Optimal Task Allocation in Vehicular Fog Networks Requiring URLLC: An Energy-Aware Perspective

Tingting Liu, *Member, IEEE*, Jun Li, *Senior Member, IEEE*, Feng Shu, *Member, IEEE*, and Zhu Han, *Fellow, IEEE*

***Abstract*—In order to make intelligent transportation systems (ITSs) come true, execution of a large amount of data needs to be migrated from the cloud centers to the edge nodes, especially in the scenarios requiring ultra reliable low latency communications (URLLC). In this article, we propose to study the energy-aware task allocation problem in the vehicular fog networks considering URLLC. Specifically, a requester who has some bursty computation tasks which cannot be finished within a required time by itself, needs to decide whether the nearby computation nodes can meet the latency and reliability requirements, and which nodes should be chosen. Given the required latency and reliability, the maximum computation capacity of each fog node is first calculated based on the martingale-theory-derived delay bound. Then, if the available fog nodes can accommodate the computation tasks, two different optimization problems concerning the energy efficiency maximization and the energy consumption minimization are constructed further. The corresponding solutions are also provided. Specifically, the optimal solution in maximizing the energy efficiency is not unique, while the optimal solution in minimizing the energy consumption is unique. Moreover, the latter solution is provided as a truncated-channel-inversion like policy. At last, numerical results are illustrated to demonstrate effectiveness of the proposed optimal task allocation schemes from the perspectives of the energy efficiency and the energy consumption.**

***Index Terms*—Computation Offloading, Energy Efficiency, Task Allocation, Truncated-Channel-Inversion Like Policy, URLLC, Vehicular Fog Networks.**

I. Introduction

RECENTLY, intelligent transportation systems (ITSs), which aim to provide high efficient traffic, safety road, autonomous driving and accurate navigation, etc., have attracted enormous attentions from both academia and industry [1], [2]. In order to implement the ITSs, a large amount of data has to be reliably processed within a very limited time. The vision of the ITSs, especially its stringent requirements on reliability and latency, can be classified as ultra-reliable and low latency communications (URLLC), one of the 5G usage pattern [3], [4]. However, at present, the processing of big data mainly depends on cloud centers which are usually located in remote areas [5]. Albeit clouds have abundant computation resources, in order to meet the high reliability and low latency requirements, the ITSs cannot process the generated data solely relying on the cloud centers.

Fog computation has been proposed to decrease the communication and computation latency in realizing the ITSs by introducing a layer of computation devices between the cloud centers and end users [6]–[10]. Usually, road side units (RSUs) and small-cell base stations (SBSs) are proposed as computation devices to help connected vehicles improve their task computation efficiency [11]–[14]. Among these works, vehicles are considered as computation sources, making use of computation resources in RSUs or SBSs. However, on the one hand, the expenditures of deploying RSUs are very high, especially in the scenarios requiring dense RSU coverage. On the other hand, it is commonly known that the SBSs are operated by the cellular network operators. It's challenging to realize URLLC between two independent systems. Nevertheless, driven by the increasing demands on connections between vehicles, vehicular manufacturing industry begins to deploy intelligent computation and communication modules into their products. Moreover, Federal Communication Commission (FCC) has issued a 75 MHz frequency band for the dedicated short-range communication (DSRC) between connected vehicles [15]. Also, the release of IEEE 802.11p protocol makes vehicle-to-vehicle (V2V) communication possible. Driven by the connections among vehicles as well as the large on-board battery capacity, it is promising that computation tasks can be alternatively executed inside the connected vehicles, not only dependent on external infrastructures or systems. Taking vehicles as computation resources may reduce the cost of realizing the ITSs, and improve computational efficiency.

Therefore, how to properly allocate the vehicular computation tasks to satisfy the URLLC has arisen as a crucial issue in realizing the ITSs. Delay, as one of the important URLLC factor, includes communication delay and queueing delay [16], [17].

Manuscript received July 22, 2019; revised November 1, 2019; accepted November 17, 2019. Date of publication November 25, 2019; date of current version September 2, 2020. This work was supported in part by the National Natural Science Foundation of China under Grants 61702258, 61771244, 61872184, and 61727802, and in part by the US MURI AFOSR MURI 18RT0073, NSF EARS-1839818, CNS-1717454, CNS-1731424, CNS-1702850, and CNS-1646607. Recommended for acceptance by K. R. Chowdhury. *(Corresponding authors: Feng Shu; Jun Li.)*

T. Liu is with the School of Information and Communication Engineering, Nanjing Institute of Technology, Nanjing 211167, China (e-mail: liutt@njit.edu.cn).

J. Li and F. Shu are with the School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: jun.li@njust.edu.cn; shufeng@njust.edu.cn).

Z. Han is with the University of Houston14743, Houston, TX 77004, USA and also with the Department of Computer Science and Engineering, Kyung Hee University, Seoul 446-701, South Korea (e-mail: zhan2@uh.edu).

This article has supplementary downloadable material available at https://ieeexplore.ieee.org, provided by the authors.

Digital Object Identifier 10.1109/TNSE.2019.2955474

In [16], the authors investigate the joint computation partitioning and resource allocation for latency sensitive applications. An efficient heuristic and an online solution are proposed to solve this problem. In [17], the authors study the joint task and network flow scheduling in order to minimize the completion time of the application. The communication delay that shows a determined characteristic has been exhaustively studied in the existing vehicular-related works [18]–[20]. Meanwhile, queueing delay is also a non-negligible issue, especially in the URLLC scenario [21]–[23]. Due to its stochastic characteristic, it is challenging to calculate the queueing delay. Instead of directly calculating the queueing delay time, queueing delay bounds which can be obtained by utilizing the effective bandwidth theory [24] or the effective capacity theory [25] are proposed to indicate the delay performance. In [26], the authors investigate the resource allocation problem in vehicular networks to maximize the sum ergodic capacity of the vehicle-to-infrastructure (V2I) links. While the V2V links need to satisfy the requirements on latency violation probability, which is derived by the effective capacity theory.

At the same time, the effective bandwidth/capacity theory is challenged by its loose result for non-Poisson processes [27]. Martingale theory is proposed as a valuable alternative in estimating the delay bound [28]. It fits any arrival and service processes, and especially, it can provide a very tight delay bound in a bursty traffic scenario. In [29], the authors provide a theoretical way to measure the end-to-end delay bound using the martingale theory in multimedia heterogeneous high-speed train networks where the link from train to track-side-access point is highly dynamic and bursty. Simulation results demonstrate that the derived delay bound are remarkably tight to the real data trace results. Furthermore, in their next work, the delay bounds are investigated in multi-hop vehicular ad hoc networks where data from vehicles is expected to be bursty [30]. It is also verified that the martingale delay bound is very tight to the real data trace. Besides providing tight system delay bound, the martingale theory can be utilized to construct optimization problems. In [31], the authors optimize the network energy efficiency subjected to a certain delay-bound violation probability which is derived within the framework of the martingale theory in machine type communication networks. In [32], the optimal task allocation scheme is derived with the objective to minimize the overall network delay-bound violation probability based on the martingale theory.

Moreover, energy-related problem is also very critical in realizing the ITSs. Energy-related problem is a systematic issue, and it requires integration and coordination of different systems. Although there is adequate energy in a vehicular network, it is still necessary to discuss the energy efficiency of computation offloading in such a system. There are two kinds of computation sources, i.e., the on-board application and the in-vehicle user equipment. Usually, the on-board applications are powered by the vehicles. Many on-board applications, such as route selection, control system, and driving strategy, are designed to improve the vehicular energy efficiency. In [33], the authors propose a timely and energy-efficient route selection algorithm based on historical driving data. In [34], a predictive control system is proposed to reduce the driving energy consumption, while maintaining a suitable distance from the preceding vehicles. In [35], the authors propose model-based driving strategies to predict and optimize the energy consumption of a trip via eco-routing, eco-driving and energy consumption prediction. In order to obtain timely and reliable feedback, almost all of the on-board applications require massive computations. As more and more on-board applications are installed in a vehicle, the massive computations will consume significant energy. Therefore, it is of great importance to design an energy-efficient computation scheme to realize on-board applications, while maintaining timely and reliable feedback. Different from the on-board applications, the in-vehicle user equipments usually suffer from energy constraint. In [36], the authors study the energy-efficient computation offloading problem for in-vehicle user equipment and provide a distributed solution based on consensus alternating direction method of multipliers (ADMM). Thus, no matter the on-board applications or the in-vehicle user equipments, improving the energy efficiency of computation is vital in a vehicular network. From the perspective of computation offloading, two commonly recognized objectives for computation offloading are reducing execution time and shifting energy consumption. Thus, at the mention of computation offloading, no matter in what kind of network, the computation source node has to make two classes of decision, i.e., what computation to offload and where to offload [37]. Many works have been done in this area. For example, in [38], the authors propose an energy-efficient computation offloading decision-making scheme in a combined fog and cloud scenario to minimize the system cost. In [39], by optimally designing the computation offloading decision, the authors can improve the energy efficiency of computation. To summarize, energy efficiency is a commonly used criterion in making computation offloading decisions.

