IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 29, 2020, accepted October 3, 2020, date of publication October 6, 2020, date of current version October 20, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3029169

A Hierarchical Vehicular-Based Architecture for Vehicular Networks: A Case Study on Computation Offloading

TINGTING LIU[1], (Member, IEEE), JUNHUA WANG[2], BAEKGYU KIM[3], (Member, IEEE), JIANG XIE[4], (Fellow, IEEE), AND ZHU HAN[5,6], (Fellow, IEEE)

[1]Nanjing Institute of Technology, Nanjing 211167, China
[2]School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China
[3]InfoTech Labs, Toyota Motor North America, Mountain View, CA 94043, USA
[4]Department of Electrical and Computer Engineering, University of North Carolina at Charlotte, Charlotte, NC 28223, USA
[5]Department of Electrical and Computer Engineering, University of Houston, Houston, TX 77004, USA
[6]Department of Computer Science and Engineering, Kyung Hee University, Seoul 446-701, South Korea

Corresponding author: Tingting Liu (liutt@njit.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61702258 and 62002166; in part by the Key Projects of Natural Science Research in Colleges and Universities of Jiangsu Province under Grant 19KJA410001; in part by the Foundation of Jiangsu Advanced Numerical Control Technology Key Laboratory under Grant SYKJ201901, Grant NSF EARS-1839818, Grant CNS-1717454, Grant CNS-1731424, Grant CNS-1702850, Grant CNS-1718666, Grant CNS-1731675, Grant CNS-1910667, Grant CNS-1910891, and Grant CNS-2025284; and in part by the China Postdoctoral Science Foundation under Grant 2020M671483.

ABSTRACT In order to realize an intelligent transportation system (ITS) which will provide smooth urban traffic, autonomous driving, accurate route navigation, etc., enormous computations need to be migrated from cloud centers to edge nodes, especially for the services requiring stringent latency. In addition to base stations and road side units (RSUs), vehicles can be alteratively considered as a kind of computation resources. In this article, a hierarchical vehicular-based architecture which consists of cloud centers and vehicles is investigated. Computation offloading performance in the hierarchical architecture is also studied. In specific, the main components in vehicular networks and their characteristics on communication and computations are presented firstly. Several communication techniques that are essential in enabling computation offloading among these components are then discussed. Secondly, a hierarchical vehicular-based architecture, which integrates the main components, is constructed. Thirdly, a case study on computation offloading in the proposed architecture is conducted. In the concerned scenario, the computation offloading problem is modelled as a multi-dimensional multiple knapsack problem (MMKP). Two algorithms are investigated, among which, the first algorithm is a greedy heuristic method providing a sub-optimal solution with a low computational complexity. The second algorithm is a modified branch and bound (B&B) method, which can obtain the best solution with a high computational complexity. Numerical results are also presented to verify the performance of the two algorithms. It can be demonstrated that the proposed architecture can migrate more computations from cloud centers to vehicular nodes, when the computations require more communication resources.

INDEX TERMS Hierarchical vehicular architecture, intelligent transportation system, computation offloading, multi-dimensional multiple knapsack problem, branch and bound algorithm.

I. INTRODUCTION

As vehicles playing an increasingly important role in people's daily life, an intelligent transportation system (ITS), which aims to provide smooth urban traffic, autonomous driving, accurate route navigation, etc., has received considerable attention from both academia and industry [1]–[8]. In order to realize the vision of ITS, various types and massive amounts of data are required to be processed reliably within a very limited time. At present, the processing of massive data mainly depends on the cloud centers, which can provide powerful computations remotely [9]–[15]. However, cloud centers which are located far away from end users, will induce high transmission delay. In this way, the ITS cannot deal with

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

emergency or provide real-time interactions among vehicular terminals, simply relying on cloud centers. By utilizing the fog computing technology which introduces a computation layer between cloud centers and end users, a large portion of computations can be offloaded to the nearby fog nodes from cloud centers, and thereby the data processing delay can be reduced dramatically [16]–[24].

Recently, the vehicular network is evolving from conventional vehicle ad-hoc networks to the internet of vehicles (IoV) [25], [26]. IoV mainly includes two research directions: vehicles' networking and vehicles' intelligentialize. Computing, which is essential in realizing IoV, is the basis of these two researches. In [27], the authors propose a computation offloading method employing vehicle-to-everything (V2X) technology. They first determine the routing of the computing tasks, and then balanced offloading strategies are generated. Finally, the optimal offloading strategy is determined. In [28], the authors propose a fog-cloud computational offloading algorithm in IoV to minimize the overall power consumption of vehicles and computation facilities. In [29], the authors investigate the optimal deployment and dimensioning of fog computing-based IoV architecture for autonomous driving. In [30], the authors propose an intelligent edge computing task offloading and migration IoV model under the software defined vehicular networks (SDVN) architecture. In [31], the authors overviews edge caching, edge computing, and edge artificial intelligent (AI), which can be further integrated to enable IoV applications.

A. VEHICLES AS COMPUTATION SOURCES

Most existing works which study computation offloading in vehicular networks, consider vehicles as computation sources. These studies mainly focus on how to optimally offload computations to the edge nodes, i.e., base stations/small-cell base stations (BSs/SBSs) or road side units (RSUs) [32]–[37]. In [32], for vehicular networks, the authors extend the original cloud radio access network (C-RAN) to integrate the device-to-device (D2D) and heterogeneous networks, forming an enhanced C-RAN (EC-RAN). The proposed EC-RAN not only improves communications quality, but also enhances vehicle computing capabilities by offloading mobile services to the cloud through BSs. In [33], the authors propose a cloud-based mobile edge computing (MEC) offloading framework in vehicular networks. With the aid of connected RSUs, an efficient predictive combination-mode relegation scheme is presented by adaptively offloading tasks to the MEC servers through direct uploading or predictive relay transmissions. In [34], the authors present a collaborative computation offloading approach based on MEC and cloud servers in vehicular networks. Computation tasks can be offloaded to the MEC and cloud servers through RSUs. In [35], the authors propose to use fog computing technology in vehicular networks to support delay-sensitive vehicular applications. The vehicular networks are divided into network layer, fog layer, and control layer by software-defined-networking (SDN) technology. Vehicular data in the network layer are offloaded to the fog layer through BSs. In [36], in order to meet the requirement of real-time vehicular applications, the authors propose an efficient resource and context aware approach for deploying containerized micro-services on on-demand fogs with the aid of RSUs. In [37], the authors study the computation offloading problem for an in-vehicle user equipment. The energy-hungry workloads are offloaded to from the vehicle to the edge nodes with the aid of RSUs. The authors provide an energy-efficient distributed solution based on consensus alternating direction method of multipliers (ADMM). Overall, no matter the on-board applications or the in-vehicle user equipment, computation offloading is vital in the vehicles as computation sources scenarios.

However, it is still challenging in realizing computation offloading to BSs/SBSs or RSUs. On the one hand, the expenditures of deploying RSUs are very high, especially when the deployment needs to be dense enough so as to provide safety-critical services, such as crash avoidance or accident warnings. On the other hand, it is not very practical to consider SBSs as computation resources for vehicular networks, since lots of computation and communication resources in SBSs have been occupied by the primary cellular network. The remaining resources which can be utilized by vehicular networks are very limited. Besides, cellular networks and vehicular networks are inherently independent on each other. It is difficult to achieve efficient cooperation between two independent systems, which in turn makes it challenging to meet the stringent real-time service requirements. Therefore, it is necessary to design a sustainable architecture which can provide massive and timely computations for vehicular networks.

