Reliability-optimal cooperative communication and computing in connected vehicle systems

Article (Accepted Version)

Reliability-Optimal Cooperative Communication and Computing in Connected Vehicle Systems

Jianshan Zhou, Daxin Tian, *Senior Member, IEEE,* Yunpeng Wang, Zhengguo Sheng, Xuting Duan, and Victor C.M. Leung, *Fellow, IEEE*

Abstract—The emergence of vehicular networking enables distributed cooperative computation among nearby vehicles and infrastructures to achieve various applications that may need to handle mass data by a short deadline. In this paper, we investigate the fundamental problems of a cooperative vehicle-infrastructure system (CVIS): how does vehicular communication and networking affect the benefit gained from cooperative computation in the CVIS and what should a reliability-optimal cooperation be? We develop an analytical framework of reliability-oriented cooperative computation optimization, considering the dynamics of vehicular communication and computation. To be specific, we propose stochastic modeling of V2V and V2I communications, incorporating effects of the vehicle mobility, channel contentions and fading, and theoretically derive the probability of successful data transmission. We also formulate and solve an execution time minimization model to obtain the success probability of application completion with the constrained computation capacity and application requirements. By combining these models, we develop constrained optimizations to maximize the coupled reliability of communication and computation by optimizing the data partitions among different cooperators. Numerical results confirm that vehicular applications with a short deadline and large processing data size can better benefit from the cooperative computation rather than non-cooperative solutions.

Index Terms—Cooperative vehicle-infrastructure systems (CVIS), vehicle-to-vehicle communication (V2V), vehicle-to-infrastructure communication (V2I), vehicular cloud computing, vehicular cooperation.

1 Introduction

RECENT advances in distributed data storage, mobile cloud computing and vehicular communication technologies have been tightly integrated into vehicles moving on roads and roadside infrastructures, which are empowering the design and implementation of *cooperative vehicle-infrastructure systems* (CVIS). In particular, along with the evolution of vehicular technologies, vehicles are envisioned to be furnished with powerful on-board storage, sensing, computing and communication facilities. They are able to provide cloud-computing-like services to other nearby connected mobile devices in their communication coverage, which has spawned a novel mobile edge-cloud computing paradigm, i.e., vehicular cloud computing [1]. In vehicular cloud computing, vehicles and roadside infrastructures, closely coupled via vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications [2], can form a small-scale cloudlet in a cooperative manner. In such a system, a vehicular user not only has an opportunity to access as well as offload its computation task to the roadside source-rich infrastructure (e.g., a road-side unit (RSU) or a central cloud) but also to access and use the underutilized storage and computing resources provided by one or several nearby vehicles. Thus, it can guarantee quality of service (QoS) for various vehicular applications. Such a computing paradigm based on V2V and V2I communications can be viewed as a computing-oriented cooperation between vehicles and between vehicles and infrastructures.

However, there exist some challenges that need to be addressed for practical and scalable realization of cooperative computing in vehicular mobility environments. On one hand, due to high mobility nature of vehicles, vehicular communication networks are quite different from traditional cellular networks and Wireless Local Area Networks (WLAN). To be specific, vehicular networks are generally characterized by transient topology between moving vehicles [3], fast channel fading and impairments resulting from vehicular mobility and Doppler shift [4], and intensive contending for channel access [5]. Hence, the inherent transient connectivity in vehicular communications makes it a challenge to reliably and efficiently exchange tremendous data (e.g., images, audio and video files, and some other multimedia streams) between vehicles and between vehicles and infrastructures. Consequently, the computation performance will be seriously constrained by the bottleneck of V2V and V2I communications in terms of reliability and efficiency performance. On the other hand, existing mobile cloud computing architectures, such as the virtual machine (VM) based cloudlets [6], the weblet model [7], the clone cloud model [8], and some other mobile edge computing paradigms [9], [10], [11], are conventionally designed for Internet-of-Things (IoT) sensors or mobile users with low mobility (e.g., smart phones, personal digital assistants and handled computers), which mainly rely on traditional cellular and WLAN networks for data transmission. Thus, these

- *J. Zhou, D. Tian, Y. Wang and X. Duan are with Beijing Advanced Innovation Center for Big Data and Brain Computing, Beijing Key Laboratory for Cooperative Vehicle Infrastructure Systems & Safety Control, School of Transportation Science and Engineering, Beihang University, Beijing 100191, China. E-mail: jianshanzhou@foxmail.com, dtian@buaa.edu.cn*
- *Z. Sheng is with Department of Engineering and Design, the University of Sussex, Richmond 3A09, UK. E-mail: z.sheng@sussex.ac.uk*
- *V. Leung is with Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, B.C., V6T 1Z4 Canada.E-mail: vleung@ece.ubc.ca*

existing mobile cloud computing or mobile edge computing technologies may not accommodate the fast-changing V2V and V2I communication topologies. Moreover, many existing computation offloading schemes for mobile cloud computing or mobile edge computing generally focus on the energy-efficiency optimization with a certain latency constraint in energy-poor or storage-poor mobile situations [12], [13]. Nonetheless, from the perspective of application demands for QoS support, reliability, low-latency and efficiency, rather than the energy consumption, are mainly concerned in the design and practical deployment of many vehicular applications, such as driving decision-making, road danger recognition, and platoon collision warning, etc. Indeed, despite the vast literature related to architecture design, offloading optimization and control in mobile cloud/edge computing [1], [14], [15], it remains an open question on how to leverage the mobile computing paradigm shift from traditional mobile networks (e.g., mobile wireless sensor networks, delay-tolerant networking, cellular networks, etc.) to vehicular networks and how to design an efficient and reliable computing mechanism for a CVIS, which jointly allows for vehicular mobility, communication and computation.

