Reliability-Aware Joint Optimization for Cooperative Vehicular Communication and Computing

Xu Han, Daxin Tian, *Senior Member, IEEE*, Zhengguo Sheng, *Senior Member, IEEE*, Xuting Duan, Jianshan Zhou, Wei Hao, *Member, IEEE*, Kejun Long, Min Chen, *Senior Member, IEEE*, and Victor C. M. Leung, *Life Fellow, IEEE*

***Abstract*— This paper comprehensively discusses the cooperative communication and computation of vehicular system. Based on the cooperative transmission, an stochastic model of vehicle-to-vehicle (V2V) communication reliability is established using probability theory. Furthermore, the computation reliability is defined as a new metric for computation offloading, and a vehicle computational performance evaluation model is also established. In order to effectively compute the required data, we combine V2V communication and vehicle computing to further characterize the coupling reliability of cooperative communications and computation systems. In addition, we propose a virtual queue model that combines queue length and vehicle privacy entropy to optimize partitioning. Finally, considering the amount of processing data and cut-off time of vehicle applications, we establish the optimal partition model of vehicle computing with the goal of maximizing the coupling reliability, and propose the coupling-oriented reliability calculation for vehicle collaboration using dynamic programming methods. Simulations show that the proposed scheme outperforms traditional approaches in terms of coupling reliability and completion rate. In addition, the allocation between local computing and data offloading is controlled by the server's privacy perception of collaboration events.**

***Index Terms*— Cooperative vehicle infrastructure system (CVIS), vehicular communication, mobile edge computing (MEC), partial offloading, dynamic programming.**

Manuscript received March 28, 2020; revised September 24, 2020; accepted November 11, 2020. Date of publication December 14, 2020; date of current version August 9, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61822101 and Grant 62061130221, in part by the Beijing Municipal Natural Science Foundation under Grant 4181002, in part by the Royal Society Kan Tong Po International Fellowship H2020-MSCA-RISE under Grant 101006411—SEEDS, in part by the Zhuoyue Program of Beihang University (Postdoctoral Fellowship), and in part by the China Postdoctoral Science Foundation under Grant 2020M680299. The Associate Editor for this article was F. Granelli. *(Corresponding author: Daxin Tian.)*

Xu Han, Daxin Tian, Xuting Duan, and Jianshan Zhou are with the Beijing Key Laboratory for Cooperative Vehicle Infrastructure Systems and Safety Control, Beijing Advanced Innovation Center for Big Data and Brain Computing, School of Transportation Science and Engineering, Beihang University, Beijing 100191, China (e-mail: 13661382800@163.com; dtian@buaa.edu.cn; duanxuting@buaa.edu.cn; jianshanzhou@foxmail.com).

Zhengguo Sheng is with the Department of Engineering and Design, University of Sussex, Brighton BN1 9RH, U.K. (e-mail: z.sheng@sussex.ac.uk).

Wei Hao and Kejun Long are with the Hunan Provincial Key Laboratory of Smart Roadway and Cooperative Vehicle-Infrastructure Systems, Changsha University of Science and Technology, Changsha 410004, China (e-mail: haowei@csust.edu.cn; longkejun@csust.edu.cn).

Min Chen is with the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: minchen2012@hust.edu.cn).

Victor C. M. Leung is with the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC V6T 1Z4, Canada (e-mail: vleung@ece.ubc.ca).

Digital Object Identifier 10.1109/TITS.2020.3038558

I. Introduction

CURRENTLY, the increasing scale of using various advanced onboard sensors has brought a growing demand for vehicle information services. Meanwhile, as more sophisticated software and algorithms are deployed on board, vehicle terminals are required to efficiently process complex programs (e.g., real-time application algorithms such as trajectory tracking, navigation positioning and environmental recognition, etc.). However, such terminals equipped with vehicles are constrained by battery capacity, storage resources and computing power. In addition, they are also constrained by limited space, volume and weight in terms of hardware resources. In the foreseeable future, the computing resources of an independent vehicle terminal will not be sufficient to fully handle the explosive growth of data from various intensive vehicle applications. The concept of the Internet of Vehicles (IoV) is emerging, which integrates available computing resources from different vehicles to construct a powerful distributed mobile computing system to maximize the utilization of potential computing resources for surrounding vehicles or roadside infrastructures [1]. Existing literature has already introduced the mobile edge computing (MEC) paradigm into Internet of Vehicles (IoV) communication systems to solve the computing resource allocation between mobile terminals [2], [3]. In such a paradigm, a vehicular user can transfer all or part of its computing tasks to a roadside resource-rich infrastructure such as a roadside unit (RSU) or Central Cloud for processing, or take advantage of underutilized storage and computing resources from one or more nearby vehicular. This type of computing paradigm based on vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications can be viewed as a computing-oriented collaboration between vehicles and infrastructure.

However, the topological dynamics caused by the high-speed movement of vehicles, the randomness of channels caused by environmental interference, and the multi-path fading of wireless signals make it difficult to maintain a reliable and efficient vehicular wireless network between vehicle clusters. In order to cope with the above-mentioned challenges, Cooperative Diversity technology in the framework of distributed wireless ad hoc networks has been introduced [4]. Cooperative Diversity is also known as Cooperative Communication, which mainly forms a spatially distributed virtual multi-antenna system by sharing physical antenna resources among mobile groups, e.g., Virtual Multiple Input Multiple Output System (Virtual MIMO), to effectively improve the reliability and channel capacity of wireless networks [5]. There are several types of cooperative communications, including fixed relaying schemes such as amplify-and-forward (AF) and decode-and-forward (DF), selective relaying schemes [4], [6], and incremental relaying schemes [4], [7], [8]. They all make use of relay signals between cooperative terminals to achieve spatial diversity. In this paper, we will apply the decode-and-forward scheme into our data offloading design. The relay selection process is out of the scope of this paper.

Computational partitioning is one of the core functions for implementing edge computing [9]. It allows a single vehicle terminal to avoid processing all the computing loads, but instead divides the whole application into several computing subtasks, and then distributes them to peer vehicles or RSU for processing according to the state of the network connection, computing load and application processing requirements. Furthermore, computational offloading is also an important function for implementing edge calculations. Current research around computational offloading is mainly aimed at optimizing system energy consumption or jointly optimizing energy and latency. The communication model is usually considered as a stable cellular communication network or a WLAN network. In fact, the inherent characteristics of vehicular networking transmission need to be considered in the scheme of vehicular computing.

As vehicular communication and computation are two coupled physical processes, the partitioning and offloading of vehicular computing tasks need to incorporate the consideration of physical characteristics of vehicular communication, e.g., the mobility of vehicles and dynamics. As previously stated, the challenges faced by end-to-end vehicular network connections also need to be addressed. Due to the mobility of vehicle networks, it's a great challenge to deliver reliable and efficient computational offloading. In the dynamic transmission environment, it is even more challenging to further ensure the coupling reliability of communication and computation.

In this paper, our aim is to address the aforementioned challenges related to vehicular cooperative communication and computation. The main contributions of our work are summarized as follows:

- We present stochastic modeling for V2V communication dynamics. An analytical model for characterizing reliability (i.e., the success probability of transferring data via the V2V connection within a deadline) of a V2V link is established.
- We propose an evaluation model to characterize the computation reliability which is defined as the probability that a vehicle successfully calculate a certain amount of data within a deadline.
- We propose a virtual queue model to optimize partitioning. Finally, we formulate constrained optimization problems based on dynamic programming by combining the reliability modeling of both V2V communications and partial offloading. The goal is to maximize the coupling reliability of vehicular cooperative communication and computing by optimizing the data workload partitions among V2V cooperators.

