An SMDP-Based Resource Allocation in Vehicular Cloud Computing Systems

Kan Zheng, *Senior Member, IEEE*, Hanlin Meng, Periklis Chatzimisios, *Senior Member, IEEE*, Lei Lei, *Member, IEEE*, and Xuemin Shen, *Fellow, IEEE*

***Abstract*—Vehicular ad hoc networks are expected to significantly improve traffic safety and transportation efficiency while providing a comfortable driving experience. However, available communication, storage, and computation resources of the connected vehicles are not well utilized to meet the service requirements of intelligent transportation systems. Vehicular cloud computing (VCC) is a promising approach that makes use of the advantages of cloud computing and applies them to vehicular networks. In this paper, we propose an optimal computation resource allocation scheme to maximize the total long-term expected reward of the VCC system. The system reward is derived by taking into account both the income and cost of the VCC system as well as the variability feature of available resources. Then, the optimization problem is formulated as an infinite horizon semi-Markov decision process (SMDP) with the defined state space, action space, reward model, and transition probability distribution of the VCC system. We utilize the iteration algorithm to develop the optimal scheme that describes which action has to be taken under a certain state. Numerical results demonstrate that the significant performance gain can be obtained by the SMDP-based scheme within the acceptable complexity.**

***Index Terms*—Resource allocation, semi-Markov decision process (SMDP), vehicular cloud computing (VCC).**

I. Introduction

RECENTLY, vehicular networks have gained extensive attention from both academia and industry. A variety of smart sensors and devices is installed on vehicles targeting at data acquisition and processing [1], [2]. Meanwhile, various wireless communication technologies can be applied to provide the intervehicle connectivity. There are usually two types of communication paradigms for vehicle services, i.e., vehicle-to-vehicle and vehicle-to-infrastructure (V2I) communications [3]. V2I communication enables vehicles to connect to the Internet via a roadside base station, e.g., either by dedicated short-range communication or cellular networks [4], [5]. By supporting the vehicle-related data gathering and processing, vehicular networks can notably improve the transport safety, relieve the traffic congestion, and increase the quality of driving experience [7]–[9]. As computing and communication technologies have been rapidly developed [6], the vehicles with powerful computing abilities are advocated to be regarded as service providers rather than being only service consumers. As a result, the concept of vehicular cloud computing (VCC) has been proposed, which jointly makes use of computation, communication, and storage resources in vehicle equipment (VE) [10], e.g., onboard computer/communication devices or mobile user equipment (MUE) carried by passengers.

In general, services in the VCC system can be divided into four types according to the function of the resources, i.e., “network-as-a-service,” “storage-as-a-service,” “sensing-as-a-service,” and “computation-as-a-service (CaaS)” [10]. Nowadays, since the computing ability of vehicles is rapidly increased in order to enable themselves to act as providers of computing services, only CaaS is the interest of this paper and is further studied.

In the current paper, we propose the deployment of a layered-cloud computing architecture for the VCC system in order to provide satisfactory services to the VEs. The proposed architecture includes not only a remote cloud (RC) such as a traditional centralized cloud but also vehicular clouds (VCs) that can be regarded as one of the computing capability providers besides the RC. Depending on the mobility of its vehicles, the VC can be either mobile or static. For example, a mobile VC consists of vehicles in movement, while a static one may include vehicles in a parking garage, parking lot, and so on.

Different from the traditional cloud computing system, the VCC system has its unique features. One of them is the variability of the available computation resources in VCs. Due to the uncertainty of the vehicle behavior, i.e., vehicles may randomly join or leave VCs, the resources in VCs are time varying. For the sake of analysis, the considered VCC system is assumed to have the following prosperities, i.e., 1) both the arrivals and the departures of service requests per vehicle follow Poisson

Manuscript received December 6, 2014; revised March 11, 2015 and May 18, 2015; accepted June 13, 2015. Date of publication September 25, 2015; date of current version November 6, 2015. This work was supported in part by the National Science Foundation of China (61331009), in part by the National High-Tech R&D Program (863 Program 2015AA01A705), in part by the National Key Technology R&D Program of China (2015ZX03002009-004), and in part by the Fundamental Research Funds for the Central Universities (2014ZD03-02).

K. Zheng and H. Meng are with the Key Laboratory of Universal Wireless Communications, Ministry of Education, Beijing University of Posts and Telecommunications, Beijing 100088, China (e-mail: kzheng@ieee.org; mhl_ml@163.com).

P. Chatzimisios is with the Department of Informatics, Alexander TEI of Thessaloniki (ATEITHE), 57400 Thessaloniki, Greece (e-mail: peris@it.teithe.gr).

L. Lei is with the State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing 100044, China (e-mail: leil@bjtu.edu.cn).

X. Shen is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: xshen@bbcr.uwaterloo.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIE.2015.2482119

distribution; 2) both the arrivals and the departures of vehicles in a VC follow Poisson distribution; 3) the number of available resources in a VC is time varying; and 4) any current action has a potential impact on a future decision.

Another obvious feature is the heterogeneity of VC resources. Vehicles are produced by different vendors and thus have inherently different computation resources. In order to deal with this issue, the virtualization technique has to be developed to abstract and slice the heterogeneous physical resources into virtual resources, which are shared by multiple VEs in the VCC system. In this paper, each vehicle in a VC is assumed to have virtualized resource units (RUs).

The main research focus in this paper is the resource allocation problem, which maximizes the total long-term expected reward of the VCC system. When a service request from a vehicle arrives at the VCC system, a decision needs to be made, i.e., whether processing it locally in a VC or transferring it to the RC. Moreover, if the request is assigned to a VC, we then have to deal with the issue of allocating RUs to serve this service request. The proposed approach is that the VCC system can achieve the corresponding reward based on the action taken. The reward consists of the income and cost, which depend on both the power consumption and processing time. The resource allocation problem is further formulated as an infinite horizon semi-Markov decision process (SMDP). The state space, action space, reward model, and transition probability distribution of the VCC system are defined and analyzed so as to obtain the optimal scheme, which determines the action taken under a certain state. The solution of optimal allocation policy, i.e., called as the SMDP-based scheme, can be found by iteration. Numerical results show that the SMDP-based scheme outperforms the other two allocation schemes, i.e., the simulated annealing (SA) and greedy allocation (GA) schemes.

The rest of this paper is organized as follows. Section II provides a relative literature survey. The VCC system model is described in Section III. The details of the SMDP formulation, the proposed model, and its corresponding solution are provided in Section IV. Section V presents the numerical results and the corresponding performance analysis. Finally, the conclusion and future work are provided in Section VI.