In this paper, our objectives are to address the aforementioned challenges related to vehicular communication and cooperative computation in CVIS, which involve two aspects in modeling and optimization. First, we present stochastic modeling for characterizing performance of V2V and V2I communications, which takes into account both the vehicle mobility and vehicular channel characteristics. Particularly, the stochastic transient V2V connectivity is considered from the physical-layer perspective and its impact on V2V connection is incorporated into the cooperative computation. Then, we integrate the dynamics of V2V and V2I communications and computation. We propose optimization models to maximize the success probability of application completion under different cooperative computation modes. The vehicular user is enabled to optimally partition, offload and execute application workload between vehicles and infrastructure with jointly considering V2V and V2I communication and computation.

The main contributions of our work are as follows:

- We present stochastic modeling for V2V and V2I communication dynamics. In V2V and V2I communications, the stochastic modeling combines the vehicle mobility with the effects from physical-layer contentions and fading. An analytical model for the performance estimation of a V2V link is developed, which can be used to evaluate the success probability of transferring data via the V2V connection within a deadline. Similarly, we also theoretically derive the success probability of V2I data transmission. Besides, an optimal V2I transmission scheduling is proposed to enhance the performance.
- We develop an optimization model to minimize the time consumption in a local application execution. Based on the model, we derive the closed-form optimal CPU clock frequencies. By using the optimal CPU clock frequencies, we obtain the maximum likelihood of successful completion using the local execution, given the available computation energy of a vehicle and the application requirements (i.e., the execution deadline and the data volume to be processed). The success probability of application completion by the local execution is defined as the computation reliability.
- Considering different cooperation modes, we formulate constrained optimization problems by combining the reliability modeling of both V2V and V2I communications and the local execution. The goal is to maximize the overall reliability of vehicular cooperative computing by optimizing the data workload partitions among V2V and V2I cooperators. The constrained optimization model developed can lead to reliability-optimal cooperation solution.

To the best of our knowledge, there exist little literature in the growing field of mobile cloud or mobile edge computing that incorporates the impacts of the vehicle mobility and of the physical-layer characteristics in wireless vehicular environments into mobile computation. We present the first step towards the modeling and optimization of reliability-oriented cooperative communication and computing in wireless vehicular accessing environments, which is expected to help bringing distributed vehicular computing power into full play in CVIS and to facilitate a transformation from traditional centralized cloud-based intelligence to distributed cooperative edge-based intelligence for connected and autonomous vehicles in the future.

The remainder of the paper is organized as follows. In Section 2, we present a review on related work. Section 3 presents the system model for a CVIS in terms of V2V and V2I communications and computation. In Section 4, we propose optimization models for cooperative computation and analyze different cooperation modes. Simulation results are supplemented in Section 5. Finally, Section 6 concludes the paper and gives future directions.

2 Related Work

The last few years have witnessed the integration of cloud-computing paradigms into the mobile systems, which promotes the development of mobile computing or mobile edge computing [14]. The key idea behind such computing paradigms is to migrate the data storage and computing tasks away from mobile users to remote central servers or local edge clouds, so as to overcome mobile devices' weakness in computing, data storage, energy capacities. Many researchers have been currently engaged in designing innovative architectures to leverage the potential of cloud computing in mobile scenarios. For instance, [6] has proposed the technological concept of a cloudlet to make the best of computing resources in local devices. [7] has presented an elastic application model that enables partitioning an application into multiple weblets and realizes dynamic configuration for the weblets. In [8], a clone-cloud execution mechanism has also been presented to augment mobile applications running on smartphones. In the context of IoT, [9] has proposed a hierarchical edge computing architecture, called EdgeIoT, to overcome the scalability issue in centralized cloud computing, in which a software defined network (SDN) cellular core is designed for information exchanging between edge nodes and a proxy VM assigned for each IoT device is running on a edge node. A computing paradigm, called Edge Mesh, has also been proposed in [10], in which decision-making tasks and data are enabled to be distributed to as well as shared among different edge devices in a network. In addition, [11] compares distributed edge/cloudlet architectures with traditional cloud architectures, revealing the advantages of distributed edge/cloudlet computing in terms of system scalability, network traffic balance and low-latency support.

Offloading decision-making plays a critical role in fully exploiting the potential of cloud computing, which has recently attracted many research efforts. Much literature focuses on the energy-efficiency optimization. [16] has proposed an adaptive mechanism to determine when to offload the computation from a client to a server while saving the energy consumption. [17] has applied the cloud offloading to mobile GPS sensing in terms of energy efficiency. A dynamic energy-efficient offloading algorithm has also been designed based on Lyapunov optimization in [18]. Both [19] and [20] aim to determine an optimal scheduling of CPU clock frequency for mobile execution and an optimal data partition for cloud execution with the goal of minimizing total computing energy cost within an execution duration. In [21], an energy-efficient cooperative computing paradigm has been proposed for mobile wireless sensor networks. In [12], the authors show that offloading computing tasks involving intensive communications may not benefit mobile users in terms of energy conservation. Multiple user computation offloading issue has also been studied in [22], which is based on a 3G/4G macro/small-cellular communication model. Many other energy-optimal offloading algorithms and architectures can also be found in [13], [15]. Nevertheless, for vehicular application situations, energy consumption is not a key design factor due to the fact that a vehicular system usually has a powerful energy supply. Instead, reliability and efficiency are major concerns in many computation-intensive and latency-critical vehicular applications. Additionally, most of computation offloading architectures presented in the aforementioned literature are based on cellular networks, the wireless channel dynamics of which is quite different from that of vehicular networks.