B. VEHICLES AS COMPUTATION RESOURCES

In addition to considering vehicles as computation sources, they can be alteratively considered as computation resources. Vehicles, especially the electric vehicles, are usually equipped with high-capacity batteries and substantial computational resources. Thus they are sufficient to perform computations and communications by themselves. The concept of vehicular fog computing (VFC) has been proposed in [38]–[41]. Generally, the vehicular fog network is an integrated network consisting a group of vehicles and multiple edge servers. In [38], the authors propose to offload computations from the BS to the vehicular fog nodes by leveraging the under-utilized computation resources of nearby vehicles. In order to stimulate the nearby vehicles, an efficient incentive mechanism based on contract and matching theory is presented. Here, vehicles are used to offload computations from an external system, i.e., the cellular networks. In [39], the authors study the effective server recruitment and reliable task offloading under information asymmetry and information uncertainty. They first propose a server recruitment mechanism based on contract theory under information asymmetry. Then, a pricing-based matching algorithm and a matching-learning-based algorithm are presented for computation offloading under complete information and information uncertainty. Here, vehicles are used to offload computations from the nearby vehicles, which can be considered to serve an internal system. In [40], the authors propose a novel contract-based incentive mechanism to make vehicles join vehicular fog computing. Distributed deep reinforcement learning is used to offload computations to the participating vehicles, and task offloading method based on the queueing model is also proposed to avoid decision collisions in multi-vehicles task offloading. Here, vehicles are considered as internal computation resources. In [41], the authors propose an optimal task offloading scheme to maximize the long-term reward of the system. Computations are offloaded to the nearby vehicles which form a dynamic VFC system to process the tasks. In this scenario, the computation offloading depends on the internal vehicular system. Thus, in many existing works, the vehicle fog networks are considered capable in supporting computation offloading from both external or internal systems.

Moreover, due to the large amount of vehicles, widespread locations, and independence of external systems, vehicles have unique advantages in performing computations [42]–[45]. The massive moving and static vehicles with considerable computation resources can provide computation services not only for vehicular networks, but also for the external systems. Although each vehicle has a limited computation capacity, a group of static vehicles, such as vehicles in a parking lot, or multiple moving vehicles with a relatively stable neighborhood can be regarded as a vehicular cloudlet. In [43], the authors propose to exploit the full potentials of parked vehicle assistance. A vehicular fog computing-aware parking reservation auction is proposed to guide the moving vehicles to the available parking places, and moreover, the fog capability of parked vehicles can be incentivized to process the delay-sensitive computing services by monetary rewards. In [44], the authors also propose to use the computation resources in the parked vehicles. The problem is modeled as allocating the limited fog resources to the delay-sensitive vehicular applications in order to minimize the service latency. A heuristic algorithm is proposed to solve the modeled problem. Then, reinforcement learning is introduce to the proposed heuristic algorithm to make efficient resource allocation decisions, leveraging the vehicles' movement and parking status. In [45], the authors construct a VFC model to enable distributed traffic management in order to minimize the response time of city-level events collected and reported by vehicles. The moving and parked vehicles are leveraged as fog nodes in modelling the optimization problem. Nevertheless, the locations of the static cloudlets are sometimes far away from end users, such that it may induce relatively larger communication delay compared to the moving cloudlet. It is of great importance to study the computation offloading problem in such a vehicular network by taking account of the distinct properties of computation and communications among moving and static cloudlets.

The main contributions of this article are summarized as follows,

1) The main components of a general vehicular network are presented and the corresponding characteristics of each vehicular component on computation and communication are analyzed. Based on the characteristics of each component, a hierarchical vehicular-based multi-layer architecture is therefore constructed, integrating cloud centers, static vehicles, moving vehicles and RSUs.
2) Since the computation tasks require distinct amount of resources on each layer, computation offloading problem is modeled as a multi-dimensional multiple knapsack problem in the proposed hierarchical architecture. A greedy heuristic method and a modified branch-and-bound (B&B) method are investigated.
3) Simulation results are provided to demonstrate the feasibility of computation offloading in the proposed hierarchical vehicular-based architecture, and also show the different performances, such as network profits and running time of the two methods. The conclusion is that when the ratio between the communication requirement and the computation requirement of a task increases, more computation tasks will be migrated from cloud centers to the vehicular fog, and vise versa.

The rest of this article is organized as follows. In section II, we analyze the main components in vehicular networks, focusing on their characteristics on computation and communications. In section III, a hierarchical vehicular-based architecture is therefore constructed. Then, we formulate a computation offloading problem, and two algorithms are investigated. In section IV, numerical results are provided to compare the performance of these two algorithms, and demonstrate the effectiveness of the proposed hierarchical vehicular-based architecture. Finally, we conclude this article in section V.

II. A HIERARCHICAL VEHICULAR-BASED ARCHITECTURE

There are various open reference vehicular system architectures especially designed for certain application scenarios. For example, architecture reference for cooperative and intelligent transportation (ARC-IT) [46] publicly opens reference system architectures for more than 100 scenarios for Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications. Generally, the common parts of these architectures include cloud centers, RSUs and vehicles. In this section, we first present the main characteristics of each component, focusing on their communication and computation properties. Then, by considering these properties, we propose a hierarchical vehicular-based architecture.

A. CLOUD CENTERS

Cloud centers, which are usually located far away from end users, with abundant computation capacities, can handle all kinds of computations, such as data exploitation or

TABLE 1. Features of the main components in vehicular networks.

Features	Cloud Centers	RSUs	Vehicles	
			Moving	Static
Communication Capacity	Low	Medium	High	Medium
Computation Capacity	High	Medium	Low	Medium
Distance to End Users	Remote	Near	Near	Medium

data mining. Cloud centers play an important role in the vehicular cloud computation (VCC) scenario [47]. In VCC, a large amount of data is transmitted to the cloud centers for city-level monitoring, management and controlling, which not only consume a lot of communication resources, but also induce a high communication delay. The communication delay between vehicles and cloud centers is non-negligible for applications that require stringent latency. Thus, it is suitable to process non-real-time computation tasks in cloud centers.

B. RSU

RSU, which is one of the critical element in vehicular networks, is typically equipped with communication and computation units that can perform wireless communication and local computations [48]. The basic role of a RSU is to collect data from adjacent vehicles, and then make local analysis. For example, RSUs deployed at a road intersection can collect vehicle locations, and provide vehicles with hazardous road warnings.

C. VEHICLES

Vehicles are usually equipped with high-capacity batteries, which are essential to support communication and computations. As the number of vehicles increases, it is promising to consider vehicles as potential computation resources. In the following, we will discuss the communication and computation characteristics of vehicles in two statuses: moving and static, respectively.

1) MOVING VEHICLES

Moving vehicles usually exhibit two routing patterns: random and scheduled, e.g., the moving patterns revealed by private cars and public transportation buses. Compared with cloud centers, moving vehicles do not have comparable computation capacities, but they are much closer to end users. In other words, moving vehicles have large communication capabilities, but low computation capabilities. Therefore, they are suitable to process delay sensitive computations with small sizes.