II. Related Work

A few works on the VCC have been carried out to enhance the service capabilities of VEs. VCC is very similar to a mobile cloud computing (MCC) system, but it brings in new characteristics. In [11], the VCC system is divided into three architectural frameworks, namely, VCs, vehicles using clouds, and hybrid clouds. Moreover, it has been pointed out that in order to form the VCs can effectively deal with services locally produced and improve the experience of VEs [12]. In [13], the parked vehicle assistance is proposed to overcome sparse/unbalanced traffic and greatly promote network connectivity by considering the parked vehicles as static cloud nodes. Also, the parked cars are utilized to sense vehicles that are not in line-of-sight in order to improve safety [14]. A two-tier data center architecture that leverages the excessive storage resources in parking lots has been studied in [15]. Furthermore,

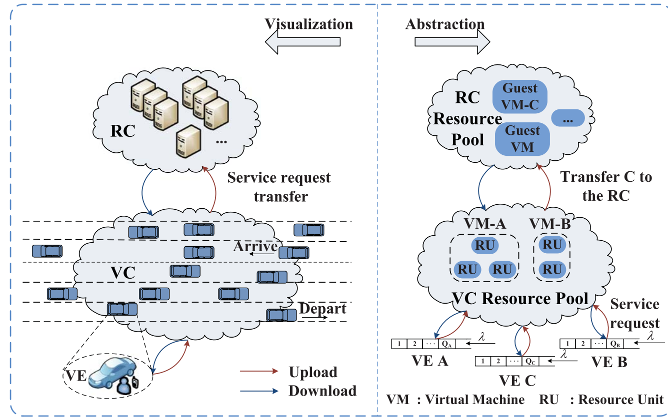


Fig. 1. Illustration of a typical VCC system.

the main focus of the works in [17] and [18] is the security of the VCC system.

There are also certain works in the literatures on the resource allocation problem to improve the computing capability of the VEs in the VCC system. A game-theoretical approach is presented for effective resource management in a roadside cloudset to provide services to several vehicles [16]. Similarly, a distributed and adaptive resource management is proposed for optimal exploitation of cognitive radio and soft-input/soft-output data fusion in vehicular access networks [19], [20], in which the energy and computing limited car smartphones are enhanced by offloading their traffic to the local or RC. However, both of them have not considered that vehicles can share the resources between each other. Consequently, a scheduling model is presented, in which the unpredictable available computation resources in the VCC system are also considered [21].

The current paper attempts to deal with the limitations of the previous works and proposes a resource allocation scheme to better serve the VEs (especially MUEs) in the VCC system that consists of RC and VCs. Although computation resource allocation in an MCC system was studied in [22], this scheme cannot be applied in the VCC system due to the variability feature of the available resources in VCs. Moreover, different from the model in [21], the requests in this paper can be allocated with more than one RUs and processed in parallel. Furthermore, although node mobility is considered in traditional MCC to achieve effective job scheduling, the total long-term expected reward of the system still cannot be obtained in a satisfactory manner as shown in [23] and [24].

III. VCC System

A. System Model

Fig. 1 shows a typical VCC system, in which vehicles in movement constitute a dynamic VC. In particular, VEs that act like smart phones can enjoy vast computing power by submitting the service requests to the VCC system in order to save the energy and enhance the processing speed. A vehicle is assumed to have one basic computation RU in the VCC system. When a service request arrives at the system, it has to make the decision of whether accepting it by the VC or transferring it

TABLE I
LIST OF IMPORTANT NOTATIONS

K	Maximal number of vehicles that the VC can support
M	Number of available RUs in the VC
N_R	Maximal number of RUs that the VC can allocate to a service request
N	Number of system states
λ_p	Arrival rate of new service requests
μ_p	Service rate of the requests
λ_v	Arrival rate of new vehicles
μ_v	Departure rate of vehicles
n_i	Number of served services allocated with i RUs
D_i	Departure of a service request allocated with i RUs
A	Arrival of a service request
B_1/B_{-1}	Arrival / departure of a vehicle
w_e	Energy income weight
w_d	Delay income weight
β_e	Price of per energy saving
β_d	Price of per delay saving
γ	Cost per transmit time
E_l	Energy consumed by executing the request at VE
D_l	Time consumed by executing the request at VE
P	Transmission power of VEs
δ_1	Transmission delay between the VE to the VC
δ_2	Transmission delay between the VC to the RC
I	Income obtained by the VCC system
ξ	Compensation to VE by the VCC system
α	Continuous-time discount factor

to the RC. If the request is assigned to the VC, the decision of allocating how many RUs to it has to be made based on current available resources. Otherwise, a transfer decision is made instead, and then, the service request may be submitted to the RC. For the sake of illustration, an example is also given in Fig. 1. Requests by VE A and VE B are accepted from the VC, while the request by VE C is obliged to be transferred to the RC. After VE A and VE B are admitted, three RUs and two RUs are allocated to them, respectively. All of the decisions are made to achieve the specified objective in the VCC system. The list of important notations of this paper is given in Table I.

Assume that there are M available RUs in the VC, which varies with the arrival and departure of vehicles. K is the maximal number of vehicles that the VC can support, i.e., the number of RUs in the VC cannot exceed K. Each arrival service request can be allocated with i RUs, where $i \in \{1, 2, \ldots, N_R\}$, $N_R \leqslant M$. The arrival rates of service requests and vehicles follow Poisson distribution with λ_p and λ_v, respectively. Let μ_p denote the computing service rate of the request in case of only one RU allocated. Then, the service time of a request is $1/i\mu_p$ in case that i RUs are allocated. In addition, the departure rate of vehicles is denoted as μ_v.

Considering the dynamic characters of the service requests and vehicle arrivals, the action of the current epoch may directly lead to a considerable change of the next state so as to have serious impacts on the system expected total reward. In other words, the action to maximize the reward of the current epoch may become unwise in the long run especially when the resources in the VC are relatively scarce. Therefore, our objective in this paper is to maximize the long-term expected total reward by properly allocating the resources in the VCC system.

B. System States

The system state s reflects the current requests with different numbers of RUs, the available resources in the VC, and the event of requests and vehicles. Therefore, the state set can be denoted by $\mathcal{S}$, i.e.,

$$\mathcal{S} = \{s | s = (n_1, n_2, \ldots, n_{N_R}, M, e)\} \tag{1}$$

where n_i is the number of service requests that have been allocated with i RUs and e represents an event in the set $e \in \mathcal{E} = \{A, D_1, D_2, \ldots, D_{N_R}, B_1, B_{-1}\}$. Here, A denotes the arrival of the service request, D_i means the departure of a request assigned with i RUs, and B_1 and B_{-1} describe the arrival and departure of a vehicle, respectively. Thus, the number of occupied RUs in the VC is $\sum_{i=1}^{N_R} i \cdot n_i$, which satisfies $\sum_{i=1}^{N_R} i \cdot n_i \leqslant M$. Moreover, the number of system states can be denoted by N.

C. Actions

In this model, several possibilities of action a can be taken in the action set $\mathcal{A}$, i.e.,

$$\mathcal{A} = \{-1, 0, 1, 2, \ldots, N_R\}. \tag{2}$$

When an event occurs, the VCC system decides which action $a(s)$ needs to be taken from the action set $\mathcal{A}_s$ based on the current state s, i.e.,

$$\mathcal{A}_s = \begin{cases} \{-1\}, & e \in \{D_1, D_2, \ldots, D_{N_R}, B_1, B_{-1}\} \\ \{0, 1, 2, \ldots, N_R\}, & e = A \end{cases} \tag{3}$$

where $a(s) = -1$ represents the cases that a service request completes and departs from the VCC system or a vehicle arrives at and leaves the VCC system, and no action is required, except that the information of the available RUs in the VCC system has to be updated. When receiving a request, one of the two actions may be chosen: either to accept with i RUs from the VC, i.e., either $a(s) = i$, or to transfer it to the RC, $a(s) = 0$.

