Received January 12, 2019, accepted January 25, 2019, date of publication February 6, 2019, date of current version March 26, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2897617

Game Theory Based Opportunistic Computation Offloading in Cloud-Enabled IoV

MINGHUI LIWANG[1], JIEXIANG WANG[1], ZHIBIN GAO[1], XIAOJIANG DU[2], (Senior Member, IEEE), AND MOHSEN GUIZANI[3], (Fellow, IEEE)

[1]School of Information Science and Engineering, Xiamen University, Xiamen 361005, China
[2]Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122, USA
[3]College of Engineering, Qatar University, Doha, Qatar

Corresponding author: Zhibin Gao (gaozhibin@xmu.edu.cn)

This work was supported in part by the Major Research Plan of the National Natural Science Foundation of China under Grant 91638204, in part by the 2015 National Natural Science Foundation of China under Grant 61401381, in part by the State Key Program of the National Natural Science Foundation of China under Grant 61731012, in part by the 2018 National Natural Science Foundation of China under Grant 61871339, and in part by the Key Laboratory of Digital Fujian on IoT Communication, Architecture and Safety Technology under Grant 2010499.

ABSTRACT With the growing popularity of the fifth-generation (5G) wireless systems and cloud-enabled Internet of Vehicles, vehicular cloud has been introduced as a novel mobile device computing mode, which enables vehicles to offload their computation-intensive tasks to neighbors. In this paper, we first present a 5G cloud-enabled scenario of vehicular cloud computing where a vehicular terminal works either as a service provider with idle computation resources or a requestor who has a computation-intensive task that can be executed either locally or offloaded to nearby providers via opportunistic vehicle-to-vehicle communications. Then, we study the following issues: 1) how to determine the appropriate offloading rate of requestors; 2) how to select the most appropriate computation service provider; 3) how to identify the ideal pricing strategy for each service provider. We address the above-mentioned problems by developing a two-player Stackelberg-game-based opportunistic computation offloading scheme under situations involving complete and incomplete information that primarily considers task completion duration and service price. We simplify the former case into a common resource assignment problem with mathematical solutions. For the latter case, Stackelberg equilibriums of the offloading game are derived, and the corresponding existence conditions are concretely discussed. Finally, a Monte-Carlo simulation-based performance evaluation shows that the proposed methods can significantly reduce the task completion duration while ensuring the profit of service providers, thus achieving mutually satisfactory computation offloading decisions.

INDEX TERMS Computation offloading, 5G cloud-enabled IoV, vehicle-to-vehicle communication, Stackelberg equilibrium.

I. INTRODUCTION

With the rapid development and widespread popularity of Internet of Things (IoT), smart mobile devices such as smartphones and smart vehicles with network access have been experiencing a boom in number and variety of industry and manufacturing. Industrial IoT (IIoT) is known as a type of industrial internet that incorporates many modern technologies including big data, machine-to-machine (M2M) communication, automation, and machine learning. The evolution of the automobile industry is a core reason why IIoT development has accelerated to an all-time high. In particular, Internet of Vehicles (IoV) can be regarded as a branch of IIoT that has recently attracted great attention and will gradually sweep the market as smart vehicles are evolving into a mainstream commodity. By the year of 2020, 97 million vehicles will be manufactured across the automobile industry worldwide, most of which will be wirelessly connected [1]. Recently, the advent of the Fifth-Generation (5G)-related technologies [2] has brought enticing prospects to IoV concerning different communication modes such as vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I)

The associate editor coordinating the review of this manuscript and approving it for publication was Chuan Huang.

communication [3], as well as the proliferation of high-data-rate applications. Furthermore, many technological advancements such as on-board cameras and embedded sensors have inspired new types of applications with advanced features, some of which are computation-insensitive such like traditional IPTV [4] that mainly supports video entertainments in IoV, but more of which are computation-intensive which demand complex computing and analysis, such like personalized navigation [5], Augmented Reality (AR) [6] and some safety services.

IoV faces many challenges principally related to vehicular mobility, limited computational resource and capability of on-board equipment and inefficient management mechanisms [7], [8]. Fortunately, the paradigm of cloud-enabled IoV supported by Mobile Cloud Computing (MCC) technology, allows vehicular terminals to borrow computation and storage resources from cloud computing centers or share excess resources with each other so that to augment the computational capabilities of mobile devices with emerging resource-hungry applications and further effectively surmount the aforementioned limitations. A promising solution involves offloading computation-intensive tasks to service providers, which efficiently alleviates the resource constraints of vehicles by migrating part of the workloads corresponding to an application to resource-rich surrogates, such as location-fixed cloud computing centers and nearby vehicles. The latter case is considered an innovative and flexible ad-hoc offloading scheme that can further be entitled as Vehicular Cloud (VC) [9], [10] by leveraging opportunistic V2V communication technology [11]. Compared with location-fixed clouds, VCs possess the advantages of infrastructure independency and economic efficiency. In a VC computing scenario, a vehicular terminal can flexibly play the part of either a service node providing computing services while charging a certain commission or a task node who has a computation request. A task vehicular terminal can offload a certain proportion of the computation-intensive tasks to one or multiple service vehicular terminals[1] when V2V connections become available. Computation results can be transferred to the task vehicular terminal either directly, or via a V2V routing path, or uploaded to the Road Side Units (RSUs) for future delivery with V2V disconnection. One significant computation offloading strategy is to implement reasonable deployment of cloud resources to meet user requirements and improve Quality of Service (QoS) while achieving mutually beneficial solutions for both parties.

We address the following technical and economical challenges faced with the proposed computation offloading framework in 5G cloud-enabled IoV as follows:

i) How to select the appropriate service provider?
ii) How to decide the proportion of data that should be offloaded to meet the requirements of service requestors in terms of task completion duration and monetary cost?
iii) What is the optimal pricing strategy for service providers through fully considering service profit and cost?

To address these issues while coping with the tremendous demands for high data transmission rates and strong computational capabilities in IoV, as well as the instabilities introduced by varying network topologies, in this paper, a novel opportunistic V2V computation offloading scheme based on game theory is established under two different circumstances. The main contributions of this paper are summarized as follows:

- A 5G cloud-enabled IoV framework is presented under which computation offloading can happen among vehicular terminals with different identities of service requestor and provider, to greatly support high user mobility while relieving the limitations on signal coverage of location-fixed cloud servers.
- Computation offloading can happen among mobile nodes where task vehicular terminals and service vehicular terminals can communicate with each other through one-hop V2V channels. Although limitations brought by opportunistic communication have impact on computation offloading procedure, it also greatly improves spectrum efficiency and support large-scale end-to-end performance in short-distance communications.
- We formulate the computation offloading scheme and pricing strategy as a Stackelberg game that can effectively model interactions between vehicles and take full consideration on vehicular mobility models, V2V contact durations, computational capabilities, channel conditions as well as service costs under both circumstances of complete information and incomplete information, where the former case is simplified into a common resource assignment problem (RAP) with mathematical solutions. For the latter case, we obtain the Stackelberg equilibrium and corresponding existing conditions.
- We provide insights based on the performance evaluation through Monte-Carlo simulation. Results show that the proposed offloading scheme can effectively reduce task completion duration and improve QoS while preserving benefits for service providers, achieving mutually satisfactory results. Furthermore, potential competitions among different service providers are also discussed according to sufficient simulations.