2) STATIC VEHICLES

Static vehicles, such as vehicles in a parking lot or along the roadside, can form a static computation cloudlet. Compared with cloud centers, their locations are not far away from end users, but not as close as RSUs. While comparing with the moving vehicles, static vehicles can provide a relatively larger computation capacity. For clarity, the communication and computation properties of each component are listed in Table 1.

D. COMMUNICATIONS AMONG THESE COMPONENTS

In order to facilitate computation offloading in vehicular networks, we will introduce several communication techniques among these components in the following.

1) VEHICLE TO VEHICLE COMMUNICATIONS

V2V communications are the direct connections between vehicles. There are two candidate techniques for V2V communications, i.e., long term evolution device to device (LTE D2D) and dedicated short-range communications (DSRC) [49].

- **LTE D2D**: LTE D2D refers to a D2D communication link underlying a cellular network. Generally, utilizing the LTE D2D method is challenged by interference problem. The discovery time before two vehicles getting connected is much larger than the effective communication time, which can be challenging for services requiring stringent latency. But, D2D communications take the advantage of adjacent physical locations between connected vehicles, and therefore provide a very high transmission rate.
- **DSRC**: V2V Communications also can be realized by a vehicular ad hoc network (VANET). As early as 2004, the standard of V2V communications has been built based on IEEE 802.11p, known as DSRC [50]. The communication frequency of VANET is assigned with 5.85GHz to 5.925 GHz, which is an unlicensed frequency band shared with WiFi. Therefore, DSRC faces some challenges, such as collisions occurring at media access control (MAC) layer. Still, V2V can provide short to medium range communication techniques for vehicles, with low deployment costs and low transmission delay.

2) VEHICLE TO RSUs COMMUNICATIONS

Vehicle to RSUs communications refer to connections between vehicles and infrastructures on roadside. Due to the widely deployment of cellular networks, it is natural to consider cellular networks as a V2I communication candidate. Another option is DSRC [49].

- **Cellular Networks**: Two transmission modes are supported by cellular networks: unicast and broadcast.

TABLE 2. Communication techniques among the main vehicular components.

Components	Communications Techniques	Introduction
Vehicle to Vehicle	LTE D2D	D2D communications take the advantage of adjacent physical locations between connected vehicles, and therefore provide a very high transmission rate.
	DSRC	DSRC is built based on IEEE 802.11p. The communication frequency of VANET is assigned with 5.85GHz to 5.925GHz. It can provide short to medium range communication techniques for vehicles, with low deployment costs and low transmission delay.
Vehicle to RSUs	Cellular Networks	Two transmission modes are supported by cellular networks: unicast and broadcast. Cellular networks have the advantage of providing seamless coverage for vehicle, but it can only support a relatively low transmission rate.
	DSRC	The advantage is that DSRC can provide a relatively high transmission rate with low deployment costs.
Cloud Centers to RSUs	Cellular Networks	Cloud centers to RSUs communications also refer to cloud-to -infrastructure (C2I) communications. The communication delay of C2I is significant, due to the long distance from RSUs to cloud centers as well as the limited communication resources.

Unicast can be used to connect a vehicle with a base station in both uplink and downlink communications. Broadcast is typically used in downlink area information distribution, such as safety pre-warning or congestion information. Besides, cellular networks can be considered as relays to support long distance communication. Cellular networks have the advantage of providing seamless coverage for vehicle, but it can only support a relatively low transmission rate.

- **DSRC**: Besides the basic property of DSRC aforementioned in V2V communications, several challenges need to be addressed when introduce DSRC to V2I communications. The traditional sparse pilot design is not practical in performing channel estimation in V2I scenario, due to the highly time-frequency selective channels. The multiple access mechanism also needs to be reconsidered. Since there are a large number of vehicles in the infrastructure covered area, and traditional carrier sense multiple access (CSMA) mechanism cannot guarantee stringent quality of service requirement for vehicular services. The advantage is that DSRC can provide a relatively high transmission rate with low deployment costs.

3) CLOUD CENTERS TO RSUs COMMUNICATIONS

Cloud centers to RSUs communications also refer to cloud-to-infrastructure (C2I) communications. Generally, all the RSUs are assumed to be connected to a backbone network with wired links which are connected to the cloud centers. Some isolated RSUs need to send data to the central RSUs which are connected to the backbone network, with the aid of moving vehicles or cellular networks. Communication delay among the neighbouring RSUs is trivial, because of the short physical distance and high transmission rate provided by the backbone networks. While the communication delay of C2I is significant, due to the long distance from RSUs to cloud centers as well as the limited communication resources.

For clarity, the communication techniques among the main vehicular components are lised in Table 2. After addressing the enabling communication techniques among these components, it is feasible to perform communications with each other. Hereafter, we propose a hierarchical vehicular-based architecture using the above-mentioned components.

E. PROPOSED HIERARCHICAL VEHICULAR-BASED ARCHITECTURE

The key challenge in designing a vehicular-based architecture is to find an efficient and flexible way to integrate various communication and computation resources. To this end, as shown in Fig. 1, a hierarchical vehicular-based architecture is proposed based on the aforementioned components. Particularly, we separate vehicles into two distinct layers by considering their different statuses. In each layer, specific communication and computation capacity is assumed. Therefore, concerning computation allocation, there exists a tradeoff between communication and computation performance in the proposed hierarchical vehicular-based architecture. It is critical to leverage the distinct components in maximizing the vehicular network profits by allocating computations.

III. CASE STUDY ON COMPUTATION OFFLOADING IN THE PROPOSED HIERARCHICAL VEHICULAR-BASED ARCHITECTURE

In the following, we investigate the computation offloading problem in the proposed hierarchical architecture. As seen in Fig. 2, we have five layers, i.e., an end users layer,

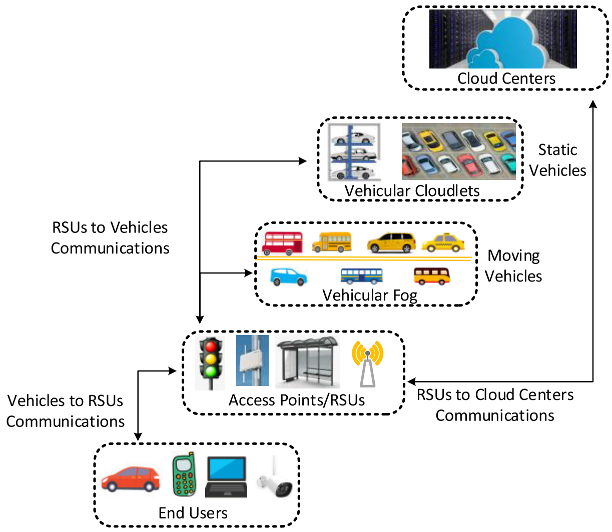


FIGURE 1. Proposed hierarchical vehicular-based architecture.