Different from the existing works in [29]–[39], in this paper, we are inspired to study the energy-aware task allocation problem in vehicular fog networks, where the computation tasks have a stringent execution time and reliability requirements. By jointly considering the channel queueing delay and the computation delay, delay-bound violation probability is derived by following the martingale theory. Specifically, one vehicular node, named as requester, has some bursty computation tasks which cannot be timely processed by itself. It needs to divide this task into small subtasks and parallelly executed in the nearby vehicular fog nodes. The requester needs to determine whether the surrounding vehicular fog nodes can accommodate these subtasks while satisfying the requirements on latency and reliability, and which nodes should be chosen to enhance the energy-aware performance further.

The proposed task allocation schemes can satisfy the requirements on latency and reliability, and moreover, it can enhance the energy-related performance of the networks. The main contributions of this paper are summarized as follows,

1) A vehicular computation offloading scenario is constructed, where the generated task with the burst nature is modeled as a Markov-modulated on off (MMOO) process, while the transmission is modeled as an independent and identically distributed (i.i.d.) process. By jointly

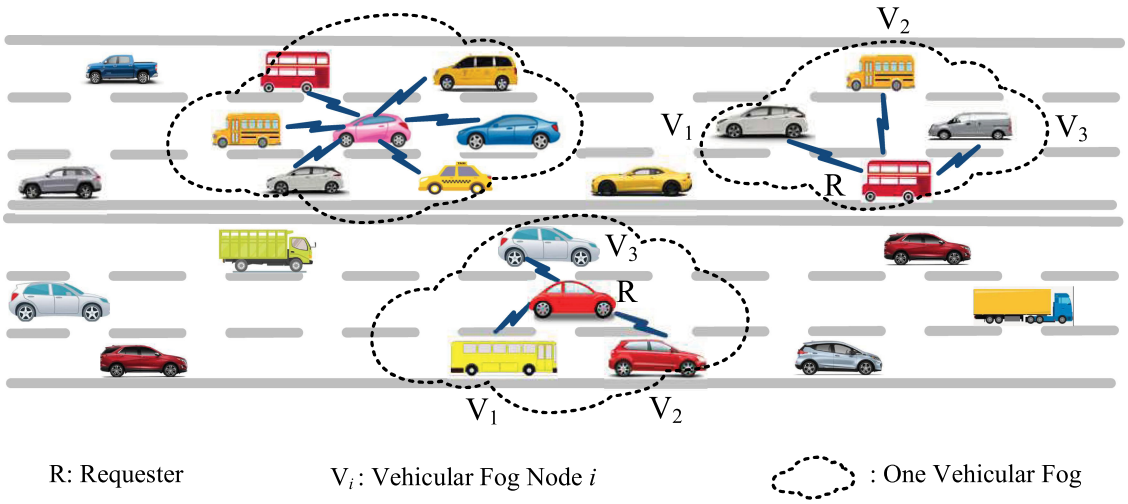


Fig. 1. A simple example in vehicular fog computation offloading.

considering the channel queueing delay and computation delay, the delay-bound violation probability is then provided by following the martingale theory.

2) Given the required time and the reliability, we calculate the maximum computation capacity of each available vehicular fog node based on the martingale-theory-derived delay bound.
3) If the summation of the nearby computation capacity is larger than the computation task, two different optimization problems, considering energy efficiency and energy consumptions, are established to enhance the network performance further.
4) At last, the corresponding task allocation solutions are provided. Especially the optimal solution for the energy consumption optimization problem is constructed as a truncated-channel-inversion like policy. When the inherent parameter of a vehicular node is larger than a predetermined threshold, this node will be suspended in order to reduce the overall network energy consumption.
5) Simulation results are provided to demonstrate the performance of the proposed schemes from the perspectives of the energy efficiency and the energy consumption.

The rest of this paper is organized as follows: the system model is presented in Section II. The optimal task allocation problem considering latency, reliability and energy is investigated in Section III. Numerical results are presented in Section IV, and conclusions are drawn in Section V.

II. System Model

In this section, we consider a vehicular fog computation scenario. A simple example of vehicular fog computation offloading is depicted in Fig. 1. In this system, there are multiple fogs, each of which consists of one requester and many fog nodes as shown in Fig. 1. In one fog, we assume there are N available vehicular fog nodes, denoted by $\mathbf{V} = \{\mathcal{V}_1, \ldots, \mathcal{V}_N\}$, within the scope of the requester $\mathcal{R}$'s communication range. $\mathcal{R}$ has multiple computation tasks which need to be executed within time T. Due to the limited local computation resources, $\mathcal{R}$ cannot finish the tasks in time by itself. It needs to offload the computation tasks to the nearby computation nodes. Therefore, $\mathcal{R}$ needs to decide whether the tasks can be finished timely within the available computation nodes and which nodes should be employed. For simplicity, arbitrary precision is assumed in task partition, and any two sub-tasks have no overlap. For example, in a autonomous driving scenario, a large amount of images can be divided in to several subsets that can be processed in parallel.

A. Data Generation Model

Usually, the requester $\mathcal{R}$ is equipped with some applications, for example, the on-board autonomous driving or entertainments in user's equipment, to enhance the driving experience and passengers' satisfactions. The bursty application data amount $a(k)$ at time k can be modeled as an MMOO process by utilizing the Monte Carlo Markov Chain (MCMC) method [40]. The MMOO process has two static status $\Pi_a \triangleq [\pi_a^0, \pi_a^1]$. On state π_a^0, there is no application data generated, i.e., $a(k) = 0$, while on state π_a^1, $a(k) = R\,bits/s$, and $R > 0$. The corresponding state transition matrix is defined as

$$\mathrm{Tr}_a \triangleq \begin{bmatrix} 1-\alpha & \alpha \\ \beta & 1-\beta \end{bmatrix}, \tag{1}$$

where α represents the transition probability from state π_a^0 to state π_a^1, while β represents the transition probability from π_a^1 to π_a^0. Accordingly, the steady state distribution of $a(k)$ is calculated as

$$\Pi_a = \left[\frac{\beta}{\alpha+\beta}, \frac{\alpha}{\alpha+\beta}\right]. \tag{2}$$

The unit of time interval is one second. Therefore, the cumulative generating data over time interval $[m, n]$ is represented as

$$A(m,n) = \sum_{k=m}^{n} a(k) * 1, \tag{3}$$

where $A(m,n)$ can be regarded as a bivariate arrival process. If $m = 0$, we use $A(0,n) \triangleq A(n)$ for brevity.

B. Transmission Service Model

When the generated computation tasks cannot be finished within the required time T by itself, the requester $\mathcal{R}$ needs to offload its tasks to the nearby vehicular fog nodes. In order to guarantee a constant receiving power P_i^c on the $\mathcal{V}_i$'s side, a truncated channel inversion policy with a cutoff channel gain g_o is employed by $\mathcal{R}$ [31]. The outage probability P_{out} can be represented as,

$$P_{\text{out}} = \Pr(g_i < g_o). \tag{4}$$

During the transmission stage, $\mathcal{R}$ provides a constant offloading rate $C_i\, bits/s$ for each $\mathcal{V}_i, i \in \{1, \dots N\}$, i.e.,

$$C_i = B \log_2\left(1 + \frac{P_i^c}{B\sigma_o^2}\right), \tag{5}$$

where B is the bandwidth allocated to one vehicular fog node and σ_o^2 is the noise variance. P_i^c is calculated as

$$P_i^c = P_{\text{tr}}(g_i) g_i d_i^{-l}, \tag{6}$$

where $P_{\text{tr}}(g_i)$ is a tunable transmission power based on the channel state g_i to guarantee a constant receiving power P_i^c, d_i denotes the distance between the requester $\mathcal{R}$ and fog node $\mathcal{V}_i$, and l is the pathloss exponent. Note that even though vehicles are moving fast, since a group of vehicles can form a relatively stable neighborhood, and the channels between vehicles will not change rapidly. Hence, the channel state information can be obtained to realize the truncated channel inversion.