The rest of this paper is organized as follows. We conduct an intensive survey of related work in Section II. Then, we present the problem overview and system models in Section III. In Section IV, we propose the opportunistic computation offloading game under circumstances of complete information and incomplete information. We next analyze the performance of the method and present the numerical results in Section V. Finally, we summarize the study and discuss our future work in Section VI.

[1]In this paper, we mainly focus on choosing one appropriate service provider.

II. RELATED WORK

Computation offloading is one of the most challenging and hot topics in mobile cloud computing environments [12], especially when latency and cost are considered. In this section, we conduct intensive and comprehensive investigation of computation offloading issues in cloud-enabled IoV.

The first paradigm widely used in cloud-enabled IoV was the remote cloud [13] but resulted in significant transmission delays, serious degradation, and low reliability [14] due to variability in network topologies, capacity limitations of wireless networks as well as delay fluctuations in transmission on backhaul and backbone networks. To overcome the aforementioned problems, Multi-access Edge Computing (MEC) technology was established, which aims at converging telecommunication and IT services [15], providing a cloud computing platform at the edge of pervasive Radio Access Networks (RAN) in close proximity to vehicles, bringing business localization along with open wireless network capabilities. In order to reduce latency and transmission costs, literature [16] studies the effectiveness of the computation transfer strategies with V2I and V2V communication modes where the tasks are adaptively offloaded to the MEC servers through direct uploading or predictive relay transmissions. Nonetheless, they didn't consider available on-board resources and remote areas without RSU coverage. To reduce the latency of the computation offloading of vehicles, a multiple vehicles computation offloading game was studied in vehicular edge networks [17]. However, MEC servers can still suffer from resource constraints as well as the signal coverage limitations of RSUs [7], [11], moreover, price can not be ignored as another cost due to that cloud computing is a pay-as-you-go service.

In VC-based systems, each vehicular terminal can access cloud servers and utilize the pay-as-you-go service for its own purpose.[2] Through applicable offloading, the resources of vehicular users are dynamically scheduled on demand [11]. The task completion duration can be effectively reduced with better QoS, meanwhile, service providers' benefits can be protected by establishing an appropriate pricing mechanism.

Although many studies have investigated device-to-device (D2D) computation offloading for smartphones, few efforts have been put forward into V2V computation offloading scheme in cloud-enabled IoV. For the purpose of fully experiencing high-rate broadband multimedia services and prolonging the battery life of smartphones, Feng *et al.* [18] developed a computation offloading scheme based on D2D communications where mobile devices can offloaded tasks to nearby neighbors. Literature [7] proposed an opportunistic task scheduling mode assisted by mobile cloudlets and performed a detailed analysis but did not consider task execution duration. Chen *et al.* [19] proposed a D2D Crowd framework for mobile edge computing, where a crowd of devices leverage network-assisted D2D collaboration15 for computation and communication resource sharing with key objective of achieving energy-efficient collaborative task executions at the network edge for mobile users. In cloud-enabled IoV, literature [20] investigated a cloud-assisted vehicular network architecture in which each cloud has its own features, and a corresponding optimal scheme was obtained by solving a Semi-Markov Decision Process aimed at maximizing the system's expected average reward. To improve network capacity and system computing capability, [21] extended the original cloud radio access network (C-RAN) to integrate local cloud services to provide a low-cost, scalable, self-organizing, and effective solution called enhanced C-RAN with essential technologies of D2D and heterogeneous networks based on a matrix game theoretical approach. Although several studies have solved the problem of high-efficiency computation offloading to some extent, the limitation of the vehicular cloud can still remain given strict requirements of inter-contact duration based on the premise that a pair of service requestor and provider shall have enough time to offload the computation-intensive tasks and applicable pricing strategy, otherwise, achieving optimal offloading results will prove difficult. To the best of our knowledge, we are one of the few works that study computation offloading between moving vehicles.

III. PROBLEM OVERVIEW AND SYSTEM MODEL

In this section, we first provide an overview of the mobile computation offloading problem. Then, the scenario employed in this paper and basic assumptions are described. Finally, the traffic model of vehicular terminals, communication model between two vehicles, and the task computation model are respectively introduced.

A. PROBLEM OVERVIEW

Two roles are mainly existed in the mobile computation offloading problem, task vehicular terminals and service vehicular terminals. For the simplicity of notation, the "vehicular terminal" is written as VT for the rest of this paper. A task VT is a service requestor who has a computation-intensive task, which can be modeled as a 7-tuple $Tn = \{P, f^T, v, \theta, D, \lambda, T^{\max}\}$, where P describes the location, f^T is defined as the computational capability expressed by CPU cycle/s, v and θ denote the velocity (km/h) and angle between the moving direction and the horizontal line of the road, respectively. The symbol D, λ, and $T^{\max}$ denote the data size (bit) of the computation-intensive task, the proportion of computational data that a task VT decides to offload, and the tolerant task duration. Task VTs may be faced with problems of insufficient computation resources, non-ideal computational capabilities and channel instabilities brought by mobility.

A service VT can be described as a 5-tuple $Sn = \{P, f^S, v, \theta, p\}$, similar to the task VT described above where f^S represents the computational capability on the service VT. Let the price for processing one bit of data be $p \in \{c + \Delta p, c + 2\Delta p, \ldots, p^{\max}\}$, where c indicates the fixed cost including communication cost and facility cost (e.g., wear

[2] In this paper, solving computation-intensive tasks is our main purpose.

and tear of smart on-board equipment) and is mapped into monetary cost with units such as US Dollars. The minimal granularity of price is denoted as Δp (e.g., one dollar can be seen as a minimal granularity where the price can only be an integral multiple of a dollar) and $p^{\max} = c + \beta \Delta p$, which represents the maximum unit price of market regulation where β is defined as a positive integer. Due to more powerful computational capability and adequate computation resources, a service provider is expected to offer computing service and request a reasonable commission.

In the mobile computation offloading problem, each task VT aims to choose an applicable service provider by communicating with service VTs in his communication range, and determine the appropriate value of λ based on a given p to ensure rapid completion of the task, that is, reducing the task completion duration as much as possible. Moreover, each service VT can also have opportunities adjusting service price p to guarantee and increase his own benefits, while achieving better QoS.

B. SCENARIO AND MODELS

We consider a dynamic traffic scenario of a two-way straight road of finite length. It is worth noting that a VT can either be a service requestor or a service provider based on the sufficient degree of the computation resources and computational capability. In a particular snapshot as we concern, the identity of each vehicle is stable and absolute, to make subsequent analysis more efficient. Consider $N + M$ VTs where N are task VTs owning computation-intensive tasks that can be either executed locally or partially offloaded to an appropriate service node. The remaining M are regarded as service VTs with relatively sufficient resources and strong computational capabilities to provide services. The resources of task VTs and service VTs are virtualized as Resource Blocks (RBs), such as CPU cycles, and aggregated into a resource pool. Several RSUs exist alongside the road, each of which has a coverage radius defined as R' in which vehicles periodically report various information such as location and velocity through V2I channels. Any two RSUs can communicate with each other directly through wired links.