The rest of this paper is organized as follows. Section II provides an overview of the related works. Section III introduces the system model in terms of V2V communications and computation offloading. In section IV, we propose an optimization model for cooperative computation, followed by the simulation results in Section V. Finally, Section VI concludes this paper.

II. Related Work and Motivation

Mobile edge computing is an emerging paradigm, whose core concept is to deploy diverse storage and computing resources close to the edge of the communication network [10], such as sensor nodes, mobile devices, roadside infrastructure, etc.. Mobile edge computing provides a high-performance computing environment that is similar to cloud computing and supports lightweight real-time processing and analysis of massive amounts of data [11]. Further, compared with the centralized cloud computing paradigm [12], mobile edge computing provides flexible management and scheduling of computing resources for edge-side nodes via a communication network, which realizes communication and computing resource coordination, thereby alleviating the impact of massive data transmission on core networks and ultimately improving service performance and user experience. Moreover, mobile edge computing is also considered to be an important part of 5G [13], [14] and 6G [15] mobile communication systems because of its promise to support the huge amount of data processing services.

Similar to typical computing offloading systems, many emerging computational offloading studies are focused on optimizing terminal energy consumption and increasing terminal battery life [16], [17]. Zhang [18] considered application data processing in both local execution and cloud execution modes, and optimized the CPU clock frequency and data transmission with the goal of minimizing power consumption, and finally determined that the application adopts the energy-efficient application execution mode in processing data. In addition to optimizing the energy consumption of mobile application execution, many studies also pay attention to the time constraints of mobile application execution [19], e.g., the constraints of application deadline [20]. Muñoz [21] studied the mobile application offload problem with delay constraints and came up with solutions to optimize utilization of communication resources and computing resources. The purpose of existing optimization models is to minimize the computing and communication energy consumption of the mobile terminal.

Many recent efforts are devoted to propose effective joint task offloading and resource allocation schemes in terms of improving offloading utility [22], decreasing the energy consumption and task completion time [23], [24], and reducing the system-wide computation overhead [25]. Sun [26] proposed a computation efficiency metric which was defined as the number of calculated data bits divided by the corresponding energy consumption, and further proposed a joint computing algorithm that combines local computing and offloading. Based on the queuing theory, Liu [27] described the queue length of computing tasks in mobile and cloud caches, and proposed reliability measurement for mobile and cloud computing, i.e., the probability that the queue length is less than the buffer capacity. However, the reliability metric in the scenario of joint calculation of local computing and offloading has not been investigated. In addition, there is a lack of joint computing methods with reliability as a constraint.

Along with the rapid development of vehicular ad hoc networks (VANETs), a significantly increasing number of privacy-related issues have entered the spotlight of research debate [28]. Most previous works focus on the protection of location privacy. The proposed methods mainly include but not limited to anonymous access [29], privacy enhancing technologies [30] and cryptographic approaches (e.g. identity-based cryptosystem [31]). Some other proposals in VANETs focus on the utility of various statistical disclosure control techniques, and the qualitative analysis of privacy leak behavior [32], [33]. Those studies that use an information-theoretic approach to measure privacy [34] serve as the cornerstone for our work.

III. System Model

A. System Analysis

We build a network scenario with multiple mobile edge computing (MEC) servers and several requesting vehicles. The servers are deployed on moving vehicles with limited wireless and computing resources. These vehicles are also referred to as service vehicles.

The computing tasks of the requesting vehicle can be executed locally by its own resources. Besides, the requesting vehicle within the communication range of service vehicles can offload the tasks to the service vehicles. There is no core scheduler in the model, and the task offloading schedule is completed by the requesting vehicle. Following existing literature such as [35], [36], we can use the following two key parameters to characterize the profile of a mobile application:

- The input data size D is the total number of data bits as the application input. These D-bit data can be partitioned and offloaded to a service vehicle which is the cloud edge for remote execution.
- The application completion deadline T denotes the maximum number of successive time slots before the mobile application must be completed. In addition, we use t (from T to 1) to represent the time slot, and these D-bit data can also be partitioned into a series of smaller pieces $s_i \in [0, D]$, where s_t denotes the number of data bits that should be transmitted to the service node in the t_i time slot.

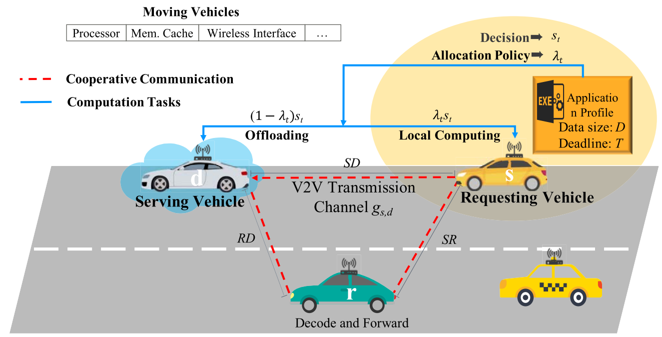


Fig. 1. Cooperative v2v computing system model.

We assume that each offloaded task is assigned to only one server. Moreover, each vehicle node has a queue buffer to store the tasks' arrivals. Denote the task queue lengths of the requesting vehicle (Source node $\mathbf{s}$) and the service vehicle (Destination node $\mathbf{d}$) in the time slot t as $Q_{\mathbf{s}}(t)$ and $Q_{\mathbf{d}}(t)$, respectively.

B. Communication Model

Suppose that each pair of V2V pair communicates through Rayleigh fading channels. We define a communication model based on cooperative transmission. The wireless link between requesting vehicle (Source node $\mathbf{s}$) and service vehicle (Destination node $\mathbf{d}$) can be represented by the following model

$$g_{\mathbf{s},\mathbf{d}} = c_{\mathbf{s},\mathbf{d}} \frac{h_{\mathbf{s},\mathbf{d}}}{SD^{k/2}} \tag{1}$$

where SD is the distance between requesting vehicle and service vehicle. Since k is the path loss coefficient, $SD^{k/2}$ describes the large-scale behavior of channel gain. $h_{\mathbf{s},\mathbf{d}}$ characterizes the fading characteristics of channels. $c_{\mathbf{s},\mathbf{d}}$ is a connection variable. $c_{\mathbf{s},\mathbf{d}} = 1$ when a connection is established between the requesting vehicle and the service node, otherwise it is 0.