Indeed, the paradigm of mobile cloud or mobile edge computing has recently been extended to connected vehicles, which promotes the evolution of vehicular cloud computing [1]. In this evolution, vehicles can be viewed as IoT resources in terms of communication and computing [23]. To leverage the potential of computing resources distributed over connected vehicles, different vehicular cloud computing architectures have been investigated, such as the mobile vehicular cloudlet [24], the cooperative fog computing [25], the fog vehicular computing [26], and the vehicular fog computing [27], etc, and many offloading mechanisms have been proposed as well. [28] has proposed a Markov decision process (MDP)-based offloading algorithm, in which the stochastic connectivity between mobile users and cloudlets is modeled by an independent homogeneous poisson point process. A heuristic mechanism for dynamic computation offloading has also been designed with consideration of on-board CPU cycle and network bandwidth costs, and applications' latency in [29]. In [30], a cooperative download scheme is proposed based on selection of carriers and data chunks at roadside access points for delay-tolerant vehicular applications. [31] has developed computation offloading strategies for mobile edge computing in vehicular networks in order to reduce the latency and transmission cost. Considering the effectiveness of V2V and V2I transmissions, [32] has proposed a dynamic relegation scheme, with which computation offloading can be adaptively performed by combining the strategies of direct uploading and predictive relaying. In [33], the management of computing resources distributed over connected vehicles has been investigated and a MDP-based task replication policy is derived.

Even thought all of the current studies above confirm that putting the underutilized computing resources distributed over connected vehicles can offer significant opportunity and value for our society, existing architectures and offloading mechanisms for vehicular cloud computing have not integrated detailed modeling of vehicular mobility and V2V/V2I transmission capability and stochastic connectivity into the computing paradigm. The interaction between vehicular communication and computing still needs to be fully modeled in terms of vehicle-infrastructure cooperation, and the coupled impacts arising from communication and computing should be incorporated into the optimization formulation of the overall system, which is the main focus of our work.

3 System Model

In this section, we present CVIS and characterize both V2V and V2I communications and computation.

3.1 System Description

In a CVIS scenario, we consider that each vehicle is well furnished with certain sensing, data storage and computing facilities, as well as radio interfaces for V2V and V2I communications. Thus, they can also be viewed as mobile sensing and computing nodes. Additionally, we also consider the presence of a source-rich infrastructure, e.g., a cloudlet or a central cloud, which can provide much more powerful computing capacity than any single vehicle. By using V2V and V2I communications, each vehicle is allowed to transfer content data (e.g., application files) to a nearby vehicle and the infrastructure at the same time [34], which establishes a distributed computing network consisting of both moving vehicles and infrastructure. Specifically, as illustrated in Fig. 1, a host vehicle can be served by a remote cloudlet, which may be far away from the vehicle, and/or by a neighboring vehicle in the V2V communication coverage, which is ready to share its underutilized computing resources to help performing computation-intensive and latency-critical tasks.

3.2 Application model

A vehicle equipped with wireless communication and with on-board storage, sensing and computing power is allowed to process a data-driven application. A vehicular application is usually a program (a task) that performs a computation on a certain amount of content data to output a result and the result is collected by the remote data center for further analysis, such as on-line on-board diagnosis, data-driven

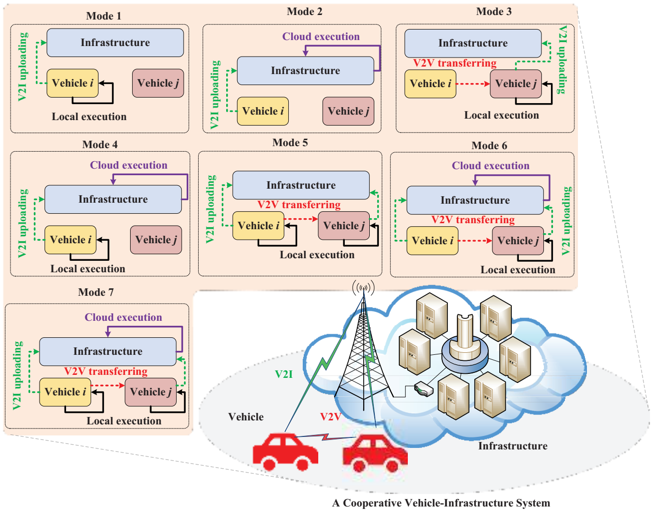


Fig. 1. A cooperative vehicle-infrastructure system where a distributed mobile application can be executed in 7 modes with each corresponding to a combination of vehicle-based and infrastructure-based executions.

analysis of driving behavior and some other applications related to pattern recognition and data mining based on a large volume of vehicular history sensing data. Another vehicular application example is the trajectory pattern mining. Conventionally, in the field of trajectory pattern mining, massive raw motion data collected by an on-board Global Position System (GPS) or by other vehicular sensors can be either pre-processed on the vehicular device locally (e.g., data reduction and filtering, adaptive nonlinear noise cancellation, etc.) or simply transferred to the cloud infrastructure (i.e., data center) for data processing remotely. In the former case, the vehicular node can compute the refined trajectory and then uploads the refined result to the data center for further pattern mining, which can save the communication resource at the sacrifice of its own computational resource. On the contrary, in the latter case, the vehicle is allowed to directly transmit its raw trajectory data to the data center and the cloud server takes the responsibility to complete all the computing tasks, which may consume a large amount of the communication resource. However, in the CVIS, the vehicular node has an option to offload parts of the computation task to a cooperative vehicular node and roadside infrastructure nearby, which may potentially provide larger computing capacity, and thus more computing effectiveness.

According to existing literature [20], [21], we use a canonical model to capture the essential of a data-driven application, which abstracts the application into three components: $\boldsymbol{A}(D_{\text{in}}, T) = (D_{\text{in}}, \alpha D_{\text{in}}, T\Delta\tau)$, where (i) D_{in} represents the total size of the input data in bits for the vehicular application, which can be partitioned into different subfiles or data blocks of heterogeneous sizes and offloaded to a peer vehicle or a cloud infrastructure for parallel processing. (ii) αD_{in} represents the size of the corresponding output after processing the D_{in}-bit data. Generally, the output size depends on the input volume, which is much smaller than that of the raw input, especially in many big data-driven applications. Thus, we assume that the scalar factor α should be $0 < \alpha < 1$. The resulting data needs to be uploaded to the cloud infrastructure. (iii) $T\Delta\tau$ denotes the completion deadline imposed by the application, in which T is the maximum number of time slots allowed to execute the application. $\Delta\tau$ is the duration of a time slot. Throughout this paper, we also use k to denote the index of the discrete time with $k \in [0, T]$.

