the multi-resource allocation problem as a semi-Markov decision processing, which is solved as a linear programming problem using the lp solver tool. Our approach has the predictive ability of the future state. The predictive feature in our approach lies in the transition probabilities from the current state to all potential next states upon receiving a new request.
- In order to verify the efficiency of our proposed multi-resource allocation in the cloudlet based MCC system, we perform the simulations of two different greedy polices and compare their blocking probabilities and average service time with our proposed algorithm's. We also examine the impacts of various reward parameters in our model.

Our extensive performance simulations show that the proposed resource allocation mechanism provides a lower request rejection rate compared to those of greedy policies. At the same time, a short time latency of mobile service is also guaranteed. The proposed multi-resource allocation algorithm can be utilized in practice by being executed offline given the various traffic parameters, the amount of system resource, and the resource price in the reward model according to the importance of the related resources. The obtained allocation decisions can be tabled for search when the traffic information and availability of system resources are profiled in real-time.

The remainder of this paper is organized as follows. Related works are reviewed in Section 2. Section 3 describes the system model and assumptions, and Section 4 proposes the strategy of an SMDP-based multi-resource allocation for cloudlet-based MCC system. Section 5 presents and discusses the performance evaluation. Finally, concluding remarks and future work are provided in Section 6.

2 Related Work

One of the main branches of research on the efficiency of mobile cloud computing focuses on application partitioning and offloading. The mobile application is partitioned into multiple tasks/components, some of which are suitable for offloading executions. The offloaded computing tasks need to be chosen carefully according to the characteristics of the mobile application, the computing ability of the cloud, and the network condition [13]. In general, the complex tasks that require higher computing capability are offloaded to the remote cloud for execution while the less complex ones running on mobile devices [14], [15], [16]. We also proposed a dynamic programming based partitioning and offloading scheme [17] that can obtain the optimal partitioning much more quickly than traditional methods used in [14], [15].

Another important factor that determines the efficiency of mobile cloud computing is resource management, which is recently investigated in [18], [19], [20] from different aspects. In [18], several types of virtual machines are configured at the distant cloud to provide the service to different mobile users according to their requirements and the availabilities of computing resource. [19] presents how to manage the cloud resources across multiple cloud domains to support continuous cloud service. Then upon [19], [20] captures the dynamic arrivals and departures of resource requests for decision making of their resource allocation. However, these works only address the computing resource allocation of the distant cloud without the consideration of the wireless bandwidth resource, which is an indispensible factor in mobile computing environments. The frequent and heavy mobile service requests cause the shortage of wireless bandwidth, which further incurs the network congestions and thus seriously impacts the mobile application performance with long waiting time and the unstable status.

The allocations of computing and radio (bandwidth) resources are jointly considered in [21], [22] within the scheme of wireless base stations and cloud service providers, instead of the cloudlet. Liang et al. [21] propose a cooperation scheme among different service providers and obtains the maximum number of applications that can run under the shared computing and base station bandwidth resources. And [22] found a one-to-one relation between the transmit power allocated on each channel and the percentage of CPU cycles assigned to the corresponding application to minimize the power consumption at the mobile side. However, the bandwidth resources investigated in [21], [22] are from the wireless base station, not the high-speed WLANs connecting the mobile devices and the cloudlet server.

Cloudlet is proposed as a practical platform for accelerating mobile cloud computing [10]. Preprocessing, caching and scheduling approaches are studied by [23] for efficient usage of the powerful and resourceful cloudlet. A bandwidth-aware admission control policy is developed in [24] for cloudlet, where a mobile service is always assigned a fixed amount of system resource without the flexibility of adapting to request traffic and resource availability of the time-varying system. Furthermore, the ability of the distant cloud

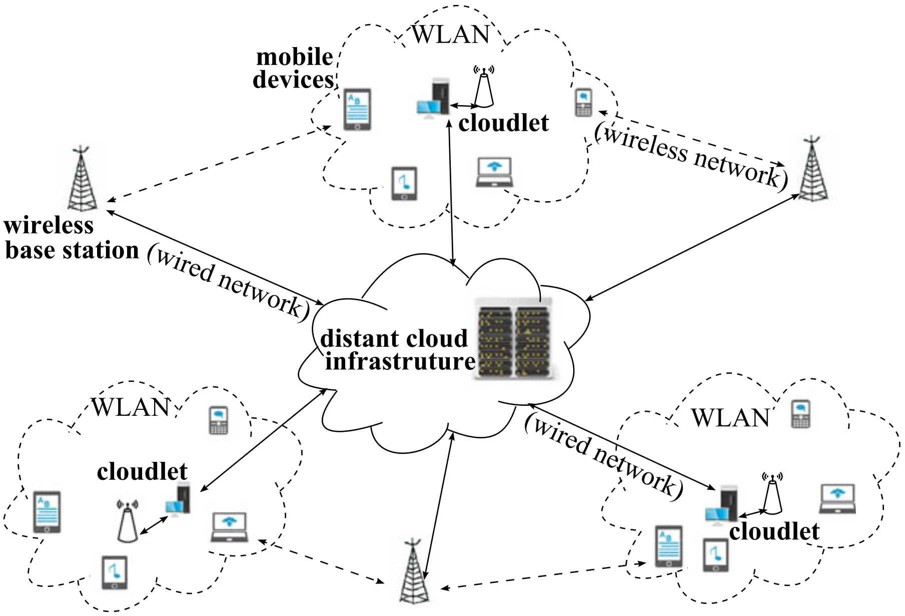


Fig. 1. The model of cloudlet-based mobile cloud computing system.

connected with the Internet is not taken into account in their studies. Xia et al. [25] address the problem of online request admission in a cloudlet with an objective of maximizing the system throughput for a certain time period, without the knowledge of future request arrival rate. In each time slot, requests are admitted/rejected one by one using a greedy strategy, according to the resources availability and admission cost which need to be updated after admitting a request.

The computing capability of the distant cloud is not considered in the models of [24] and [25]. What they studied are actually admission control problem, and they cannot adaptively adjust the allocated wireless bandwidth and computing resource of the cloudlet and the remote cloud for mobile service requests according to the availability of the current system resource. Our adaptive resource allocation approach can increase the allocated amount of wireless bandwidth and computing resource to speed up the application execution when the resource is sufficient, and to reserve the resource for future requests when the request traffic is heavy. To the best of our knowledge, none of the works in the published literatures addressed the issue of multi-resource allocation coordinately at both the local cloudlet and the distant cloud with the goal of maximizing the system revenue while meeting the requirements of QoS for mobile users.

3 System Model

3.1 System Description

We consider the cloudlet-based MCC environments as shown in Fig. 1. The cloudlet provides wireless LAN (WLAN) connections for mobile devices within its working range, and the distant cloud infrastructure is connected with cloudlets through the high-speed wired network. The mobile device can run mobile applications locally, or offload some workload to the cloudlet or to the distant cloud for faster execution. In our model, compared to the conventional way of accessing to the distant cloud through wireless base station (e.g., 4G or LTE), the mobile devices connect to the distant cloud through cloudlets as long as WLAN is available between the mobile device and cloudlet, which enables much faster data transmission and lower cost of access to the distant cloud. The base station connection can also be utilized only when the cloudlet is already being maximally used, or users are out of the cloudlet range.

Upon arrival of a mobile service request, the system decides whether to run it at the mobile device or offload it to the cloudlet according to the offloading decision that has already been made based on the network performance and the characteristics of the application [17]. For the task or subcomponent of a mobile application that is determined to be offloaded, a light-weight virtual machine [26] will be instantiated and assigned to it at either the cloudlet server or the remote cloud data center (distant cloud). In the meanwhile, the execution completion of the application can be sped up if more bandwidth or more VMs are allocated for data transmission or cloud computation, respectively [18].

The computing resource of the cloudlet is adequate for running multiple mobile applications simultaneously, but not as sufficient as that of the remote cloud data center, such as Amazon EC2 [27], Google Compute Engine [28], and Microsoft Azure [29], where the computing resource is always available as long as the mobile users purchase/subscribe the service. Thus, the number of available VMs at the cloudlet server is usually much less than that at the distant cloud data center, and there will be cases that the tasks offloaded to the cloudlet need to be further offloaded to the distant cloud data center via wired network because no more available VMs available at the cloudlet.