Several basic assumptions in this paper are listed as follows:

- The velocity and direction of every VT remains unchanged during a short offloading period (e.g., a few seconds) [1].
- One service VT can provide services for several task VTs within his communication range while ignoring the interference and interactions among these services.
- Computation results can be delivered non-destructively through V2V or V2I channels, and the feedback duration can be ignored [22], [23].

1) TRAFFIC MODEL

We denote the set of N task VTs and M service VTs as $T = \{Tn_i | i \in \{1, \ldots, N\}\}$ and $S = \{Sn_j | j \in \{1, \ldots, M\}\}$. The velocity of every VT in the above two sets is defined

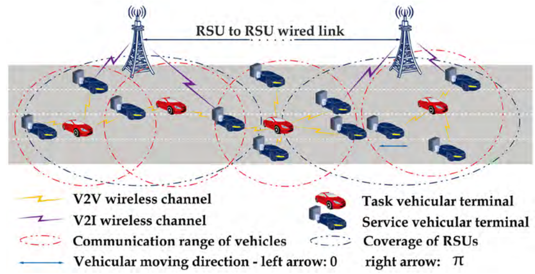

FIGURE 1. The 5G cloud-enabled IoV scenario.

as $v_k \in [v_{\min}, v_{\max}], k \in \{1, \ldots, N + M\}$, in which $v_{\min}$ and $v_{\max}$ are the minimum and maximum velocity of a VT. For the angle $\theta_k \in \{0, \pi\}$, both task VTs and service VTs maintain uniform linear motion during computation offloading [1]. Suppose that the initial location of VT $k, k \in \{Tn_1, \ldots, Tn_N, Sn_1, \ldots Sn_M\}$ at time $t_0 = 0$ is $P_k(t_0) = (x_k, y_k)$; the location after duration Δt can be written as $P_k(t_0 + \Delta t) = (x_k^{\Delta t}, y_k^{\Delta t})$.

$$\begin{cases} x_k^{\Delta t} = x_k + v_k \Delta t \cos \theta_k \\ y_k^{\Delta t} = y_k + v_k \Delta t \sin \theta_k \end{cases} \tag{1}$$

2) COMMUNICATION MODEL

Each pair of task VTs and service VTs can opportunistically communicate only when they are within the communication range R, which can be defined as $\left\| P_{Tn_i}(t) - P_{Sn_j}(t) \right\| \leq R$, where $P_{Tn_i}(t)$ and $P_{Sn_j}(t)$ are the locations of task VT i and service VT j at time t. For simplicity, the data transmission rate (bit/s) r_j between Tn_i and Sn_j is considered a fixed average value related to several factors including channel condition, packet loss retransmission, transmission power, and outage probability [24], [25].

Based on the traffic model discussed above, assuming that Tn_i has a computation-intensive task at time $t_0 = 0$, the set of candidate service VTs within the communication range of Tn_i can be denoted as $S_i = \{Sn_a | \, a \in \{1, \ldots, m\}\}$. Correspondingly, the Euclidean distance between Tn_i and Sn_a at time Δt can be calculated as (2):

$$d_{Sn_a}(\Delta t) = \sqrt{(x_{Tn_i}^{\Delta t} - x_{Sn_a}^{\Delta t})^2 + (y_{Tn_i}^{\Delta t} - y_{Sn_a}^{\Delta t})^2} \tag{2}$$

As for the straight road, due to that every vehicle maintains uniform linear motion for a small period of time during offloading [1], the contact duration Δt_a of Tn_i and Sn_a can be ascertained by solving the system of linear equations with one unknown in (3):

$$R^2 = (x_{Tn_i}^{\Delta t_a} - x_{Sn_a}^{\Delta t_a})^2 + (y_{Tn_i}^{\Delta t_a} - y_{Sn_a}^{\Delta t_a})^2 \tag{3}$$

3) COMPUTATION MODEL

The task completion duration consists of the time consumed by three procedures: 1) the delivery of task contents; 2) task execution; and 3) task resulting feedback. Considering a situation where a task is computation-intensive and the V2V mode is unavailable, the duration of local processing for the

TABLE 1. Variables and notations.

Variables	Explanations
P	Position of VT
R	Communication range of VT
N, M	Number of task VTs and service VTs
v	Velocity of VT
θ	Moving direction of VT
K	The mapping between data size and CPU cycles
D	Data size of a computation-intensive task
r	Data transmission rate between task VT and service VT
f^T, f^S	Computational capability of task VT and service VT
t_i^L	The task completion duration of Tn_i on local side
t_i^C	The task completion duration of Tn_i in collaborative computing mode
c	Unit cost of service VT for executing one bit of data
Δp	Minimal granularity of service price
p	Unit price for executing one bit of data
ω	Resulting feedback duration
α_1, α_2	Weight parameters
λ	Offloading rate (bit)
U_T, U_S	Utility of task VT and service VT
S_i	Set of service VTs within communication range of Tn_i

task in Tn_i is denoted by $t_i^L = KD_i/f_i^T$, where f_i^T represents the local processing speed and t_i^L is typically less than the tolerant duration $T_i^{\max}$ but results in an unsatisfactory task completion duration, K is denoted as a constant describing the mapping between data size and CPU cycles [22]–[24].

Let λ_a be the proportion of data that Tn_i chooses to offload to service VT $Sn_a \in S_i$, the task duration executed in parallel by two VTs can be computed as (4), where ω expresses the resulting feedback duration that is the same for all $Tn_i, i \in \{1, \ldots, N\}$ due to that the size of the task result is much smaller than the input data size and can be further neglected:

$$t_i^C = \max(K\lambda_a/f_a^S + \lambda_a/r_a, K(D_i - \lambda_a)/f_i^T) + \omega \qquad (4)$$

Obviously, the inequation $t_i^C \leq t_i^L$ will always be true. The variables and notations used in this paper are summarized in Table 1.

IV. THE OPPORTUNISTIC MOBILE COMPUTATION OFFLOADING GAME

We design a computation offloading Stackelberg game, which is a non-cooperative and leader-follower based approach that can reasonably and efficiently describe the sequential interactions among vehicles of different identities under circumstances of complete information and incomplete information. Furthermore, due to that the game and solution are appropriate for every pair of task nodes as well as service nodes, the subscript referring to one specific VT will be ignored. Consider an offloading game composed of two players: a task VT ($\mathcal{T}$) and one of the service VTs ($\mathcal{S}$) within the communication range of the task VT. We denote the computation offloaded to the service VT as $\lambda \in [0, D]$ (bit). If $\lambda = 0$, the task VT processes the entire task locally, whereas $\lambda = D$ means that all data are processed by the service VT. The offloading rate can be chosen as a tradeoff between the task completion duration and the commission that task VT has to pay for service VT.

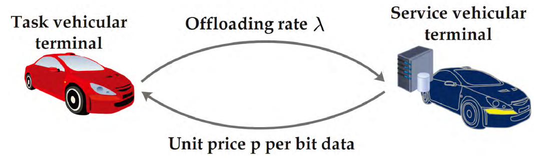

FIGURE 2. Illustration of the mobile computation offloading game.

As mobile cloud computing is considered as a kind of pay-as-you-go service, service VTs can provide computing service on the premise of obtaining certain commissions from the task VTs, as monetary benefits. An illustration of the mobile computation offloading game is shown in Fig. 2.