D. Rewards

Given an action a, the system reward under the current state s is denoted by

$$r(s, a) = k(s, a) - g(s, a) \tag{4}$$

where $k(s, a)$ is the instant revenue of the VCC system by taking action a under state s in case that event e occurs, which consists of both the income and cost of the VCC system. Since the main benefits of the VCC system are to save the power consumption and speed up the processing rate of VEs [26], the income has to include the effects of both of them [27]. Meanwhile, the cost of the system is the transfer expense to send and receive the request. $g(s, a)$ is the expected system cost before the next decision epoch. Furthermore, $k(s, a)$ of the VCC system can be described by (5), shown at the bottom of the page. The details of the revenue function are explained as follows.

1) When a service request is admitted to the VC, the instant revenue $[w_e\beta_e(E_l - P\cdot\delta_1) + w_d\beta_d(D_l - 1/i\mu_p - \delta_1)]$ can be earned by the system. $(E_l - P\cdot\delta_1)$ and $(D_l - 1/i\mu_p - \delta_1)$ are the saved energy and time when processing the computing task in the VC, respectively. β_e and β_d are the price of per unit energy and time. Different weights, i.e., w_e and w_d, can be predefined according to different purposes, where $w_e + w_d = 1$. The transfer expense is denoted as $\gamma\delta_1$, which is the cost of the VCC system to receive the computing task from the VEs and send back the results. More specially, since the request has already been accepted by the VC, the VE can enjoy the service by transmitting the task to the VC and receiving the feedback from it, which consumes $P\cdot\delta_1$ energy and δ_1 time at this stage. For the purpose of analysis, the transmitted power and received power are assumed to be identical [28]. If the request is allocated with i RUs by the VCC system, the service time spent for finishing the task is $1/i\mu_p$.
2) The service request may be transferred to the RC, which may happen in case of no sufficient available resources in the VC. By this way, the VCC system gains I revenue at the cost of transfer expense including $\gamma\delta_2$ and $\gamma\delta_1$. Here, $\gamma\delta_2$ is the cost to transmit the computing task to the RC and receive the feedback from it. Since the RC is assumed to have the powerful computing capability, the revenue I can be calculated by $[w_e\beta_e(E_l - P\cdot\delta_1) + w_d\beta_d(D_l - \delta_1 - \delta_2)]$ without considering the processing time. Moreover, due to the large end-to-end communication delay [16], it is not wise for the VCC system to send the requests to the RC if the resources in the VC are sufficient.
3) The VCC system has no revenue when a service leaves the system or a vehicle joins the system.
4) The VCC system also gains no revenue under the states in which the VC has spare RUs to be allocated and a vehicle leaves the system.
5) When all of the RUs have already been exhausted and at exactly this moment a vehicle with a RU leaves the VC, the VCC system has to compensate the request occupying this RU with ξ price. This is because there is no spare RU in the system to be allocated to guarantee the number of RUs of the request whose RU is leaving.

TABLE II
TRANSFORMATION OF ACTIONS AND CORRESPONDING STATES

No.	Actions	State Transition
1	$a = 0$	$s_{next1} = \{1, 1, 1, M, A\}$, $s_{next2} = \{1, 1, 1, M, D_i\}$, $i \in \{1, 2, 3\}$ $s_{next3} = \{1, 1, 1, M, B_1/B_{-1}\}$,
2	$a = 1$	$s_{next1} = \{2, 1, 1, M, D_i\}$, $i \in \{1, 2, 3\}$ $s_{next2} = \{2, 1, 1, M, A\}$ $s_{next3} = \{2, 1, 1, M, B_1/B_{-1}\}$
3	$a = 2$	$s_{next1} = \{1, 2, 1, M, D_i\}$, $i \in \{1, 2, 3\}$ $s_{next2} = \{1, 2, 1, M, A\}$ $s_{next3} = \{1, 2, 1, M, B_1/B_{-1}\}$
4	$a = 3$	$s_{next1} = \{1, 1, 2, M, D_i\}$, $i \in \{1, 2, 3\}$ $s_{next2} = \{1, 1, 2, M, A\}$ $s_{next3} = \{1, 1, 2, M, B_1/B_{-1}\}$

Next, the expected system cost is defined by

$$g(s, a) = c(s, a)\tau(s, a) \tag{6}$$

where $\tau(s, a)$ is the expected service time from the current state to the next state in case that action a is taken under state s and $c(s, a)$ is the cost rate of $\tau(s, a)$ in case that action a is selected. Moreover, $c(s, a)$ can be characterized by the number of occupied RUs in the VC due to its limited computing capability, i.e.,

$$c(s, a) = \sum_{i=1}^{N_R} i \cdot n_i. \tag{7}$$

IV. SMDP-BASED SCHEME FOR VCC

In our analysis, the state transition is determined by the action a under state s. Let us consider the system state $s = (1, 1, 1, M, A)$ as an example, and the corresponding state transition under different actions is shown in Table II. Furthermore, the state transition probability under different actions plays an important role on the acquired optimal policy. Thus, in this section, we first derive the state transition probability matrix. Then, the reward function is revised since a discounted model is utilized. Finally, we provide the optimal policy that can be found by utilizing the value iteration algorithm.

$$k(s,a) = \begin{cases} [w_e\beta_e(E_l - P\cdot\delta_1) + w_d\beta_d(D_l - 1/i\mu_p - \delta_1)) - \gamma\delta_1], & a = i, e = A \\ I - \gamma(\delta_2 + \delta_1), & a = 0, e = A \\ 0, & a = -1, e \in \{D_1, D_2, \ldots, D_{N_R}, B_1\} \\ 0, & a = -1, e \in \{B_{-1}\}, \sum_{i=1}^{N_R} i\cdot n_i \neq M \\ -\xi, & a = -1, e \in \{B_{-1}\}, \sum_{i=1}^{N_R} i\cdot n_i = M \end{cases} \tag{5}$$