Thus, the accumulative transmission process from $\mathcal{R}$ to $\mathcal{V}_i, i \in \{1, \dots N\}$ can be represented as

$$S_i(m, n) = \sum_{k=m}^{n} s_i(k) * 1, \tag{7}$$

where $s_i(k) * 1$ is the offloading amount at time k. Specifically, it is

$$s_i(k) = \begin{cases} C_i, & P_{\text{suc}}, \\ 0, & 1 - P_{\text{suc}}, \end{cases} \tag{8}$$

where P_{suc} represents the probability of successful transmission, i.e.,

$$P_{\text{suc}} = 1 - P_{\text{out}}. \tag{9}$$

Observing (8), it can be seen that the transmission realizations $s_i(1), s_i(2), \dots$ can be regarded as i.i.d. variables with a non-negative distribution which is given by

$$\Pi_s \triangleq [\pi_s^0, \pi_s^1] = [1 - P_{\text{suc}}, P_{\text{suc}}]. \tag{10}$$

On state π_s^0, there is no transmission, while on state π_s^1, $s_i(k) = C_i$.

In order to make the problem non-trivial, we assume that

$$\mathbb{E}[a(k)] < \mathbb{E}[s_i(k)] < \sup a(k), \forall i. \tag{11}$$

It means that the expectation of $s_i(k)$ is larger than the expectation of $a(k)$, but it is smaller than the peak rate of $a(k)$. In this scenario, the arrival data may experience some queueing delay before it is transmitted successfully.

C. Delay Violation Probability

Every vehicular fog node $\mathcal{V}_i, i = 1, \dots, N$ is assumed to have a basic computation capability of $f_o\, cycles/s$ which can be adjusted by a tuning parameter $\gamma_i, i = 1, \dots, N$. Generally, a task can be represented by a three-field notation which contains the task size X_i in the unit of $bits$, the completion deadline in the unit of $seconds$, and the computation intensity L in the unit of $cycles/bit$ [41], [42]. The three-filed notation describes the nature of the applications, and the three parameters can be evaluated by task profilers [37], [43]. Furthermore, these parameters can be used to evaluate the computation and communication demands, as well as the execution latency [41]. When the task size X_i and the computation intensity L are known by the requester $\mathcal{R}$, the computation time D_i^C consumed on $\mathcal{V}_i$'s side can be calculated as

$$D_i^C = \frac{X_i L}{\gamma_i f_o}. \tag{12}$$

Note that the final computation outcomes are usually much smaller than the raw computation data X_i. For example, in a autonomous driving scenario, when a vehicle is approaching an intersection, a large amount of images, i.e., the raw computation data, will be processed, while the final computation outcome will be a simple instruction, such as turning left/right, stopping or going ahead. Thus, in order to make the analysis tractable, we do not consider the latency caused by the outcome feedback from $\mathcal{V}_i$ to $\mathcal{R}$.

The computation task is expected to be finished within time T which includes the computation delay determined by (12) and the channel queueing delay which has a stochastic characteristic and cannot be determined precisely. The channel queueing delay is defined as follows,

Definition 1: **Channel Queueing Delay:** the channel queueing delay D_i^Q from the requester $\mathcal{R}$ to the computation node $\mathcal{V}_i$ is defined as the waiting time in the network before the data is successfully transmitted from $\mathcal{R}$ to $\mathcal{V}_i$. D_i^Q is given by

$$D_i^Q \triangleq \inf\{k \geq 0 | A(n-k) \leq S_i(n)\}. \tag{13}$$

Before we set out to analyze the property of channel queueing delay, we first construct two supermartingales [28] as follows.

$$M_A(n) \triangleq h_a(a(n)) e^{\theta(A(n) - nK_a)}, n \geq 0, \tag{14}$$

and

$$M_S(n) \triangleq h_s(s(n)) e^{\theta(nK_s - S(n))}, n \geq 0. \tag{15}$$

$(A)_n$ is an arrival martingale process, if for every $\theta > 0$, there is a K_a and a function $h_a : \text{rng}(a(n)) \to \mathbb{R}^+$ such that the process $M_A(n)$ is a supermartingale. $\text{rng}(\cdot)$ is a range operator.

Similarly, $(S)_n$ is a service martingale process, if for every $\theta > 0$, there is a K_s and a function $h_s : \mathrm{rng}(s(n)) \to \mathbb{R}^+$ such that the process $M_S(n)$ is a supermartingale.

In this way, the arrival and service processes are characterized by two supermartingales with parameters $h_a(a(n))$, K_a, $h_s(s(n))$, and K_s, respectively. Next, we present an auxiliary definition on 'Threshold' which is useful for the following discussions.

Definition 2: **Threshold [28]:** We define the threshold H as

$$H \triangleq \min\{h_a(a)h_s(s)|a - s > 0\}. \tag{16}$$

H is the smallest value of $h_a(a)h_s(s)$ that the arrival a is larger than the service s.

In the following, we will give the delay-bound violation probability in Theorem 1. Since the channel queueing delay of multiple computation nodes are independent from each other, in Theorem 1 and its corresponding proof, we add the aforementioned supermartingale parameters with index i to indicate the specific link from the requester $\mathcal{R}$ to a fog node $\mathcal{V}_i$, i.e., $h_i^a(a(n))$, K_i^a, $h_i^s(s(n))$, K_i^s, and H_i, correspondingly. First in first out (FIFO) scheduling policy is employed in this paper. Similar to [28], the delay-bound violation probability in the concerned scenario is presented in Theorem 1.

Theorem 1: The delay-bound violation probability is calculated as

$$\begin{aligned}&\Pr\Big(D_i^Q \geq \big(T - D_i^C\big)\Big)\\ &\quad\leq \frac{\mathbb{E}\big[h_i^a(a(0))\big]}{H_i} e^{-\theta_i^* K_i^s (T - D_i^C)}\\ &\quad= \frac{\mathbb{E}\big[h_i^a(a(0))\big]}{H_i} e^{-\theta_i^* K_i^s (T - \frac{X_i L}{\gamma_i f_o})},\end{aligned} \tag{17}$$

where $H_i = \min\{h_i^a(a)h_i^s(s_i)|a - s_i > 0\}$.

Proof: Please refer to Appendix A. ■

The relationships between the martingale parameters and the physical layer parameters also can be found in Appendix A.

Definition 3: **Reliability:** reliability is defined as user's maximal endurable delay-bound violation probability.

It can be seen from Definition 3 that (17) should be smaller than the reliability. In the following section, based on the relationship between (17) and the reliability, we will discuss the maximum computation capacity of the available vehicular fog nodes, and further discuss how to enhance the network energy-aware performance, i.e., energy efficiency and energy consumption.

III. Energy-Aware Optimal Task Allocation

Given the reliability requirement ε and latency time T, we have

$$\Pr\Big(D_i^Q \geq \big(T - D_i^C\big)\Big) \leq \varepsilon. \tag{18}$$

Substituting (17) into (18), we have

$$\frac{\mathbb{E}\big[h_i^a(a(0))\big]}{H_i} e^{-\theta_i^* K_i^s (T - \frac{X_i L}{\gamma_i f_o})} \leq \varepsilon, \tag{19}$$

and then we arrive at

$$X_i \leq \frac{\gamma_i f_o}{L}\left(T + \frac{\ln \frac{\varepsilon H_i}{E[h_i^a(a(0))]}}{\theta_i^* K_i^s}\right), \forall i. \tag{20}$$

Therefore, the maximum computation capacity $X_i^{\max}$ of each vehicular fog node is represented as

$$X_i^{\max} \triangleq \frac{\gamma_i f_o}{L}\left(T + \frac{\ln \frac{\varepsilon H_i}{E[h_i^a(a(0))]}}{\theta_i^* K_i^s}\right), \forall i. \tag{21}$$

If the computation task size X is larger than any of the nearby fog nodes' computation capacity, this task should be divided into several small tasks and computed parallelly. Thereby, we need to figure out whether the nearby vehicular fog nodes can accommodate the task X, and how to optimally allocate the computation task. Therefore, in this section, we first figure out the maximum fog computation capacity considering the network computation energy constraint. Then energy efficient and energy consumption-aware task allocation problems are investigated to enhance the performance of the considered networks further.

A. Maximum Computation Capacity With Computation Energy Constraint

Recalling that the computation node has a tuning parameter γ_i, the overall computation capacity can be maximized by properly adjusting the parameter γ_i. Moreover, in practical, each node has a different hardware architecture which may induce different energy expenditures. We use a parameter $\kappa_i, i = 1, \ldots, N$ related to the hardware architecture to characterize this difference. Since the execution of each task is associated with some kinds of rewards. If the execution expenditures are larger than its rewards, it is not worth to process this task. Here, we assume that the overall execution energy expenditure of one task is constrained by $E_{\max}$. That is to say, if the energy expenditure is larger than $E_{\max}$, it is not worth to process this task. The problem is therefore formulated as

$$\begin{aligned}&\max_{\gamma_i} \sum_{i=1}^{N} \frac{\gamma_i f_o}{L}\left(T + \frac{\ln \frac{\varepsilon H_i}{E[h_i^a(a(0))]}}{\theta_i^* K_i^s}\right),\\ &\text{s.t.,} \quad \sum_{i=1}^{N} L X_i^{\max} \kappa_i (\gamma_i f_o)^2 \leq E_{\max},\\ &\qquad \gamma_i \geq 0,\end{aligned} \tag{22}$$

where the first constraint represents that the overall energy consumption in the vehicular fog cannot exceed the maximum energy $E_{\max}$. The optimal solution γ_i^* is provided in Theorem 2.