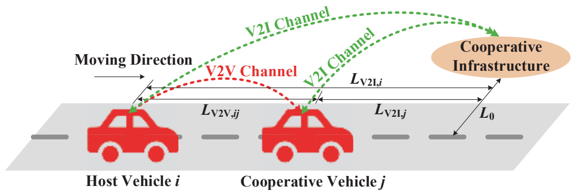


Fig. 2. Mobility scenario.

In addition, we assume that such an application model, similar to the MapReduce [35], supports distributed computing (which is indeed common in many mobile-sensing cloud-computing based situations [21]). That is, it can be divided into a series of sub-tasks or sub-jobs which can be executed by using partitions of the input data in a parallel and independent manner. Hence, the partitions of the input data need to be distributed by a host vehicular node to other cooperative nodes. It is noticed that even though there exist some situations where applications cannot be divided into parallel smaller computing tasks and need to be executed on a single node because of dependencies in the input data, such applications can be processed in batches. Namely, different batches of the applications can also be distributed to different computing nodes, so as to gain benefit from a distributed system.

3.3 Vehicular Mobility Formulation

Without loss of generality, we consider two vehicles connected via V2V communication moving on a straight roadway, where the host vehicle is indexed by i while the cooperative vehicle is labeled by j. Let s_l, v_l, a_l be the vehicle l's longitudinal position, speed and acceleration, respectively, where $l \in \{i, j\}$. As in most literature in the fields of transportation engineering, we consider that the kinematics parameters are constant in each time interval $[k\Delta\tau, (k+1)\Delta\tau)$, since the time-slot duration $\Delta\tau$ is sufficiently small. Thus, the discretized implementation of the vehicle mobility can be represented by a general double-integrator model as follows:

$$\begin{cases} s_l(k+1) = s(k) + \Delta\tau v_l(k) + \frac{(\Delta\tau)^2}{2} a_l(k); \\ v_l(k+1) = v_l(k) + \Delta\tau a_l(k), \end{cases} \tag{1}$$

where $l \in \{i, j\}$, and $k = 0, 1, \ldots, T$.

As shown in Fig. 2, denote by $L_{\text{V2V},ij}$ the initial inter-vehicle distance between vehicles i, j, while $L_{\text{V2I},i}$ and $L_{\text{V2I},j}$ are the initial longitudinal relative distances between i and the infrastructure and between j and the infrastructure, respectively. Besides, the vertical relative distance between the infrastructure and the roadway is denoted by L_0. Using these notations, we can easily describe the time-varying headway distance between i and j, $L_{i,j}(k)$, as well as the time-varying relative distance between i (or j) and the infrastructure, $L_i(k)$ (or $L_j(k)$), as follows

$$\begin{cases} L_{i,j}(k+1) = L_{\mathrm{V2V},ij} + \Delta s_{i,j}(k) + \Delta\tau\Delta v_{i,j}(k) \\ \qquad\qquad + \frac{(\Delta\tau)^2}{2}\Delta a_{i,j}(k); \\ L_l(k+1) = \sqrt{\left[L_{\mathrm{V2I},l} - s_l(k+1)\right]^2 + L_0^2}; l \in \{i,j\}, \end{cases} \tag{2}$$

where we let $\Delta s_{i,j}(k) = s_j(k) - s_i(k)$, $\Delta v_{i,j}(k) = v_j(k) - v_i(k)$, and $\Delta a_{i,j}(k) = a_j(k) - a_i(k)$.

3.4 Modeling of V2V Communication Reliability

3.4.1 V2V physical-layer transmission

For V2V communication, we consider the employment of the exclusive Dedicated Short-Range Communication (DSRC), which is operated on the frequency band between 5.850 to 5.925 GHz as allocated by U.S. FCC. According to the DSRC specification, the physical layer of the vehicular wireless access follows the IEEE 802.11p standard and adopts the orthogonal frequency division multiplexing (OFDM) protocol as in the IEEE 802.11a physical layer [36]. According to the field-test measurements in [37] and much existing literature such as [34], the fast fading in typical V2V channels can be captured by the Nakagami distribution. Hence, we denote by $m\left(d_{i,j}(k)\right)$ the fading parameter, which depends on the relative distance between the transmitter i and the receiver j, i.e., $d_{i,j}(k) = |L_{i,j}(k)|$, and by $\omega\left(d_{i,j}(k)\right)$ the average received power in the fading envelope, which can be further estimated by $\omega\left(d_{i,j}(k)\right) = \mathbb{E}\left[P\left(d_{i,j}(k)\right)\right]$. $P\left(d_{i,j}(k)\right)$ is the received signal strength, which is related to the relative distance $d_{i,j}(k)$ and can be well approximated by using a dual-slope piecewise-linear model [37]

$$P\left(d_{i,j}(k)\right) = \begin{cases} P\left(d_0\right) - 10\eta_1 \log_{10}\left(\frac{d_{i,j}(k)}{d_0}\right) + X_{\sigma_1}, d_0 \le d_{i,j}(k) \le d_c; \\ P\left(d_0\right) - 10\eta_1 \log_{10}\left(\dfrac{d_c}{d_0}\right) - \\ \quad 10\eta_2 \log_{10}\left(\dfrac{d_{i,j}(k)}{d_c}\right) + X_{\sigma_2} \end{cases}, d_{i,j}(k) > d_c, \tag{3}$$

where $P(d_0)$ is the known reference power received at the reference distance d_0. η_1 and η_2 are two different path loss exponents. d_c is a critical distance which can be estimated as $d_c = 4h_ih_j/\lambda_{\mathrm{WAVE}}$ given the heights of i's and j's antennas, h_i, h_j, and the wavelength of the electromagnetic wave at 5.9 GHz, λ_{WAVE}. X_{σ_1} and X_{σ_2} are two zero-mean normally random variables with standard deviations σ_1 and σ_2, respectively.