Consider a three-node scenario in Fig.1, a source node $\mathbf{s}$ wants to communicate with a destination node $\mathbf{d}$ with the help of a relay node $\mathbf{r}$. Assuming that the relay node can perform perfect decoding when the received signal-to-noise ratio (SNR) exceeds a threshold, the mutual information of this cooperative link can be shown as

$$I_{\mathbf{s},\mathbf{d}} = \begin{cases} \frac{1}{2}\log(1 + 2SNR\left|g_{\mathbf{s},\mathbf{d}}\right|^2), & if \ \left|g_{\mathbf{s},\mathbf{r}}\right|^2 < q(SNR) \\ \frac{1}{2}\log(1 + SNR\left|g_{\mathbf{r},\mathbf{d}}\right|^2 + SNR'\left|g_{\mathbf{s},\mathbf{d}}\right|^2), & \\ & if \ \left|g_{\mathbf{s},\mathbf{r}}\right|^2 \geq q(SNR) \end{cases} \tag{2}$$

where SNR and SNR' are the signal-to-noise ratio in the path $\mathbf{s} \rightarrow \mathbf{d}$ and $\mathbf{r} \rightarrow \mathbf{d}$ respectively. $q(SNR) = \frac{2^{2R}-1}{SNR}$ and R is the data rate in $bits/s/Hz$ defined by Quality of Service (QoS) requirement. Since $h_{\mathbf{s},\mathbf{d}}$ is assumed to be a complex Gaussian variables with zero mean and unit variance, $\left|g_{\mathbf{s},\mathbf{d}}\right|^2 = \left|\frac{h_{\mathbf{s},\mathbf{d}}}{SD^{k/2}}\right|^2$ is an exponentially distributed variable with parameter $SD^{k/2}$.

According to the results in [4], we have the outage probability of cooperative transmission between the source node $\mathbf{s}$

and destination node $\mathbf{d}$ as

$$P_{\mathbf{s},\mathbf{d}}(R) = Pr(I_{s,d} < R) \approx \frac{1}{2} SD_t^k (SR_t^k + \frac{SNR}{SNR'} RD_t^k) \frac{(2^{2R}-1)^2}{SNR_t^2} \quad (3)$$

where SR is the distance between $\mathbf{s}$ and $\mathbf{r}$, RD is the distance between $\mathbf{r}$ and $\mathbf{d}$.

The application needs to determine how much data will be executed by the local and service vehicles, respectively, in each time slot t, with an objective to minimize the total unreliability on the mobile device. Specifically, the distribution policy under a given threshold is determined by the following decision rule

$$\begin{cases} Local\ execution, & \lambda_t \\ Service\ Vehicle\ execution, & 1-\lambda_t \end{cases} \quad (4)$$

where λ_t is the proportionality coefficient in time slot t.

Therefore, the data size of the local execution $s_t^{\mathbf{s}}$ and the service vehicle execution $s_t^{\mathbf{d}}$ in time slot t are respectively expressed as $s_t^{\mathbf{s}} = \lambda_t s_t$ and $s_t^{\mathbf{d}} = (1-\lambda_t)s_t$. Further, we can rewrite (3) as follow

$$P_{\mathbf{s},\mathbf{d}}(s_t) = Pr(I_{\mathbf{s},\mathbf{d}}(t) < s_t) \approx \frac{1}{2} SD_t^k (SR_t^k + \frac{SNR_t}{SNR_t'} RD_t^k) \frac{(2^{2(1-\lambda_t)s_t}-1)^2}{SNR_t^2} \quad (5)$$

where $P_{\mathbf{s},\mathbf{d}}(s_t)$ represents that the outage probability for a task with data size s_t transmitting by cooperative transmission in time slot t.

C. Computation Model

The top of Fig.1 shows the architecture of the vehicle node, which contains a processor, data storage components such as memory and cache, a single-server First Input First Output (FIFO) queue to store arriving tasks pending for execution, and a wireless interface. We denote B as bandwidth and W as number of CPU cycles.

Let W indicate the number of CPU cycles needed for an application. For a given input data size L, it can be derived as

$$W = LX \quad (6)$$

We assume that the probability distribution function (PDF) of X is $P_X(x)$, and its cumulative distribution function (CDF) is defined as $F_X(x) = Pr[X \leq x]$ and its complementary cumulative distribution function (CCDF) denoted as $F_X^c(x) = 1 - F_X(x)$. Therefore, the CCDF of the workload W is given by $F_W^c(w) = F_X^c(w/L)$.

As shown in [35], the number of CPU cycles per bit can be modeled by a Gamma distribution. The PDF of the Gamma distribution is given by

$$P_X(x) = \frac{1}{\beta\Gamma(\alpha)} (\frac{x}{\beta})^{\alpha-1} e^{-\frac{x}{\beta}} \quad (7)$$

The CDF of the Gamma distribution is given by

$$F_X(x) = \frac{1}{\Gamma(\alpha)} \Gamma(\alpha, \beta x) \quad (8)$$

while the Gamma distribution has the following two inferences: 1) $\Gamma(\alpha, z) = \int_z^{\infty} x^{\alpha-1} e^{-x} \mathrm{d_x}$. 2) For any positive integer $\alpha \geq 1$, it satisfies $\Gamma(\alpha) = (\alpha-1)!$.

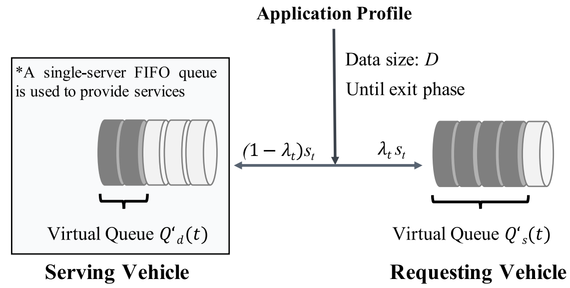


Fig. 2. Virtual Queue in time slot t.

We use the soft deadline to characterize probabilistic performance, that is, the statistical CPU scheduling model which assumes the application completion needs to meet its deadline with the probability p by allocating W_p CPU cycles. The parameter p is the application completing probability (ACP). In other words, the probability of an application requires no more than the allocated W_p should satisfy $(F_W^c)^{-1}(p) = Pr[W \leq Wp] \geq p$.

In this work, if W_p allocated by the requesting vehicle is not enough to support the application to complete the computing task, the probability of success is calculated as follow

$$F_W(W_{\mathbf{s}}(t)) = 1 - P_r[W \leq W_{\mathbf{s}}(t)] = 1 - \int_{W_{\mathbf{s}}(t)/\lambda_t s_t}^{\infty} \frac{1}{\beta^{\alpha}\Gamma(\alpha)} (\frac{W_{\mathbf{s}}(t)}{\lambda_t s_t})^{\alpha-1} e^{-\frac{W_{\mathbf{s}}(t)}{\lambda_t s_t \beta}} d_{s_t} \quad (9)$$

D. Virtual Queue Model

According to Little's law, the average queuing delay is proportional to the average queue length [37]. Besides, the queue length/queuing delay violation will ultimately undermines the reliability of task computing. For example, if a finite-size queue buffer is over-loaded, the incoming tasks will be dropped. Based on the system provided before (sections III-A - III-C), we propose the concept of a virtual queue (as Fig.2 shows). The initial time slot is indexed by $t = T + 1$. In this work, the end-to-end delay constraint is indirectly represented by the virtual queue length of the service vehicle.

For the current decision period t, $\mathbb{A}(Q'_{\mathbf{s}}(t), Q'_{\mathbf{d}}(t))$ is used to store the virtual queue status of the network, where the real queue length of requesting vehicle $\mathbf{s}$ and the real queue length of service vehicle $\mathbf{d}$ are defined as $Q_{\mathbf{s}}(t)$ and $Q_{\mathbf{d}}(t)$, respectively.