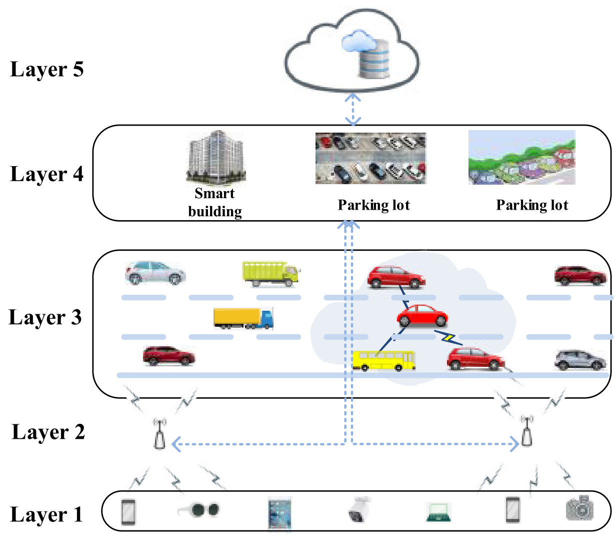


FIGURE 2. System model.

an access points layer and 3 computation resources layers which are vehicular fog, vehicular cloudlet and cloud centers. As discussed in the previous section, these three computation resources layers are distinct from each other, concerning the communication and computation characteristics. The cloud centers in layer 5 have abundant computation capacity, while they are always far away from end users. The vehicular cloudlet layer in layer 4 includes parking lots and smart buildings. They have a relatively smaller computation capacity compared to the cloud centers, while they are more closer to the end users. The vehicular fog layer, i.e., layer 3, includes moving vehicles. Although they cannot provide stable computation capacity, but they are close to end users. Multiple RSUs in layer 2, bridging the end users in layer 1 and different resource layers, are deployed as access points along the roadside. Note that we assume RSUs as access points here rather than computation resources. This assumption will help to simplify the functionalities of RSUs, and focus on utilizing the computation capabilities inside the proposed architecture.

In what follows, we elaborate three computational offloading stages in the proposed architecture. In the first stage, end users transmit their extra computation tasks, which cannot be executed locally, to the nearby RSUs. In the second stage, RSUs allocate computation tasks to different layers according to a certain criteria, e.g., network profits maximization. In the third stage, each computation layer transmits the computation results to end users with the aid of RSUs. The first and the third stages need to handle some communications-related issues. The second stage is related to the computation offloading issue. Considering the different properties of each layer, computation offloading problem needs to be carefully designed. Thus, we conduct a case study on the second stage, i.e., the computation offloading problem.

A. SYSTEM MODEL AND PROBLEM DESCRIPTION

As we have mentioned before, the cloud centers in layer 5 have abundant computation capacity, while they are always far away from end users. The vehicular cloudlet layer in layer 4 includes parking lots and smart buildings. They have a relatively smaller computation capacity compared to the cloud centers, while they are more closer to the end users. The vehicular fog layer, i.e., layer 3, includes moving vehicles. Although they cannot provide stable computation capacity, but they are close to end users. We denote the vehicular fog layer as $\mathcal{L}_3$, the vehicular cloudlet layer as $\mathcal{L}_4$, and the cloud center layer as $\mathcal{L}_5$. The communication capacity of the jth layer is denoted as $r_j, j = 3, 4, 5$, and the corresponding computation capacity of the jth layer is $m_j, j = 3, 4, 5$. A set of tasks upon allocation is denoted as $a_i, i = 1, \dots, N$. The communication resource required by each task on the jth layer is denoted as r_j^i, and the computation resource required on the jth layer is denoted as m_j^i. The profits arising from completing each task are represented as $p_i, i = 1, \dots, N$. The objective function is to maximize the network profits by fully making use of the communication and computation resources of each layer, i.e.,

$$\max_{x_{i,j}} \sum_{i=1}^{N} \sum_{j=3}^{5} p_i x_{i,j}, \tag{1}$$

$$\text{s.t. (C1)} \sum_{i=1}^{N} r_j^i \leq r_j, \quad \forall j, \tag{2}$$

$$\text{(C2)} \sum_{i=1}^{N} m_j^i \leq m_j, \quad \forall j, \tag{3}$$

$$\text{(C3)} \sum_{j=3}^{5} x_{i,j} \leq 1, \quad \forall i, \tag{4}$$

$$\text{(C4)}\ x_{i,j} \in \{0, 1\}, \quad \forall i, j, \tag{5}$$

where $x_{i,j}$ represents the allocation indicator. The four constraints are explained in the following.

TABLE 3. A greedy heuristic algorithm.

Input: Task set A, profit set P, required transmission and computation resource, transmission and computation capacity of each layer.
Output: Allocation matrix $[X]$
Steps:

1: *Step 1: Initialize* Rank the knapsacks in an increasing order of their distances from end users, and record the sequence in a vector. Denote a set F as the tasks that have not been selected and a set U as the tasks that have been selected.
2: *Step 2: Solving 2d-KP for each knapsack*
3: **for** $j = 3 : 5$ **do**
4: Fill the knapsack $\mathcal{L}_j$ by selecting tasks from set F with modified B&B algorithm (which is to solve 2d-KP). Record the selected items in set U; and update matrix $[X]$. Update F.
5: **end for**

- C1: The communication capacity constraint. It means that the total communication consumptions by the tasks allocated to the jth layer cannot exceed the layer communication capacity r_j;
- C2: The computation capacity constraint. Similar to C1, the total computation costs by the tasks allocated to the jth layer cannot exceed the layer computation capacity m_j;
- C3: Each task can be allocated to one layer at most;
- C4: Each task should be allocated as a whole.

The above problem can be treated as a multi-dimensional multiple knapsack problem (MMKP) [51], where the different layers can be considered as independent knapsacks, while each task can be considered as an item with two dimensional parameters, i.e., the consumptions of communications and computations. A traditional knapsack problem is known as an NP-hard problem. Compared to the traditional knapsack problem, the computation offloading problem in the proposed architecture faces the following challenge. The two dimensional parameters of a task vary depending on the selected knapsack, i.e., the particular layer. For example, in the cloud center, a task will be assigned with less communication resource but more computation resources. However, in the vehicular fog layer, the task will be allocated with more communication resources but less computation resource.

B. SOLUTIONS

In this section, we present two algorithms to solve the MMKP. **Algorithm 1** is a greedy heuristic method shown in Table 3. It can get a sub-optimal solution with low computational complexity. The second one is a modified branch-and-bound (B&B) method [52], [53] shown in Fig. 3. This algorithm can be utilized to get the best solution with a high computational complexity.

Algorithm 1 is a greedy heuristic method shown in Table 3. At Step 1, the knapsacks will be ordered according to a certain criteria. Here, we order the knapsacks in an increasing order concerning their distances from end users. In this way, we will first fill the vehicular fog layer, and then the vehicular cloudlet layer, and at last the cloud center layer. At Step 2, each knapsack will be filled with tasks by solving a two dimensional knapsack problem (2d-KP). The modified B&B method which is applied to solve the 2d-KP will be discussed later.