The wireless bandwidth resource in the model refers to the WLAN connections between the mobile devices and the cloudlet, and one wireless bandwidth unit refers to the minimum bandwidth required to support mobile computing offloading, for example, 50, 100 Kbps, etc. Then, the total bandwidth available can be expressed as the integer multiple of the bandwidth unit. For the manageability of the model computation, we assume a single mobile service requires at least one basic unit of WLAN bandwidth or channel, and only the integral numbers of basic bandwidth

TABLE 1
Notations

Name	Description
B	The total number of wireless bandwidth units provided by cloudlet
M	The number of VMs at cloudlet
W	The maximal number of wireless bandwidth units that the system provides to one service
T	The maximal number of cloudlet VMs that the system provides to one service
λ	The arrival rate of mobile request
μ_i^j	The departure rate of service that is allocated i units of wireless bandwidth resource and j cloudlet VMs
ν_i	The departure rate of service that is allocated i units of wireless bandwidth resource and distant cloud VMs
x_i^j	The number of ongoing service that is allocated i units of wireless bandwidth resource and j cloudlet VMs
y_i	The number of ongoing service that is allocated i units of wireless bandwidth resource and distant cloud VMs
A_n	The event of a new mobile request arrival
D_i^j	The departure event of the service that is allocated i units of wireless bandwidth resource and j cloudlet VMs
F_i	The departure event of the service that is allocated i units of wireless bandwidth resource and distant cloud VMs
a_i^j	The action to accept the request by allocating i wireless bandwidth and j cloudlet VMs
a_i	The action to accept the request by allocating i wireless bandwidth and distant cloud VMs
E_a	The income to accept a mobile service request
E_r	The penalty to reject a mobile service request
C_t	The cost of the time unit in service time
c_b	the cost rate of occupying bandwidth resource
c_e	the cost rate of occupying cloudlet computing resource
P_b	The blocking probability requirement of the mobile cloud computing system

units are allocated for wireless resource. Note that, a long time running service or continuously active service can be separated into multiple tasks/computing modules in our model. The tasks requiring low data transmission rate and high computing capability will be offloaded to the cloudlet, and each such task is considered as a mobile request. In this way, the power of mobile device can be saved and the application execution time shortened. For each task/computing module, the wireless bandwidth to transmit data and the computing resource to process data will not undergo obvious change over the relatively short period of task execution time. The minimum resource requirement of each task/request is thus relatively fixed.

In this model, the number of VMs that can be supported at the cloudlet is denoted as M, and the number of VMs at the distant cloud data center is assumed to be infinite compared with the small scale of the cloudlet. The number of wireless bandwidth units provided by the cloudlet is denoted as B. The minimum requirement for cloud computing is assumed to be one cloud VM and one wireless bandwidth unit. The notations used in this paper are summarized in Table 1.

3.2 Traffic Model

The arrival of mobile request for offloading application is assumed to follow a Poisson process with mean rate λ. If a mobile request is accepted by the cloudlet, then the service departure from the cloudlet is assumed to follow exponential distribution with rate μ_i^j,[1] and the mean service time at the cloudlet for this mobile request is $\frac{1}{\mu_i^j}$, where i ($i \in \{1, 2, \ldots, W\}$) stands for the allocated number of wireless bandwidth units and j ($j \in \{1, 2, \ldots, T\}$) the number of assigned VMs at the cloudlet server. Here, W is the maximal number of bandwidth units the system provides to one mobile request, while T is the maximal number of VMs allowed for one mobile request. On the other hand, if the system determines to offload the task to the distant cloud data center, T VMs will be assigned to fully support the fast execution of the mobile application. Then, the service departure from the distant cloud is assumed to follow exponential distribution with rate ν_i, and the mean service time at the distant cloud is $\frac{1}{\nu_i}$, where i is the number of allocated wireless bandwidth units. Note that, we assume the distant cloud can allocate the maximal number (T) of VMs that the cloudlet can provide to one mobile service, since data center at the distant cloud usually owns much more available server machines as long as the service is paid by mobile users.

3.3 Problem Statement

The decision making procedure of multi-resource allocation is described in Fig. 2. When a new request arrives, the system determines whether to accept it or not according to the current request traffic and the utilization of the wireless bandwidth and computing resources at the cloudlet. If the request is acceptable, the system will assign this new service request to the cloudlet or the distant cloud with a certain number of wireless resource and VMs of the cloudlet/distant cloud. The objective of our multi-resource allocation system for MCC is to make an optimal decision about whether to accept the mobile service request, and where to run the mobile application with how much allocated wireless bandwidth and computing resources if the request is accepted in order to maximize the system benefits and to guarantee the QoS of mobile users.

[1] Paper [30] studied the service time distribution with the experiments of nearly 200,000 processes from an academic environment. The process service time were split into short processes and long processes in terms of CPU and disk processing time. And it was found that short processes appeared to have nearly equal service times, while long processes appeared to have exponential service times. And for our model, the processes offloaded to the cloudlet or remote cloud server are mostly with the demand of high computing resources (i.e., long process) and thus the service time of tasks utilizing the same resources can be assumed to follow exponential distribution.

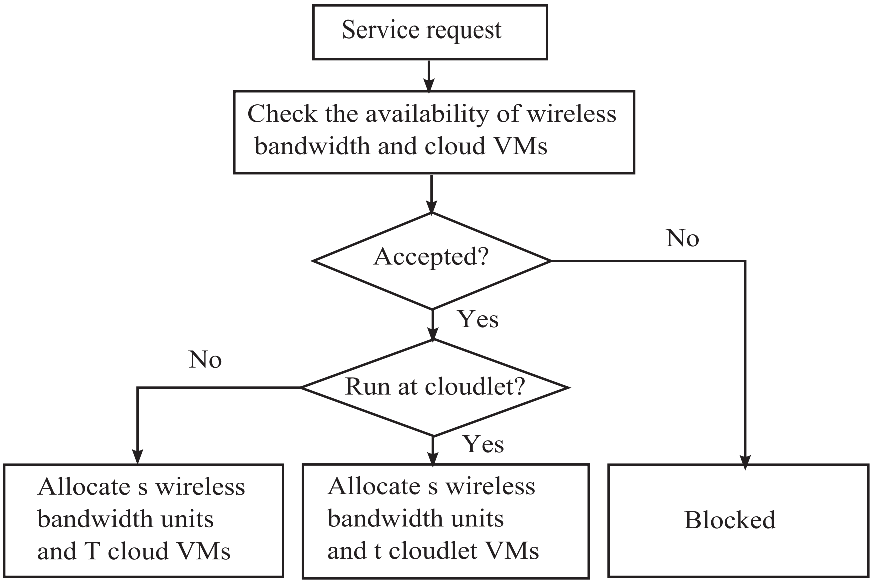

Service request
Check the availability of wireless bandwidth and cloud VMs
Accepted?
No
Yes
Run at cloudlet?
No
Yes
Allocate s wireless bandwidth units and T cloud VMs
Allocate s wireless bandwidth units and t cloudlet VMs
Blocked

Fig. 2. Multi-resource allocation in cloudlet-based mobile cloud computing system.

4 SMDP-Based Multi-Resource Allocation

SMDP is a generalization of the Markov Decision Process (MDP) [31], where the transition time between decision epochs is a continuous time random variable with the same probability distribution, which depends on: 1) the current system state, 2) the taken action, and 3) the potential next state. In our model, the multi-resource allocation problem in cloudlet-based MCC system is formulated as an infinite horizon optimal control of a finite-state SMDP under the average cost criterion [12]. At each state of the request arrival, the resource allocation decision is about whether to accept the service and how to allocate the bandwidth and computing resources. Different allocation decisions result in different next potential states, and thus the different system rewards. Among all possible decisions, the optimal policy is obtained by maximizing the long-term expected average system reward under the QoS requirements (e.g., low blocking probability and short service time).

4.1 State and Action

In the cloudlet-based MCC system, the total number of ongoing services occupying i wireless bandwidth units and j cloudlet VMs is denoted as x_i^j, while the total number of ongoing services occupying i wireless bandwidth units and the demanded computing resource at the distant cloud is denoted as y_i. The sum of the system wireless bandwidth units and cloudlet VMs being used by all the ongoing services should be equal to or less than the total bandwidth (B) and cloudlet VMs (M) the system can provide, respectively:

$$\sum_{j=1}^{T}\sum_{i=1}^{W}(ix_i^j)+\sum_{i=1}^{W}(iy_i)\leq B, \tag{1}$$

and

$$\sum_{j=1}^{T}\sum_{i=1}^{W}(jx_i^j)\leq M. \tag{2}$$

A_n represents a new request arrival event, while D_i^j and F_i stand for a departure event of a service occupying i bandwidth units and j cloudlet VMs, and a departure event of a service occupying i bandwidth units and T distant cloud VMs, respectively. An event in state s is defined as $e(s)\in\{A_n,D_i^j,F_i\}$ $(i\in\{1,2,\ldots,W\},j\in\{1,2,\ldots,T\})$.