Now, let A be the signal envelope received by vehicle j from vehicle i, which is a random variable following the Nakagami distribution characterized by $m\left(d_{i,j}(k)\right)$ and $\omega\left(d_{i,j}(k)\right)$. We can express the cumulative density function (CDF) of the received signal strength as

$$\Pr\left\{A^2 \le x\right\} = \frac{\gamma\left(m\left(d_{i,j}(k)\right), \frac{m(d_{i,j}(k))}{\omega(d_{i,j}(k))}x\right)}{\Gamma\left(m\left(d_{i,j}(k)\right)\right)}, \tag{4}$$

where $\Gamma\left(m\left(d_{i,j}(k)\right)\right)$ is the Gamma function characterized by $m(d_{i,j}(k))$, $\Gamma\left(m\left(d_{i,j}(k)\right)\right) = \int_0^\infty s^{m(d_{i,j}(k))-1}e^{-s}ds$, and $\gamma\left(m\left(d_{i,j}(k)\right), \frac{m(d_{i,j}(k))}{\omega(d_{i,j}(k))}x\right)$ is the complementary incomplete Gamma function, $\gamma\left(m\left(d_{i,j}(k)\right), \frac{m(d_{i,j}(k))}{\omega(d_{i,j}(k))}x\right) = \int_0^{\frac{m(d_{i,j}(k))}{\omega(d_{i,j}(k))}x} s^{m(d_{i,j}(k))-1}e^{-s}ds$.

Based on (4), we can further derive the CDF of the received signal-to-noise-ratio (SNR) at vehicle j as follows

$$\Pr\left\{\frac{A^2}{\psi} \le x\right\} = \frac{\gamma\left(m\left(d_{i,j}(k)\right), \frac{m(d_{i,j}(k))}{\omega(d_{i,j}(k))}\psi x\right)}{\Gamma\left(m\left(d_{i,j}(k)\right)\right)}, \tag{5}$$

where ψ is the thermal noise strength at j. In the following, for the sake of simplicity, we use $m(k)$ and $\omega(k)$ to represent $m\left(d_{i,j}(k)\right)$ and $\omega\left(d_{i,j}(k)\right)$, respectively.

In the DSRC physical layer, four types of modulation modes, i.e., BPSK, QPSK, 16-QAM and 64-QAM, and three types of forward error correction coding rates, i.e., 1/2, 2/3 and 3/4, are usually available, which can result in different transmission rates on the V2V channels. Let $\mathcal{C} = \{c_1, c_2, \ldots, c_M\}$ denote the finite set of alternative transmission rates, where M is the maximum number of the transmission rates. Therefore, we adopt an adaptive modulation scheme to enhance the V2V communication on the DSRC channel as in [34], [38], in which the implementation of each transmission rate c_l is according to the SNR at the receiver j. Specifically, the SNR at the receiver can be divided into M non-overlapping intervals by different thresholds ϕ_l ($l = 1, 2, \ldots, M+1$), where $\phi_l < \phi_{l+1}$ for $l = 1, 2, \ldots, M$, $\phi_1 = 0$ and $\phi_{M+1} = +\infty$. If the SNR at the receiver satisfies $\phi_l < A^2/\psi \le \phi_{l+1}$, then the transmission rate c_l is implemented in the DSRC physical layer. With (5), we derive the probability of implementing the transmission rate c_l as follows:

$$\begin{aligned} p\left(C = c_l; m\left(k\right), \omega\left(k\right)\right) &= \Pr\left\{\phi_l < \frac{A^2}{\psi} \le \phi_{l+1}\right\} \\ &= \frac{\gamma\left(m\left(k\right), \frac{m(k)}{\omega(k)}\psi\phi_{l+1}\right) - \gamma\left(m\left(k\right), \frac{m(k)}{\omega(k)}\psi\phi_l\right)}{\Gamma\left(m\left(k\right)\right)}. \end{aligned} \tag{6}$$

3.4.2 V2V channel contention

To model the V2V channel contention, we assume that the number of vehicles in the neighborhood of vehicle i contending to access the V2V channel, N, follows a Poisson distribution

$$p\left(N = n\right) = \frac{(R\rho)^n}{n!}\exp(-R\rho), \tag{7}$$

where ρ denotes the vehicle density and R is vehicle i's carrier sensing range. It is worth pointing out that the Poisson distribution of vehicular nodes has been validated based on the measurements in [39] and also widely adopted in current literature such as [28].

Additionally, we assume that the IEEE 802.11 distributed coordination function (DCF) scheme is used for the DSRC MAC scheduling and the RTS and CTS schemes are used for the hidden terminal transmission eliminating. Thus, given the contention window of the exponential backoff in each competitive vehicle, W, we can get the average transmission probability of vehicles, b, by using the mean approximation: $b = 1/(0.5W+1)$. Since there exist N vehicles in i's neighborhood, the probability that the V2V channel is idle can be expressed as $p_{\mathrm{idle}} = (1-b)^N$, while the probability of successful transmission among the vehicles is $p_{\text{suc}} = \mathrm{C}_N^1 b(1-b)^{N-1}$. Consequently, the probability of transmission colliding is $p_{\text{col}} = 1 - p_{\text{idle}} - p_{\text{suc}}$. Therefore, we can derive the throughput from i to j as follows

$$\Omega(C,N) = \frac{b(1-b)^{N-1}Z}{p_{\text{idle}}T_{\text{slot}} + p_{\text{col}}T_{\text{col}} + p_{\text{suc}}\left[T_{\text{MAC}} + \frac{Z}{\overline{C}}\right]}, \tag{8}$$