In **Algorithm 2**, task set A, network profit set P, required transmission resource r_j^i, required computation resource m_j^i, transmission and computation resources of each layer are considered as inputs. The allocation matrix is the output. As shown in Fig. 3 (A), at Step 1, the algorithm is initialized with input parameters. Since there are 3 resource layers, denoting the cloud centers, the vehicular cloudlet, and the vehicular fog with $\mathcal{L}_3$, $\mathcal{L}_4$, and $\mathcal{L}_5$, respectively, the optimization problem can be decomposed into 3 subproblems, and each subproblem can be considered as a 2d-KP. At Step 2, for each knapsack, 2d-KP is solved by utilizing a modified B&B algorithm independently. The optimal allocation solution is determined by each layer, respectively. Then, the tasks which are allocated to multiple layers are selected. If no such task exists, a feasible solution is obtained, and the algorithm will be terminated. If there are several multi-allocated tasks, then go to Step 3 and perform the modified B&B algorithm among these tasks. Step 3, i.e., the modified B&B, is a critical part of **Algorithm 2** and it is illustrated in Fig. 3 (B). After determining the multi-allocated tasks to a specific layer, the layer which is not fully allocated will go back to Step 2. Until each layer is filled with tasks, then this algorithm is terminated.

Fig. 3 (B) is a simple example of the modified B&B method, i.e., Step 3 of **Algorithm 2**. Assuming there are 10 tasks to be allocated, at first in (1) shown in Fig. 3 (B), 6 out of 10 tasks are assigned to each layer independently. It can be seen in (1) that three tasks, i.e., a_3 is allocated to both $\mathcal{L}_3$ and $\mathcal{L}_4$, a_4 and a_5 are allocated to both $\mathcal{L}_4$ and $\mathcal{L}_5$. Since each task should be allocated to only one layer. In Step 3, the modified B&B method illustrated in (2) is employed to determine the specific allocation layer of tasks a_3, a_4 and a_5. The result as shown in (3) is that a_3 is allocated to $\mathcal{L}_3$, a_4 and a_5 are allocated to $\mathcal{L}_4$. Note that the allocation criteria is maximizing the network profits. So in $\mathcal{L}_4$, one more task can be assigned, and two more tasks can be assigned to $\mathcal{L}_5$. Repeating Step 2, the result is shown in (4) that task a_8 is assigned to both $\mathcal{L}_4$ and $\mathcal{L}_5$. Then go to Step 3, i.e., conducting (5), the result is shown in (6) that a_8 is allocated to $\mathcal{L}_4$. At last in (7), each layer is filled with tasks, and no task is multi-allocated. Then the algorithm is terminated.

In **Algorithm 1**, the modified B&B method solving 2d-KP at Step 2 will be conducted by 3 times, while in **Algorithm 2**, the modified B&B method solving 2d-KP at Step 2 will be executed for at least 3 times. **Algorithm 2** has a very high computation complexity but it can achieve a better performance. Thus, **Algorithm 2** is suitable for allocating computation tasks which do not have stringent time requirements, and seek for the maximum network profits. **Algorithm 1**,

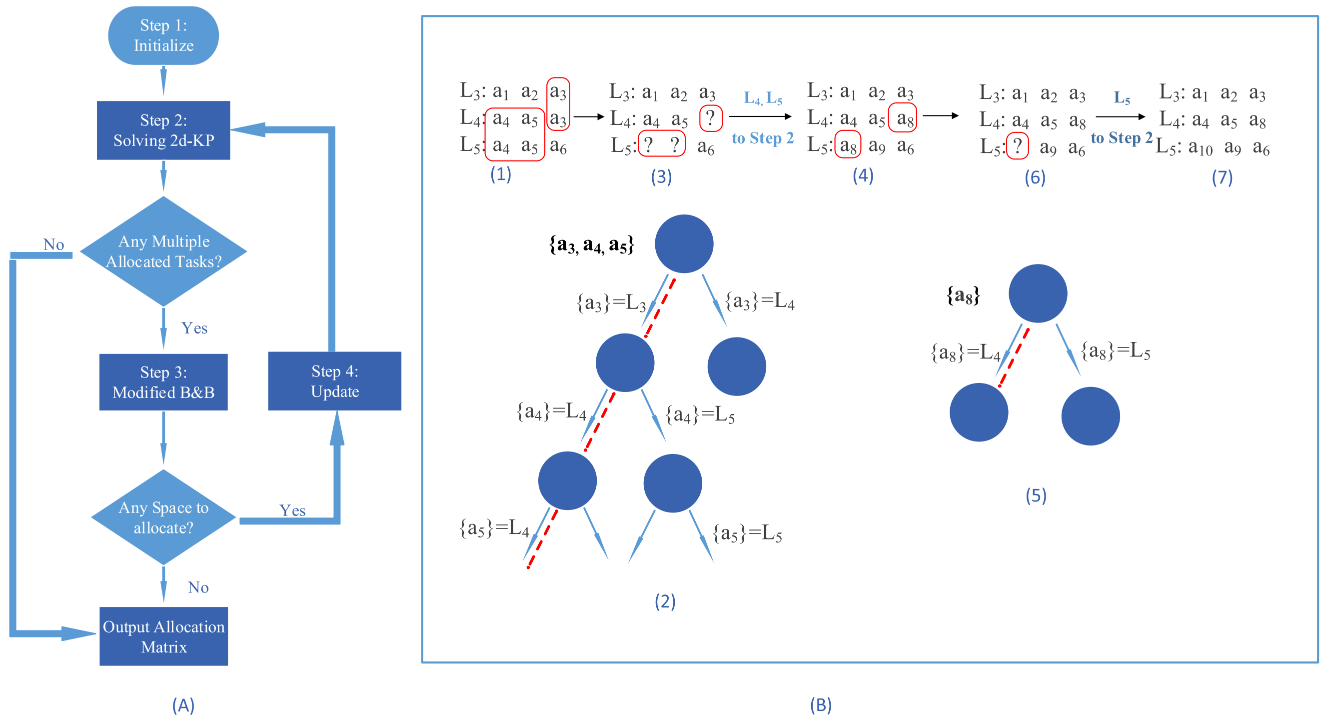


FIGURE 3. **(A): Algorithm 2. (B): A simple example of Step 3, i.e., the modified B&B method.**

TABLE 4. **Simulation parameters.**

Parameters	Values
Communication rate from RSUs to vehicular fog	1.5 Gbps
Communication rate from RSUs to vehicular cloudlet	80 Mbps
Communication rate from RSUs to cloud centers	30 Mbps
Computation capacity in fog	$200 * 10^8$ cycles/s
Computation capacity in cloudlet	$400 * 10^8$ cycles/s
Computation capacity in cloud centers	$4000 * 10^8$ cycles/s
computation tasks	[5,30]
size of each task	$[1, 50]$G cycles

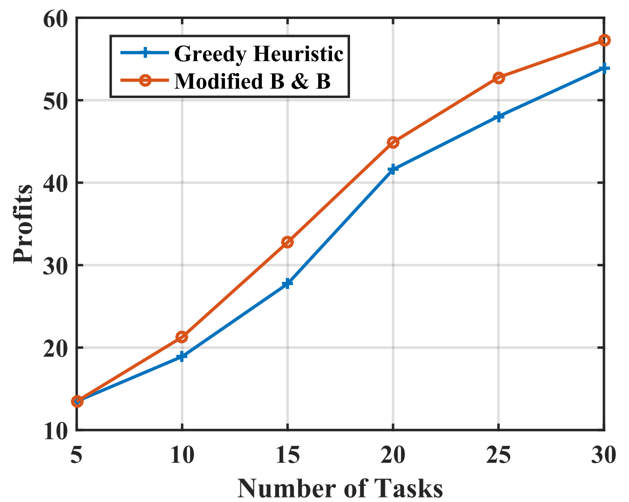


FIGURE 4. **Network profits using the modified B&B algorithm and the greedy heuristic algorithm.**

which has a relatively low computation complexity and is a sub-optimal solution, can be used as allocation method for computation tasks requiring stringent service time.