A. UTILITY FUNCTIONS

We consider a mobile computation offloading Stackelberg game denoted by $\mathcal{G}$ in which the service VT chooses its quoted price per bit $p \in \{c + \Delta p, c + 2\Delta p, \ldots, p^{\max}\}$. Then the task VT decides whether to offload and how much data will be offloaded based on the quoted price. The data transmission rate (bit/s) and the inter-contact duration between a task VT and a service VT are denoted as r and Δt, respectively. What needs to be emphasized is that the actual amount of data allowed to be offloaded depends on the smaller value between D and $r\Delta t$. For simplicity, we adjust the value $\lambda \in [0, \min(D, r\Delta t)]$.

For a task VT, the revenue function is denoted as the duration saved from vehicular cloud computing mode:

$$\alpha_1(KD/f^T - \max(K\lambda/f^S + \lambda/r, K(D-\lambda)/f^T))$$

where f^T and f^S represent the respective computational capability of the two players. The first term and second term represent t^L and t^C respectively as mentioned in *Computation Model*. The cost function is mainly regarded as the total payment for the computation service: $\alpha_2(p \cdot \lambda)$. Above all, the utility function of a task VT can be shown as (5):

$$U_T(p, \lambda) = \alpha_1(KD/f^T - \max(K\lambda/f^S + \lambda/r, K(D-\lambda)/f^T)) - \alpha_2(p\lambda) \qquad (5)$$

where α_1 and α_2 are the positive weight coefficients that satisfy $\alpha_1 = 1 - \alpha_2$ depending on the personal preferences of task VTs.

Accordingly, the revenue function of a service VT is calculated as $p \cdot \lambda$; the cost function can be similarly defined as $c \cdot \lambda$. The utility function of a service VT can be written as:

$$U_S(p, \lambda) = p\lambda - c\lambda \qquad (6)$$

In summary, we consider the mobile computation offloading Stackelberg game denoted by $\mathcal{G}^{SE} = \langle\{\mathcal{T}, \mathcal{S}\}, \{\lambda, p\}, \{U_T, U_S\}\rangle$ in which the service VT first chooses its quoted price p, and then the task VT chooses the offloading rate λ based on the observed p. For simplicity, the follower is assumed to accurately obtain the action of the leader.

The Stackelberg equilibrium of the offloading game denoted by (p^{SE}, λ^{SE}) is given as follows:

$$\lambda^{SE}(p) = \arg\max_{\lambda\in[0,\min(D,r\Delta t)]} U_T(p,\lambda) \tag{7}$$

$$p^{SE} = \arg\max_{p\in(c,p^{\max}]} U_S(p,\lambda^{SE}(p)) \tag{8}$$

In the Stackelberg equilibrium of the offloading game, a service VT acts as a leader and can first choose the quoted price to maximize its utility given by (8) while considering the response of the task VT as the follower. Then, a proportion of offloading data is determined to maximize the utility in (7) by the task VT based on the observed quoted price.

Before discussing the analysis, we briefly examine function $t^C = \max(K\lambda/f^S + \lambda/r, K(D-\lambda)/f^T)$ shown in (4). For the physical significance of t^C, the task completion duration depends on the larger processing latency between the task VT and service VT. In the case of ignoring any other constraints, t^C can reach its minimum value when $K\lambda/f^S + \lambda/r = K(D-\lambda)/f^T$. Let the value that makes $K\lambda/f^S + \lambda/r = K(D-\lambda)/f^T$ be $\lambda^0 = KrDf_S/((f^T+f^S)Kr + f^Tf^S)$.

B. MOBILE OFFLOADING COMPLETE INFORMATION GAME (MOCIG)

In this work, when both players have accurate information about the features, strategies, and utility functions of each other, the task VT can fully grasp the unit cost c_a of each service VT $Sn_a \in S_i$, which causes the service VTs to lose their bargaining capabilities. A service VT Sn_a can only possibly be chosen if the quoted price $p_a = c_a + \Delta p$. Consequently, the above-mentioned mobile offloading complete information game can be simplified into a common RAP with mathematical solutions.

For a task VT Tn_i, the set of service VTs in its communication range is denoted by $S_i = \{Sn_a | a \in \{1,\ldots,m\}\}$. The utility functions of Tn_i can be represented as a set $UTn_i = \{U_{Ta}(c_a+\Delta p, \lambda) | a \in \{1,\ldots,m\}\}$ where

$$U_{T_a}(c_a+\Delta p, \lambda_a) = \alpha_1(KD_i/f_i^T - \max(K\lambda_a/f_a^S + \lambda_a/r_a, K(D_i-\lambda_a)/f_i^T)) \tag{9}$$

The most appropriate service VT and offloading data size can be obtained by finding the largest value greater than zero among the elements in set UTn_i. The relevant service VT and offloading data size will be the appropriate solution for the game, and the algorithm is shown in Table 2.

The calculation of $U_{Ta}^{SE} = \arg\max_{\lambda_a\in[0,\min(D_i,r_a\Delta t_a)]} U_{Ta}(c_a + \Delta p, \lambda_a)$ is described in Table 3.

C. MOBILE OFFLOADING INCOMPLETE INFORMATION GAME (MOIIG)

In realistic applications, participants in a game are usually not able to fully know the features, strategies, and utility functions of each other. In the incomplete information scenario, we consider the Stackelberg equilibrium of the game $\mathcal{G}^{SE} = \langle\{\mathcal{T},\mathcal{S}\}, \{\lambda, p\}, \{U_T, U_S\}\rangle$mentioned before as follows.

TABLE 2. Algorithm of mobile offloading complete information game.

Steps	Operations
1	parameters initialization;
2	determine the service set $S_i = \{Sn_a \mid a \in \{1,\ldots,m\}\}$;
3	$\forall a \in \{1,\ldots,m\}$, calculate Δt_a by (3);
4	$\forall a \in \{1,\ldots,m\}$, calculate $U_{Ta}^{SE} = \arg\max_{\lambda_a\in[0,\min(D_i,r_a\Delta t_a)]} U_{Ta}(c_a+\Delta p,\lambda_a)$;
5	***if*** $\exists U_{Tb}^{SE}, b \in \{1,\ldots,m\}, s.t. \forall a \in \{\{1,\ldots,m\}\backslash b\}$, $U_{Tb}^{SE} \geq U_{Ta}^{SE}$ and $U_{Tb}^{SE} > 0$, ***then*** Sn_b and λ_b will be the appropriate provider and the offloading data size, proceed to step 7;
6	***else*** the Tn_i will process all data locally, proceed to step 7;
7	***end if***;
8	***end***

TABLE 3. The calculation of U_{Ta}^{SE}.