where Z is the average packet size of each vehicle on the V2V channel. T_{slot} is the unit duration of a time slot in the DCF backoff procedure. T_{col} is the average time of a collided transmission, which can be further expressed as $T_{\text{col}} = T_{\text{RTS}} + T_{\text{DIFS}} + T_{\text{slot}}$ given the time interval of the DCF interframe space (DIFS), T_{DIFS}, and the time interval reserved for transmission of the RTS message, T_{RTS}. T_{MAC} can also be expressed as $T_{\text{MAC}} = T_{\text{RTS}} + T_{\text{DIFS}} + 3T_{\text{SIFS}} + T_{\text{CTS}} + T_{\text{ACK}} + 4T_{\text{slot}}$, where T_{SIFS}, T_{CTS}, T_{ACK} are the pre-specified time intervals reserved for the short interframe space (SIFS) as well as transmissions of the CTS and ACK messages. $\overline{C}$ denotes the average transmission rate among the N vehicles, which can be obtained with (6)

$$\overline{C} = \frac{1}{N}C + \frac{N-1}{N}\int_0^R \frac{\sum_{l=1}^{M} c_l p(C = c_l; m(r), \omega(r))}{R} dr, \tag{9}$$

where $m(r)$ is the V2V channel fading parameter and $\omega(r)$ is the average received signal strength, when the distance between a transmitter-receiver pair is r.

3.4.3 V2V transmission performance

Based on equations (9), we are able to evaluate the performance of the V2V transmission between vehicle i and j. Let $T_{i,j}$ ($T_{i,j} \geq 0$) be the allowable deadline (the maximum number of time slots $\Delta\tau$) for the V2V connection between i and j. Then, noting that the relative transmission distance between i and j, $d_{i,j}(k)$, can be updated by using the discretized vehicle dynamics (1) and (2), we can derive the data amount that can transferred from i to j during the time interval $[0, T_{i,j}\Delta\tau]$ by

$$\theta(T_{i,j}) = \sum_{k=1}^{T_{i,j}} \Omega(C,N)\Delta\tau. \tag{10}$$

From equation (10), C and N are two independent random variables, so that $\theta(T_{i,j})$ is also a random variable. The calculation of (10) indeed involves random processes, which is intractable. Thus, we turn to analyze the mathematical expectation and the variance of $\theta(T_{i,j})$ from a probabilistic perspective. The linearity of the expected value leads to the following lemma

Lemma 1. *Given $m(k)$ and $\omega(k)$ based on $d_{i,j}(k) = |L_{i,j}(k)|$ where $L_{i,j}(k)$ is determined by (1) and (2) for $k = 1, 2, \ldots, T_{i,j}$, the expected volume of data that can be transmitted from i to j within the time period of $T_{i,j}$ is*

$$\mathbb{E}\left[\theta(T_{i,j})\right] = \sum_{k=1}^{T_{i,j}} \mathbb{E}\left[\Omega(C,N)\right]\Delta\tau, \tag{11}$$

in which $\mathbb{E}\left[\Omega(C,N)\right]$ can be determined by

$$\begin{aligned}&\mathbb{E}\left[\Omega(C,N)\right]\\ &= \sum_{n=1}^{N_{\max}} \sum_{l=1}^{M} \Omega(C,N) p\left(C = c_l; m(k), \omega(k)\right) p(N = n),\end{aligned} \tag{12}$$

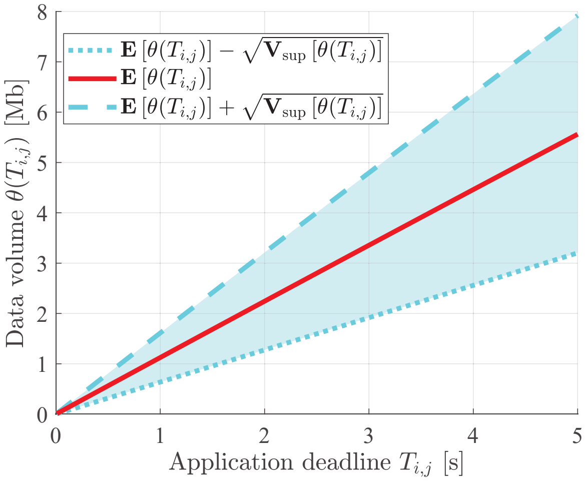


Fig. 3. The average data volume transmitted over a V2V link and the upper bound of the corresponding variance within different deadlines $T_{i,j}$.

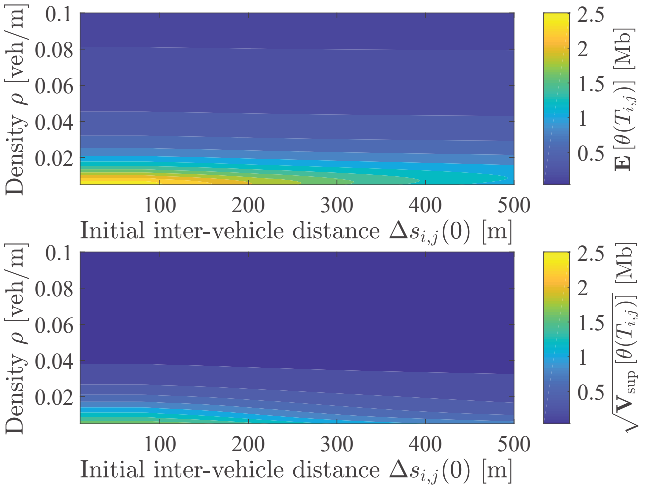


Fig. 4. The average data volume over a V2V link and the upper bound of the corresponding variance under different vehicle densities ρ and initial inter-vehicle distance $\Delta s_{i,j}(0)$.

where $N_{\max}$ is the possible maximum number of vehicles existing in the application scenario.

Based on Lemma 1, we derive an upper bound for the variance of $\theta(T_{i,j})$, $\mathbb{V}\left[\theta(T_{i,j})\right]$, by using inequality analysis as follows. The proof is available in the online supplemental material.