IV. SIMULATION RESULTS

The numerical results of computation offloading in the proposed architecture is conducted on Matlab. The communication rates are considered as communication resources. The simulation parameters are listed in Tab. 4. For simplify, we assume, the allocated transmission rates to each task are uniformly generated within $[1, 200]$ Mbps, $[1, 20]$ Mbps, and $[1, 10]$ Mbps for the fog layer, cloudlet layer and cloud centers, respectively.

In Fig. 4, we vary the number of tasks from 5-30 to verify the network profits by using the aforementioned two algorithms. It can be seen that as the number of tasks increasing, the total network profits of each algorithm increase. But the trend of the network profits is not always increasing. When the three layers are filled with tasks, and no task can be allocated, then the network profits will keep the same. Also, it can be seen that the modified B&B algorithm always achieves

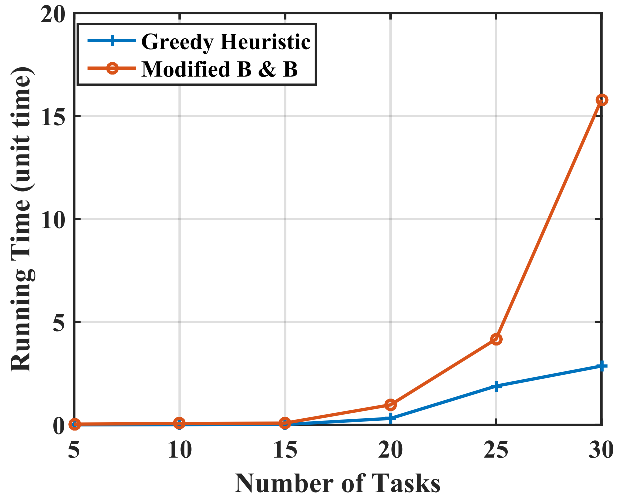


FIGURE 5. Running time of the modified B&B algorithm and the greedy heuristic algorithm.

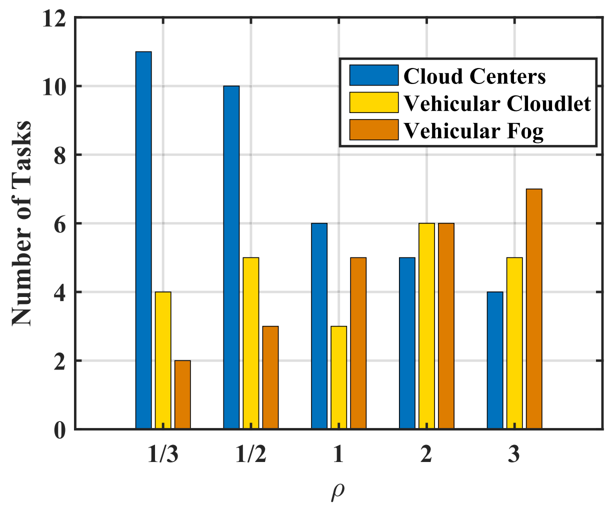


FIGURE 6. Number of tasks assigned to each layer when ρ varies.

higher profits compared to the heuristic greedy algorithm. Even though, the greedy heuristic algorithm also solves 2d-KP for each knapsack. The computation tasks are allocated to layers in an increasing order of the distances from the end users, and thereby the optimal network profits cannot be obtained under this allocation order. While in the modified B&B algorithm, one computation task is allocated to multiple layers at the very beginning, and then determine the optimal layer which can achieve the maximum network profits. So the proposed modified B&B algorithm can gradually converge to the optimal network profits.

Fig. 5 compares the running time between the two algorithms. It can be seen that as the computation tasks increase, the running time of the modified B&B algorithm increases dramatically, while the running time of the heuristic greedy algorithm increases more gently as the number of tasks increases.

In some scenarios, a computation task requires few communication resources, but needs to consume more computation resources. We use parameter ρ to represent the ratio between the communication requirement and the computation requirement of a task. We conduct simulations under the scenario that ρ equals to $\frac{1}{3}$, $\frac{1}{2}$, 1, 2, and 3. Fig. 6 demonstrates that as ρ increases, i.e., the communication requirement increases, more computation tasks will be migrated from cloud centers to the vehicular fog which has abundant communication resources.

V. CONCLUSION

In this article, we discuss the main vehicular components, such as vehicles, RSUs and cloud centers in the vehicular system. The communication and computation characteristics of each component are addressed. These components exhibit different strengths in communication and computation aspects. Based on these properties, a hierarchical vehicular-based architecture is proposed to combine various communication and computation resources efficiently and flexibly. Then, the computation offloading problem which is crucial in realizing ITS is investigated in the proposed architecture. The computation offloading problem can be modelled as a MMKP. Two algorithms are investigated. One of which is a greedy heuristic method with sub-optimal performance and low computational complexity. The other one is a modified B&B algorithm with better performance and a high computational complexity. Numerical results are provided to demonstrate the effectiveness of the proposed algorithms in maximizing the network profits and their differences in running time. Furthermore, as communication requirements increase, the proposed architecture can migrate more computations from cloud centers to vehicular fog.

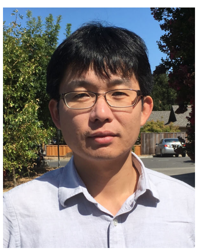
REFERENCES

[1] G. Dimitrakopoulos and P. Demestichas, "Intelligent transportation systems," *IEEE Veh. Technol. Mag.*, vol. 5, no. 1, pp. 77–84, Mar. 2010.
[2] L. Cai, J. Pan, L. Zhao, and X. Shen, "Networked electric vehicles for green intelligent transportation," *IEEE Commun. Standards Mag.*, vol. 1, no. 2, pp. 77–83, Jul. 2017.
[3] L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, "Big data analytics in intelligent transportation systems: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 1, pp. 383–398, Jan. 2019.
[4] H. Zhou, W. Xu, J. Chen, and W. Wang, "Evolutionary V2X technologies toward the Internet of vehicles: Challenges and opportunities," *Proc. IEEE*, vol. 108, no. 2, pp. 308–323, Feb. 2020.
[5] I. Ullah, S. Yan, and R. Malaney, "Location verification for emerging wireless vehicular networks," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10261–10272, Dec. 2019.
[6] M. Chowdhury and K. Dey, "Intelligent transportation systems—A frontier for breaking boundaries of traditional academic engineering disciplines [education]," *IEEE Intell. Transp. Syst. Mag.*, vol. 8, no. 1, pp. 4–8, Spring 2016.
[7] K. C. Dey, A. Mishra, and M. Chowdhury, "Potential of intelligent transportation systems in mitigating adverse weather impacts on road mobility: A review," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 3, pp. 1107–1119, Jun. 2015.
[8] L. F. Herrera-Quintero, J. C. Vega-Alfonso, K. B. A. Banse, and E. C. Zambrano, "Smart ITS sensor for the transportation planning based on IoT approaches using serverless and microservices architecture," *IEEE Intell. Transp. Syst. Mag.*, vol. 10, no. 2, pp. 17–27, Apr. 2018.