Steps	Operations
Input	$D_i, \Delta t_a, r_a, c_a, \lambda_a^0$ *and* Δp
Output	λ_a
1	***if*** $r_a\Delta t_a < \lambda_a^0$;
2	***if*** $\dfrac{\partial U_{T_a}(c_a+\Delta p,\lambda_a)}{\partial\lambda_a} > 0$;
3	$\lambda_a = r_a\Delta t_a$;
4	***else*** $\lambda_a = 0$;
5	***end if***;
6	***else*** $r_a\Delta t_a \geq \lambda_a^0$;
7	***if*** $\dfrac{\partial U_{T_a}(c_a+\Delta p,\lambda_a)}{\partial\lambda_a} > 0$;
8	$\lambda_a = \lambda_a^0$;
9	***else*** $\lambda_a = 0$;
10	***end if***;
11	***end if***;
12	***end***

Theorem 1: The mobile computation offloading game has a SE$(p^{\max}, r\Delta t)$ if

$$\begin{cases} r\Delta t < \lambda^0 & \text{(10a)} \\ \partial U_T(p^{\max},\lambda)/\partial\lambda > 0, \quad \lambda\in[0,r\Delta t] & \text{(10b)} \end{cases}$$

Proof: When $\lambda \geq 0$, we have

$$\partial U_S/\partial p = \lambda \geq 0 \tag{11}$$

According to (11), we have

$$\begin{aligned} &\arg\max_{p\in\{c+\Delta p,\ldots,p^{\max}\}} U_S(p,\lambda) \\ &= U_S(p^{\max},\lambda) \\ &\geq U_S(p^{\max}-\Delta p,\lambda) \geq \ldots \geq U_S(c+\Delta p,\lambda) \end{aligned} \tag{12}$$

When (10a) holds, we have

$$U_T(p^{\max},\lambda) = \left(\alpha_1 K/f^T - \alpha_2 p^{\max}\right)\lambda, \quad \lambda\in[0,r\Delta t] \tag{13}$$

When (10b) holds, we have

$$\begin{aligned} &\arg\max_{\lambda\in[0,r\Delta t]} U_T(p^{\max},\lambda) \\ &= U_T(p^{\max},r\Delta t) > U_T(p^{\max},r\Delta t-1) \\ &> U_T(p^{\max},2) > \ldots > U_T(p^{\max},1) \end{aligned} \tag{14}$$

Thus, we have a $SE(p^{\max}, r\Delta t)$ if (10a) and (10b) holds. □

Remark 1: In the Stackelberg game $\mathcal{G}$, the task VT decides the offloading proportion based on the observed quoted price of the service VT. When the task VT is insensitive to payment and the actual amount of data that can be offloaded is rather small (smaller than λ^0), the service VT decides to charge $p^{\max}$ per bit, and the task VT chooses to offload the proportion of the task that can be transmitted during contact.

Theorem 2: The mobile computation offloading game has a $SE(p^{\max}, \lambda^0)$ if

$$\begin{cases} r\Delta t \geq \lambda^0 & (15a) \\ \partial U_T(p^{\max}, \lambda)/\partial\lambda > 0, \quad \lambda \in [0, \lambda^0] & (15b) \end{cases}$$

Proof: When $\lambda \geq 0$, we have equation set (11)–(13). When (15a) holds, we have

$$\partial U_T(p^{\max}, \lambda)/\partial\lambda = -\alpha_1(K/f^S + 1/r) - \alpha_2 p^{\max} < 0, \quad \lambda \in [\lambda^0, \min(0, r\Delta t)] \quad (16)$$

If (15b) holds, we have

$$\arg\max_{\lambda\in[0,\min(D,r\Delta t)]} U_T(p^{\max}, \lambda) = U_T(p^{\max}, \lambda^0) \quad (17)$$

where $U_T(p^{\max}, \lambda^0) > U_T(p^{\max}, 1) > \ldots > U_T(p^{\max}, \lambda^0 - 1)$ and $U_T(p^{\max}, \lambda^0) > U_T(p^{\max}, \lambda^0 + 1) > \ldots > U_T(p^{\max}, \min(D, r\Delta t))$.

Thus, we have a $SE(p^{\max}, \lambda^0)$ if (15a) and (15b) hold. □

Remark 2: In the Stackelberg game $\mathcal{G}$, when a task VT is insensitive to payment and the actual amount of data that can be offloaded is rather large (larger than λ^0), the service VT decides to charge $p^{\max}$ per bit, and the task VT can reach an ideal choice when it decides to offload data λ^0.

Theorem 3: The mobile computation offloading game has a $SE(\lfloor K\alpha_1/\Delta p f^T\alpha_2\rfloor \cdot \Delta p, r\Delta t)$ if

$$\begin{cases} r\Delta t < \lambda^0 & (10a) \\ \partial U_T(p^{\max}, \lambda)/\partial\lambda < 0, \quad \lambda \in [0, r\Delta t] & (18a) \\ \left\lfloor K\alpha_1/\Delta p f^T\alpha_2 \right\rfloor \cdot \Delta p > c & (18b) \end{cases}$$

Proof: As mentioned above, under the cases of **Theorem 1** and **Theorem 2**, we have (11) and (12). □

If (10a) and (18a) each holds, we have

$$\arg\max_{\lambda\in[0,r\Delta t]} U_T(p^{\max}, \lambda) = U_T(p^{\max}, 0) = 0 \quad (19)$$

When (10a), (18a), and (18b) hold, we have

$$\arg\max_{p\in\{c+\Delta p,\ldots,p^{\max}\}} U_S(p, \lambda) = U_S(\left\lfloor K\alpha_1/\Delta p f^T\alpha_2 \right\rfloor \cdot \Delta p, \lambda) \quad (20)$$

and

$$\begin{aligned} &\arg\max_{\lambda\in[0,r\Delta t]} U_T(\lfloor K\alpha_1/\Delta p f^T\alpha_2\rfloor \cdot \Delta p, \lambda) \\ &\quad = U_T(\lfloor K\alpha_1/\Delta p f^T\alpha_2\rfloor \cdot \Delta p, r\Delta t) \\ &\quad > U_T(\lfloor K\alpha_1/\Delta p f^T\alpha_2\rfloor \cdot \Delta p, r\Delta t - 1) \\ &\quad > U_T(\lfloor K\alpha_1/\Delta p f^T\alpha_2\rfloor \cdot \Delta p, r\Delta t - 2) \\ &\quad > \ldots > U_T(\lfloor K\alpha_1/\Delta p f^T\alpha_2\rfloor \cdot \Delta p, 1) \end{aligned} \quad (21)$$

Under the condition of holding (10a), (18a) and (18b), we have a $SE(\lfloor K\alpha_1/\Delta p f^T\alpha_2\rfloor \cdot \Delta p, r\Delta t)$.

Remark 3: When a service VT chooses to charge the maximum price, the task VT will not offload any data to the service VT due to the decreasing monotonicity of its utility function (the better choice for the task VT is to execute the computation task locally). The above situation will lead to a poor result for the service VT with no benefits. Consequently, the service VT will need to change the pricing strategy to increase its benefits to the greatest extent on the premise of a nonzero benefit while ensuring positive income for the task VT.