Assuming that the number of tasks in the system at the beginning of time slot t is same as the previous time slot, i.e. $Q(t) = s_{t+1}$. Then the priori virtual queue $Q'_{\mathbf{s}}(t)$ is determined by s_{t+1} and a priori parameter λ_{t+1}(following the decision rule provided in (4)), it can be derived as

$$Q'_{\mathbf{s}}(t) = \lambda_{t+1} s_{t+1} \quad (10)$$

Similarly, the priori virtual queue length of $\mathbf{d}$ at the beginning of time slot t is expressed as

$$Q'_{\mathbf{d}}(t) = (1-\lambda_{t+1})s_{t+1} + \theta_{t+1} \quad (11)$$

let $\theta_i = \Theta_i^+$, where $(a)^+ := \max(a, 0)$ and Θ_i is defined as $\sum_i^T \left[(1-\lambda_i)s_i - \frac{W_{\mathbf{d}}(i)}{X_{\mathbf{d}}}\right]$. $W_{\mathbf{d}}(i)$ represents the number of CPU cycles that the service vehicle **d** provides for the application in time slot t, and $W_{\mathbf{d}}(i) \in \{W_{\mathbf{d}}(i) \mid W_{\mathbf{d}H}, W_{\mathbf{d}L}\}$, which depend on the privacy exposure rate of **d**. $W_{\mathbf{d}H}$ represents large CPU cycles, $W_{\mathbf{d}L}$ represents small CPU cycles. $X_{\mathbf{d}}$ is the number of CPU cycles required by the application for the service vehicle to calculate the data per bit per second.

Instead of providing a privacy protection mechanism, the proposed system attempts to make progress in quantifying the privacy perception of vehicles. To this end, we refer to privacy entropy to further define privacy exposure rate.

$$H_{\mathbf{d}} = -\sum_{i=t+1}^{T} \hat{p}_i \log_2 \hat{p}_i \tag{12}$$

where $\hat{p}_i$ is expressed as the probability that each message is associated with the service vehicle during the communication process, which is updated as time changes. Assuming that the purpose of the attacker is to determine the willingness and behavior of the task recipient to the request, then in each time slot t in the system, the service vehicle is guessed as the true provider of the service with a certain probability.

In the absence of any optimal scheduling, the service vehicle considers that the probability of a service event occurring in each time slot is equal, i.e., vehicle **d** is determined as the service provider in each time slot, in which case its maximum entropy

$$max\{H_{\mathbf{d}}\} = -\log_2 \frac{1}{T} \tag{13}$$

$Pri_{\mathbf{d}}$ is proposed to measure the privacy level of the service vehicle **d**, which represents the subjective perception of privacy [38] by the service vehicle for the offloading event. $Pri_{\mathbf{d}}$ is defined as the normalization of entropies as below

$$Pri_{\mathbf{d}} = \frac{H_{\mathbf{d}}}{max\{H_{\mathbf{d}}\}} \tag{14}$$

Therefore, $1 - Pri_{\mathbf{d}}$ indicates the privacy exposure rate determined by the subjective perception of the service vehicle **d**. An increase in the value of $Pri_{\mathbf{d}}$ means the offloading event is protected, that is, the privacy leakage rate is lowered and the probability of being attacked is reduced, and vice versa. In this model, $W_{\mathbf{d}}(i)$ in time slot i is taken as $W_{\mathbf{d}H}$ when $1 - Pri_{\mathbf{d}}$ is less than a given threshold ϕ, otherwise $W_{\mathbf{d}}(i)$ is taken as $W_{\mathbf{d}L}$.

Suppose that the initial threshold λ_T is subject to standard uniform distribution (i.e., $\lambda_T \sim U(0, 1)$). We denote the number of data bits that are remained to be offloaded at the beginning of the time slot t by l_t. According to the previous statement, we have $Q'_{\mathbf{s}}(T) = \lambda_T l_T$ and $Q'_{\mathbf{d}}(T) = (1-\lambda_T)l_T$.

To achieve load balancing of the complete system, the the proportionality coefficient λ_t in time slot t is determined as follow

$$\frac{\lambda_t}{1-\lambda_t} = \frac{Q'_{\mathbf{d}}(t)}{Q'_{\mathbf{s}}(t)} \Rightarrow \lambda_t = \frac{Q'_{\mathbf{d}}(t)}{Q'_{\mathbf{s}}(t) + Q'_{\mathbf{d}}(t)} \tag{15}$$

According to the length of the queue $Q'_{\mathbf{s}}(t)$, $Q'_{\mathbf{d}}(t)$ in (10) and (11), (15) is reduced to

$$\lambda_t = \frac{(1-\lambda_{t+1})s_{t+1} + \theta_{t+1}}{s_{t+1} + \theta_{t+1}} \tag{16}$$

IV. Stochastic Optimization Model for V2V Transmission and Computing

A. Optimization Algorithm for Joint Reliability

We consider the optimization of transmission scheduling of D-bits application data in T time slots and denote t as discrete time index in descending order (from T to 1). The initial time slot is indexed by $t = T + 1$. We also denote by $f = [f_T, f_{T-1}, \ldots, f_1]^T$ a feasible scheduling solution and by $\mathbb{F}$ the corresponding feasible region. Then, since SD, SR, RD, SNR are random variables, and λ_t changes over time, we propose the optimization model for V2V transmission scheduling based on (5) and (9). The goal is to maximize the estimated success probability of computation offloading, with the derivation process and final expression shown in (17), as shown at the bottom of the next page.

For the sake of simplicity, let

$$P(s_t) \equiv \left[1 - \frac{SD_t^k(SR_t^k + \frac{SNR_t}{SNR'_t}RD_t^k)\cdot[2^{2(1-\lambda_t)s_t} - 1]^2}{2SNR_t^2}\right] \times \left[1 - \frac{\Gamma(\alpha, \frac{W_{\mathbf{s}}(t)}{\beta\lambda_t s_t})}{\Gamma(\alpha)}\right] \tag{18}$$

Therefore, the optimization model can be abstracted into another form like

$$\max_{\{s_i\}_{i=1}^T} : \prod_{t=1}^{T} P(s_t)$$
$$s.t. \begin{cases} \sum_{t=1}^{T} s_t = D, \\ \forall s_t > 0. \end{cases} \tag{19}$$

From (19), the optimal solution depends on the random variables of the channel in the initial time slot $t = T + 1$, i.e., the value of SD_t, SR_t, RD_t, SNR_t. Applying the dynamic programming principle to (19), we denote the number of data bits that without offloaded at the beginning of the time slot t by l_t. Thus, we can have $l_{t+1} = l_t - s_t$ for $t = T-1, \ldots, 2, 1$, and $l_1 = D$. We also denote the optimal number of data bits to be scheduled in time slot t by s_t^*. Let $f_t(l_t, s_t)$ indicates the contribution of stages $t, t+1, \ldots, T$ to objective function if system starts in state l_t at stage t, immediate decision is s_t, and optimal decisions are made thereafter. Besides, let $f_t^*(l_t)$ be the optimal value of the objective functions in (19) under the system conditions. $f_t^*(l_t) = f_t(l_t, s_t^*)$, and the recursive relationship will always be of the form $f_t^*(l_t) = \max_{s_t} f_t(l_t, s_t)$. Then, the dynamic programming solution is used to solve the optimization problem of the objective function in (19).