Theorem 2: Solving for γ_i with constraint that $\gamma_i \geq 0$ yields the optimal solution as

$$\gamma_i^* = \frac{f_o^{-1}\kappa_i^{-\frac{1}{2}}E_{\max}^{\frac{1}{3}}}{\left(\sum_{i=1}^{N}\kappa_i^{-\frac{1}{2}}(T+\Xi_i)\right)^{\frac{1}{3}}}, \tag{23}$$

where

$$\Xi_i = \frac{\ln\frac{\varepsilon H_i}{E[h_i^a(a(0))]}}{\theta_i^* K_i^s}.$$

Proof: Please refer to Appendix B. ∎

Substituting (23) into (21), the maximum computation capacity X_{CVF} of the available fog nodes is calculated as

$$\begin{aligned} X_{\mathrm{CVF}} &= \sum_{i=1}^{N} X_i^{\max} \\ &= \frac{1}{L}\sum_{i=1}^{N}\frac{(T+\Xi_i)\kappa_i^{-\frac{1}{2}}E_{\max}^{\frac{1}{3}}}{\left(\sum_{i=1}^{N}\kappa_i^{-\frac{1}{2}}(T+\Xi_i)\right)^{\frac{1}{3}}} \\ &= \frac{E_{\max}^{\frac{1}{3}}}{L}\sum_{i=1}^{N}\frac{\kappa_i^{-\frac{1}{2}}(T+\Xi_i)}{\left(\sum_{i=1}^{N}\kappa_i^{-\frac{1}{2}}(T+\Xi)\right)^{\frac{1}{3}}} \\ &= \frac{1}{L}E_{\max}^{\frac{1}{3}}\left(\sum_{i=1}^{N}\kappa_i^{-\frac{1}{2}}(T+\Xi_i)\right)^{\frac{2}{3}}. \end{aligned} \tag{24}$$

Observing (24), it can be seen that only all the nearby fog nodes, i.e., N nodes, are involved in computation, the maximum computation capacity can be achieved. Moreover, the maximum computation capacity in (24) is irrelevant to the basic computation capability f_o and the tuning parameter γ_i.

B. Energy-Aware Task Allocation

When the generated computation task is larger than the maximum computation capacity shown in (24), i.e., the summation of all the available computation resources, it can be concluded that this task cannot be finished within the required time T by satisfying the reliability ε. When the computation task X is smaller than the maximum computation capacity, by observing (24), the computation task can be accommodated while reducing the overall energy consumption $E_{\max}$, or suspending some fog nodes. There are multiple criteria in choosing the suspended nodes. In this subsection, we focus on discussing two different schemes.

1) Scheme 1: Energy Efficiency-Aware: Since the computation task is smaller than the maximum computation capacity, the requester $\mathcal{R}$ needs to determine which nodes can be employed. In Scheme 1, after determining the optimal $\gamma_i, i = 1,\ldots,N$ in (23) and the maximum computation capacity in (24), the optimal nodes determination problem is then constructed as follows,

$$\max_{\mathcal{I}} \frac{X}{\sum_{j\in\mathcal{I}} L X_j^{\max}\kappa_j(\gamma_j f_o)^2}, \tag{25}$$

$$\text{s.t.,} \quad \frac{1}{L}E_{\max}^{\frac{1}{3}}\left(\sum_{j\in\mathcal{I}}\kappa_j^{-\frac{1}{2}}(T+\Xi_j)\right)^{\frac{2}{3}} \geq X. \tag{26}$$

The objective function in (25) is to maximize the energy efficiency in choosing the optimal node set $\mathcal{I}$, while the constraint is to guarantee that the chosen set $\mathcal{I}$ can accommodate the computation task X. The optimal solution is therefore provided in Theorem 3.

Theorem 3: The optimal set $\mathcal{I}$ in Scheme 1 is determined by

$$\mathcal{I}^* \triangleq \arg\min\left\{\mathcal{I}\middle|\sum_{j\in\mathcal{I}}\kappa_j^{-\frac{1}{2}}(T+\Xi_j) \geq \left(\frac{X}{\frac{1}{L}E_{\max}^{\frac{1}{3}}}\right)^{\frac{3}{2}}\right\}. \tag{27}$$

Proof: Please refer to Appendix C. ∎

In Scheme 1, there may have multiple computation sets $\mathcal{I}$ that satisfy Theorem 3. Given an optimal computation node set $\mathcal{I}^*$, we get the optimal task allocation X_j as

$$X_j = \begin{cases} X_j^{\max}, & j \in \mathcal{I}^*, \\ 0, & j \notin \mathcal{I}^*. \end{cases} \tag{28}$$

2) Scheme 2: Energy Consumption-Aware: In Scheme 2, when the network maximum computation capacity is larger than the computation task X, the requester $\mathcal{R}$ needs to redetermine the optimal node set $\mathcal{I}$ at the very beginning, i.e., at the stage that determines the tuning parameter γ_j. The optimal γ_j for the chosen node can be rewritten as

$$\gamma_j = \frac{f_o^{-1}\kappa_j^{-\frac{1}{2}}E_{\max}^{\frac{1}{3}}}{\left(\sum_{j\in\mathcal{I}}\kappa_j^{-\frac{1}{2}}(T+\Xi_j)\right)^{\frac{1}{3}}}. \tag{29}$$

Observing (29) and (23), though the numerator in (29) is the same with the one in (23). However, the denominator in (23), i.e., $\sum_{i=1}^{N}\kappa_i^{-\frac{1}{2}}(T+\Xi_i)$, is the summation of N fog nodes, while the denominator in (29), i.e., $\sum_{j\in\mathcal{I}}\kappa_j^{-\frac{1}{2}}(T+\Xi_j)$, is the summation of the optimal node set $\mathcal{I}$. Moreover, the set $\mathcal{I}$ is a subset of the N fog nodes.

The energy consumption E_j on unit cycle is represented as

$$E_j = \frac{LX_j\kappa_j(\gamma_j f_o)^2}{X_j} = L\kappa_j(\gamma_j f_o)^2. \tag{30}$$

Substituting (29) into (30), we get

$$E_j = \frac{LE_{\max}^{\frac{2}{3}}}{\left(\sum_{j\in\mathcal{I}}\kappa_j^{-\frac{1}{2}}(T+\Xi_j)\right)^{\frac{2}{3}}}. \tag{31}$$

Observing (31), when the node set $\mathcal{I}$ is determined, E_j is actually irrelative to the index j. In order to make the notations accurate, we use $E_{\mathcal{I}}$ denote E_j in the following discussions. With the purpose to minimize the overall energy consumption of the

chosen fog nodes, the optimization problem can be constructed as

$$\min_{\mathcal{I}} |\mathcal{I}| E_{\mathcal{I}}, \tag{32}$$

$$\text{s.t.,} \quad \frac{1}{L} E_{\max}^{\frac{1}{3}} \left(\sum_{j \in \mathcal{I}} \kappa_j^{-\frac{1}{2}} \left(T + \Xi_j\right) \right)^{\frac{2}{3}} \geq X. \tag{33}$$

The corresponding solution is provided in Theorem 4.

Theorem 4: The optimal fog node set $\mathcal{I}^*$ in Scheme 2 is determined by

$$\mathcal{I}^* \triangleq \quad \arg\min \left\{ |\mathcal{I}| \Big| \sum_{j \in \mathcal{I}} \kappa_j^{-\frac{1}{2}} \left(T + \Xi_j\right) \geq \left(\frac{X}{\frac{1}{L} E_{\max}^{\frac{1}{3}}} \right)^{\frac{3}{2}} \right\}. \tag{34}$$

Proof: Please refer to Appendix D. ■

Theorem 4 indicates that the optimal computation set $\mathcal{I}^*$ should meet the following two conditions:

1) The chosen nodes can accommodate the computation task X.
2) Among all the candidates, choose the set with the minimal size. If the candidates have the same sizes, choose the one with a larger value of $\sum_{j \in \mathcal{I}} \kappa_j^{-\frac{1}{2}} (T + \Xi_j)$.