The decision process is on a state space $\mathcal{S}$, where each state s $(s\in\mathcal{S})$ describes the numbers of ongoing services occupying various resources and the current event in the system, which is denoted by $s=[x_1^1,\ldots,x_W^T,y_1,\ldots y_W,e(s)]$.

At each state s (transition epoch), action a can be chosen from the set of allowable actions at state s $\mathcal{A}_s$. Let action space $\mathcal{A}=\bigcup_{s\in\mathcal{S}}\mathcal{A}_s:\{-1,0,a_i^j,a_i\}$ $(i\in\{1,2,\ldots,W\},j\in\{1,2,\ldots,T\})$, where -1 represents a service departure, 0 represents a rejection of new service request, a_i^j represents an action to accept the request by allocating i wireless bandwidth units and j cloudlet VMs, and a_i represents acceptance of the request by allocating i wireless bandwidth units and T distant cloud VMs. Note T is the maximum number of VMs allowed to allocate to one mobile service request.

The cumulative event rate $\gamma(s,a)$ is the sum of rates of all constituent processes from state s to others after selecting action a [31], and the expected sojourn time $\tau(s,a)$ is the average time duration from the current state s to others after selecting action a. For each possible combination of event $e(s)$ and selected action a, $\gamma(s,a)$ and $\tau(s,a)$ can be calculated as follows:

$$\gamma(s,a)=\begin{cases}\lambda+\sum_{j=1}^{T}\sum_{i=1}^{W}x_i^j\mu_i^j+\sum_{i=1}^{W}y_i\nu_i, & e(s)=A_n,\\ & a=0\\ \lambda+\sum_{j=1}^{T}\sum_{i=1}^{W}x_i^j\mu_i^j+\sum_{i=1}^{W}y_i\nu_i+\mu_w^t, & e(s)=A_n,\\ & a=a_w^t\\ \lambda+\sum_{j=1}^{T}\sum_{i=1}^{W}x_i^j\mu_i^j+\sum_{i=1}^{W}y_i\nu_i+\nu_w, & e(s)=A_n,\\ & a=a_w\\ \lambda+\sum_{j=1}^{T}\sum_{i=1}^{W}x_i^j\mu_i^j+\sum_{i=1}^{W}y_i\nu_i-\mu_w^t, & e(s)=D_w^t\\ \lambda+\sum_{j=1}^{T}\sum_{i=1}^{W}x_i^j\mu_i^j+\sum_{i=1}^{W}y_i\nu_i-\nu_w, & e(s)=F_w\end{cases} \tag{3}$$

and

$$\tau(s,a) = \frac{1}{\gamma(s,a)}, \tag{4}$$

where λ is the request arrival rate of service request, $\sum_{j=1}^{T}\sum_{i=1}^{W} x_i^j \mu_i^j$ represents the departure rates of the ongoing services utilizing the cloudlet resource, and $\sum_{i=1}^{W} y_i \nu_i$ represents the departure rates of services utilizing the distant cloud resource. With event A_n, if action a_w^t or a_w is selected, one more cloudlet or distance cloud service is admitted, which increases $\gamma(s,a)$ by rate μ_w^t or ν_w. In case of event D_w^t or F_w, one service is completed and the corresponding resource is released, which decreases $\gamma(s,a)$ by departure rate μ_w^t or ν_w, respectively.

4.2 State Transition Probability

The state transition probability $p(k|s,a)$ is defined as the probability that the system will be in state k at the next decision epoch, if action a is chosen at the current state s. There are three cases to consider depending on the event type A_n(new request arrival), D_w^t(service departure from cloudlet), and F_w(service departure from distant cloud) at the current state s, respectively.

4.2.1 If the Current State $s = [x_1^1, \ldots, x_1^T, x_W^1, \ldots, x_W^T, y_1, \ldots, y_W, A_n]$

For the current state with event A_n, the candidate action can be to reject the request or to allocate a certain number of system resources. According to the selected action and the next state, the transition probability $p(k|s,a)$ to the next state k can be given as:

$$p(k|s,a) = \begin{cases} \frac{\lambda}{\gamma(s,a)}, & e(k) = A_n \\ \frac{(x_i^j+1)\mu_i^j}{\gamma(s,a)}, & e(k) = D_i^j, a = a_i^j \\ \frac{x_i^j\mu_i^j}{\gamma(s,a)}, & e(k) = D_i^j, a \neq a_i^j \\ \frac{(y_i+1)\nu_i^j}{\gamma(s,a)}, & e(k) = F_i, a = a_i \\ \frac{y_i\nu_i}{\gamma(s,a)}, & e(k) = F_i, a \neq a_i. \end{cases} \tag{5}$$

(i) When the event $e(k)$ is a new request arrival A_n, the transition probability $p(k|s,a)$ equals the arrival rate λ over the total cumulative event rate $\gamma(s,a)$. (ii) When the event $e(k)$ is a service departure D_i^j using cloudlet resource, $p(k|s,a)$ equals the total departure rate over $\gamma(s,a)$. For the service that just accepted one more request by allocating i units wireless bandwidth and j cloudlet VMs at the current state s (i.e., $a = a_i^j$), the total departure rate is μ_i^j multiplied by the number of this type of ongoing service $(x_i^j + 1)$ accounting for the acceptance of one more request, which gives rise to the expression given in the second line of (5). For the other actions, $p(k|s,a)$ equals the total departure rates $x_i^j\mu_i^j$ over $\gamma(s,a)$. (iii) Similar to the case in (ii), the remaining two cases in (5) can be obtained when we consider the distant cloud instead of the cloudlet.

4.2.2 If the Current State $s = [x_1^1, \ldots, x_1^T, x_S^1, \ldots, x_S^T, y_1, \ldots, y_W, D_w^t]$

For the state with the event D_w^t, the action can only be -1, meaning a service departure. The transition probability is calculated as:

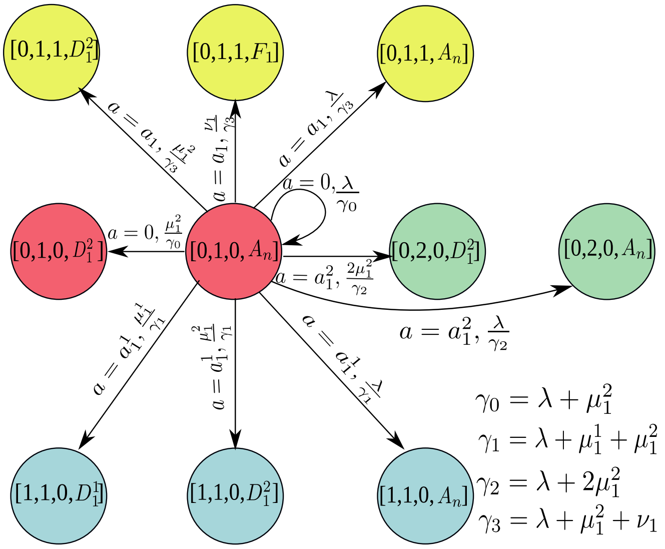

[0,1,1,D_1^2]
[0,1,1,F_1]
[0,1,1,A_n]
[0,1,0,D_1^2]
[0,1,0,A_n]
[0,2,0,D_1^2]
[0,2,0,A_n]
[1,1,0,D_1^1]
[1,1,0,D_1^2]
[1,1,0,A_n]
$a = a_1, \frac{\mu_1^2}{\gamma_3}$
$a = a_1, \frac{\nu_1}{\gamma_3}$
$a = a_1, \frac{\lambda}{\gamma_3}$
$a = 0, \frac{\lambda}{\gamma_0}$
$a = 0, \frac{\mu_1^2}{\gamma_0}$
$a = a_1^2, \frac{2\mu_1^2}{\gamma_2}$
$a = a_1^2, \frac{\lambda}{\gamma_2}$
$a = a_1^1, \frac{\mu_1^1}{\gamma_1}$
$a = a_1^1, \frac{\mu_1^2}{\gamma_1}$
$a = a_1^1, \frac{\lambda}{\gamma_1}$
$\gamma_0 = \lambda + \mu_1^2$
$\gamma_1 = \lambda + \mu_1^1 + \mu_1^2$
$\gamma_2 = \lambda + 2\mu_1^2$
$\gamma_3 = \lambda + \mu_1^2 + \nu_1$

Fig. 3. State transition diagram for state $[0, 1, 0, A_n]$.

$$p(k|s,a) = \begin{cases} \frac{\lambda}{\gamma(s,a)}, & e(k) = A_n \\ \frac{(x_i^j-1)\mu_i^j}{\gamma(s,a)}, & e(k) = D_i^j, D_i^j = D_w^t \\ \frac{x_i^j\mu_i^j}{\gamma(s,a)}, & e(k) = D_i^j, D_i^j \neq D_w^t \\ \frac{y_i\nu_i}{\gamma(s,a)}, & e(k) = F_i. \end{cases} \tag{6}$$

Since the current event is a service departure D_w^t, the number of service occupying w bandwidth units and t cloudlet VMs decreases by 1 in the next state k. For the next state with event A_n, $p(k|s,a)$ equals a new service arrival rate λ over $\gamma(s,a)$. For the next state with event D_i^j or F_i, $p(k|s,a)$ equals the departure rate of related service multiplied by the number of such type of ongoing services in new state k over $\gamma(s,a)$. Note that, for the service that just completed one task, the number of this type of ongoing services in k should be $(x_i^j - 1)$.