Theorem 1. $\mathbb{V}\left[\theta(T_{i,j})\right] \leq \mathbb{V}_{\text{sup}}\left[\theta(T_{i,j})\right]$ *always holds, where*

$$\mathbb{V}_{\text{sup}}\left[\theta(T_{i,j})\right] = T_{i,j}\Delta\tau \sum_{k=1}^{T_{i,j}} \mathbb{E}\left[\Omega^2(C,N)\right]\Delta\tau - \left(\mathbb{E}\left[\theta(T_{i,j})\right]\right)^2. \tag{13}$$

To show how the application deadline $T_{i,j}$, the vehicle density on the road ρ and the initial space headway $\Delta s_{i,j}(0)$ affect the V2V data transmission, we carry out numerical evaluations by combining the models of vehicle mobility, the V2V physical-layer and the channel contending. The

TABLE 1
V2V fading parameter and data rate settings

$\Delta s_{i,j}(k)$ [m]	5.5	13.9	35.5	90.5	230.7	588.0	$+\infty$
$m(k)$	4.07	2.44	3.08	1.52	0.74	0.84	0.5

ϕ_l [dB]	5	6	8	11	15	20	25
c_l [Mbps]	3	4.5	6	9	12	18	24

TABLE 2
DSRC-based V2V communication configuration

Z [Bytes]	1000
W	32
R [m]	300
T_{SLOT} [µs]	13
$T_{\mathrm{SIFS}}, T_{\mathrm{DIFS}}$ [µs]	32
$T_{\mathrm{ACK}}, T_{\mathrm{CTS}}$ [µs]	37
T_{RTS} [µs]	53
ψ [dBm]	-96
h_i, h_j [m]	1
η_1, η_2	2.1, 3.8
d_0 [m]	100
$G_t, G_r, \alpha_{\mathrm{loss}}$	1
P_t [dBm]	23

channel fading parameter $m(k)$ is associated with the inter-vehicle distance, and the physical-layer data rate c (i.e., the selection of the modulation scheme) is adapted according to the SNR condition. The settings on $m(k)$ and c are detailed in Table 1 according to [37], [40]. The reference power at d_0 is evaluated by using the well-known two-ray ground-reflection model $P(d_0) = P_t G_t G_r \frac{(h_i h_j)^2}{(d_0^4 \alpha_{\mathrm{loss}})}$ with the gains at the transmitter and the receiver $G_t = G_r$ and the system path loss factor α_{loss}. The DSRC-based vehicular communication configurations are given in Table 2 according to the existing studies [37], [41], [42].

Fig. 3 shows the average data volume transmitted from vehicle i to vehicle j, $\mathbb{E}\left[\theta(T_{i,j})\right]$, and the upper bound of the corresponding variance, $\mathbb{V}_{\mathrm{sup}}\left[\theta(T_{i,j})\right]$, given different deadlines $T_{i,j}$, where the mobility-related parameters are initialized as $\Delta\tau = 1\,\mathrm{ms}$, $v_i(0) = 30\,\mathrm{km/h}$, $v_j(0) = 90\,\mathrm{km/h}$, $s_i(0) = s_j(0) = 0\,\mathrm{m}$, and $L_{\mathrm{V2V},ij} = 150\,\mathrm{m}$. In the simulation, the accelerations of both the vehicles are constantly zero, i.e., indicating that i and j are moving at a constant speed. As can be seen from Fig. 3, the expected data volume that can be successfully transferred via the V2V link is monotonically increasing with specifying a larger deadline. Nonetheless, the variance of the transmitted data volume also increases in such a situation, which implies that the uncertainty in actually completing the data transmission over the V2V link is enhanced. This is due to the vehicle mobility, i.e., the vehicles' motion increasing the outage possibility of the V2V connection within a relatively large period. In Fig. 4 where we specify the application deadline as $T_{i,j} = 500\,\mathrm{ms}$ while varying the vehicle density and the initial inter-vehicle distance, it can be observed that a higher vehicle density or/and a larger initial space headway can reduce the data transmission over the V2V link. The main reason is that the channel contending is intensified with increasing the vehicle density and the channel fading also becomes more serious with increasing the transmission distance, which, as a consequence, degrades V2V performance.

Now, using both Lemma 1 and Theorem 1, we characterize the stochastic V2V communication performance. Let $D_{i,j}$ denote the amount of content data that needs to be transferred from vehicle i to vehicle j within the time window $T_{i,j}$, as the input data size of the application partitioned and offloaded to j. We define the *V2V communication reliability* as the success probability of transmitting $D_{i,j}$ data bits via i-to-j connection meanwhile satisfying the given deadline $T_{i,j}$. Mathematically, the V2V communication reliability can be formulated by $\mathrm{Prob}\left\{\theta(T_{i,j}) > D_{i,j}\right\}$. However, it is difficult or even impossible to obtain the actual probability distribution of $\theta(T_{i,j})$ in reality. Here, we further use a utility function to characterize the unknown $\mathrm{Prob}\left\{\theta(T_{i,j}) > D_{i,j}\right\}$ by resorting to the CDF of the normal distribution. The normal distribution is widely applied in many fields related to stochastic processes because of the central limit theorem. Specifically, we formulate the utility function

$$p_{\mathrm{V2V}}(D_{i,j}, T_{i,j}) = \begin{cases} 1 - \int_{-\infty}^{D_{i,j}} \frac{e^{-\frac{(x-\mathbb{E}[\theta(T_{i,j})])^2}{2\mathbb{V}_{\mathrm{sup}}[\theta(T_{i,j})]}}}{\sqrt{2\pi\mathbb{V}_{\mathrm{sup}}[\theta(T_{i,j})]}} \mathrm{d}x, D_{i,j} < \mathbb{E}\left[\theta(T_{i,j})\right]; \\ 0, D_{i,j} \geq \mathbb{E}\left[\theta(T_{i,j})\right]. \end{cases} \tag{14}$$

as a metric of the V2V communication reliability. From (14), it can be seen that $p_{\mathrm{V2V}}(D_{i,j}, T_{i,j})$ is a monotonously decreasing function of $D_{i,j}$, shaped by the expected V2V link capacity $\mathbb{E}\left[\theta(T_{i,j})\right]$ and the supper bound of the variance $\mathbb{V}_{\mathrm{sup}}\left[\theta(T_{i,j})\right]$. Logically, a V2V link is more likely to successfully transfer the data with smaller size by a larger deadline. Thus, the reliability metric $p_{\mathrm{V2V}}(D_{i,j}, T_{i,j})$ increases by reducing $D_{i,j}$ and/or increasing $T_{i,j}$.