In order to make the following discussions clear, we define

$$\delta_i \triangleq \kappa_i^{\frac{1}{2}} (T + \Xi_i)^{-1}. \tag{35}$$

In (35), κ_i is a parameter that determines the energy consumption in $\mathcal{V}_i$. A larger κ_i will induce a larger energy consumption in processing the same computation cycles. $T + \Xi_i$ represents the remaining computation time after being successfully transmitted through wireless channel. Since the total time is T, it is easy to figure out that a larger $T + \Xi_i$ associates with a smaller queueing delay. In other words, from the perspective of the requester $\mathcal{R}$, a computation node with a smaller δ_i, i.e., a product of $\kappa_i^{\frac{1}{2}}$ and $(T + \Xi_i)^{-1}$, is more preferable than the nodes with larger $\delta_j, j \neq i$.

For $\{\delta_1, \ldots, \delta_n, \ldots, \delta_N\}$, let $\nu(\cdot)$ denote a permutation of δ_i, such that $\delta_{\nu(1)} \leq \delta_{\nu(2)} \leq \ldots \leq \delta_{\nu(N)}$. The computation node set $\mathcal{I}^*$ can be further determined by the following proposition.

Proposition 1: The fog node set $\mathcal{I}^*$ derived in Theorem 4 can be rewritten as

$$\mathcal{I}^* = \arg\min \left\{ M \Big| \sum_{j=\nu(1)}^{\nu(M)} \delta_j^{-1} \geq \left(\frac{X}{\frac{1}{L} E_{\max}^{\frac{1}{3}}} \right)^{\frac{3}{2}} \right\}. \tag{36}$$

Proof: Please refer to Appendix E. ■

Proposition 1 indicates that the computation set $\mathcal{I}^*$ excludes the nodes with larger δ_i. In this way, the network performance can be enhanced by suspending the nodes with larger δ_i. The cutoff threshold δ_o is therefore represented as,

$$\delta_o \triangleq \kappa_{\nu(M)}^{\frac{1}{2}} \left(T + \Xi_{\nu(M)}\right)^{-1}. \tag{37}$$

The solution γ_j^* and allocated task $X_j^{\max}$ are given by

$$\gamma_j^* = \begin{cases} \dfrac{f_o^{-1} \kappa_j^{-\frac{1}{2}} E_{\max}^{\frac{1}{3}}}{\left(\sum_{j \in \mathcal{I}^*} \delta_j^{-1} \right)^{\frac{1}{3}}}, & \delta_j \leq \delta_o, \\ 0, & \delta_j > \delta_o, \end{cases} \tag{38}$$

and

$$X_j^{\max} = \begin{cases} \dfrac{E_{\max}^{\frac{1}{3}} \delta_j^{-1}}{L (\sum_{j \in \mathcal{I}^*} \delta_j^{-1})^{\frac{1}{3}}}, & \delta_j \leq \delta_o, \\ 0, & \delta_j > \delta_o, \end{cases} \tag{39}$$

respectively.

The solution of Scheme 2 employs a truncated-channel-inversion like policy. It can be seen that when the inherent parameter δ_j is larger than a cutoff threshold δ_o, the nodes will be suspended to improve the network performance.

C. Some Discussions

When there is a data burst from the applications, the data will first be pushed into a queue. Then, the scheduling policy will help determine when to allocate the data based on their priorities. In this paper, FIFO is utilized as a scheduling policy. Based on which, the delay violation probability can be derived. Then, the proposed two schemes will be utilized to give the optimal energy-aware task allocation results. That is to say, the two schemes will be triggered after the data burst is pushed into a queue.

Given β is fixed, changing the value of α which is the transition probability from state π_α^1 to π_α^0, is equivalent to changing the data generation model. When α decreases, the steady probability of state 0 grows and the steady probability of state 1 drops. That means there are less data generated on average. Based on some preliminary simulations, it can be inferred that as data generation model changes, i,e, α decreases or less data generated on average, less fog nodes may be involved.

The number of fog nodes involved is jointed affected by many factors, such as the data generation model, channel status, total computation amount, computation capacity of each node, as well as the chosen allocation scheme. In this paper, we assume that all the other factors are fixed, and then we investigate the involved fog nodes under different allocation criterion. Explicit analyse on the relationship between the involved fog nodes and data generation model, channel status, as well as other factors will be thoroughly investigated in our future works.

IV. Numerical Results

In this section, we conduct numerical simulations to verify effectiveness of the proposed task allocation schemes. The FIFO scheduling policy is employed in this paper. Rayleigh fading channel is assumed, and the small-scale channel gain g_i obeys the exponential distribution with parameter 1. The general simulation parameters are listed in Table I. We will provide the specific simulation parameters when the scenario is changed.

TABLE I
SIMULATION PARAMETERS

Parameter	Value
Bandwidth	$B = 75$MHz [44]
Transmit Power	$P_{\text{tr}} = 33$dBm [45]
Distance	$d_i = [2,\ 90]$m [46]
Pathloss Exponent	$l = 4$
Noise Density	$\sigma_o^2 = -174$dBm
Fog Node Number	$N = 6$
Basic Computation Capacity	$f_o = 1$GHz
Hardware-related Parameter	$\kappa_i = [10^{-11},\ 9 * 10^{-11}]$ [47]
Computation Deadline	$T = 0.1\mu$s
Reliability	$\varepsilon = 10^{-9}$
computation intensity	$L = 1$
Channel Cutoff Threshold	$g_o = [0.05, 0.1, 0.4, 0.7]$

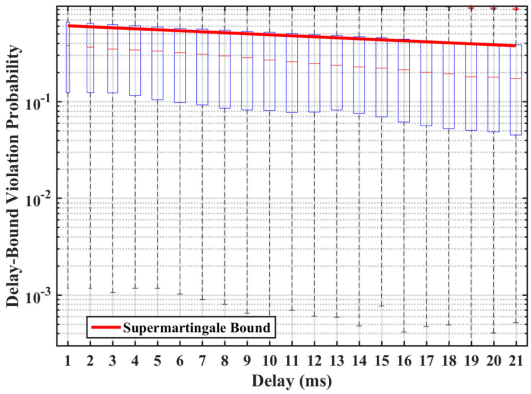


Fig. 2. Comparisons between the supermartingale delay bound and real simulation results under the FIFO scheduling policy.

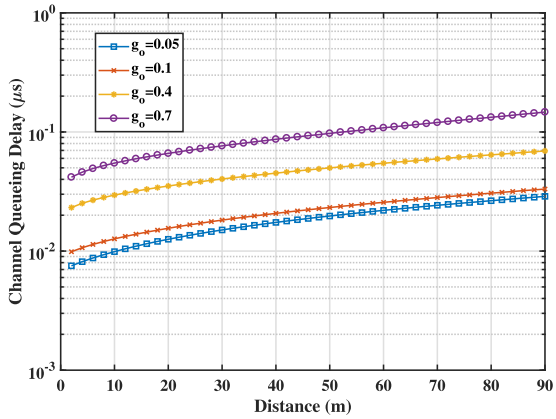


Fig. 3. Channel queueing time of nodes with increasing distances from requester $\mathcal{R}$, when channel cutoff threshold $g_o = 0.05, 0.1, 0.4,$ and0.7.

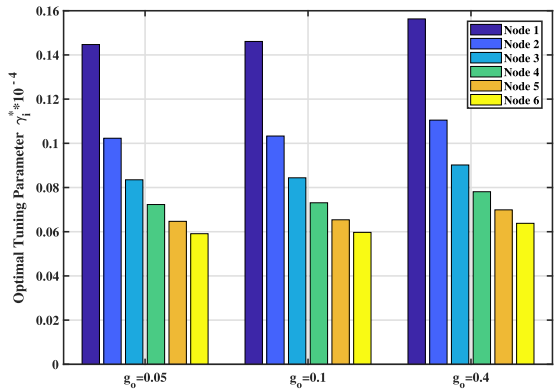


Fig. 4. Optimal tuning parameter γ_i^*, when channel cutoff threshold $g_o = 0.05, 0.1,$ and 0.4.

A. *Theoretical Supermartingale Delay Bound Versus Simulation Results*

At first, we conduct a simplified simulation and run 5000 realizations to calculate the delay-bound violation probability. We assume that $\alpha = 0.3$, $\beta = 0.4$, and data generation rate equals to 6 Mega bits per second (Mbps). Service rate is 5 Mbps. The channel cutoff threshold is assumed to be 0.6242 and the corresponding successful transmission probability is $P_{\text{suc}} = 0.5357$.

To verify the tightness of the derived bound, Fig. 2 compares the supermartingale delay-bound violation probability with the real simulation results shown as the box plot. It is obvious that as delay time increases, the delay-bound violation probability decreases. Also, it can be verified in Fig. 2 that the delay bound derived from the martingale theory is very tight to the real simulation results. Thus, it can be concluded that the supermartingale delay bound is feasible to conduct accurate delay performance analysis, especially in the URLLC scenarios.