A. Transition Probability

Under a given state s and an action a, the expected service time between two continuous decision epoch is denoted by $\tau(s,a)$. Thus, the mean event rate for specific s and a values is the sum of rates of all of the events in the VCC system, which can be expressed by

$$\sigma(s,a) = \tau(s,a)^{-1}$$
$$= \begin{cases} (M+1)\lambda_p + \lambda_v + \mu_v + \sum_{j=1}^{N_R} jn_j\mu_p, & e = B_1, a = -1 \\ (M-1)\lambda_p + \lambda_v + \mu_v + \sum_{j=1}^{N_R} jn_j\mu_p, & e = B_{-1}, a = -1 \\ M\lambda_p + \lambda_v + \mu_v + \sum_{j=1}^{N_R} jn_j\mu_p + i\mu_p, & e = A,\ a = i \\ & i \in \{0,1,\ldots,N_R\} \\ M\lambda_p + \lambda_v + \mu_v + \sum_{j=1}^{N_R} jn_j\mu_p - i\mu_p, & e = D_i,\ a = -1 \\ & i \in \{0,1,\ldots,N_R\} \end{cases} \tag{8}$$

where μ_v is the departure rate of vehicles and $(M\lambda_p + \lambda_v)$ is the total arrival rate of requests and vehicles. Since λ_p is the arrival rate for requests per vehicle, the arrival rate of requests of the VCC system can be denoted by $M\lambda_p$. The departure rate of requests is explained as follows. When a vehicle joins or leaves the VCC system, the total number of occupied RUs by the existing requests is not changed, which can be denoted by $\sum_{j=1}^{N_R} jn_j$. Thus, the departure rate of vehicles can be computed as $\sum_{j=1}^{N_R} jn_j\mu_p$. When a request arrives, the number of occupied RUs can be given by $(\sum_{j=1}^{N_R} jn_j + i)$ no matter which action is taken by the VCC system. Thus, the corresponding departure rate is computed as $(\sum_{j=1}^{N_R} jn_j\mu_p + i\mu_p)$. When a request is served and leaves the system, the number of occupied RUs becomes $(\sum_{j=1}^{N_R} jn_j - i)$. The departure rate of request is $(\sum_{j=1}^{N_R} jn_j\mu_p - i\mu_p)$.

Next, $P(s'|s,a)$ is defined as the transition probability from state s to state s' under an action a, which can be calculated under different events, i.e.,

- State $s = (n_1,\ldots,n_{N_R},M,A)$

$$P(s'|s,a)$$
$$= \begin{cases} \frac{M\lambda_p}{\sigma(s,a)}, & a = 0, s' = (n_1,\ldots,n_{N_R},M,A) \\ \frac{n_i i\mu_p}{\sigma(s,a)}, & a = 0, s' = (n_1,\ldots,n_{N_R},M,D_i) \\ \frac{\lambda_v}{\sigma(s,a)}, & a = 0, s' = (n_1,\ldots,n_{N_R},M,B_1) \\ \frac{\mu_v}{\sigma(s,a)}, & a = 0, s' = (n_1,\ldots,n_{N_R},M,B_{-1}) \\ \frac{(n_i+1)i\mu_p}{\sigma(s,a)}, & a = i, s' = (n_1,\ldots,n_i+1,\ldots,n_{N_R},M,D_i) \\ \frac{n_m m\mu_p}{\sigma(s,a)}, & a = i, m \neq i, \\ & s' = (n_1,\ldots,n_i+1,\ldots,n_{N_R},M,D_m) \\ \frac{M\lambda_p}{\sigma(s,a)}, & a = i, s' = (n_1,\ldots,n_i+1,\ldots,n_{N_R},M,A) \\ \frac{\lambda_v}{\sigma(s,a)}, & a = i, s' = (n_1,\ldots,n_i+1,\ldots,n_{N_R},M,B_1) \\ \frac{\mu_v}{\sigma(s,a)}, & a = i, s' = (n_1,\ldots,n_i+1,\ldots,n_{N_R},M,B_{-1}). \end{cases} \tag{9}$$

- State $s = (n_1,\ldots,n_{N_R},M,D_i)$

$$P(s'|s,a)$$
$$= \begin{cases} \frac{M\lambda_p}{\sigma(s,a)}, & a = -1, s' = (n_1,\ldots,n_i-1,\ldots,n_{N_R},M,A) \\ \frac{(n_i-1)i\mu_p}{\sigma(s,a)}, & a = -1, s' = (n_1,\ldots,n_i-1,\ldots,n_{N_R},M,D_i) \\ \frac{n_m m\mu_p}{\sigma(s,a)}, & a = -1, m \neq i, \\ & s' = (n_1,\ldots,n_i-1,\ldots,n_N,M,D_m) \\ \frac{\lambda_v}{\sigma(s,a)}, & a = -1, s' = (n_1,\ldots,n_i-1,\ldots,n_{N_R},M,B_1) \\ \frac{\mu_v}{\sigma(s,a)}, & a = -1, s' = (n_1,\ldots,n_i-1,\ldots,n_{N_R},M,B_{-1}). \end{cases} \tag{10}$$

- State $s = (n_1,\ldots,n_{N_R},M,B_1)$

$$P(s'|s,a)$$
$$= \begin{cases} \frac{\lambda_v}{\sigma(s,a)}, & a = -1, s' = (n_1,\ldots,n_{N_R},M+1,B_1) \\ \frac{u_v}{\sigma(s,a)}, & a = -1, s' = (n_1,\ldots,n_{N_R},M+1,B_{-1}) \\ \frac{(M+1)\lambda_p}{\sigma(s,a)}, & a = -1, s' = (n_1,\ldots,n_{N_R},M+1,A) \\ \frac{n_i i\mu_p}{\sigma(s,a)}, & a = -1, s' = (n_1,\ldots,n_{N_R},M+1,D_i). \end{cases} \tag{11}$$

- State $s = (n_1,\ldots,n_{N_R},M,B_{-1})$

$$P(s'|s,a)$$
$$= \begin{cases} \frac{\lambda_v}{\sigma(s,a)}, & a = -1, s' = (n_1,\ldots,n_{N_R},M-1,B_1) \\ \frac{u_v}{\sigma(s,a)}, & a = -1, s' = (n_1,\ldots,n_{N_R},M-1,B_{-1}) \\ \frac{(M-1)\lambda_p}{\sigma(s,a)}, & a = -1, s' = (n_1,\ldots,n_{N_R},M-1,A) \\ \frac{n_i i\mu_p}{\sigma(s,a)}, & a = -1, s' = (n_1,\ldots,n_{N_R},M-1,D_i). \end{cases} \tag{12}$$

B. Discounted Reward Model

Assume that the time between two decision epochs is exponentially distributed, i.e.,

$$F(t|s,a) = 1 - e^{-\sigma(s,a)t}, \quad \text{for } t > 0. \tag{13}$$

Since the system state does not change between decision epochs, the expected discounted reward is defined based on the discounted reward model found in [29] and [30]

$$\begin{aligned} r(s,a) &= k(s,a) - c(s,a)E_s^a\left\{\int_0^\tau e^{-\alpha t}dt\right\} \\ &= k(s,a) - c(s,a)E_s^a\left\{\frac{[1-e^{-\alpha\tau}]}{\alpha}\right\} \\ &= k(s,a) - c(s,a)/\left[\alpha + \sigma(s,a)\right] \end{aligned} \tag{14}$$

where α is a continuous-time discount factor.