4.2.3 If the Current State $s = [x_1^1, \ldots, x_1^T, x_W^1, \ldots, x_W^T, y_1, \ldots, y_W, F_w]$

For the state with event F_w, a departure of the service occupying w bandwidth units and T distant cloud VMs, the action can only be chosen as -1. The transition probability is calculated as:

$$p(k|s,a) = \begin{cases} \frac{\lambda}{\gamma(s,a)}, & e(k) = A_n \\ \frac{x_i^j\mu_i^j}{\gamma(s,a)}, & e(k) = D_i^j \\ \frac{(y_i-1)\nu_i}{\gamma(s,a)}, & e(k) = F_i, F_i = F_w \\ \frac{y_i\nu_i}{\gamma(s,a)}, & e(k) = F_i, F_i \neq F_w. \end{cases} \tag{7}$$

With the transition probabilities calculated above, an SMDP chain can be built up for multi-resource allocation problem. Here, an example is given in order to display the details of the state transitions. In the example, W and T are respectively set to 1 and 2, respectively, and, therefore, the state s is expressed as $[x_1^1, x_1^2, y_1, e(s)]$. Figs. 3, 4, and 5 respectively presents all the state transitions along with the selected actions and the state transition probabilities starting from the state $[0, 1, 0, A_n]$, $[1, 1, 1, D_1^1]$, and $[1, 2, 1, F_1]$.

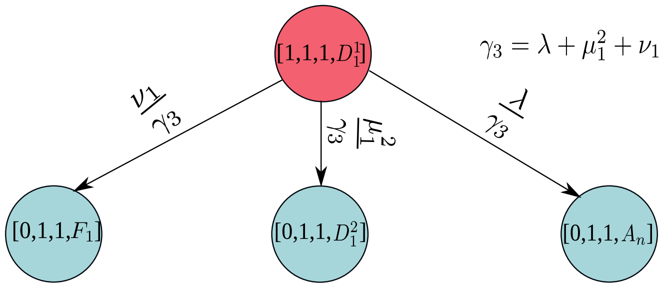

[1,1,1,D_1^1]
$\gamma_3 = \lambda + \mu_1^2 + \nu_1$
$\frac{\nu_1}{\gamma_3}$
$\frac{\mu_1^2}{\gamma_3}$
$\frac{\lambda}{\gamma_3}$
[0,1,1,F_1]
[0,1,1,D_1^2]
[0,1,1,A_n]

Fig. 4. State transition diagram for state $[1, 1, 1, D_1^1]$.

4.3 System Reward

In order to find the optimal resource allocation policy that maximizes the MCC system benefits, we define a real-valued function $r(s, a)$ as the system reward for selecting action a at state s. Following the definition of system reward in [12], $r(s, a)$ can be calculated as the sum of the lump income of decision making and the continuous cost of resource usage in our model:

$$r(s, a) = k(s, a) - \tau(s, a) \times o(s, a), \tag{8}$$

where $k(s, a)$ is the lump reward portion and $o(s, a)$ is the system cost per time unit for selecting action a at state s. In the definitions of $k(s, a)$ and $o(s, a)$, we consider the information including the mobile service request traffic, the usages of wireless and cloudlet computing resource, and the significance of accepting/rejecting one single request. The detailed definitions of $k(s, a)$ and $o(s, a)$ are described next.

$k(s, a)$ is defined as,

$$k(s, a) = \begin{cases} E_a - \frac{C_t}{\mu_i^j}, & a = a_i^j \\ E_a - \frac{C_t}{\nu_i}, & a = a_i \\ -E_r, & a = 0 \\ 0, & a = -1, \end{cases} \tag{9}$$

where E_a represents the income of accepting a mobile service request, and $-E_r$ represents the penalty of rejecting a service request. C_t denotes the cost per unit service time. Therefore, $\frac{C_t}{\mu_i^j}$ and $\frac{C_t}{\nu_i}$ respectively represent the cost of the mean service time incurred by accepting a new request with cloudlet and distant cloud computing resource. Considering the physical proximity and one-hop network latency of the cloudlet, utilizing the computing resource of the nearby cloudlet usually brings much shorter service time for the execution of offloaded mobile applications, than utilizing that of the distant cloud with same allocated bandwidth units and VMs [10]. Therefore, the system prefers to run the mobile tasks at the local cloudlet rather than the distant cloud, if cloudlet VMs are available.

o(s, a) describes the cost of occupying the wireless bandwidth and computing resource per unit time, which is defined as,

$$o(s, a) = c_b \left(\sum_{j=1}^{T} \sum_{i=1}^{W} (i x_i^j) + \sum_{i=1}^{W} (i y_i) \right) + c_e \left(\sum_{j=1}^{T} \sum_{i=1}^{W} (j x_i^j) \right), \tag{10}$$

where c_b denotes the cost rate of occupying bandwidth resource, and c_e denotes the cost rate of occupying the computing resource of cloudlet. They are both set as 1 by default in the model, and it is possible to make adjustments according to the prices of wireless resource and the cloudlet computational resource in reality. $o(s, a)$ is determined by these two rates of occupying resources and the number of being utilized resources after taking action a at s, and can be seen as the price of occupying the system resources given the current resource usage. The wireless bandwidth and computing resources will become more expensive as there are more mobile service requests coming. The computation cost at the distant cloud resource is neglected intentionally in the cost model, since its computing resource is too ample to be a resource bottleneck of the system.

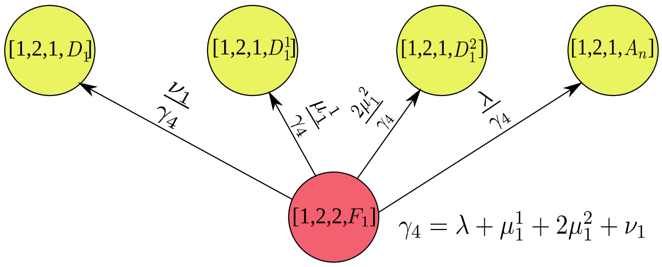

[1,2,1,D_1]
[1,2,1,D_1^1]
[1,2,1,D_1^2]
[1,2,1,A_n]
$\frac{\nu_1}{\gamma_4}$
$\frac{\mu_1^1}{\gamma_4}$
$\frac{2\mu_1^2}{\gamma_4}$
$\frac{\lambda}{\gamma_4}$
[1,2,2,F_1]
$\gamma_4 = \lambda + \mu_1^1 + 2\mu_1^2 + \nu_1$

Fig. 5. State transition diagram for state $[1, 2, 1, F_1]$.

4.4 Calculation of SMDP-Based Multi-Resource Allocation

The objective of the optimal multi-resource allocation is to maximize the average reward of the formulated SMDP model. According to [12], the proposed SMDP model belongs to the unichain case, and the optimization problem of maximizing the average reward can be formulated as below:

$$maximize \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}_s} (r(s, a) z(s, a)) \tag{11}$$

subject to the constraints:

$$\sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}_s} (\tau(s, a) z(s, a)) = 1 \tag{12a}$$

$$\sum_{a \in \mathcal{A}_k} z(k, a) - \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}_s} (p(k|s, a) z(s, a)) = 0, k \in \mathcal{S} \tag{12b}$$

$$z(s, a) \geq 0, s \in \mathcal{S} \; and \; a \in \mathcal{A}_s \tag{12c}$$

$$\sum_{s \in \mathcal{S}} \sum_{a=0} (\tau(s, a) z(s, a)) \leq P_b, \tag{12d}$$

where $z(s, a)$ represents the optimal basic solution at each state s. In the SMDP problem, $\tau(s, a) z(s, a)$ represents the long-run fraction of decision epochs at which the system is in state s and action a is selected [31]. Therefore, (12a) requires that the sum of all the fractions must be equal to 1. (12b) represents the balance equations requiring that for any state the long-run average number of transitions out of the state per time unit must be equal to the one into the state per time unit. (12c) restricts the optimal basic solution $z(s, a)$ is non-negative, and (12d) corresponds to the QoS requirement that the blocking probability of mobile service requests must be less than or equal to a constant P_b.