B. *Optimal γ_i and Maximum Computation Capacity*

We assume $\alpha = 0.1$, $\beta = 0.9$, $R = 3.1 * 10^9$ bps, $T = 0.1\ \mu$s, and reliability $\varepsilon = 10^{-9}$. To evaluate the channel queueing delay time of nodes in different distances, Fig. 3 plots the delay curves when the channel cutoff threshold varies. It is shown that the channel queueing time gradually grows as the distance increases. This is due to the fact that when the distance increases, the offloading rate, i.e., C_i in (5) decreases. As the channel cutoff threshold g_o grows from 0.05 to 0.7, given the node's distance, the channel queueing delay time increases dramatically. Also it is obvious that all the channel queueing time are below $0.1\ \mu$s when $g_o = 0.05, 0.1,$ and 0.4. Fig. 3 reveals the fact that the distance d_i, which determines the offloading rate in (5), and channel cutoff threshold g_o, which determines the successful transmission probability in (9), makes a profound impact on the channel queueing delay performance.

Given the required time $T = 0.1\ \mu$s which includes the channel queueing time and the computation time, as seen in Fig. 3, when the cutoff channel threshold is smaller than 0.4, the channel queueing delay time may meet the deadline requirement. We assume 6 vehicular fog nodes with $d_i = 4, 6, 8, 10, 12,$ and14 m away from the requester, the same hardware architecture, i.e., $\kappa = 10^{-11}$, and the maximum energy $E_{\max} = 10$ w. Fig. 4 and Fig. 5 illustrate the bar plots of the optimal tuning parameter γ_i^* and the maximum computation capacity $X_i^{\max}$ of each node. Observing the two figures, it can be seen that the nodes which are closer to the

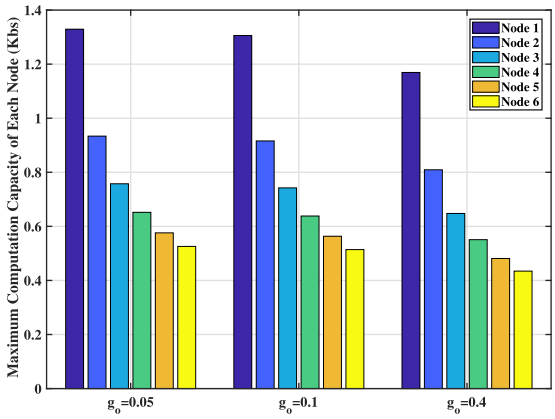


Fig. 5. Maximum computation capacity of each node, when channel cutoff threshold $g_o = 0.05, 0.1, \text{and } 0.4$.

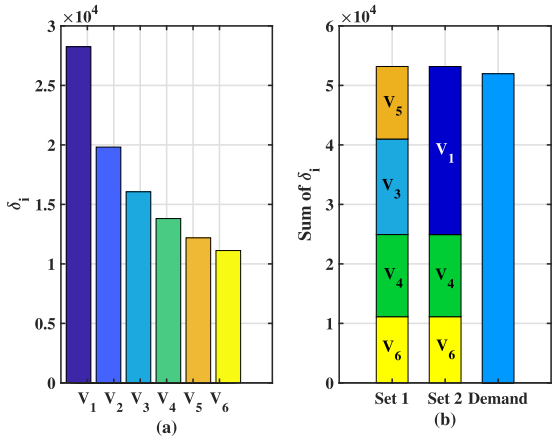


Fig. 6. δ_i^{-1} of each vehicular node, and the comparisons between the summation of $\{\mathcal{V}_3, \mathcal{V}_4, \mathcal{V}_5, \mathcal{V}_6\}$ and $\{\mathcal{V}_1, \mathcal{V}_4, \mathcal{V}_6\}$, given $g_o = 0.1$.

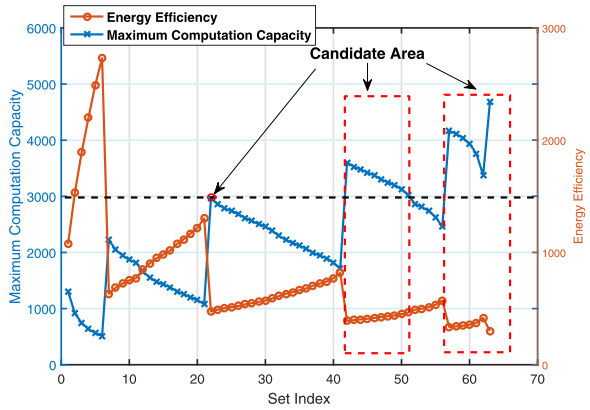


Fig. 7. Maximum computation capacity and energy efficiency of all the set combinations among these 6 fog nodes, given $g_o = 0.1$.

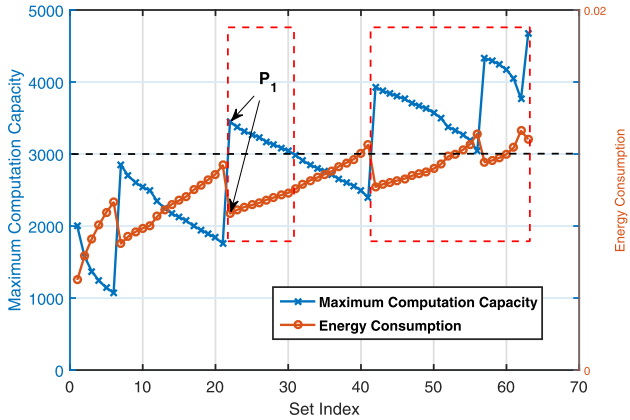


Fig. 8. Maximum computation capacity and energy consumption of all the set combinations among these 6 fog nodes, given $g_o = 0.1$.

requester are allocated with higher tuning parameters γ_i, and they have larger computation capacities.

In Fig. 4, when the channel cutoff threshold g_o increases, the optimal tuning parameter of one vehicular node also increases. That is due to the reason that, when g_o increases, according to (9), the successful transmission probability decreases. This will make the channel queueing delay time increase, and then make the remaining computation time, i.e., $T + \Xi_i$ decrease. According to (23), a smaller denominator leads to a larger γ_i^*.

In Fig. 5, it is observed that when the channel cutoff threshold g_o increases, the maximum computation capacity of one vehicular node decreases. This can be verified by simply substituting (23) into (21).

Moreover, the variation of channel cutoff threshold can be caused by the channel status variation. As shown in Fig. 5, when channel cutoff threshold g_o grows, the maximum computation capacity of each node decreases. In this case, in order to complete the computation task, more nodes may be involved.

C. Performance of the Energy Efficiency-Aware Scheme

In this subsection, we follow the same simulation setup in Section IV-B, i.e., there are 6 vehicular fog nodes and the channel cut off threshold is set to $g_o = 0.1$. $\kappa_i = [1, 2, 3, 4, 5.062, \quad 6] * 10^{-11}$. The computation task which needs to be offloaded is $X = 3000$ bits. By substituting X into (27), we can get the minimum demanding value of $\sum_{i \in \mathcal{I}} \delta_i^{-1} = 5.1962 * 10^4$.

To verify the conclusion drawn in Theorem 3, Fig. 6(a) plots δ_i^{-1} of each vehicular node. The δ_i^{-1} value of each node is $\{2.8249, 1.9817, 1.6060, 1.3808, 1.2189, 1.1121\} * 10^4$. It can be verified that set 1 with nodes $\{\mathcal{V}_3, \mathcal{V}_4, \mathcal{V}_5, \mathcal{V}_6\}$ and set 2 with nodes $\{\mathcal{V}_1, \mathcal{V}_4, \mathcal{V}_6\}$ have the same summations, i.e., $\delta_3^{-1} + \delta_4^{-1} + \delta_5^{-1} + \delta_6^{-1} = \delta_1^{-1} + \delta_4^{-1} + \delta_6^{-1} = 5.3178 * 10^4$. Fig. 6(b) makes a comparison between the summation of two node sets and the minimum demanding summation value. It can be seen that set 1 and set 2 are all larger than the demanding value. Thereby, the two sets can be chosen as candidates. Moreover, by substituting the summation of δ_i^{-1} of the two sets into (25), it can be verified that the two sets have the same energy efficiency with a value of 571.1569. Thus, it can be verified that the optimal solution in maximizing the energy efficiency is not unique.