C. Solution

A discounted model is applied to obtain the maximum total long-term expected discounted reward [30]. With a stationary policy π: $\mathcal{S} \to \mathcal{A}$, the total long-term expected discounted reward can be given by

$$v_\alpha^\pi(s) = E_s^\pi\left[\sum_{n=0}^{\infty} e^{-\alpha\sigma_n} r(s_n,a_n)|s_0 = s\right]. \tag{15}$$

The function is the expected sum of rewards that can be obtained throughout the process when starting from state s. Therefore, the maximal expected total discounted reward can be obtained, i.e.,

$$v_\alpha^*(s) = v_\alpha^{\pi^*}(s) = \max_\pi v_\alpha^\pi(s). \tag{16}$$

Based on the transition probability, (16) can be rewritten in the form of Bellman equation, i.e.,

$$v(s) = \max_{a\in\mathcal{A}_s}\left[r(s,a) + \lambda\sum_{s'\in\mathcal{S}} p(s'|s,a)v(s')\right] \tag{17}$$

where $\lambda = \sigma(s,a)/(\alpha+\sigma(s,a))$. In order to further uniform the continuous-time Markov decision process, another new parameter is introduced, i.e., $y = K\lambda_p + \lambda_v + \mu_v + K\cdot N_R\cdot\mu_p$. Then, the normalized component (17) is given by

$$\tilde{r}(s,a) = r(s,a)\frac{\alpha+\sigma(s,a)}{\alpha+y} \tag{18}$$

$$\tilde{\lambda} = \frac{y}{(y+\alpha)} \tag{19}$$

$$\tilde{p}(s'|s,a) = \begin{cases} 1-\frac{[1-p(s|s,a)]\sigma(s,a)}{y}, & s'=s \\ \frac{p(s'|s,a)\sigma(s,a)}{y}, & s'\neq s. \end{cases} \tag{20}$$

Thus, after normalization, (17) can be rewritten as

$$\tilde{v}(s) = \max_{a\in\mathcal{A}_s}\left[\tilde{r}(s,a) + \tilde{\lambda}\sum_{s'\in S}\tilde{p}(s'|s,a)\tilde{v}(s')\right]. \tag{21}$$

Since the proposed model is the infinite SMDP with finite state and action spaces, the value iteration can be used to solve the optimization problem given by (21). A detailed description is provided in Algorithm 1.

Algorithm 1

Step 1: Set $\tilde{v}(s) = 0$ for each state s. Specify $\varepsilon > 0$, and set $k = 0$.
Step 2: For each state s, compute $\tilde{v}^{k+1}(s)$ by

$$\tilde{v}^{k+1}(s) = \max_{a\in\mathcal{A}_s}\left[\tilde{r}(s,a) + \tilde{\lambda}\sum_{s'\in\mathcal{S}}\tilde{p}(s'|s,a)\tilde{v}^k(s')\right].$$

Step 3: If $\|\tilde{v}^{k+1} - \tilde{v}^k\| < \varepsilon(1-\tilde{\lambda})/2\tilde{\lambda}$, go to **Step 4**. Otherwise, increase k by 1 and go back to **Step 2**.
Step 4: For each $s\in\mathcal{S}$, compute the stationary optimal policy and stop

$$d_\varepsilon^*(s) \in \arg\max_{a\in\mathcal{A}_s}\left[\tilde{r}(s,a) + \tilde{\lambda}\sum_{s'\in\mathcal{S}}\tilde{p}(s'|s,a)\tilde{v}^{k+1}(s')\right].$$

In our paper, the norm function is defined as $\|\tilde{\mathbf{v}}\| = \max|\tilde{v}(s)|$ for $s\in\mathcal{S}$. Since the operation in **step 2** corresponds to a contraction mapping, the convergence of the value iteration is ensured by Banach fixed-point theorem [29]. Thus, the function $\tilde{v}^k(s)$ converges in norm to $v_{\tilde{\lambda}}{}^*(s)$. Note that the convergence rate of the value iteration algorithm is linear with the rate $\tilde{\lambda}$.

TABLE III
SYSTEM PARAMETERS IN THE VCC SYSTEM

Parameter	Value	Parameter	Value
N_R	3	K	3-13
λ_p	1-9	μ_p	8
λ_v	4-8	μ_v	8
w_e	0.5	w_d	0.5
β_e	2	β_d	2
γ	2	E_l	20
D_l	20	P_l	4
δ_1	2	δ_2	5
α	0.1	ξ	18

V. NUMERICAL RESULTS AND ANALYSIS

This section provides the numerical results in order to evaluate the performances of the proposed computation resource allocation scheme. For comparison purposes, we perform performance comparison between the proposed scheme and the following two allocation schemes, i.e.,

1) **GA scheme**: By this scheme, the VCC system always allocates the maximal number of RUs to achieve the highest system reward at the decision epoch [25].
2) **SA scheme**: As being one of the typical heuristic algorithms, the SA scheme is widely used to be able to find near-optimal solutions to optimization problems [31]. However, since the objective function value of each new policy has to be obtained especially in case of a large number of system states, it is hard to be implemented due to its high computational complexity.

In order to obtain optimal results for resource allocation in the VCC system, the SMDP-based scheme has the polynomial complexity of $O(N^2)$ [30]. On the other hand, the complexities of the GA and SA schemes are of $O(N)$ and $O(N^3)$, respectively [32].

The main parameters used in our analysis are provided in Table III. The maximum number of RUs allocated to a service request is $N_R = 3$, i.e., a service can be assigned one, two, or three RUs, which depends on the available resources in the VC. Some parameters such as the arrival rate of service requests and vehicles as well as the maximal number of vehicles K that the VCC system supports can be adjusted for evaluation.

Case 1, *case 2*, and *case 3* represent that the VCC system assigns the request to the VC and allocates it with one, two, and three RUs, respectively. Moreover, one special case, i.e., *case 0*, denotes that the VCC system transfers the request to the RC. Figs. 2–4 show the action probability of the service requests under different λ_p values.

As shown in Fig. 2, when the arrival rate of the requests per vehicle is low, the VCC system has abundant resources in the VC to be allocated so that the probability of assigning service requests to the VC is higher than the probability of transferring to the RC. Moreover, when the VCC system assigns one request to the VC, it tends to allocate as many RUs as possible to maximize the system reward. This results in the situation that the probability of *case 3* is the highest while those of *case 1* and *case 2* are lower and that of *case 0* is the lowest. Such situation gradually begins to change when the arrival rate of requests per vehicle increases. The VCC system is inclined to make conservative decisions since the reward of accepting a new request with one or two RUs is more attractive. In other words,

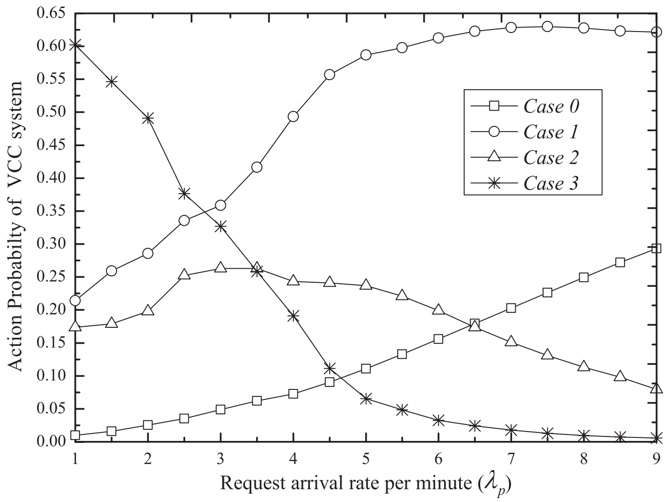


Fig. 2. Action probabilities under different arrival rates of service requests per vehicle ($\lambda_v = 7$, and $K = 10$).