Consequently, the contribution of stages $f_t(l_t, s_t)$ for this problem is

$$f_t(l_t, s_t) = P(s_t) \cdot \max \prod_{i=t+1}^{T} P(s_i) \tag{20}$$

where the maximum is taken over $s_{t+1}, \ldots, s_T$ such that

$$l_1 = D; l_{t+1} = l_t - s_t, \quad t = T-1, \ldots, 2, 1 \tag{21}$$

when $t = 1$, D-bit application data is waiting for an optimized transmission schedule, i.e., the number of data bits that are remained to be offloaded at the beginning of the time slot T is $l_1 = D$. Moreover, $\forall s_i > 0$ for $t = T, T-1, \ldots, 1$. Thus,

$$f_t^*(l_t) = \max_{0<s_t\le l_t} f_t(l_t, s_t) \tag{22}$$

where $f_t(l_t, s_t) = P(s_t) \cdot f_{t+1}^*(l_{t+1})$, with $f_{T+1}^*(l_{T+1})$ defined to be 1. Thus, according to (20) and (22), the optimization model can be rearranged into a series of recursive equations, i.e., the recursive relationship relating the $f_1^*, f_2^*, \cdots, f_T^*$ functions, as follows

$$f_t^*(l_t) = \begin{cases} P(l_T), & t = T \\ f_{t+1}^*(l_{t+1}) \cdot \max_{0<s_t\le l_t} P(s_t), & 1 \le t < T \end{cases} \tag{23}$$

Although $t = T$ is the first step of the optimization process, it is the last time slot in practice. Therefore, the whole remaining data bit, l_T, must be transmitted in the last time slot to meet the deadline T imposed on the computation offload. Thus, the optimal number of data bits scheduled in time slot $t = T$ is $s_1^*(l_T, SD_T, SR_T, RD_T, SNR_T, SNR'_T) = l_T$. Besides, the queue length of the requesting vehicle **s** in slot $t = T$ is $Q_T = l_T$ and the initial threshold λ_T is subject to a uniform distribution (i.e., $\lambda_T \sim U(0,1)$). Then, we can calculate the expected optimal objective function in time slot $t = T$ by

$$f_T^*(l_T) = P(l_T) = \left[1 - \frac{\Gamma(\alpha, \frac{W_{\mathbf{s}}(T)}{\beta\lambda_T l_T})}{\Gamma(\alpha)}\right] \times \left[1 - \frac{SD_T^k(SR_T^k + \frac{SNR_T}{SNR'_T}RD_T^k)\cdot[2^{2(1-\lambda_T)l_T} - 1]^2}{2SNR_T^2}\right] \tag{24}$$

In addition, a practical implementation of the reliability-oriented vehicular computation offloading based on the dynamic programming approach is described in Algorithm 1, where the output $f^*(1,1)$ is the optimal reliability of the system.

Algorithm 1 Reliability-Oriented V2V Computation Offloading

Input: α, β, k, privacy tolerance threshold ϕ, λ_{t+1}, $W_{\mathbf{dH}}$, $W_{\mathbf{dL}}$ and T-dimensional arrays SNR, SNR', SD, SR, RD

Output: $f^*(1,1)$

```
Initialize the countdown time index t as t = T;
if t = T then
    for i = 1 to D + 1 do
        l(i,t) ← i − 1, s*(i,t) ← l(i,t) ;
        Initialize λ(i,t), φ and W_d(i,t) ← W_dH ;
        calculate f*(i,t) according to (24);
    end
end
else
    for t = T − 1 to 1 do
        for i = 1 to D + 1 do
            let tmp(i) ← 0; l(i,t) ← i − 1;
            for j = 1 to l(i,t) + 1 do
                s(j,t) ← j − 1, next ← l(i,t) − s(j,t);
                nextn ← find(l(:,t+1) − next = 0);
                calculate λ(i,t) according to (16) where
                  λ_{t+1} ← λ(nextn,t+1);
                calculate P(s(j,t)) according to (18) ;
                f(i,t) ← f*(nextn,t+1) · P(s(j,t));
                if f(i,t) > tmp(i) then
                    let f*(i,t) ← f(i,t), s*(i,t) ← s(j,t);
                    replace tmp(i) with f(i,t);
                    calculate Pri_d(i,t);
                    if Pri_d(i,t) < φ then
                        let W_d(i,t) ← W_dL;
                    end
                    else
                        let W_d(i,t) ← W_dH;
                    end
                    calculate θ(i,t) and λ(i,t);
                end
            end
        end
    end
end
return f*(1,1);
```

B. Complexity Analysis

Since the algorithm involves two stages (The process of path restoration is omitted in the algorithm.), we analyze the complexity in a sequential way. Firstly, D is defined as input n. When $t = T$, the variables s_t, l_t, f_t have linear complexity

$$\max_{\{s_i\}_{i=1}^T} : \prod_{t=1}^{T}\left\{\left[1 - \frac{1}{2}SD_t^k(SR_t^k + \frac{SNR_t}{SNR'_t}RD_t^k)\frac{(2^{2(1-\lambda_t)s_t} - 1)^2}{SNR_t^2}\right] \times \left[1 - \int_{W_{\mathbf{s}}(t)/\lambda_t s_t}^{\infty} \frac{1}{\beta^\alpha\Gamma(\alpha)}(x)^{\alpha-1}e^{-\frac{x}{\beta}}d_x\right]\right\}$$
$$= \prod_{t=1}^{T}\left\{\left[1 - \frac{SD_t^k(SR_t^k + \frac{SNR_t}{SNR'_t}RD_t^k)[2^{2(1-\lambda_t)s_t} - 1]^2}{2SNR_t^2}\right] \times \left[1 - \frac{\Gamma(\alpha, \frac{\beta W_{\mathbf{s}}(t)}{\lambda_t s_t})}{\Gamma(\alpha)}\right]\right\}$$
$$s.t. \begin{cases} \sum_{t=1}^{T} s_t = D, \\ \forall s_t > 0. \end{cases} \tag{17}$$

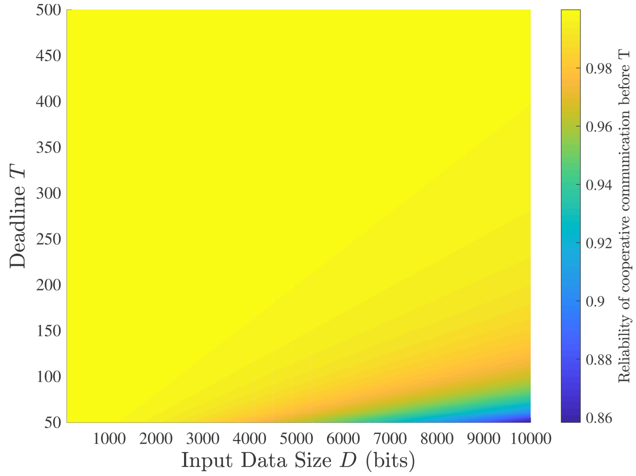


Fig. 3. Reliability of independent communication.