Theorem 4: The mobile computation offloading game has a $SE(\lfloor K\alpha_1/\Delta p f^T\alpha_2\rfloor \cdot \Delta p, \lambda^0)$ if

$$\begin{cases} r\Delta t \geq \lambda^0 & (15a) \\ \partial U_T(p^{\max}, \lambda)/\partial\lambda < 0, \quad \lambda \in [0, \lambda^0] & (22a) \\ \left\lfloor K\alpha_1/\Delta p f^T\alpha_2 \right\rfloor \cdot \Delta p > c & (18a) \end{cases}$$

Proof: Similar to that of Theorem 3. □

*Remark 4: When the actual amount of data that can be offloaded is rather large (15a), similar to **Theorem 3**, the service VT has to modify its strategy and the task VT is in favor of offloading λ^0 bit data.*

Theorem 5: The mobile computation offloading game has a $SE(c + \Delta p, 0)$ if

$$\begin{cases} \partial U_T(p^{\max}, \lambda)/\partial\lambda < 0, \lambda \in [0, \lambda^0] & (22a) \\ \left\lfloor K\alpha_1/\Delta p f^T\alpha_2 \right\rfloor \cdot \Delta p \leq c & (23a) \end{cases}$$

Proof: When (22a) and (23a) hold, we have

$$U_S\left(\left\lfloor K\alpha_1/\Delta p f^T\alpha_2 \right\rfloor \cdot \Delta p, \lambda\right) \leq 0 \quad (24)$$

and

$$\arg\max_{\lambda\in[0,\min(D,r\Delta t)]} U_T(c + \Delta p, \lambda) = U_T(c + \Delta p, 0) = 0 \quad (25)$$

Under the condition of holding (22a) and (23a), we have a $SE(c + \Delta p, 0)$. □

Remark 5: The service VT will always choose the strategy that benefits itself. When a minimum quoted price cannot meet the requirements of the task VT, the service VT will not provide any computing service and the task VT will decide to execute the computation-intensive task locally.

V. PERFORMANCE EVALUATION

In this section, we evaluate the performance of the proposed mobile offloading scheme through Monte Carlo simulation on MATLAB platform. Notably, the offloading scheme of computation-intensive tasks in application layer is not limited by and also put no restrictions on the underlying protocols. Three baseline methods are introduced as follows, and compared with the proposed methods in this simulation.

- **Local Computing Scheme (LCS)**: where each task VT processes the computation-intensive task locally without offloading.

TABLE 4. Simulation parameters.

Parameters	Value	Parameters	Value
P	Uniform distribution	R(m)	200
N	10	M	5–30
Road width(m)	20	Road length(m)	1000
v(km/h)	80–110	θ	$\{0, \pi\}$
K	18000 [23]	D(MB)	4–7
f^T(CPU cycle/s)	4*10^9–4*10^{10}	r(MB/s)	3–6
f^S(CPU cycle/s)	1*10^{11}–7*10^{11}	Δp	1*10^{-9}
c	5*10^{-8}–9*10^{-8}	β	1–100
ω	0	α_1,α_2	0–1

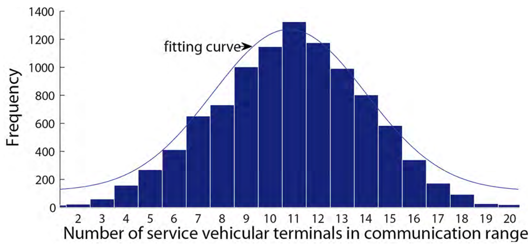

FIGURE 3. Histogram describes the number of service VTs that can communicate with one task VT.

- **First Come First Serve (FCFS)**: where each task VT chooses the first accessing service VT and considers λ^0 as the offloading rate with p^{max} as the quoted price.
- **Fastest Processing Scheme (FPS)**: where each task VT chooses the service VT with the most powerful computational capability, and considers λ^0 as the offloading rate with p^{max} as the quoted price.

We first introduce the simulation condition as well as parameters and then the performance evaluation and discussions are presented.

A. PARAMETER SETTING

In a map of a finite-length straight road with four lanes, a VT will return once it reaches the road boundary. Three RSUs are uniformly spaced alongside the road with a coverage radius of $R' = 300$ m. We have compared the performance of MOCIG, MOIIG, and aforementioned baseline methods on the task completion duration, the knockdown unit price, the value of utility functions, as well as the impact on different value of α_1 and α_2. In order to prove the generalization of the proposed methods and cover as many as possible traffic conditions, at least 1000 simulations have been conducted to verify the statistical characteristics and before each simulation, some of the parameters will be reset as Table 4.

B. SIMULATION RESULTS

Fig. 3 shows the statistical histogram describes the frequency of service VT numbers that can communicate with one task VT through 1000 simulations. We clearly see that the Monte Carlo Simulation can fully consider various traffic conditions that may appear during vehicles' moving and has favorable generalization characteristics that approximate a Gaussian distribution with a mean value of $\mu = 11$, around which can be seen as the most common situations.

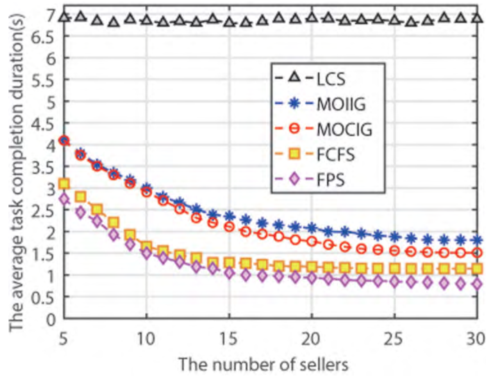

FIGURE 4. Performance on task completion duration ($N = 10$, $\alpha_1 = \alpha_2 = 0.5$).

We ran 1000 simulations for each value of M from 5 to 30 and analyzed the average task completion duration of 10 task VTs and the average unit knockdown price of service VTs, in the simulation environment. As shown in Fig. 4, the data executed locally without computation offloading in LCS takes much more time than vehicular cloud computing mode. The average task completion duration decreases with increasing density of service VTs in the proposed scenario due to task VTs having more options and being inclined to choose service providers that are more favorable to themselves in both MOCIG and MOIIG. In FCFS and FPS mode, the task duration is less than the proposed two methods due to offloading rate λ^0 but results in non-ideal utilities as shown in Fig. 6 without a bargain procedure. In MOCIG mode, service providers have no bargaining capabilities and can only be passively accepted; the task completion duration is therefore slightly smaller than that in MOIIG mode due to less service cost. Similarly, the average knockdown unit price shown in Fig. 5 decreases following an increase in potential competition among service providers in both proposed methods. As for MOIIG mode, service providers can earn higher profits due to bargaining capabilities. In FCFS and FPS, quoted prices are set as p^{max} and no bargain procedure exists, which lead to the highest average unit knockdown prices and U_S but lowest U_T in Fig. 6, and can hardly bring both players with win-win situations.

In Fig. 6 and Fig. 7, we analyze the variation tendency in the average value of utility functions U_T and U_S along with the increasing number of service VTs. In Fig. 6, the curves describe the average utility of the task VTs in MOIIG and MOCIG which present rising tendency due to better options. As for MOIIG mode, task VTs need to pay more to service providers with bargaining capabilities, which accordingly results in a lower value of U_T compared with MOCIG mode. The average value of U_T in FCFS and FPS is far less than ones in the proposed methods. As for the utilities of service VTs shown in Fig. 7, the average value of U_S remains unchanged in MOCIG mode but clearly

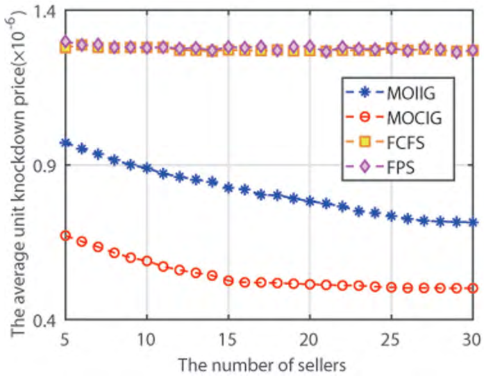

FIGURE 5. Performance on unit knockdown price ($N = 10$, $\alpha_1 = \alpha_2 = 0.5$).