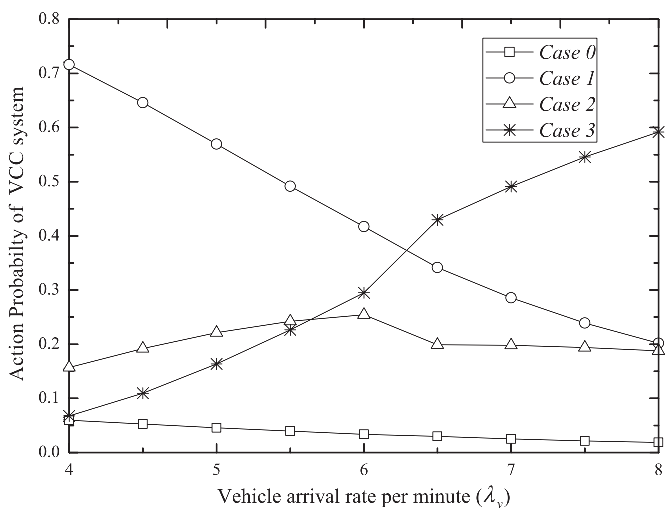


Fig. 3. Action probabilities under different arrival rates of vehicles ($\lambda_p = 2$, and $K = 10$).

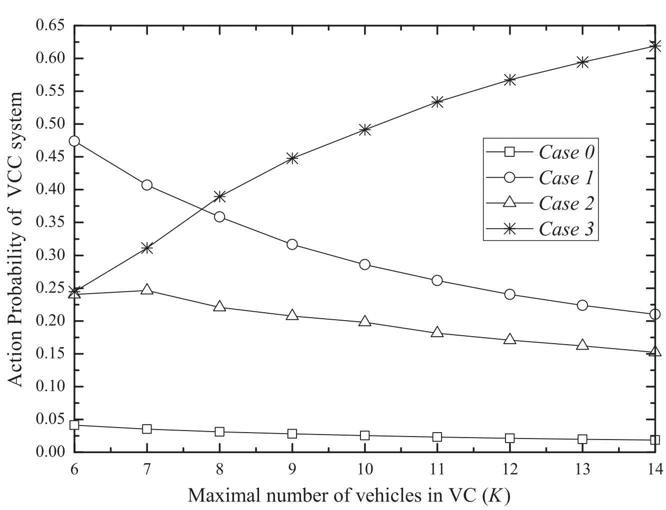


Fig. 4. Action probabilities under different numbers of maximal vehicles in the VCC system ($\lambda_p = 2$, $\lambda_v = 7$).

accepting a request with three RUs in the VC and then being forced to transfer the new request to the RC is not a wise choice any longer. Thus, the probability of *case 3* becomes small, while

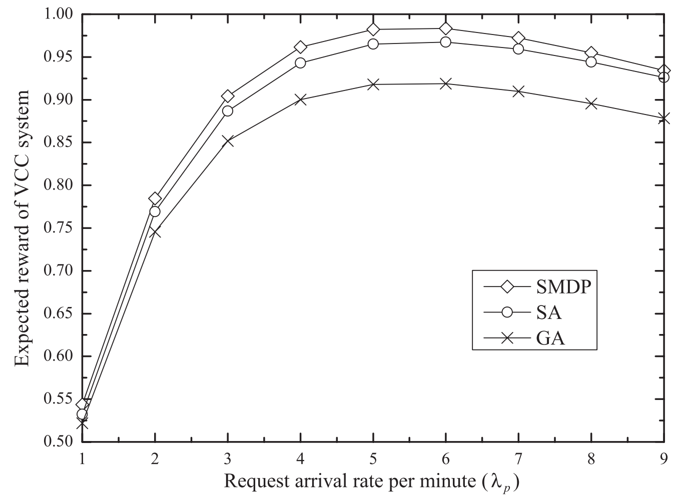


Fig. 5. System reward for different arrival rates of service requests per vehicle ($\lambda_v = 7$, and $K = 10$).

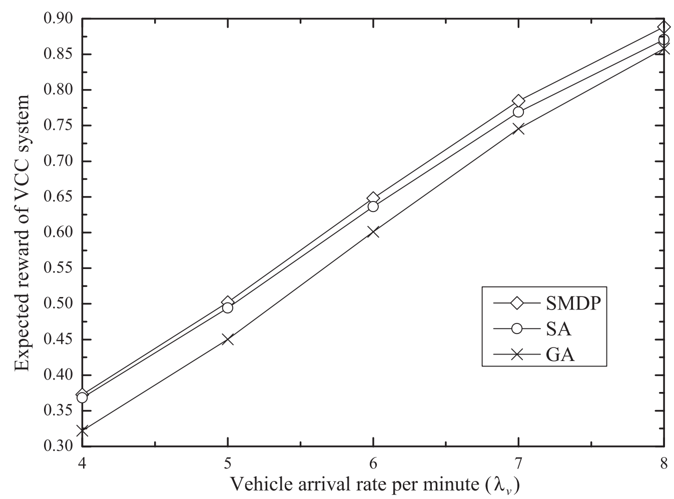


Fig. 6. System reward for different arrival rates of vehicles ($\lambda_p = 2$, and $K = 10$).

those of *case 1* and *case 2* increase at the beginning. When the arrival rate is high, the VCC system tends to only allocate one RU to the request so that the probability of *case 2* also begins to decrease.

In Fig. 3, with the increase of the arrival rate of vehicles, the probabilities of *case 2* and *case 3* become larger, while those of *case 0* and *case 1* begin to decrease. It is because the resources in the VC have the trend to become sufficient when the arrival rate of the vehicle increases. When the arrival rate is high, the probability of *case 2* also decreases because of relatively abundant resources. Fig. 4 illustrates that the probability of *case 3* becomes larger with the increase of the maximal number of vehicles that the VCC system supports, which is also due to more RUs to be allocated to the request.