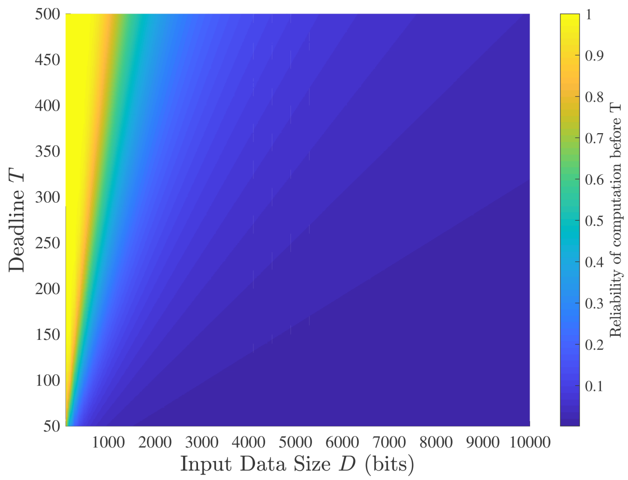


Fig. 4. Reliability of independent computation with $SNR = 30$dB.

with the data size D. So the time complexity is $O(3D) = O(n)$, where $O(n)$ means the upper bound for the complexity grows with order n. In the second stage, when t is from $T-1$ to 1, the reverse dynamic programming is used to solve the optimal reliability. Firstly, traverse all possible state variables l_t and the decision variables s_t allowed by each l_t, the time complexity is $O(TD^2) = O(n^2)$ which is affected by the Data size. Secondly, find l_{t+1} in previous stage that matches the result of the state transition equation through the binary search method, the time complexity is $O(\log D) = O(\log n)$. Therefore, our proposed algorithm has a total complexity in $O(n^2 \log n)$.

V. Numerical Results

In this section, we present our simulation results of the joint offloading and computation scheme. We conduct different simulation experiments, where the parameter and communication configurations are shown in Table I. In particular, we set $SNR_H = 15$dB and $SNR_L = 10$dB respectively to simulate the good and the bad SNR conditions, and randomly select the values of SNR and SNR' between 10dB and 15dB in each time slot t to simulate dynamic conditions.

TABLE I

Simulation Parameters and Communication Configuration

D [bits]	10^4
T	500
k	2
α	2
β	0.008
B [MHz]	10
$W_{\mathbf{s}}$ [GHz/s]	0.2
$W_{\mathbf{d}H}$[GHz/s]	$(0.02, 0.2)$
$W_{\mathbf{d}L}$ [GHz/s]	1
$X_{\mathbf{d}}$ [MHz/bit/s]	0.25
SNR[dB]	$[10, 15]$
SNR'[dB]	$[10, 15]$
$\{SD_0, SR_0, RD_0\}$[m]	$\{15, 12.5, 12.5\}$

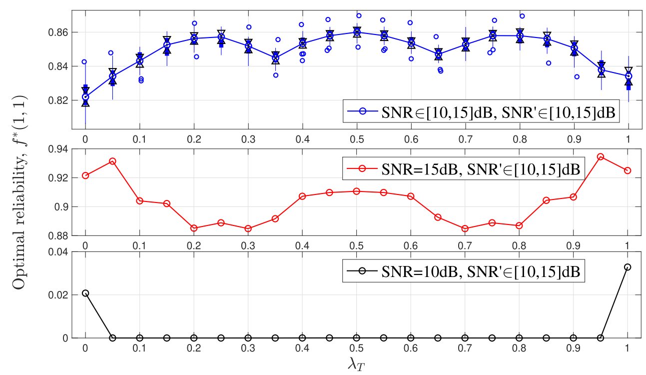


Fig. 5. Optimal reliability $f^*(1,1)$ under different initial offloading decision coefficient λ_T.

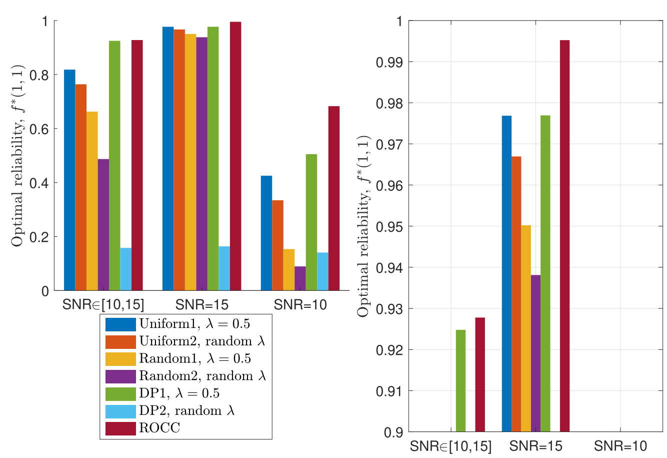
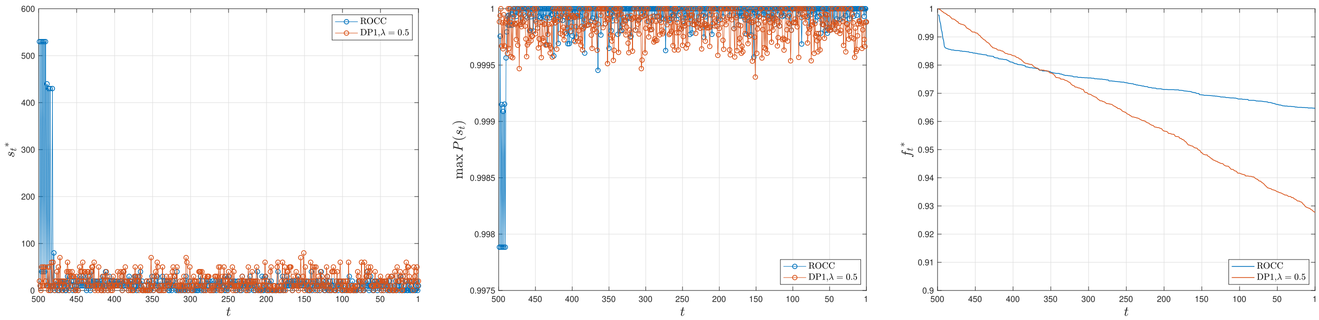


Fig. 6. Reliability performance comparison.

There are two basic situations in actual implementation: (1) All computing tasks are transmitted to the service vehicle via a cooperative communication link. (2) Vehicles with calculation requirements can decide to performance local execution for all the data instead of computation offloading. We evaluate the proposed reliability of above situations by imposed deadline and input data in Fig.3 and Fig.4, where the size of the application input data D is varied from 100 bits to 10^4 bits and the given deadline from 50 to 500 unit time



(a) Optimal computation scheduling (b) Stage reliability index (c) Optimal reliability

Fig. 7. Optimal data computation scheduling under dynamic conditions.

slots. As can be seen from Fig.3, with a larger processing data by a given deadline, the reliability of communication performs worse. Similarly, the CPU is likely to be unable to complete data processing tasks when the application data is large. For example, it can be seen from Fig.4 that when the local CPU needed to successfully process 1500 bits of data in 500 time slots, the reliability of independent computation is lower than 0.5.

Fig.5 provides the influence of the initial offloading decision coefficient λ_T on the optimal reliability of the system under different channel conditions. The three sub-figuresi from top to bottom shows the optimal reliabilities of the system when $\mathbf{s} \to \mathbf{d}$ is an uncertain channel, a channel with SNR_H, and a channel with SNR_L. By averaging the results of multiple simulations, we can draw the following conclusions: (1) For the random channel, the proposed optimal reliability performance is the best when the initial offloading decision coefficient λ_T is set to 0.5; (2) when the channel state can maintain a stable level, $\lambda_T = 1$ is a better choice. The above conclusions provide guidance for the subsequent experiments. We will select the optimal λ_T for different simulation environments to reduce the interference of various factors on the experimental results and to obtain optimal reliability.