TABLE 2
System Default Parameters

E_a	E_r	C_t	P_b	λ	μ_1^1	μ_1^2	μ_2^1	μ_2^2	ν_1	ν_2	M	B
6.0	3.0	1.0	1.0×10^{-3}	12.0	3.0	4.0	5.0	6.0	3.5	5.5	10	10

The optimization problem of (11) and (12a)-(12d) can be solved as a linear programming problem of $z(s, a)$ [32]. Suppose an optimal feasible solution of $z(s, a)$ exists for each s and a ($a \in \mathcal{A}_s$), then the probability of selecting action a at state s, denoted by $p(s, a)$, can be calculated as:

$$p(s, a) = \frac{z(s, a)}{\sum_{a' \in \mathcal{A}_s} z(s, a')}. \tag{13}$$

An optimal policy is composed of all the probabilities of randomly selecting the actions, which can be calculated through (13) at every state of our SMDP model. The policy calculation can be executed and recorded offline, whose results are searched online with the real-time system inputs, such as the request traffics, the resource availability, and the blocking probability requirements of the request traffic.

5 Performance Evaluation

In this section, we evaluate the efficiency of the proposed SMDP-based multi-resource allocation strategy by investigating the simulation results of the obtained optimal policy (i.e., set of action selecting decisions), the blocking probability, the mean service time, and the system reward. The performance is evaluated under various mobile service request traffics, wireless network conditions and system model parameters. The advantages of the proposed multi-resource allocation algorithm are clearly revealed by comparing with two different kinds of greedy admission control methods. The simulations are written using MATLAB, in which a free API of lp_solve_5.5.2.0 [33] is embedded for solving the linear programming problem.

The system parameter assumptions in our simulation are: the maximum number of VMs that the cloudlet can provide to one mobile service request is two, and the number of wireless bandwidth units that the system can assign to one mobile service request is up to two. Hence, the state s in our model is in the format of [x_1^1, x_1^2, x_2^1, x_2^2, y_1, y_2, $e(s)$]. The default values of other parameters used in the system model are listed in Table 2. Here, the system reward parameter (E_a, E_r, and C_t) values having direct impact on the system reward are selected with the purpose of mimicking the ratio values that could be used in the real system. And if the weights of the parameters are considered differently by the system, the values should be set accordingly. For example, if the system considers the event of rejecting a mobile service request having much more negative impact, the value of E_r could be set higher (such as 10.0). Similar for P_b (0.001), which is the upper bound of blocking probability of the system, if the system does not consider the blocking probability that seriously, the optimization constraint (12d) can be relaxed by setting a higher value of P_b (such as 0.01). The default request arrival rate λ is set to a relative high value of 12.0. From our simulation we found, when λ is low (e.g., less than 6.0), the blocking probability approaches 0 (since the resource is too sufficient to refuse any service request). The heavy service request traffic helps to investigate and verify the efficiency of our approach under a simulation environment with insufficient resource.

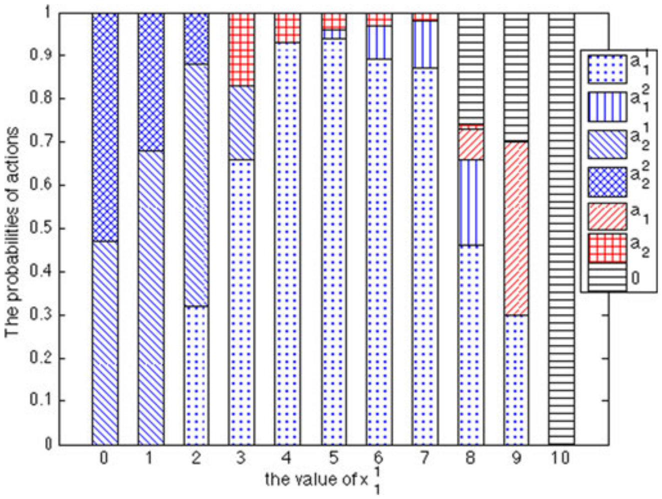


Fig. 6. The probabilities of actions against the value of x_1^1 for s [x_1^1,0,0,0,0,0,A_n] ($M = 10$, $B = 10$, $\lambda = 12$).

As the criteria to evaluate the QoS of the proposed resource allocation algorithm, the service request blocking probability ($P_{blocking}$) and the mean service time ($T_{service}$) are calculated as below to show their simulation values:

$$P_{blocking} = \frac{the\ No.\ of\ blocked\ new\ requests}{the\ No.\ of\ total\ new\ requests} \tag{14}$$

and

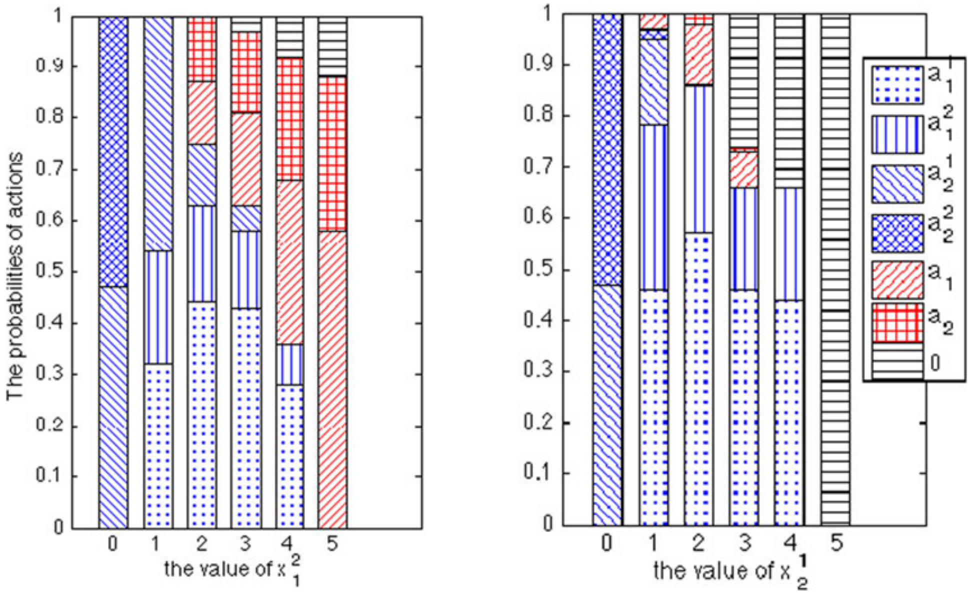
$$T_{service} = \frac{the\ total\ service\ time}{the\ No.\ of\ accepted\ requests}, \tag{15}$$

where all the statistical data is the average of ten times simulation results and each simulation lasts 10,000 seconds.

5.1 Optimal Policies

The optimal resource allocation policy for the cloudlet based MCC system is calculated as described in Section 4 with the default values of system parameters given in Table 2. The optimal policy comprises of all the optimal actions chosen at their corresponding states, which are calculated according to (11) using linear programming. Below we will show the selected actions of the proposed resource allocation strategy under three types of simple but representative states.

For state $s = [x_1^1$,0,0,0,0,0,$A_n]$, the probabilities of choosing action a at state s are presented in Fig. 6, where x_1^1 varies from 0 to 10 and the combination sum of probabilities of selecting any action is 1. For instance, when x_1^1 is 3, it means there are three ongoing services, each of which is using 1 cloudlet VM and 1 wireless bandwidth unit. Since the system has totally 10 cloudlet VMs and 10 wireless bandwidth units resource, state [3,0,0,0,0,0,A_n] indicates that 30 percent of the wireless bandwidth and 30 percent cloudlet computing resource are



(a) (b)

Fig. 7. The probabilities of actions against the values of x_1^2 for s [0,x_1^2,0,0,0,0,A_n] and x_2^1 for s [0,0,x_2^1,0,0,0,A_n] (M =10, B = 10, $\lambda = 12$).

being occupied with x_1^1 equal to three and all other state equal to zeros, as there comes a new service request arrival event A_n. It can be observed from Fig. 6 that, for a new mobile request, the probability of selecting action a_1^1 is 65, 20 percent for action a_2^1(using two wireless bandwidth units and one cloudlet VM), 15 percent for action a_2 (using two wireless bandwidth units and distant cloud VMs).