Next, we compare the performance of various VCC systems that utilize different resource allocation schemes, i.e., the SMDP-based, SA, and GA schemes. Figs. 5–7 illustrate the trend of the total expected reward under different conditions. In Fig. 5, when the arrival rate of the requests per vehicle increases, the expected total reward of the VCC system attains higher values because more requests are accepted and served by the VC. However, when the arrival rate of the requests

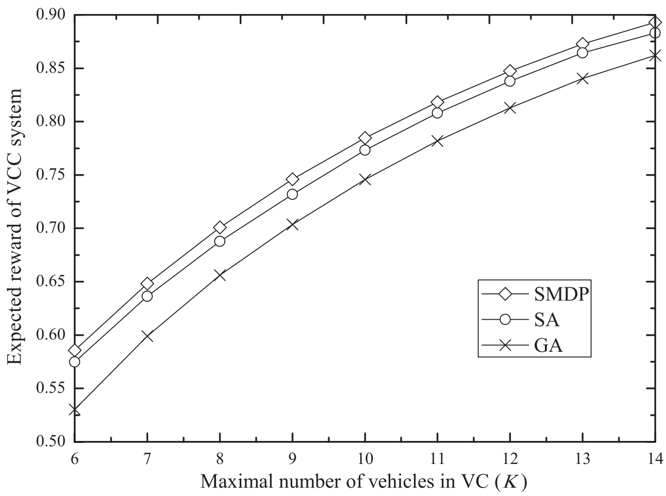


Fig. 7. System reward for different numbers of maximal vehicles ($\lambda_p = 2$, and $\lambda_v = 7$).

is high, the expected total reward starts to decrease since the probability of transfer to the RC becomes larger. It is clear that our proposed SMDP-based scheme outperforms the others. When the arrival rate is low, the total reward difference between different schemes is not too much. In this case, the VCC system tends to allocate as many RUs as possible to the request due to the existence of sufficient resources in the VC. Then, the optimal policy obtained by the SMDP-based scheme is similar to the other considered schemes. However, when the arrival rate is increased a little, the advantage of the SMDP-based scheme becomes obvious. When the arrival rate is high, e.g., $\lambda_p = 5$, 7% performance improvement can be achieved by the SMDP-based scheme compared to the GA scheme. Meanwhile, the performance of the SMDP-based scheme is always a little better than that of the SA scheme.

In Fig. 6, with the increase of the arrival rate of vehicles, the expected total reward of the three considered schemes becomes larger because the resources in the VC are more sufficient. Meanwhile, a similar trend can also be found when the maximal number of vehicles that the VC can support is increased as shown in Fig. 7.

VI. Conclusion and Future Work

In this paper, we have proposed a computation resource allocation scheme for a VCC system, which is formulated as an infinite horizon SMDP. An optimal decision-making scheme is obtained through the iteration algorithm in order to maximize the long-term expected total reward of the VCC system. The numerical results show a significant expected reward performance gain over others, e.g., compared with the GA scheme, nearly 7% performance gain when either λ_p is high or K is low. Moreover, the complexity of the SMDP-based scheme is lower than that of the SA scheme.

In our future work, we are planning to investigate the effects of parameter tolerance to the optimal scheme in the VCC system, which may lead to develop more robust and practical schemes. This becomes a more challenging work if we take into account that the system size of a VCC system is rapidly increased.

References

[1] Y. Lee and K. Rajashekara, "Power electronics and motor drives in electric, hybrid electric, and plug-in hybrid electric vehicles," *IEEE Trans. Ind. Electron.*, vol. 55, no. 6, pp. 2237–2245, Jun. 2008.

[2] K. Chan, T. Dillon, and E. Chang, "An intelligent particle swarm optimization for short-term traffic flow forecasting using on-road sensor systems," *IEEE Trans. Ind. Electron.*, vol. 60, no. 10, pp. 4714–4725, Aug. 2013.

[3] M. Baguena Albaladejo, C. Calafate, J. Cano, and P. Manzoni, "An adaptive anycasting solution for crowd sensing in vehicular environments," *IEEE Trans. Ind. Electron.*, to be published.

[4] J. Kenney, "Dedicated short-range communications (DSRC) standards in the united states," *Proc. IEEE*, vol. 99, no. 7, pp. 1162–1182, Jul. 2011.

[5] G. Araniti, C. Campolo, M. Condoluci, A. Iera, and A. Molinaro, "LTE for vehicular networking: A survey," *IEEE Commun. Mag.*, vol. 51, no. 5, pp. 148–157, May 2013.

[6] M. Jin, X. Zhou, E. Luo, and X. Qing, "Industrial-QoS-oriented remote wireless communication protocol for the Internet of construction vehicles," *IEEE Trans. Ind. Electron.*, vol. 62, no. 11, pp. 7103–7113, Nov. 2015.

[7] F. Dressler, H. Hartenstein, O. Altintas, and O. Tonguz, "Inter-vehicle communication: Quo vadis," *IEEE Commun. Mag.*, vol. 52, no. 6, pp. 170–177, Jun. 2014.

[8] J. A. F. F. Dias, J. J. P. C. Rodrigues, C. Mavromoustakis, and F. Xia, "A cooperative watchdog system to detect misbehavior nodes in vehicular delay-tolerant networks," *IEEE Trans. Ind. Electron.*, to be published.

[9] C. C. Lin and D. J. Deng, "Optimal two-lane placement for hybrid VANET-sensor networks," *IEEE Trans. Ind. Electron.*, to be published.

[10] L. Gu, D. Zeng, and S. Guo, "Vehicular cloud computing: A survey," in *Proc. IEEE Globecom Workshops*, 2013, pp. 403–407.

[11] R. Hussain, J. Son, H. Eun, S. Kim, and H. Oh, "Rethinking vehicular communications: Merging VANET with cloud computing," in *Proc. IEEE CloudCom*, 2012, pp. 606–609.

[12] E. Lee, M. Gerla, and S. Oh, "Vehicular cloud networking: Architecture and design principles," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 148–155, Feb. 2014.

[13] N. Liu, M. Liu, W. Lou, G. Chen, and J. Cao, "PVA in VANETs: Stopped cars are not silent," in *Proc. IEEE INFOCOM*, 2011, pp. 431–435.

[14] D. Eckhoff *et al.*, "Cooperative awareness at low vehicle densities: How parked cars can help see through buildings," in *Proc. IEEE GLOBECOM*, 2011, pp. 1–6.

[15] L. Gu *et al.*, "Leverage parking cars in a two-tier data center," in *Proc. IEEE WCNC*, 2013, pp. 4665–4670.

[16] R. Yu, Y. Zhang, W. Xia, and K. Yang, "Toward cloud-based vehicular networks with efficient resource management," *IEEE Netw. Mag.*, vol. 27, no. 5, pp. 49–55, Sep. 2013.

[17] G. Yan *et al.*, "Towards secure vehicular clouds," in *Proc. CISIS*, 2012, pp. 370–375.

[18] G. Yan *et al.*, "Security challenges in vehicular cloud computing," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 1, pp. 284–294, Mar. 2013.

[19] N. Cordeschi, D. Amendola, and E. Baccarelli, "Reliable adaptive resource management for cognitive cloud vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 64, no. 6, pp. 1–10, Aug. 2014.

[20] N. Cordeschi *et al.*, "Distributed and adaptive resource management in cloud-assisted cognitive radio vehicular networks with hard reliability guarantees," *Veh. Commun.*, vol. 2, no. 1, pp. 1–12, Aug. 2014.