[9] S. Bitam, A. Mellouk, and S. Zeadally, "VANET-cloud: A generic cloud computing model for vehicular ad hoc networks," *IEEE Wireless Commun.*, vol. 22, no. 1, pp. 96–102, Feb. 2015.
[10] S. Olariu, "A survey of vehicular cloud research: Trends, applications and challenges," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 6, pp. 2648–2663, Jun. 2020.
[11] M. Sookhak, F. R. Yu, Y. He, H. Talebian, N. Sohrabi Safa, N. Zhao, M. K. Khan, and N. Kumar, "Fog vehicular computing: Augmentation of fog computing using vehicular cloud computing," *IEEE Veh. Technol. Mag.*, vol. 12, no. 3, pp. 55–64, Sep. 2017.
[12] A. Khurana, "Bringing big data systems to the cloud," *IEEE Cloud Computing*, vol. 1, no. 3, pp. 72–75, Sep. 2014.
[13] J. Zhang, Z. Chen, Z. Xu, M. Du, W. Yang, and L. Guo, "A distributed collaborative urban traffic big data system based on cloud computing," *IEEE Intell. Transp. Syst. Mag.*, vol. 11, no. 4, pp. 37–47, Oct. 2019.
[14] X. Li, J. Yuan, H. Ma, and W. Yao, "Fast and parallel trust computing scheme based on big data analysis for collaboration cloud service," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 8, pp. 1917–1931, Aug. 2018.
[15] J. Sun, Y. Zhang, Z. Wu, Y. Zhu, X. Yin, Z. Ding, Z. Wei, J. Plaza, and A. Plaza, "An efficient and scalable framework for processing remotely sensed big data in cloud computing environments," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 7, pp. 4294–4308, Jul. 2019.
[16] Q. Yuan, H. Zhou, J. Li, Z. Liu, F. Yang, and X. S. Shen, "Toward efficient content delivery for automated driving services: An edge computing solution," *IEEE Netw.*, vol. 32, no. 1, pp. 80–86, Jan. 2018.
[17] S.-C. Hung, H. Hsu, S.-Y. Lien, and K.-C. Chen, "Architecture harmonization between cloud radio access networks and fog networks," *IEEE Access*, vol. 3, pp. 3019–3034, 2015.
[18] Z. Chang, L. Liu, X. Guo, and Q. Sheng, "Dynamic resource allocation and computation offloading for IoT fog computing system," *IEEE Trans. Ind. Informat.*, early access, Mar. 6, 2020, doi: 10.1109/TII.2020.2978946.
[19] W. Zhang, Z. Zhang, and H.-C. Chao, "Cooperative fog computing for dealing with big data in the Internet of vehicles: Architecture and hierarchical resource management," *IEEE Commun. Mag.*, vol. 55, no. 12, pp. 60–67, Dec. 2017.
[20] A. V. Dastjerdi and R. Buyya, "Fog computing: Helping the Internet of Things realize its potential," *Computer*, vol. 49, no. 8, pp. 112–116, Aug. 2016.
[21] L. Liu, Z. Chang, X. Guo, S. Mao, and T. Ristaniemi, "Multiobjective optimization for computation offloading in fog computing," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 283–294, Feb. 2018.
[22] M. Gao, R. Shen, J. Li, S. Yan, Y. Li, J. Shi, Z. Han, and L. Zhuo, "Computation offloading with instantaneous load billing for mobile edge computing," *IEEE Trans. Services Comput.*, early access, May 25, 2020, doi: 10.1109/TSC.2020.2996764.
[23] M. Gao, R. Shen, S. Yan, J. Li, H. Guan, Y. Li, J. Shi, and Z. Han, "Heterogeneous computational resource allocation for C-RAN: A contract-theoretic approach," *IEEE Trans. Services Comput.*, early access, Apr. 15, 2019, doi: 10.1109/TSC.2019.2911509.
[24] Z. Chang, W. Guo, X. Guo, Z. Zhou, and T. Ristaniemi, "Incentive mechanism for edge-computing-based blockchain," *IEEE Trans. Ind. Informat.*, vol. 16, no. 11, pp. 7105–7114, Nov. 2020.
[25] F. Yang, S. Wang, J. Li, Z. Liu, and Q. Sun, "An overview of Internet of vehicles," *China Commun.*, vol. 11, no. 10, pp. 1–15, Oct. 2014.
[26] H. Zhou, W. Xu, J. Chen, and W. Wang, "Evolutionary V2X technologies toward the Internet of vehicles: Challenges and opportunities," *Proc. IEEE*, vol. 108, no. 2, pp. 308–323, Feb. 2020, doi: 10.1109/JPROC.2019.2961937.
[27] X. Xu, Y. Xue, X. Li, L. Qi, and S. Wan, "A computation offloading method for edge computing with Vehicle-to-Everything," *IEEE Access*, vol. 7, pp. 131068–131077, 2019.
[28] X. Wang, X. Wei, and L. Wang, "A deep learning based energy-efficient computational offloading method in Internet of Vehicles," *China Commun.*, vol. 16, no. 3, pp. 81–91, Mar. 2019.
[29] C. Yu, B. Lin, P. Guo, W. Zhang, S. Li, and R. He, "Deployment and dimensioning of fog computing-based Internet of vehicle infrastructure for autonomous driving," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 149–160, Feb. 2019.
[30] Z. Lv, D. Chen, and Q. Wang, "Diversified technologies in internet of vehicles under intelligent edge computing," *IEEE Trans. Intell. Transp. Syst.*, early access, Sep. 7, 2020, doi: 10.1109/TITS.2020.3019756.
[31] J. Zhang and K. B. Letaief, "Mobile edge intelligence and computing for the Internet of vehicles," *Proc. IEEE*, vol. 108, no. 2, pp. 246–261, Feb. 2020.
[32] 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.
[33] 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.
[34] J. Zhao, Q. Li, Y. Gong, and K. Zhang, "Computation offloading and resource allocation for cloud assisted mobile edge computing in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 7944–7956, Aug. 2019.
[35] C. Lin, G. Han, X. Qi, M. Guizani, and L. Shu, "A distributed mobile fog computing scheme for mobile delay-sensitive applications in SDN-enabled vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5481–5493, May 2020.
[36] H. Sami, A. Mourad, and W. El-Hajj, "Vehicular-OBUs-as-on-demand-fogs: Resource and context aware deployment of containerized micro-services," *IEEE/ACM Trans. Netw.*, vol. 28, no. 2, pp. 778–790, Apr. 2020.
[37] Z. Zhou, J. Feng, Z. Chang, and X. Shen, "Energy-efficient edge computing service provisioning for vehicular networks: A consensus ADMM approach," *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 5087–5099, May 2019.
[38] Z. Zhou, P. Liu, J. Feng, Y. Zhang, S. Mumtaz, and J. Rodriguez, "Computation resource allocation and task assignment optimization in vehicular fog computing: A contract-matching approach," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3113–3125, Apr. 2019.
[39] Z. Zhou, H. Liao, X. Zhao, B. Ai, and M. Guizani, "Reliable task offloading for vehicular fog computing under information asymmetry and information uncertainty," *IEEE Trans. Veh. Technol.*, vol. 68, no. 9, pp. 8322–8335, Sep. 2019.
[40] J. Zhao, M. Kong, Q. Li, and X. Sun, "Contract-based computing resource management via deep reinforcement learning in vehicular fog computing," *IEEE Access*, vol. 8, pp. 3319–3329, 2020.
[41] Q. Wu, H. Liu, R. Wang, P. Fan, Q. Fan, and Z. Li, "Delay-sensitive task offloading in the 802.11p-based vehicular fog computing systems," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 773–785, Jan. 2020.
[42] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, "Vehicular fog computing: A viewpoint of vehicles as the infrastructures," *IEEE Trans. Veh. Technol.*, vol. 65, no. 6, pp. 3860–3873, Jun. 2016.
[43] Y. Zhang, C.-Y. Wang, and H.-Y. Wei, "Parking reservation auction for parked vehicle assistance in vehicular fog computing," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3126–3139, Apr. 2019.
[44] S. Lee and S. Lee, "Resource allocation for vehicular fog computing using reinforcement learning combined with heuristic information," *IEEE Internet Things J.*, May 21, 2020, doi: 10.1109/JIOT.2020.2996213.
[45] Z. Ning, J. Huang, and X. Wang, "Vehicular fog computing: Enabling real-time traffic management for smart cities," *IEEE Wireless Commun.*, vol. 26, no. 1, pp. 87–93, Jun. 2019.
[46] U. S. D. Transportation. *Architecture Reference for Cooperative and Intelligent Transportation*. Accessed: Oct. 14, 2019. [Online]. Available: https://local.iteris.com/arc-it/
[47] M. Whaiduzzaman, M. Sookhak, A. Gani, and R. Buyya, "A survey on vehicular cloud computing," *J. Netw. Comput. Appl.*, vol. 40, pp. 325–344, Apr. 2014.
[48] D. Niyato and E. Hossain, "A unified framework for optimal wireless access for data streaming over Vehicle-to-Roadside communications," *IEEE Trans. Veh. Technol.*, vol. 59, no. 6, pp. 3025–3035, Jul. 2010.
[49] K. Zheng, Q. Zheng, P. Chatzimisios, W. Xiang, and Y. Zhou, "Heterogeneous vehicular networking: A survey on architecture, challenges, and solutions," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2377–2396, 4th Quart., 2015.
[50] E. Uhlemann, "The US and europe advances V2 V deployment [connected vehicles]," *IEEE Veh. Technol. Mag.*, vol. 12, no. 2, pp. 18–22, Jun. 2017.
[51] M. Silvano and P. Toth, "Algorithms for knapsack problems," *North-Holland Math. Stud.*, vol. 132, pp. 213–257, Jul. 1987.
[52] S. Martello and P. Toth, "Solution of the zero-one multiple knapsack problem," *Eur. J. Oper. Res.*, vol. 4, no. 4, pp. 276–283, Apr. 1980.
[53] J. Wang, T. Liu, K. Liu, B. Kim, J. Xie, and Z. Han, "Computation offloading over fog and cloud using multi-dimensional multiple knapsack problem," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Abu Dhabi, United Arab Emirates, Dec. 2018, pp. 1–7.