The results indicate that, when there are sufficient system resources available (i.e., when x_1^1 is less than 30 percent), the system prefers to select action a_2^1 or a_2^2. As the resource utilization increases, the probability of choosing action a_1^1 increases with decreasing probability of selecting a_2^1 and a_2^2. When both of the usages of wireless bandwidth and cloudlet VMs approach 90 percent, the probability of using distant cloud resource becomes much larger. At the same time, the probability of request rejection increases in order to reserve some resource for the possible coming requests. These actions are automatically chosen in the way to maximize the system reward (high throughput and low service time) under the QoS requirement (low service rejection rate).

And when state s = [0,x_1^2,0,0,0,0,A_n], the simulation results of $p(s,a)$ is shown in Fig. 7a. As x_1^2 increases from 0 to 5, representing the number of ongoing service occupying one wireless bandwidth unit and two cloudlet VMs is increasing, the probabilities of choosing action a_1 and a_2 increase, which means more mobile requests are accepted by allocating distant cloud VMs instead of the close cloudlet VMs. The reason of this trend is that, as x_1^2 becomes larger, relatively less cloudlet computing resource is available than wireless bandwidth, and thus the optimal choice tends to choose the distant cloud for cloud computing while reserving some cloudlet resource for possible future high-priority tasks. On the other hand, Fig. 7b shows the simulation results when the system state s = [0,0,x_2^1,0,0,0,A_n], as x_2^1 becomes larger, the optimal decision tends to use one wireless bandwidth unit rather than two because of the shortage of wireless resource. And when the value of x_2^1 is 5 (i.e., all the wireless bandwidth units assigned), the system no longer has the capability to accept the mobile service request, and thus the optimal action chosen at that moment will be rejection.

5.2 Impacts of Request Traffics and System Parameters

Different optimal policies are achieved under various service request arrival rates. Fig. 8 shows the probabilities of all possible actions for the sample state [0,0,0,0,0,0,A_n] with arrival rate increasing from 10 to 14 and 10 units of wireless bandwidth plus 10 cloudlet VMs available. When the request arrival rate is 10.0, the probability of action a_2^2 is 1, which means the system will assign 2 units of bandwidth and two cloudlet VMs for a new arriving request with 100 percent. As the request arrival rate increases, the probability of choosing a_2^2 decreases, while the probability of choosing

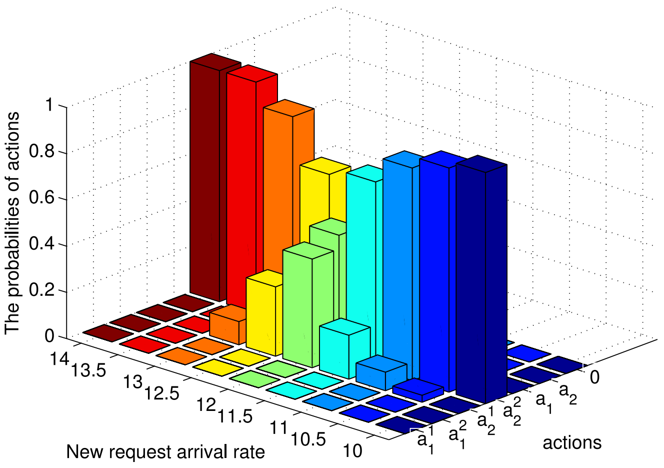


Fig. 8. The probabilities of actions against the request arrival rate for s [0,0,0,0,0,0,A_n] ($M = 10$, $B = 10$).

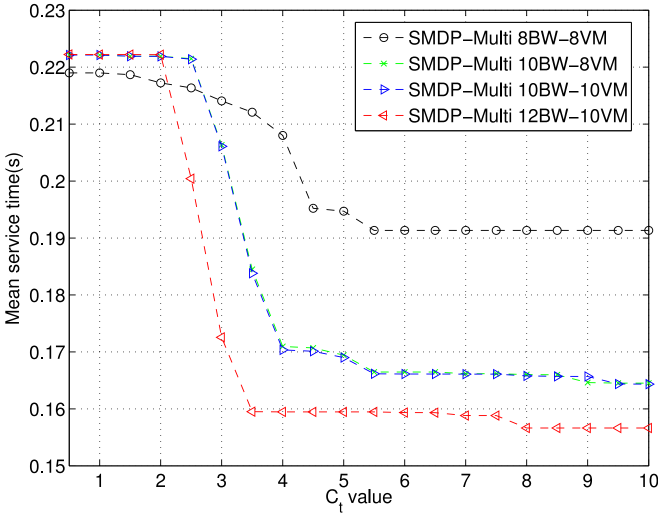


Fig. 9. The time latency under various C_t value ($\lambda = 12$, $P_b = 5.0 \times 10^{-3}$).

action a_2^1 increases. When the arrival rate becomes as high as 12, the probability of choosing a_1 starts to increase, and the probability of choosing a_2^1 decreases, which means the system starts to utilize the distant cloud resource. When the arrival rate becomes 14, the system will choose one bandwidth unit and distant cloud VMs for the new arrival service with 100 percent in order to save more cloudlet VMs for possible future requests.

From this experiment we can derive three features of our multi-resource allocation approach. (i) The system prefers to assign cloudlet VMs for the new request when the request traffic is low in order to guarantee short service time for mobile applications. (ii) As the request traffic increases, the system allocates less cloudlet VMs for the new service request. When the request traffic becomes heavy enough, the system moves more tasks from cloudlet to the distant cloud in order to reserve more cloudlet resource for the possible coming new request. (iii) More wireless bandwidth resource is assigned to the users when the request traffic is light. On the other hand, less bandwidth resource is assigned to the request when the traffic is heavy. That makes sense because the heavy traffic means the probability of the arrival is high. In order to satisfy more mobile users requests, some bandwidth should be reserved for the possible future requests by assigning less bandwidth to the current ones.

Next, C_t, the cost per unit service time, is adjusted to show its impact to the blocking probability ($P_{blocking}$) and the mean service time ($T_{service}$) of the system under four different scenarios of system resource: eight wireless bandwidth units and eight cloudlet VMs; 10 wireless bandwidth units and eight cloudlet VMs; 10 wireless bandwidth units and 10 cloudlet VMs; and 12 wireless bandwidth units and 10 cloudlet VMs. As shown in Fig. 9, $T_{service}$ is reduced when C_t is set larger for different scenarios because, according to (8) and (9), higher value of C_t means higher penalty of time latency and thus lower system reward. On the other hand, as shown in Fig. 10, the constraint of $P_{blocking}$ has to be relaxed in order to maximize the system reward, but is still guaranteed to be under the upper bound (5.0×10^{-3}) to satisfy QoS requirement.

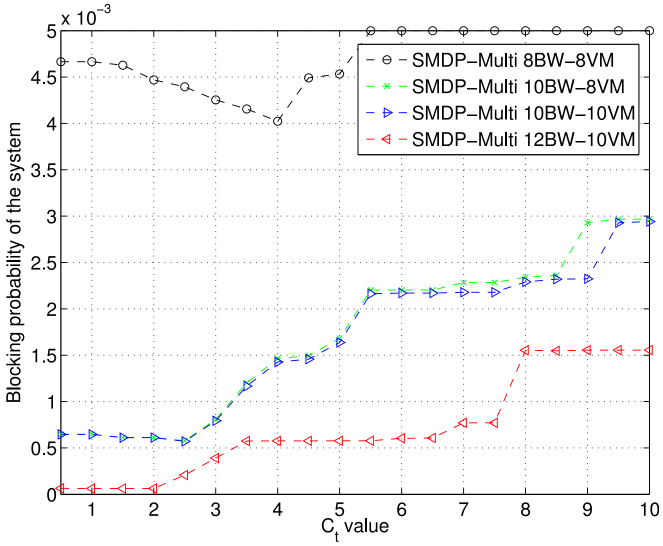


Fig. 10. The blocking probability under various C_t value ($\lambda = 12$, P_b = 5.0×10^{-3}).

Therefore, when mobile applications are more sensitive to the service time latency, the requirement can be met through setting a higher value of C_t; in contrary, to achieve higher system throughput (lower blocking probability), lower value of C_t is preferred. Besides, from Figs. 9 and 10, it can be clearly found that the system performance, in terms of mean service time and blocking probability, is improved as the available system resources increase.