[21] P. Ghazizadeh, "Resource Allocation in Vehicular Cloud Computing," Ph.D. dissertation, Comput. Sci., Old Dominion Univ., Norfolk, VA, USA, 2014.

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

[23] P. Ghosh, R. Nirmalya, and S. K. Das, "Mobility-aware efficient job scheduling in mobile grids," in *Proc. IEEE CCGRID*, May 2007, pp. 701–706.

[24] A. Gomes, A. Ziviani, L. Lima, and M. Endler, "DICHOTOMY: A resource discovery and scheduling protocol for multihop ad hoc mobile grids," in *Proc. IEEE CCGRID*, May 2007, pp. 719–724.

[25] C. Thomas, H. Cormen, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed. Cambridge, U.K., MIT Press, 2009.

[26] L. Lei *et al.*, "Challenges on wireless heterogeneous networks for mobile cloud computing," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 34–44, Jun. 2013.

[27] M. Nasseri, M. Alam, and R. C. Green, "MDP based optimal policy for collaborative processing using mobile cloud computing," in *Proc. IEEE CloudNet*, 2013, pp. 123–129.

[28] K. Kumar and Y. H. Lu, “Cloud computing for mobile users: Can offloading computation save energy?” *Computer*, vol. 43, no. 4, pp. 51–56, Apr. 2010.

[29] S. Mine and M. Puterman, *Markovian Decision Process*. Amsterdam, The Netherlands: Elsevier, 1970.

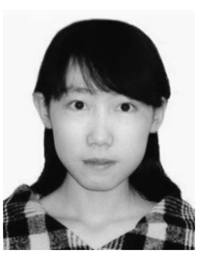
[30] M. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York, NY, USA: Wiley, 2005.

[31] P.B. Hansen, “Simulated annealing,” Elect. Eng. Comput. Sci., Tech. Reps., Paper 170, 1992. [Online]. Available: http://surface.syr.edu/eecs_techreports/170

[32] T. Cormen *et al.*, *Introduction to Algorithms*. Cambridge, MA, USA: MIT Press, 2001.

Kan Zheng (SM’09) received the B.S., M.S., and Ph.D. degrees from Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 1996, 2000, and 2005, respectively.

He is currently a Professor with BUPT. He has rich experience in the research and standardization of new emerging technologies. He is the author of more than 200 journal articles and conference papers in the field of wireless networks, M2M networks, VANET, etc. He holds Editorial Board positions for several journals. He has organized several Special Issues in prestigious journals including IEEE Communications Surveys and Tutorials, *IEEE Communications Magazine*, and IEEE Systems Journal.



Hanlin Meng received the B.S. degree from the School of Automation, Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2013, where she is currently working toward the M.S. degree in the Key Laboratory of Universal Wireless Communications, Ministry of Education.

Her research interests include mobile cloud computing, vehicular networks, software-defined networks, and corresponding resource allocation.

Periklis Chatzimisios (S’02–M’05–SM’12) received the B.Sc. degree from Alexander TEI of Thessaloniki (ATEITHE), Thessaloniki, Greece, in 2000 and the Ph.D. degree from Bournemouth University, Poole, U.K., in 2005.

He serves as an Associate Professor with the Computing Systems, Security and Networks Research Laboratory, Department of Informatics, ATEITHE. Recently, he has been a Visiting Academic/Researcher with the University of Toronto, Toronto, ON, Canada, and Massachusetts Institute of Technology, Cambridge, MA, USA. He is the author/editor of eight books and more than 100 peer-reviewed papers and book chapters on the topics of performance evaluation and standardization activities of mobile/wireless communications, quality of service/quality of experience, and vehicular networking. His published research work has received more than 1500 citations by other researchers.

Dr. Chatzimisios is involved in several standardization activities serving as a member of the Standards Development Board for the IEEE Communications Society (since 2010) and lately as an active member of the IEEE Research Groups on IoT Communications and Networking Infrastructure and on Software-Defined and Virtualized Wireless Access. He is also very active in IEEE activities such as serving as the Vice Chair of the Emerging Technical Subcommittee on Big Data and the Secretary of the IEEE Technical Committee on Cognitive Networks (during 2012–2014). He has served as an organizing/TPC committee member for more than 150 conferences and as the founder/organizer/cochair for many workshops which are coallocated with major IEEE conferences. He also holds Editorial Board positions for several IEEE/non-IEEE journals, and he is the Director (Codirector during 2012–2014) for the e-letter of the IEEE Technical Committee on Multimedia Communications (MMTC).

Lei Lei (M’13) received the B.S. and Ph.D. degrees in telecommunications engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 2001 and 2006, respectively.

From July 2006 to March 2008, she was a Postdoctoral Fellow with the Computer Science Department, Tsinghua University, Beijing. She was with the Wireless Communications Department, China Mobile Research Institute, from April 2008 to August 2011. She has been an Associate Professor with the State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing, since September 2011. Her current research interests include performance evaluation, quality of service, and radio resource management in wireless communication networks.

Xuemin (Sherman) Shen (F’09) received the B.Sc. degree in electrical engineering from Dalian Maritime University, Dalian, China, in 1982 and the M.Sc. and Ph.D. degrees in electrical engineering from Rutgers University, New Brunswick, NJ, USA, in 1987 and 1990, respectively.

He is a Professor and University Research Chair with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. He was the Associate Chair for Graduate Studies from 2004 to 2008. His research focuses on resource management in interconnected wireless/wired networks, wireless network security, social networks, smart grid, and vehicular ad hoc and sensor networks.

Dr. Shen is a Registered Professional Engineer of Ontario, Canada, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, and a Distinguished Lecturer of the IEEE Vehicular Technology Society and IEEE Communications Society. He is an elected member of the IEEE ComSoc Board of Governor and the Chair of the Distinguished Lecturers Selection Committee. He served as the Technical Program Committee Chair/Cochair of IEEE INFOCOM’14 and IEEE VTC’10 Fall, the Symposia Chair of IEEE ICC’10, the Tutorial Chair of IEEE VTC’11 Spring and IEEE ICC’08, the Technical Program Committee Chair of IEEE Globecom’07, the General Cochair of ACM Mobihoc’15, Chinacom’07, and QShine’06, and the Chair of the IEEE Communications Society Technical Committee on Wireless Communications and P2P Communications and Networking. He also serves/served as the Editor-in-Chief of IEEE Network, *Peer-to-Peer Networking and Application*, and *IET Communications*; a Founding Area Editor of the IEEE Transactions on Wireless Communications; an Associate Editor of the IEEE Transactions on Vehicular Technology, *Computer Networks*, *ACM/Wireless Networks*, etc.; and the Guest Editor of IEEE JSAC, *Wireless Communications*, *IEEE Communications Magazine*, *ACM Mobile Networks and Applications*, etc. He received the Excellent Graduate Supervision Award in 2006 and the Outstanding Performance Award in 2004, 2007, 2010, and 2014 from the University of Waterloo, the Premier’s Research Excellence Award in 2003 from the Province of Ontario, Canada, and the Distinguished Performance Award in 2002 and 2007 from the Faculty of Engineering, University of Waterloo.