In this simulation, this carefully designed parameter κ_i can be used to verify Theorem 3. We give the explicit values here to help readers calculate the minimum demanding value and the summations of δ_i^{-1} in different sets. It can

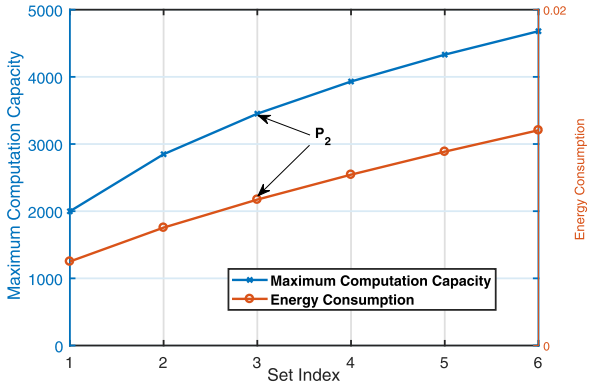


Fig. 9. Maximum computation capacity and energy consumption of the proposed scheme, given $g_o = 0.1$.

be noted that there are two different sets with the same summation value, and further Theorem 3 can be validated.

Fig. 7 plots the maximum computation capacity and the corresponding energy efficiency of all the set combinations among these 6 vehicular fog nodes. The curve with crosses denotes the maximum computation capacity, and the corresponding Y-axis is on the left hand side of Fig. 7. The curve with circles represents the energy efficiency, and the corresponding Y-axis is on the right hand side of Fig. 7. There are totally 63 sets, the dotted box highlights the candidate area, within which, the summation of computation capacity is larger than the demanding capacity $X = 3000$ bits, the set with the maximum energy efficiency can be found correspondingly.

D. Performance of the Energy Consumption-Aware Scheme

In this subsection, we follow the same simulation setup in Section IV-C. To demonstrate the conclusions drawn in Theorem 4 and Proposition 1, Figs. 8 and 9 depict the energy consumption under different node set combinations.

Fig. 8 plots the maximum computation capacity and energy consumption of all the set combinations among 6 vehicular nodes. Note that in this scheme, the maximum computation capacity is not the same as that in Fig. 7. This is due to the reason that, in this scheme, the optimal γ_i^* is determined in (29) which is associated to the chosen set $\mathcal{I}$. However, in Fig. 7, γ_i^* is determined by (23) which is associated with all the fog nodes. Therefore, they have different γ_i^*, and further, they have different $X_i^{\max}$ according to (21). The dotted box highlights the candidate area. The set with the minimum energy consumption, i.e., P_1, is the optimal solution.

Fig. 9 plots the maximum computation capacity and the corresponding energy consumption followed by the rule in Proposition 1. In specific, we sort the nodes with an increasing order of δ_i, and then compare the summation of δ_i^{-1} to the threshold shown in (36). The computation capacity grows as the node number increases, and the energy consumption also increases. Since the demanding computation $X = 3000$, the first 3 nodes can meet the computation requirement. At the same time, they can provide the minimum energy consumption.

Moreover, comparing Figs. 8 and 9, P_1 in Fig. 8 has the same value with P_2 in Fig. 9. Observing Fig. 8, P_1 is the optimal solution among all the set combinations, and P_2 is the optimal solution using our proposed scheme. Thus, the optimal solution can be found by the proposed scheme which is easy to be implemented.

V. Conclusion

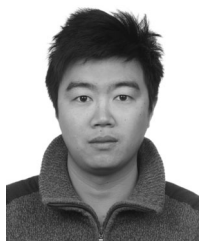
In this paper, we provide a new idea on task allocation in vehicular fog networks. In order to realize the ITSs, queueing delay, especially in the scenario requiring URLLC, is a non-negligible issue. Due to its stochastic characteristics, queueing delay time cannot be preciously calculated. The martingale theory is utilized to analyze the queueing delay performance instead. Thereby, we conduct researches on task allocation problem in a vehicular fog computation offloading scenario based on the martingale-theory-derived delay bound. The requester needs to determine whether the demanding computations can be satisfied by the available computation resources, and which nodes should be chosen to further optimize the energy-aware performance. The optimal solution to maximize the energy efficiency is not unique. Meanwhile, the optimal solution to minimize the overall energy consumption is unique, and can be constructed as a truncated-channel-inversion like policy. Simulation results are provided to demonstrate effectiveness of the proposed schemes.

References

[1] G. Dimitrakopoulos and P. Demestichas, "Intelligent transportation systems," *IEEE Veh. Technol. Mag.*, vol. 5, no. 1, pp. 77–84, Mar. 2010.
[2] 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.
[3] M. Shafi *et al.*, "5G: A tutorial overview of standards, trials, challenges, deployment, and practice," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 6, pp. 1201–1221, Jun. 2017.
[4] M. I. Ashraf, C.-F. Liu, M. Bennis, and W. Saad, "Towards low-latency and ultra-reliable vehicle-to-vehicle communication," in *Proc. Eur. Conf. Netw. Commun.*, Jun. 2017, pp. 1–5.
[5] W. Md, S. Mehdi, G. Abdullah, and B. Rajkumar, "A survey on vehicular cloud computing," *J. Netw. Comput. Appl.*, vol. 40, pp. 325–344, 2014.
[6] 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.
[7] M. Sookhak *et al.*, "Fog vehicular computing: Augmentation of fog computing using vehicular cloud computing," *IEEE Veh. Technol. Mag.*, vol. 12, no. 3, pp. 55–64, Sep. 2017.
[8] J. Wang, T. Liu, K. Liu, B. Kim, L. J. Xie, and Z. Han, "Computation offloading over fog and cloud using multi-dimensional multiple knapsack problem," in *Proc. IEEE Global Commun. Conf.: Mobile Wireless Netw.*, Abu Dhabi, United Arab Emirates, Dec. 2018, pp. 1–7.
[9] L. Liu, Z. Chang, and X. Guo, "Socially aware dynamic computation offloading scheme for fog computing system with energy harvesting devices," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1869–1879, Jun. 2018.
[10] 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.
[11] J. Liu, J. Wan, B. Zeng, Q. Wang, H. Song, and M. Qiu, "A scalable and quick-response software defined vehicular network assisted by mobile edge computing," *IEEE Commun. Mag.*, vol. 55, no. 7, pp. 94–100, Jul. 2017.
[12] J. Liu *et al.*, "High-efficiency urban traffic management in context-aware computing and 5G communication," *IEEE Commun. Mag.*, vol. 55, no. 1, pp. 34–40, Jan. 2017.