5.3 Performance Comparisons

In order to verify the performance of our SMDP-based multi-resource allocation strategy, $P_{blocking}$ and $T_{service}$ of our approach are compared with the ones of the Greedy policy [34], which is typically used for admission control in wireless networks. Here, two types of Greedy policies are simulated for cloudlet-based MCC systems as follows:

- Once the bandwidth and the cloudlet VMs are available, they are allocated to a new arriving request; Note that, as defined in Section 3, we assume the minimum requirement of any mobile request is one wireless bandwidth unit and one cloudlet or distant cloud VM.
- In Greedy-1, the minimum bandwidth and minimum number of cloudlet VMs required by a single service are assigned whenever there are resources available. In Greedy-2, the maximum number of wireless bandwidth units (W) and the maximum number of cloudlet VMs (T) the system can provide are allocated to a single service request if they are available.
- If available, the cloudlet computing resource is always the first to be allocated; otherwise, the distant cloud is utilized to serve the mobile request.
- The greedy resource allocation methods are programmed using MATLAB by event driven method. And all the simulation results are obtained with the same system parameters as used in our proposed algorithm.

The simulations are repeated 10 times by which the average values and standard deviations are obtained for blocking probabilities and service time of our proposed

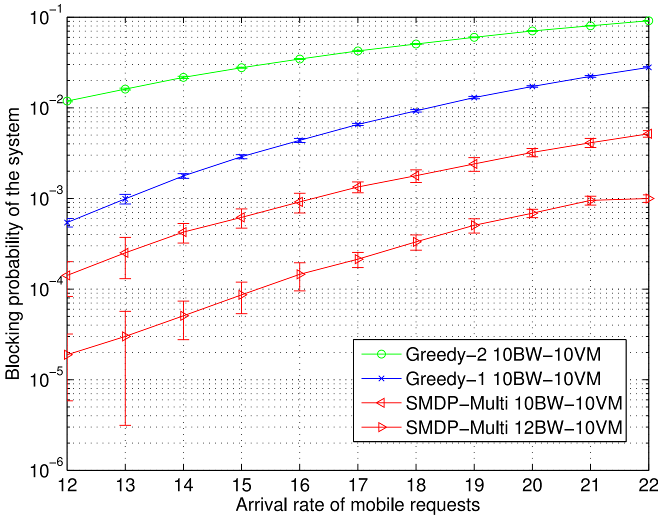


Fig. 11. The blocking probability under various request arrival rates.

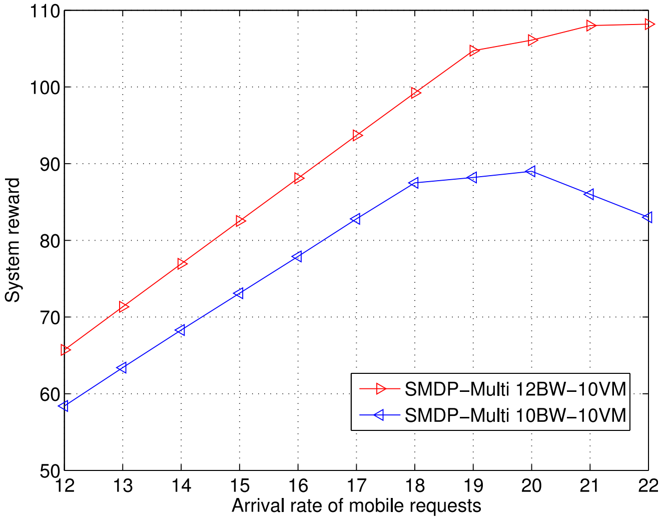


Fig. 13. The system reward under various request arrival rates.

strategy and two different Greedy approaches. The 95 percent confidence intervals are also obtained based on 10-time simulation results and attached in the corresponding figures. The simulation results of $P_{blocking}$ as calculated according to (14) are shown in Fig. 11 under various service request arrival rates. It is obvious that our approach shows much better performance than the Greedy methods when there are 10 units of wireless bandwidth and 10 cloudlet VMs available in the system (10BW-10VM). It is also shown that for the case when the number of available wireless bandwidth units increases to 12 (12 BW-10 VM), the proposed resource allocation strategy results in much lower blocking probability even with high service request arrival rate.

Similarly, Fig. 12 presents the simulation results of $T_{service}$ as calculated according to (15) along with 95 percent confidence interval. $T_{service}$ keeps the same value of 0.33 and 0.17 seconds respectively for Greedy-1 and Greedy-2 as the arrival rate λ varies. The proposed resource allocation strategy can guarantee $T_{service}$ less than 0.2 seconds initially with low arrival rate, and then becomes larger (around 0.22) as the arrival rate of the new request continues increasing. Although it is indicated that $T_{service}$ in our approach does not perform as well as Greedy-2's, it is totally worthy especially in a service congested environment to reduce $P_{blocking}$ around 90 percent with a slight expense in $T_{service}$.

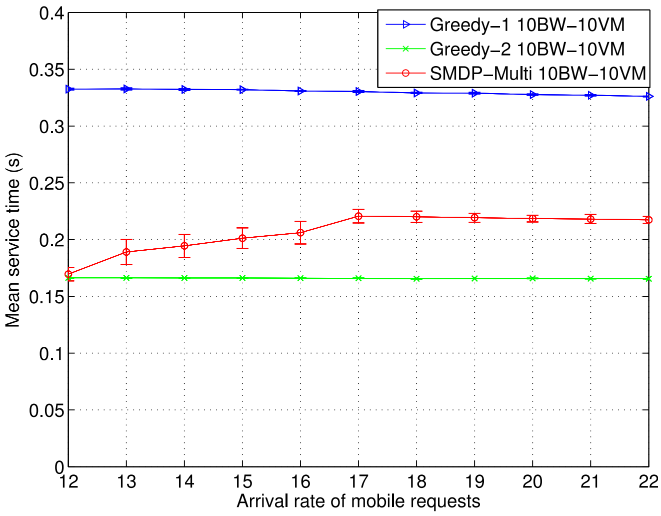


Fig. 12. The mean service time under various request arrival rates.

The system reward $r(s,a)$ computed based on (8) is measured against the arrival rate of mobile requests under two scenarios of system resource availability. As shown in Fig. 13, the system rewards increase linearly as the service request traffic becomes heavy until the arrival rate reaches around 19. Interestingly, the slope of $r(s,a)$ becomes smaller as the traffic rate becomes larger than 19 for the case that there are 12 wireless bandwidth units and 10 cloudlet VMs available (12 BW-10 VM). On the other hand, the slope of $r(s,a)$ decreases when the traffic rate is larger than 20 for the case that there are 10 wireless bandwidth units and 10 cloudlet VMs available (10 BW-10 VM). The changing trend of system reward is resulted from that the system can achieve larger reward by accepting more requests when the service request arrival rate becomes bigger and the system resource are ample. However, once the cloudlet computing resource is no longer enough for the quickly coming service requests, the system starts to use the computing resource of the distant cloud, which introduces more latency for the mobile application and thus higher penalty to the system reward. With even more service requests coming at a given time, there will be more requests that got rejected, and the negative influence applied to the total reward will result in reward declining.

6 Conclusion and Future Work

The ability to provide cloud service is critical for modern mobile cloud computing system [35]. In this paper, we present a novel multi-resource allocation approach for cloudlet-based Mobile Cloud Computing system. According to the current traffic of mobile requests and the availability of the system resource, our algorithm allows the system to adaptively allocate an optimal amount of allocated wireless bandwidth, cloudlet computing resource, and distant cloud computing resource for cloud computing of mobile applications. As the service request from mobile user increases, the approach ensures the high quality of cloud service is not affected much by efficiently using both local cloudlet and distant cloud resource. Furthermore, we also show how to adjust the built reward model for service latency sensitive application to meet various system requirements.

Regarding the service request from mobile users, we are using the Poisson process as many others adopted. We intend to continue to study further resource allocation problem in MCC when real traffic data is available, for instance, a proper mathematical model or data based like histogram based model. Another future work is to investigate the admission control approach with the consideration of user mobility. In a physically larger venue with multi-cloudlet setup, mobile service transfer for interdomain will be a significant factor. How to allocate the resource of a cluster of cloudlets considering the handoff caused by the user mobility is an interesting topic.

Acknowledgments

This research was supported in part by the US National Science Foundation (NSF) I/UCRC grant (No. 1160872).