TINGTING LIU (Member, IEEE) received the B.S. degree in communication engineering and the Ph.D. degree in information and communication engineering from the Nanjing University of Science and Technology, Nanjing, China, in 2005 and 2011, respectively. She was a Postdoctoral Researcher with the Nanjing University of Science and Technology. From 2017 to 2018, she was a Visiting Scholar with the University of Houston, USA. She is currently an Associate Professor with the Nanjing Institute of Technology, China. Her research interests include game theory, blockchain, caching-enabled systems, edge computing, network quality of service, device-to-device networks, and cognitive radio networks. She served as a TPC Member of ICC 2019.

JUNHUA WANG received the B.S. and Ph.D. degrees in computer science from Chongqing University, in 2014 and 2019, respectively. From 2017 to 2018, she was a Visiting Scholar with the University of Houston, USA. Since 2019, she has been working with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics. Her research interests include mobile computing, vehicular *ad hoc* networks, and wireless networks.



BAEKGYU KIM (Member, IEEE) received the Ph.D. degree in computer science from the University of Pennsylvania, USA. He is currently a Principal Researcher at InfoTech Labs, Toyota Motor North America. His research interests include software platform technologies for connected cars and model-based software development for high-assurance systems.

JIANG XIE (Fellow, IEEE) received the B.E. degree from Tsinghua University, Beijing, China, the M.Phil. degree from the Hong Kong University of Science and Technology, and the M.S. and Ph.D. degrees from the Georgia Institute of Technology, all in electrical and computer engineering. She joined the Department of Electrical and Computer Engineering, University of North Carolina at Charlotte (UNC-Charlotte), as an Assistant Professor in August 2004, where she is currently a Full Professor. Her current research interests include resource and mobility management in wireless networks, mobile computing, Internet of Things, and cloud/edge computing. She is a Senior Member of the ACM. She received the Graduate Teaching Excellence Award from the College of Engineering at UNC-Charlotte, in 2007, the U.S. National Science Foundation Faculty Early Career Development (aCAREER) Award in 2010, the Best Paper Award from the IEEE/WIC/ACM International Conference on Intelligent Agent Technology, in 2010, and the Best Paper Award from the IEEE Global Communications Conference in 2017. She is an Editorial Board Member of the IEEE/ACM Transactions on Networking and *Journal of Network and Computer Applications* (Elsevier).

ZHU HAN (Fellow, IEEE) received the B.S. degree in electronic engineering from Tsinghua University, in 1997, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Maryland, College Park, in 1999 and 2003, respectively.

From 2000 to 2002, he was a Research and Development Engineer at JDSU, Germantown, MD, USA. From 2003 to 2006, he was a Research Associate with the University of Maryland. From 2006 to 2008, he was an Assistant Professor with Boise State University, Boise, ID. He is currently a John and Rebecca Moores Professor with the Department of Electrical and Computer Engineering and the Department of Computer Science, University of Houston, Houston, TX. His research interests include wireless resource allocation and management, wireless communications and networking, game theory, big data analysis, security, and smart grid. He has been a Fellow of AAAS, since 2019, and an ACM Distinguished Member, since 2019. He received the NSF Career Award, in 2010; the Fred W. Ellersick Prize of the IEEE Communications Society, in 2011; the EURASIP Best Paper Award for the *EURASIP Journal on Advances in Signal Processing*, in 2015; the IEEE Leonard G. Abraham Prize in the field of communications systems (Best Paper Award for IEEE Journal on Selected Areas in Communications JSAC), in 2016; and several best paper awards in IEEE conferences. He is also the winner of the 2021 IEEE Kiyo Tomiyasu Award, for his outstanding early to mid-career contributions to technologies holding the promise of innovative applications, with the following citation for contributions to game theory and distributed management of autonomous communication networks. He was an IEEE Communications Society Distinguished Lecturer, from 2015 to 2018. He has been a 1% highly cited researcher, since 2017, according to Web of Science.