Next, we compare the proposed scheme (marked as 'ROCC') with six other methods: (1) Uniform scheme which (marked as 'Uniform1') schedules equal data bits in each time slot and half of them to service vehicle; (2) Uniform scheme which (marked as 'Uniform2') schedules equal data bits in each time slot and randomly schedules them to service vehicle; (3) Random scheme which (marked as 'Random1') schedules random data bits in each time slot and half of them to service vehicle; (4) Random scheme which (marked as 'Random2') schedules equal data bits in each time slot and randomly schedules them to service vehicle; (5) Dynamic programming scheme which (marked as 'DP1') schedules data bits through a standard dynamic programming and half of them to service vehicle; (6) Dynamic scheme which (marked as 'DP2') schedules data bits through a standard dynamic programming and randomly schedules them to service vehicle. Extensive Monte Carlo simulations have been carried out with 1000 replications per initial state condition. The numerical results are given in Fig.6. It can be seen in the figure that the proposed scheme ROCC has a better optimal reliability performance than any other method under different channel states (bad SNR conditions, good SNR conditions, dynamic conditions) and limited computational resources. Specifically, compared with the dynamic programming scheme (DP1) with better performance, the proposed scheme is 35.21% higher when the channel state is bad. When the channel state is good, the proposed scheme is 1.87% higher than DP1 and when the channel is under dynamic conditions the proposed scheme has a 0.32% advantage against DP1. This confirms the advantage of our proposed method in stochastic and limited communication and computation situations.

Fig.7a shows the optimal data computation schedule s_t^* for ROCC and DP1 under dynamic conditions. It can be seen from the figure that in the initial 20-time slots, ROCC has an obvious trend of oscillation. This oscillation period indicates that the algorithm is adjusting the system configuration according to the channel conditions so that the dynamic adjustment of the distribution coefficient in the later period can maintain the stability of the system. So as to obtain the optimal system reliability under the premise of maintaining the load balance between vehicles after entering the stable period. Fig.7b illustrates the stage reliability index, which is similar to Fig.7a. It is obvious that driven by the proposed ROCC algorithm, the system is divided into two phases: an oscillation period and a stable period. After a very short period of oscillation, the algorithm can maintain the overwhelming majority of stage reliability index at a higher level. e.g., after $t = 480$, the stage reliability index of ROCC is mostly better than the stage reliability index of DP1. Moreover, Fig.7c compares the changes in system optimal reliability of two optimization schemes over the deadline T.

VI. Conclusion

In this paper, we have explored the reliability-oriented modeling of cooperative communication and computational offloading. Then we have formulated constrained optimization problems by combining the reliability modeling of both V2V communications and partial offloading. In addition, we have proposed a virtual queue model that combines queue length and vehicle privacy entropy to optimize partitioning. To gain a better insight, we have proposed to solve it via dynamic programming method. Simulation results have revealed the effectiveness and advantage of our method in guaranteeing the coupling reliability performance and improving completion rate (e.g., Compared with 6 methods based on uniform, random or dynamic programming, the optimal reliabilities of the proposed schemes are 13.39%, 21.45%, 40.05%, 90.50%, 0.32%, 488.50% higher, respectively.). In addition, since the proposed optimal reliability decreases as the service vehicle's privacy sensitivity to collaborative events increases, it is very important to consider the load balancing of the system during the optimization process.

References

[1] S. K. Datta, J. Haerri, C. Bonnet, and R. F. D. Costa, "Vehicles as connected resources: Opportunities and challenges for the future," *IEEE Veh. Technol. Mag.*, vol. 12, no. 2, pp. 26–35, Jun. 2017.

[2] A. C. Baktir, A. Ozgovde, and C. Ersoy, "How can edge computing benefit from software-defined networking: A survey, use cases, and future directions," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2359–2391, 4th Quart., 2017.

[3] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 450–465, Feb. 2018.

[4] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, "Cooperative diversity in wireless networks: Efficient protocols and outage behavior," *IEEE Trans. Inf. Theory*, vol. 50, no. 12, pp. 3062–3080, Dec. 2004.

[5] A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity-part II: Implementation aspects and performance analysis," *IEEE Trans. Commun.*, vol. 51, no. 11, pp. 1939–1948, Nov. 2003.

[6] A. S. Ibrahim, A. K. Sadek, W. Su, and K. J. Liu, "Cooperative communications with relay-selection: When to cooperate and whom to cooperate with?" *IEEE Trans. Wireless Commun.*, vol. 7, no. 7, pp. 2814–2827, Jul. 2008.

[7] D. Michalopoulos and G. Karagiannidis, "Distributed switch and stay combining (DSSC) with a single decode and forward relay," *IEEE Commun. Lett.*, vol. 11, no. 5, pp. 408–410, May 2007.

[8] K.-S. Hwang, Y.-C. Ko, and M.-S. Alouini, "Performance analysis of incremental opportunistic relaying over identically and non-identically distributed cooperative paths," *IEEE Trans. Wireless Commun.*, vol. 8, no. 4, pp. 1953–1961, Apr. 2009.

[9] C. Young *et al.*, "Protium, an infrastructure for partitioned applications," in *Proc. 8th Workshop Hot Topics Operating Syst.*, May 2001, pp. 47–52.

[10] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.

[11] P. Dai, K. Liu, X. Wu, H. Xing, Z. Yu, and V. C. S. Lee, "A learning algorithm for real-time service in vehicular networks with mobile-edge computing," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6.

[12] 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.

[13] T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili, "Collaborative mobile edge computing in 5G networks: New paradigms, scenarios, and challenges," *IEEE Commun. Mag.*, vol. 55, no. 4, pp. 54–61, Apr. 2017.

[14] 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.

[15] Y. Zhang, B. Di, P. Wang, J. Lin, and L. Song, "HetMEC: Heterogeneous multi-layer mobile edge computing in the 6G era," *IEEE Trans. Veh. Technol.*, vol. 69, no. 4, pp. 4388–4400, Apr. 2020.

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

[17] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1628–1656, 3rd Quart., 2017.

[18] 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.

[19] B. C. Kavitha, R. Vallikannu, and K. S. Sankaran, "Delay-aware concurrent data management method for IoT collaborative mobile edge computing environment," *Microprocessors Microsyst.*, vol. 74, Apr. 2020, Art. no. 103021. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0141933119305666

[20] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2795–2808, Oct. 2016.

[21] O. Muñoz, A. Pascual-Iserte, and J. Vidal, "Optimization of radio and computational resources for energy efficiency in latency-constrained application offloading," *IEEE Trans. Veh. Technol.*, vol. 64, no. 10, pp. 4738–4755, Oct. 2015.

[22] T. X. Tran and D. Pompili, "Joint task offloading and resource allocation for multi-server mobile-edge computing networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 856–868, Jan. 2019.

[23] J. Zhang, W. Xia, F. Yan, and L. Shen, "Joint computation offloading and resource allocation optimization in heterogeneous networks with mobile edge computing," *IEEE Access*, vol. 6, pp. 19324–19337, 2018.

[24] H. Sun, H. Yu, G. Fan, and L. Chen, "Energy and time efficient task offloading and resource allocation on the generic IoT-fog-cloud architecture," *Peer-to-Peer Netw. Appl.*, vol. 13, no. 2, pp. 548–563, Mar. 2020, doi: 10.1007/s12083-019-00783-7.

[25] Q.-V. Pham, L. B. Le, S.-H. Chung, and W.-J. Hwang, "Mobile edge computing with wireless backhaul: Joint task offloading and resource allocation," *IEEE Access*, vol. 7, pp. 16444–16459, 2019.