References

[1] H. Rim, S. Kim, Y. Kim, and H. Han, "Transparent method offloading for slim execution," in *Proc. Int. Symp. Wireless Pervasive Comput.*, Jan. 2006, pp. 1–6.

[2] Y. Liu and M. J. Lee, "Security-aware resource allocation for mobile cloud computing systems," in *Proc. IEEE Int. Conf. Comput. Commun. Netw.*, Aug. 2015, pp. 1–8.

[3] Y. Liu and M. J. Lee, "An adaptive resource allocation algorithm for partitioned services in mobile cloud computing," in *Proc. IEEE Symp. Service-Oriented Syst. Eng.*, Mar. 2015, pp. 209–215.

[4] H. Wu, Q. Wang, and K. Wolter, "Tradeoff between performance improvement and energy saving in mobile cloud offloading systems," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2013, pp. 728–732.

[5] K. Yang, S. Ou, and H. H. Chen, "On effective offloading services for Resource-constrained mobile devices running heavier mobile Internet applications," *IEEE Commun. Mag.*, vol. 46, no. 1, pp. 56–63, Jan. 2008.

[6] C. Xian, Y. Lu, and Z. Li, "Adaptive computation offloading for energy conservation on battery-powered systems," in *Proc. Int. Conf. Parallel Distrib. Syst.*, 2007, pp. 1–8.

[7] T. Verbelen, P. Simoens, F. D. Turck, and B. Dhoedt, "Cloudlets: Bringing the cloud to the mobile user," in *Proc. 3rd ACM Workshop Mobile Cloud Comput. Services*, 2012, pp. 29–36.

[8] M. Satyanarayanan, G. Lewis, E. Morris, S. Simanta, J. Boleng, and K. Ha, "The role of cloudlets in hostile environments," *IEEE Pervasive Comput.*, vol. 12, no. 4, pp. 40–49, Oct.–Dec. 2013.

[9] J. Weissman and S. Ramakrishnan, "Using proxies to accelerate cloud applications," in *Proc. Conf. Hot Topics Cloud Comput.*, 2009, pp. 1–5.

[10] M. Satyanarayanan, P. Bahl, R. Caceres, and M. Davies, " The case for VM-based cloudlets in mobile computing," *IEEE Pervasive Comput.*, vol. 8, no. 4, pp. 14–23, Oct.–Dec. 2009.

[11] S. Clinch, J. Harkes, A. Friday, N. Davies and M. Satyanarayanan, "How close is close enough? Understanding the role of cloudlets in supporting display appropriation by mobile users," in *Proc. IEEE Int. Conf. Pervasive Comput. and Commun.*, 2012, pp. 19–23.

[12] M. Puterman, "Model formulation," in *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York, NY USA: Wiley, 2005.

[13] K. Kumar and Y.-H. Lu, "Cloud computing for mobile users: Can offloading computation save energy?" *IEEE Comput.*, vol. 43, no. 4, pp. 51–56, Apr. 2010.

[14] B. G. Chun, and S. Ihm, "CloneCloud: Elastic execution between mobile device and cloud," in *Proc. 6th Conf. Comput. Syst.*, Apr. 2011, pp. 301–314.

[15] E. Cuervo, A. Balasubramanian, and D.-K. Cho, "MAUI: Making smartphones last longer with code offload," in *Proc. 8th Int. Conf. Mobile Syst.*, Jun. 2010, pp. 49–62.

[16] M. S. Gordon, D. A. Jamshidi, S. Mahlke, Z. M. Mao, and X. Chen, "COMET: Code offload by migrating execution transparently," *Proc. 10th USENIX Conf. Operating Syst. Des. Implementation*, Oct. 2012, pp. 93–106.

[17] Y. Liu and M. J. Lee, "An effective dynamic programming offloading algorithm in mobile cloud computing system," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Apr. 2014, pp. 1891–1895.

[18] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, "Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 945–953.

[19] H. Liang, L.X. Cai, D. Huang, X. Shen, and D. Peng, "A SMDP-based service model for Inter-domain resource allocation in mobile cloud networks," *IEEE Trans. Veh. Technol.*, vol. 61, no. 5, pp. 2222–2232, Jun. 2012.

[20] H. Liang, T. Xing, L. X. Cai, D. Huang, D. Peng, and Y. Liu, "Adaptive computing resource allocation for mobile cloud computing," *Int. J. Distrib. Sensor Networks*, vol. 2013, Apr. 2013.

[21] R. Kaewpuang, D. Niyato, P. Wang, and E. Hossain, "A framework for cooperative resource management in mobile cloud computing," *IEEE Trans. Sel. Areas Commun.*, vol. 31, no. 12, pp. 2685–2700, Dec. 2013.

[22] P. D. Lorenzo, S. Barbarossa, and S. Sardellitti, "Joint optimization of radio resources and code partitioning in mobile cloud computing," *CoRR*, 2013.

[23] H. Wang, "Accelerating mobile-cloud computing using a cloudlet," M.S. thesis, Dept. Elect. Comput. Eng., Univ. Rochester, Rochester, NY, USA, 2013.

[24] D. T. Hoang, D. Niyato, and P. Wang, "Optimal admission control policy for mobile cloud computing hotspot with cloudlet," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Apr. 2012, pp. 3145–3149.

[25] Q. Xia, W. Liang, and W. Xu, "Throughput maximization for online request admissions in mobile cloudlets," in *Proc. IEEE 38th Conf. Local Comput. Netw.*, 2013, pp. 589–596.

[26] Y. Li, W. Li, and C. Jiang, "A survey of virtual machine system: Current technology and future trends," in *Proc. Int. Symp. Electron. Commerce Security*, Jul. 2010, pp. 332–336.

[27] (2015). *Amazon EC2* [Online]. Available: http://aws.amazon.com/ec2/

[28] (2015). *Google Compute Engine* [Online]. Available: https://cloud.google.com/products/compute-engine/

[29] (2015). *Microsoft Windows Azure* [Online]. Available: https://www.windowsazure.com/en-us/

[30] C. G. Rommel, "The probability of load balancing success in a homogeneous network," *IEEE Trans. Softw. Eng.*, vol. 17, no. 9, pp. 922–933, Sep. 1991.

[31] H. C. Tijms, "Semi-Markov decision processes," in *A First Course in Stochastic Models*, Amsterdam, The Netherlands: Wiley, 2003.

[32] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, "Linear programming and the simplex method," in *The Art of Scientific Computing*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 1992, pp. 423–436.

[33] (2015). *lp_solve 5.5.2.0* [Online]. Available: http://lpsolve.sourceforge.net/5.5/

[34] T. Javidi and D. Teneketzis, "An approach to connection admission control in single-hop multiservice wireless networks with QoS requirements," *IEEE Trans. Veh. Technol.*, vol. 52, no. 4, pp. 1110–1124, Jul. 2003.

[35] K. Kumar, J. Liu, Y. H. Lu, and B. Bhargava, "A survey of computation offloading for mobile systems," *Mobile Netw. Appl.*, vol. 18, no. 1, pp. 129–140, Feb. 2013.

Yanchen liu received the BS and MS degrees in 2004 and 2009, respectively, from National University of Defense Technology in China. He is currently working toward the PhD degree in the Department of Electrical Engineering, City University of New York. His recent research interests are in the area of resource allocation and management in wireless networks, cloud computing and mobile cloud computing, database management systems, data processing, and distributed systems.

Myung J. Lee received the BS and MS degrees from Seoul National University in Korea and the PhD degree from Columbia University in electrical/electronics engineering. He is currently a professor in the Department of Electrical and Computer Engineering City, University of New York. He is also an adjunct professor at GIST. His recent research interests include wireless sensor networks, ad hoc networks, mobile cloud computing, VANET, Security, and Vehicle-to-Grid applications. He published intensively in these areas including a book (*Green IT: Technologies and Applications, Springer*) (ed.) and more than 25 US and international patents. He is a technical editor for *IEEE Communications Magazine*. He also actively contributes to international standard organizations IEEE and ZigBee (currently the chair of IEEE 802.15 TG8 PAC). His research group developed the first NS-2 simulator for IEEE 802.15.4, a standard NS-2 distribution widely used for wireless sensor network researches. He received the Best Paper Award at IEEE CCNC 2005 and CUNY Excellence Performance Award.

Yanyan Zheng received the MS degree in electrical engineering from Stanford University. She is a software engineer at Google Inc. Her research interests are in the area of multimedia streaming, resource allocation, and optimization. She recently became interested in mobile search and mobile media in applications.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.