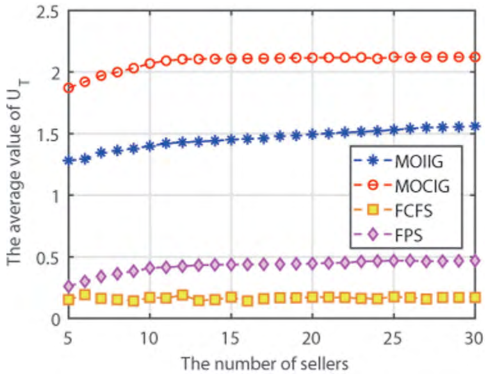

FIGURE 6. Performance on the value of U_T ($N = 10$, $\alpha_1 = \alpha_2 = 0.5$).

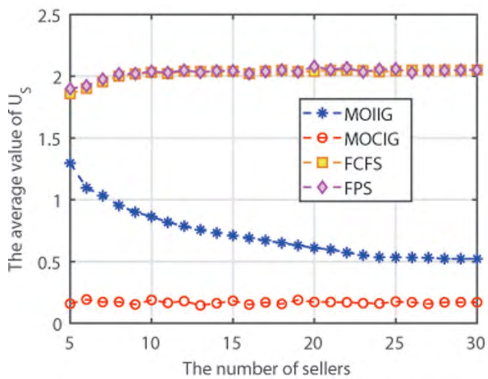

FIGURE 7. Performance on the value of U_S ($N = 10$, $\alpha_1 = \alpha_2 = 0.5$).

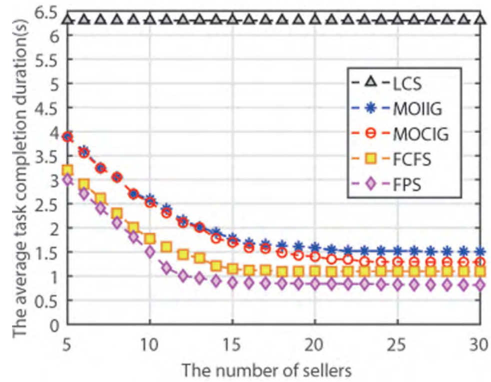

FIGURE 8. Performance on task completion duration ($N = 1$, $\alpha_1 = \alpha_2 = 0.5$).

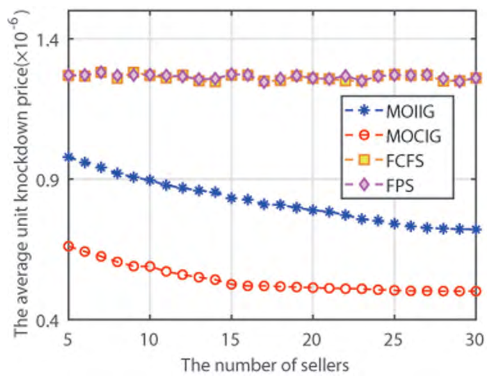

FIGURE 9. Performance on unit knockdown price ($N = 1$, $\alpha_1 = \alpha_2 = 0.5$).

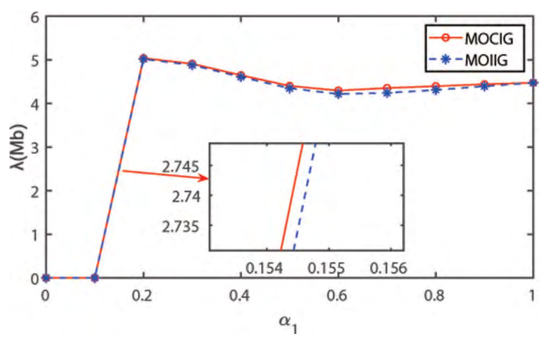

FIGURE 10. Relationship between λ and α_1 ($N = 1$).

declines in MOIIG mode as the number of service providers increases owing to more and more competitors who can also provide computing services. Without a bargaining procedure, the U_S in FCFS and FPS could be much larger for service VTs.

In Fig. 8 and Fig. 9, we mainly pay attention to the average task completion duration and unit knockdown price of one specific task requestor with increasing service provider density. In other words, we simulate various situations that may occur as the vehicle moves forward. The task completion duration without offloading in LCS is obviously much larger than that in MOCIG, MOIIG as well as FCFS and FPS modes which is similar in Fig. 4. With the availability of options, the average task completion duration of the specific task VT decreases by choosing an appropriate service provider who is more beneficial. For FCFS and FPS mode, the average task completion duration can be smaller due to the optimal offloading rate λ^0 without considering price but becomes non-ideal compared to the proposed methods given increasing service provider options. As can be seen in Fig. 9, a requestor tends to choose a cheaper service provider under the same conditions due to increasing competition among service VTs. Because of limited bargaining capabilities, the knockdown unit price in MOCIG can be far below that in MOIIG mode. Owing to the quoted price p^{max}, service providers in FCFS and FPS mode can obtain the highest benefits.

Fig. 10 shows the correlativity of offloading rate λ and weight coefficient α_1 of one specific task VT through 1000 simulations for each value in set $\alpha_1 \in \{0, 0.1, 0.2, \ldots, 1\}$. For the investigated task VT, the larger the value of α_1 , the more sensitive the task VT is to task completion duration. In one of the bounding cases $\alpha_1 = 0$, the task VT will not offload any data because the unwillingness to pay any commission ($\alpha_2 = 1$). With an increase in α_1, a service requestor becomes price insensitive and generally chooses a more appropriate offloading rate to achieve higher utility. When $\alpha_1 = 1$, the service requestor is willing to pay any price if the task completion duration can be reduced.

VI. CONCLUSION

In this paper, we propose a Stackelberg game based V2V computation offloading scheme in 5G cloud-enabled IoV under circumstances of complete information and incomplete information, while fully considering significant factors such as wireless channel conditions, vehicular mobilities, limitations on computational resources as well as capabilities and service prices, etc. Mathematical solutions show that the offloading game can always offer appropriate resource assignments and Stackelberg equilibriums, respectively. Moreover, simulation results demonstrate that the proposed scheme can efficiently reduce task completion durations while protecting service providers' benefits, achieving mutually satisfactory computation offloading decisions for both players. In our future work, we plan to consider some more challenging cases including multi-player cooperative game where one task can be assigned to several service providers, graph-based task offloading mechanism, as well as scenarios with more complicated vehicular mobilities.

REFERENCES

[1] Z. Su, Q. Xu, Y. Hui, M. Wen, and S. Guo, "A game theoretic approach to parked vehicle assisted content delivery in vehicular ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 7, pp. 6461–6474, Jul. 2017.

[2] N. J. Gomes *et al.*, "Boosting 5G through ethernet: How evolved fronthaul can take next-generation mobile to the next level," *IEEE Veh. Technol. Mag.*, vol. 13, no. 1, pp. 74–84, Mar. 2018.