[26] H. Sun, F. Zhou, and R. Q. Hu, "Joint offloading and computation energy efficiency maximization in a mobile edge computing system," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 3052–3056, Mar. 2019.

[27] C.-F. Liu, M. Bennis, and V. H. Poor, "Latency and reliability-aware task offloading and resource allocation for mobile edge computing," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2017, pp. 1–7.

[28] Z. Lu, G. Qu, and Z. Liu, "A survey on recent advances in vehicular network security, trust, and privacy," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 2, pp. 760–776, Feb. 2019.

[29] D. Manivannan, S. S. Moni, and S. Zeadally, "Secure authentication and privacy-preserving techniques in vehicular ad-hoc networks (VANETs)," *Veh. Commun.*, vol. 25, Oct. 2020, Art. no. 100247. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2214209620300188

[30] P. Asuquo *et al.*, "Security and privacy in location-based services for vehicular and mobile communications: An overview, challenges, and countermeasures," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4778–4802, Dec. 2018.

[31] J. Sun, C. Zhang, Y. Zhang, and Y. Fang, "An identity-based security system for user privacy in vehicular ad hoc networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 21, no. 9, pp. 1227–1239, Sep. 2010.

[32] Y. Xi, W. Shi, and L. Schwiebert, "Mobile anonymity of dynamic groups in vehicular networks," *Secur. Commun. Netw.*, vol. 1, no. 3, pp. 219–231, May 2008, doi: 10.1002/sec.28.

[33] X. Zhang, X. Gui, F. Tian, S. Yu, and J. An, "Privacy quantification model based on the bayes conditional risk in location-based services," *Tsinghua Sci. Technol.*, vol. 19, no. 5, pp. 452–462, Oct. 2014.

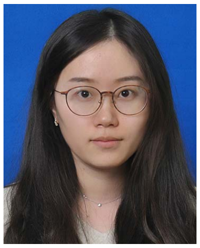
[34] A. Lesne, "Shannon entropy: A rigorous notion at the crossroads between probability, information theory, dynamical systems and statistical physics," *Math. Struct. Comput. Sci.*, vol. 24, no. 3, Jun. 2014, Art. no. e240311.

[35] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. O. Wu, "Energy-optimal mobile cloud computing under stochastic wireless channel," *IEEE Trans. Wireless Commun.*, vol. 12, no. 9, pp. 4569–4581, Sep. 2013.

[36] C. You, K. Huang, H. Chae, and B.-H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1397–1411, Mar. 2017.

[37] J. D. C. Little, "A proof for the queuing formula: L= λ w," *Oper. Res.*, vol. 9, no. 3, pp. 383–387, 1961.

[38] S. Lederer, A. K. Dey, and J. Mankoff, *A Conceptual Model and A Metaphor of Everyday Privacy in Ubiquitous Computing Environments*. (Computer Science Division), Los Angeles, CA, USA: Univ. California, 2002.



Xu Han received the B.E. and M.Sc. degrees in software engineering from Jilin University. She is currently pursuing the Ph.D. degree with the School of Transportation Science and Engineering, Beihang University, Beijing, China. Her current research interests include wireless communication, privacy protection in the Internet of Vehicles, and intelligent transportation systems.

Daxin Tian (Senior Member, IEEE) is currently a Professor with the School of Transportation Science and Engineering, Beihang University, Beijing, China. His current research interests include mobile computing, intelligent transportation systems, vehicular ad hoc networks, and swarm intelligence. He is also a member of the IEEE Intelligent Transportation Systems Society and the IEEE Vehicular Technology Society.

Zhengguo Sheng (Senior Member, IEEE) received the B.Sc. degree from the University of Electronic Science and Technology of China (UESTC) in 2006, and the M.S. (Hons.) and Ph.D. degrees from Imperial College London in 2007 and 2011, respectively. He has been a Senior Lecturer with the Department of Engineering and Design, University of Sussex, since 2015. His current research interests include the Internet of Things (IoT), connected vehicles, and cloud/edge computing.

Xuting Duan is currently an Assistant Professor with the School of Transportation Science and Engineering, Beihang University, Beijing, China. His current research interest includes vehicular ad hoc networks.

Jianshan Zhou received the B.Sc., M.Sc., and Ph.D. degrees in traffic information engineering and control from Beihang University, Beijing, China, in 2013, 2016, and 2020, respectively. From 2017 to 2018, he was a Visiting Research Fellow with the School of Informatics and Engineering, University of Sussex, U.K. He is currently a Post-Doctoral Research Fellow supported by the "Zhuoyue" Program of Beihang University. He is the author or coauthor of more than 20 international scientific publications. His research interests include modeling and optimization of vehicular communication networks and air-ground cooperative networks, analysis and control of connected autonomous vehicles, and intelligent transportation systems. He has received the first prize in the Science and Technology Award from the China Intelligent Transportation Systems Association (China-ITS) in 2017, and the first prize in the Innovation and Development Award from the China Association of Productivity Promotion Centers (CAPPC) in 2020. He also received the national scholarships in 2017 and 2019, the outstanding top-ten Ph.D. candidate prize from Beihang University in 2018, and the Outstanding China-SAE Doctoral Dissertation Award in 2020. He serves/served as the Technical Program Session Chair in IEEE EDGE 2020 and the Youth Editorial Board Member of the Unmanned Systems Technology.

Wei Hao (Member, IEEE) received the Ph.D. degree from the Department of Civil and Environmental Engineering, New Jersey Institute of Technology. He is currently a Professor with the Changsha University of Science and Technology, China. His research interests include traffic operations, ITS, planning for operations, traffic modeling and simulation, connected automated vehicles, and travel demand forecasting.

Kejun Long is currently a Professor with the Changsha University of Science and Technology, China. His research interests include planning for operations, traffic information engineering and control, traffic safety and driving behavior, ITS, traffic modeling and simulation, and connected automated vehicles.

Min Chen (Senior Member, IEEE) is currently a Professor with the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China. His current research interests include cyber physical systems, IoT sensing, 5G networks, mobile cloud computing, SDN, healthcare big data, medical cloud privacy and security, body area networks, emotion communications, and robotics.

Victor C. M. Leung (Life Fellow, IEEE) is currently a Distinguished Professor of Computer Science and Software Engineering with Shenzhen University, Shenzhen, China, and a Professor Emeritus with The University of British Columbia (UBC), Vancouver, Canada. Before he retired from UBC at the end of 2018, he was a Professor of Electrical and Computer Engineering and holder of the TELUS Mobility Research Chair there. He has coauthored more than 1300 journal/conference papers and book chapters. His research interests include wireless networks and mobile systems. He is a Fellow of the Royal Society of Canada, the Canadian Academy of Engineering, and the Engineering Institute of Canada. He received the IEEE Vancouver Section Centennial Award, the UBC Killam Research Prize in 2011, the Canadian Award for Telecommunications Research in 2017, and the IEEE TCGCC Distinguished Technical Achievement Recognition Award in 2018. His coauthored papers that won the 2017 IEEE ComSoc Fred W. Ellersick Prize, the 2017 IEEE Systems Journal Best Paper Award, the 2018 IEEE CSIM Best Journal Paper Award, and the 2019 IEEE TCGCC Best Journal Paper Award. He serves on the editorial boards of the IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING, IEEE TRANSACTIONS ON CLOUD COMPUTING, IEEE ACCESS, IEEE *Network*, and several other journals. He is named in the current Clarivate Analytics list of "Highly Cited Researchers."