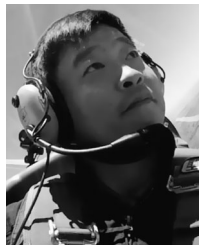
[13] R. Yu, J. Ding, X. Huang, M. T. Zhou, S. Gjessing, and Y. Zhang, "Optimal resource sharing in 5G-enabled vehicular networks: A matrix game approach," *IEEE Trans. Veh. Technol.*, vol. 65, no. 10, pp. 7844–7856, Oct. 2016.
[14] M. Shojafar, N. Cordeschi, and E. Baccarelli, "Energy-efficient adaptive resource management for real-time vehicular cloud services," *IEEE Trans. Cloud Comput.*, vol. 7, no. 1, pp. 196–209, Jan. 2019.
[15] U. F. C. Commission *et al.*, "FCC report and order 03-324: Amendment of the commission's rules regarding dedicated short–range communication services in the 5.850-5.925 GHz band," 2003.
[16] L. Yang, B. Liu, J. Cao, Y. Sahni, and Z. Wang, "Joint computation partitioning and resource allocation for latency sensitive applications in mobile edge clouds," in *Proc. IEEE 10th Int. Conf. Cloud Comput.*, 2017, pp. 246–253.
[17] Y. Sahni, J. Cao, and L. Yang, "Data-aware task allocation for achieving low latency in collaborative edge computing," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3512–3524, Apr. 2019.
[18] J. P. Champati and B. Liang, "Semi-online algorithms for computational task offloading with communication delay," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 4, pp. 1189–1201, Apr. 2017.
[19] F. Gao, S. E. Li, Y. Zheng, and D. Kum, "Robust control of heterogeneous vehicular platoon with uncertain dynamics and communication delay," *IET Intell. Transp. Syst.*, vol. 10, no. 7, pp. 503–513, Sep. 2016.
[20] Y. Qi, H. Wang, L. Zhang, and B. Wang, "Optimal access mode selection and resource allocation for cellular-VANET heterogeneous networks," *IET Commun.*, vol. 11, no. 13, pp. 2012–2019, Aug. 2017.
[21] Z. Hou, C. She, Y. Li, T. Q. S. Quek, and B. Vucetic, "Burstiness-aware bandwidth reservation for ultra-reliable and low-latency communications in tactile internet," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 11, pp. 2401–2410, Nov. 2018.
[22] C. She, Z. Chen, C. Yang, T. Q. S. Quek, Y. Li, and B. Vucetic, "Improving network availability of ultra-reliable and low-latency communications with multi-connectivity," *IEEE Trans. Commun.*, vol. 66, no. 11, pp. 5482–5496, Nov. 2018.
[23] C. She, C. Yang, and T. Q. S. Quek, "Joint uplink and downlink resource configuration for ultra-reliable and low-latency communications," *IEEE Trans. Commun.*, vol. 66, no. 5, pp. 2266–2280, May 2018.
[24] C.-S. Chang and J. A. Thomas, "Effective bandwidth in high-speed digital networks," *IEEE J. Sel. Areas Commun.*, vol. 13, no. 6, pp. 1091–1100, Aug. 1995.
[25] D. Wu and R. Negi, "Effective capacity: A wireless link model for support of quality of service," *IEEE Trans. Wireless Commun.*, vol. 2, no. 4, pp. 630–643, Jul. 2003.
[26] C. Guo, L. Liang, and G. Y. Li, "Resource allocation for low-latency vehicular communications: An effective capacity perspective," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 4, pp. 905–917, Apr. 2019.
[27] G. L. Choudhury, D. M. Lucantoni, and W. Whitt, "Squeezing the most out of ATM," *IEEE Trans. Commun.*, vol. 44, no. 2, pp. 203–217, Dec. 1996.
[28] F. Poloczek and F. Ciucu, "Service-martingales: Theory and applications to the delay analysis of random access protocols," in *Proc. IEEE Conf. Comput. Commun.*, Hong Kong, China, Apr. 2015, pp. 945–953.
[29] Y. Hu, H. Li, Z. Chang, and Z. Han, "Scheduling strategy for multimedia heterogeneous high-speed train networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 4, pp. 3265–3279, Apr. 2017.
[30] Y. Hu, H. Li, Z. Chang, and Z. Han, "End-to-end backlog and delay bound analysis for multi-hop vehicular ad hoc networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 10, pp. 6808–6821, Oct. 2017.
[31] L. Zhao, X. Chi, and Y. Zhu, "Martingales-based energy-efficient D-ALOHA algorithms for MTC networks with delay-insensitive/ URLLC terminals co-existence," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1285–1298, Apr. 2018.
[32] T. Liu, J. Li, F. Shu, and Z. Han, "Quality-of-Service driven resource allocation based on martingale theory," in *Proc. IEEE Global Commun. Conf.: Commun. QoS, Rel. Model.*, Abu Dhabi, United Arab Emirates, Dec. 2018, pp. 1–6.
[33] A. M. Bozorgi, M. Farasat, and A. Mahmoud, "A time and energy efficient routing algorithm for electric vehicles based on historical driving data," *IEEE Trans. Intell. Vehicles*, vol. 2, no. 4, pp. 308–320, Dec. 2017.
[34] S. Zhang, Y. Luo, K. Li, and V. Li, "Real-time energy-efficient control for fully electric vehicles based on an explicit model predictive control method," *IEEE Trans. Veh. Technol.*, vol. 67, no. 6, pp. 4693–4701, Jun. 2018.
[35] L. Thibault, G. De Nunzio, and A. Sciarretta, "A unified approach for electric vehicles range maximization via eco-routing, eco-driving, and energy consumption prediction," *IEEE Trans. Intell. Vehicles*, vol. 3, no. 4, pp. 463–475, Dec. 2018.
[36] 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.
[37] S. Melendez and M. P. McGarry, "Computation offloading decisions for reducing completion time," in *Proc. 14th IEEE Annu. Consum. Commun. Netw. Conf.*, Jan. 2017, pp. 160–164.
[38] Q. Li, J. Zhao, Y. Gong, and Q. Zhang, "Energy-efficient computation offloading and resource allocation in fog computing for internet of everything," *China Commun.*, vol. 16, no. 3, pp. 32–41, Mar. 2019.
[39] F. Guo, H. Zhang, H. Ji, X. Li, and V. C. M. Leung, "An efficient computation offloading management scheme in the densely deployed small cell networks with mobile edge computing," *IEEE/ACM Trans. Netw.*, vol. 26, no. 6, pp. 2651–2664, Dec. 2018.
[40] J. S. Dagpunar, *Simulation and Monte Carlo: With Applications in Finance and MCMC*. New York, NY, USA: Wiley, 2008.
[41] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surv. Tut.*, vol. 19, no. 4, pp. 2322–2358, Oct.–Dec. 2017.
[42] J. Liu and Q. Zhang, "Offloading schemes in mobile edge computing for ultra-reliable low latency communications," *IEEE Access*, vol. 6, pp. 12 825–12 837, 2018.
[43] A. P. Miettinen, "Energy efficiency of mobile clients in cloud computing," in *Proc. Usenix Conf. Hot Topics Cloud Comput.*, 2010, p. 4.
[44] J. B. Kenney, "Dedicated short-range communications (DSRC) standards in the United States," *Proc. IEEE*, vol. 99, no. 7, pp. 1162–1182, Jul. 2011.
[45] K. Hong, D. Xing, V. Rai, and J. Kenney, "Characterization of DSRC performance as a function of transmit power," in *Proc. 6th ACM Int. Workshop Veh. InterNETw., China*, Sep. 2009, pp. 63–68.
[46] Y. L. Morgan, "Notes on DSRC & WAVE standards suite: Its architecture, design, and characteristics," *IEEE Commun. Surv. Tut.*, vol. 12, no. 4, pp. 504–518, Oct.–Dec. 2010.
[47] 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.

Tingting Liu (M'12) 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 Postdoctor in the Nanjing University of Science and Technology. From 2017 to 2018, she was a Visiting Scholar with the University of Houston, Houston, TX, USA. She is currently an Associate Professor with the Nanjing Institute of Technology. Her research interests include game theory, blockchain, caching-enabled systems, edge computing, network quality of service, device-to-device networks, and cognitive radio networks.



Jun Li (M'09–SM'16) received the Ph.D degree in electronic engineering from Shanghai Jiao Tong University, Shanghai, China, in 2009. From January 2009 to June 2009, he worked with the Department of Research and Innovation, Alcatel Lucent Shanghai Bell, Shanghai, as a Research Scientist. From June 2009 to April 2012, he was a Postdoctoral Fellow with the School of Electrical Engineering and Telecommunications, the University of New South Wales, Australia. From April 2012 to June 2015, he was a Research Fellow with the School of Electrical Engineering, University of Sydney, Australia. Since 2015, he has been a Professor with the School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing, China. His research interests include network information theory, channel coding theory, wireless network coding, and cooperative communications.

Feng Shu (M'09) received the Ph.D., M.S., and B.S. degrees from Southeast University, Nanjing, China, in 2002, XiDian University, Xian, China, in 1997, and Fuyang Teaching College, Fuyang, China, in 1994, respectively. From September 2009 to September 2010, he was a Visiting Postdoctor with the University of Texas at Dallas. In October 2005, he joined the School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing, China, where he is currently a Professor and supervisor of Ph. D. and graduate students. He is also with the Fujian Agriculture and Forestry University and awarded with Mingjiang Scholar Chair Professor in Fujian Province. He has authored or coauthored about 200 papers, of which more than 100 are in archival journals including more than 80 papers on IEEE Journals and more than 110 SCI-indexed papers. He holds 10 Chinese patents. His research interests include wireless networks, wireless location, and array signal processing.



Zhu Han (S'01–M'04–SM'09–F'14) received the B.S. degree in electronic engineering from Tsinghua University, Beijing, China, in 1997, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Maryland, College Park, MD, USA, in 1999 and 2003, respectively.

From 2000 to 2002, he was an R&D Engineer with JDSU, Germantown, MD, USA. From 2003 to 2006, he was a Research Associate with the University of Maryland, MD, USA. From 2006 to 2008, he was an Assistant Professor with Boise State University, ID, USA. He is currently a John and Rebecca Moores Professor with the Electrical and Computer Engineering Department as well as with the Computer Science Department, University of Houston, Houston, TX, USA. He is also a Chair professor with National Chiao Tung University, ROC, Hsinchu, Taiwan. His research interests include wireless resource allocation and management, wireless communications and networking, game theory, big data analysis, security, and smart grid. He was a recipient of the NSF Career Award in 2010, the Fred W. Ellersick Prize of the IEEE Communication Society in 2011, the EURASIP Best Paper Award for the *Journal on Advances in Signal Processing* in 2015, IEEE Leonard G. Abraham Prize in the field of Communications Systems (best paper award in IEEE JSAC) in 2016, and several best paper awards in IEEE conferences. He was an IEEE Communications Society Distinguished Lecturer from 2015-2018, AAAS fellow since 2019, and ACM distinguished Member since 2019. He is one of the highly cited researcher since 2017 according to Web of Science.