[3] G.-P. A. W. Group. (2018). *A Study on 5G V2X Deployment*. [Online]. Available: https://5g-ppp.eu/wp-content/uploads/2018/02/5G-PPP-Automotive-WGWhite-Paper_Feb.2018.pdf

[4] Y. Xiao, X. Du, J. Zhang, F. Hu, and S. Guizani, "Internet protocol television (IPTV): The killer application for the next-generation Internet," *IEEE Commun. Mag.*, vol. 45, no. 11, pp. 126–134, Nov. 2007.

[5] V. Antrobus, B. Gary, and K. Claudia, "Driver-passenger collaboration as a basis for human-machine interface design for vehicle navigation systems," *Ergonomics*, vol. 60, no. 3, pp. 321–332, 2017.

[6] A. S. Ali and S. Osvaldo, "Energy-efficient resource allocation for mobile edge computing-based augmented reality applications," *IEEE Wireless Commun. Lett.*, vol. 6, no. 3, pp. 398–401, Jun. 2017.

[7] M. Chen, Y. Hao, C.-F. Lai, D. Wu, Y. Li, and K. Hwang, "Opportunistic task scheduling over co-located clouds in mobile environment," *IEEE Trans. Serv. Comput.*, vol. 11, no. 3, pp. 549–561, May/Jun. 2018.

[8] A. Ashok, P. Steenkiste, and F. Baic, "Vehicular cloud computing through dynamic computation offloading," *Comput. Commun.*, vol. 120, pp. 125–137, May 2018.

[9] X. Du, M. Guizani, Y. Xiao, and H.-H. Chen, "Transactions papers a routing-driven elliptic curve cryptography based key management scheme for heterogeneous sensor networks," *IEEE Trans. Wireless Commun.*, vol. 8, no. 3, pp. 1223–1229, Mar. 2009.

[10] T. Mekki, I. Jabri, A. Rachedi, and M. B. Jemaa, "Vehicular cloud networks: Challenges, architectures, and future directions," *Veh. Commun.*, vol. 9, pp. 268–280, Jul. 2017.

[11] D. Xu *et al.*, "A survey of opportunistic offloading," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2198–2236, 3rd Quart., 2018.

[12] L. Tang and S. He, "Multi-user computation offloading in mobile edge computing: A behavioral perspective," *IEEE Netw.*, vol. 32, no. 1, pp. 48–53, Jan./Feb. 2018.

[13] T. Taleb and A. Ksentini, "Follow me cloud: Interworking federated clouds and distributed mobile networks," *IEEE Netw.*, vol. 27, no. 5, pp. 12–19, Sep. /Oct. 2013.

[14] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, 4th Quart. 2017.

[15] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1657–1681, 3rd Quart. 2017.

[16] K. Zhang, Y. Mao, S. Leng, Y. He, and Y. Zhang, "Mobile-edge computing for vehicular networks: A promising network paradigm with predictive offloading," *IEEE Veh. Technol. Mag.*, vol. 12, no. 2, pp. 36–44, Jun. 2017.

[17] Y. Liu, S. Wang, J. Huang, and F. Yang, "A computation offloading algorithm based on game theory for vehicular edge networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6.

[18] J. Feng, L. Zhao, J. Du, X. Chu, and F. R. Yu, "Computation offloading and resource allocation in D2D-enabled mobile edge computing," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6.

[19] X. Chen, L. Pu, L. Gao, W. Wu, and D. Wu, "Exploiting massive D2D collaboration for energy-efficient mobile edge computing," *IEEE Wireless Commun. Mag.*, vol. 24, no. 4, pp. 64–71, Aug. 2017.

[20] H. Meng, K. Zheng, P. Chatzimisios, H. Zhao, and L. Ma, "A utility-based resource allocation scheme in cloud-assisted vehicular network architecture," in *Proc. IEEE Int. Conf. Commun. Workshop (ICCW)*, Jun. 2015, pp. 1833–1838.

[21] R. Yu, J. Ding, X. Huang, M.-T. Zhou, S. Gjessing, and Y. Zhang, "Optimal resource sharing in 5G-enabled vehicular networks: A matrix game approach," *IEEE Trans. Veh. Technol.*, vol. 65, no. 10, pp. 7844–7856, Oct. 2016.

[22] Y. Wang, M. Sheng, X. Wang, L. Wang, and J. Li, "Mobile-edge computing: Partial computation offloading using dynamic voltage scaling," *IEEE Trans. Commun.*, vol. 64, no. 10, pp. 4268–4282, Oct. 2016.

[23] Y. Yu, J. Zhang, and B. L. Khaled, "Joint subcarrier and CPU time allocation for mobile edge computing," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2016, pp. 1–6.

[24] D. Xu *et al.*, "A survey of opportunistic offloading," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2198–2236, 3rd Quart. 2018.

[25] Z. Su, Y. Hui, T. H. Luan, and S. Guo, "Engineering a game theoretic access for urban vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 6, pp. 4602–4615, Jun. 2017.

MINGHUI LIWANG received the B.S. degree in computer science and technology from Huaqiao University, Xiamen, in 2013. She is, on a master-doctor continuous study course, currently pursuing the Ph.D. degree with the School of Information Science and Engineering, Xiamen University. Her current research interests include wireless communication systems and the Internet of Vehicles.

JIEXIANG WANG received the bachelor's degree in electrical engineering and automation, Xiamen University, Xiamen, in 2017. He is currently pursuing the master's degree with the Department of Communication Engineering, School of Information Science and Engineering, Xiamen University. His current research interests include image processing and machine learning.

ZHIBIN GAO received the B.S. degree in communication engineering, the M.S. degree in radio physics, and the Ph.D. degree in communication engineering from Xiamen University, Xiamen, Fujian, China, in 2003, 2006, and 2011, respectively, where he is currently a Senior Engineer of communication engineering. His current research interests include wireless communication, mobile network resource management, and signal processing.

XIAOJIANG DU (S'99–M'03–SM'09) received the B.E. degree from Tsinghua University, Beijing, China, in 1996, and the M.S. and Ph.D. degrees from the University of Maryland, College Park, MD, USA, in 2002 and 2003, respectively, all in electrical engineering. He is currently a Professor with the Department of Computer and Information Sciences, Temple University, Philadelphia, PA, USA. His research interests include security systems, wireless networks, and computer networks. He has published over 300 journal and conference papers in these areas. He has been awarded more than U.S. $6 M in research grants from the U.S. National Science Foundation, the Army Research Office, and the Air Force Research Laboratory. He serves on the Editorial Board of three international journals.

MOHSEN GUIZANI (S'85–M'89–SM'99–F'09) received the B.S. (Hons.) and M.S. degrees in electrical engineering and the M.S. and Ph.D. degrees in computer engineering from Syracuse University, Syracuse, NY, USA, in 1984, 1986, 1987, and 1990, respectively. He is currently a Professor with the Department of Computer Science and Engineering, Qatar University, Qatar. He has authored nine books and more than 500 publications in refereed journals and conferences. He is a Senior Member of ACM. He received the 2017 IEEE Communications Society WTC Recognition Award and the 2018 Ad-Hoc Technical Committee Recognition Award for his contribution to outstanding research in wireless communications and Ad-Hoc Sensor networks. He served as the IEEE Computer Society Distinguished Speaker. He is also an IEEE ComSoc Distinguished Lecturer.