3.5 Modeling of V2I Communication Reliability

3.5.1 V2I channel model

For modeling the V2I communication, we consider that the V2I connection is provided based on the cellular network. That is, the wireless access infrastructure of the roadside cloudlet can be a cell base station in reality. However, quite different from the exclusive DSRC-based V2V communication, the cellular licensed radio channel for the V2I communication can be usually assumed to follow the Rayleigh fading [43], [44], especially when the radio signal propagation is operated in built-up urban environments. Let $g_{\mathrm{V2I}}\left(d_l(k)\right)$ be the distance-dependent cellular V2I channel gain, in which $d_l(k)$ is the relative distance between vehicle l ($l \in \{i, j\}$) and the infrastructure, i.e., $d_l(k) = |L_l(k)|$ based on (2). Formally, the cellular V2I channel capacity between vehicle $l \in \{i, j\}$ and the infrastructure can be formulated as

$$\pi_l(k) = \frac{B}{M} \log_2\left(1 + w_l g_{\mathrm{V2I}}^2\left(d_l(k)\right)\right), \tag{15}$$

where B is the available bandwidth, M is the number of vehicles that are also accessing the infrastructure, and w_l denotes the normalized power of l for V2I communication.

3.5.2 V2I transmission performance

Given the path loss exponent β, the Rayleigh fading makes $g_{\mathrm{V2I}}^2\left(d_l(k)\right)$ follow an exponential distribution with the parameter $d_l^{\beta}(k)$. In this sense, denoting by $u_l(k)$ the data volume that needs to be transmitted to the infrastructure in time slot k, i.e., the number of bits offloaded from vehicle l to the infrastructure in time slot k, the probability of successful transmission is

$$\begin{aligned} & h_l(u_l(k);M) = \text{Prob}\left\{\pi_l(k) \geq \frac{u_l(k)}{\Delta\tau}\right\} = \\ & \exp\left(-\frac{2^{\frac{u_l(k)M}{B\Delta\tau}}-1}{w_l} d_l^{\beta}(k)\right). \end{aligned} \tag{16}$$

Let the total number of data bits transmitted by vehicle l to the infrastructure be $D_{\text{out},l}$, and the allowable deadline be $T_{\text{V2I},l}$. We can propose a V2I transmission schedule on $D_{\text{out},l}$ across $T_{\text{V2I},l}$ time slots by appropriately partitioning the total data file into a series of smaller files $\{u_l(k) \geq 0, k = 1, 2, \ldots, T_{\text{V2I},l}\}$, such that $D_{\text{out},l} = \sum_{k=1}^{T_{\text{V2I},l}} u_l(k)$. Based on the multiplication principle, we derive the overall success probability of V2I transmissions

$$\begin{aligned} & H\left(D_{\text{out},l}, T_{\text{V2I},l}; M\right) = \prod_{k=1}^{T_{\text{V2I},l}} h_l(u_l(k);M) = \\ & \exp\left(-\frac{\sum_{k=1}^{T_{\text{V2I},l}} d_l^{\beta}(k) 2^{\frac{u_l(k)M}{B\Delta\tau}}}{w_l} + \frac{\sum_{k=1}^{T_{\text{V2I},l}} d_l^{\beta}(k)}{w_l}\right). \end{aligned} \tag{17}$$

3.5.3 Optimal V2I transmission scheduling

Since the V2I transmission is independently performed by a vehicular node, this individual node can independently optimize its V2I data transmission by optimally scheduling $\{u_l(k) \geq 0, k = 1, 2, \ldots, T_{\text{V2I},l}\}$ transmitted across $T_{\text{V2I},l}$ time slots. Formally, the reliability-oriented optimization model for the optimal V2I data transmission scheduling of vehicle $l \in \{i, j\}$ can be formulated as follows

$$\begin{aligned} & \max_{\{u_l(k)\}} : H\left(D_{\text{out},l}, T_{\text{V2I},l}; M\right) \\ & s.t. \begin{cases} \sum_{k=1}^{T_{\text{V2I},l}} u_l(k) = D_{\text{out},l}, \\ u_l(k) \geq 0, \ \forall k \in \{1, 2, \ldots, T_{\text{V2I},l}\} \end{cases} \end{aligned} \tag{18}$$

To further analyze the optimization model above, we equivalently transform it to the following one

$$\begin{aligned} & \min_{\{u_l(k)\}} : G\left(\boldsymbol{u}_l\right) = \sum_{k=1}^{T_{\text{V2I},l}} d_l^{\beta}(k) 2^{\frac{u_l(k)M}{B\Delta\tau}} \\ & s.t. \begin{cases} \sum_{k=1}^{T_{\text{V2I},l}} u_l(k) = D_{\text{out},l}, \\ u_l(k) \geq 0, \ \forall k \in \{1, 2, \ldots, T_{\text{V2I},l}\} \end{cases} \end{aligned} \tag{19}$$

where $\boldsymbol{u}_l = \left[u_l(1), u_l(2), \ldots, u_l(T_{\text{V2I},l})\right]^{\text{T}}$ denotes a data scheduling solution.

To derive the optimal V2I transmission scheduling, we first provide the following lemma based on model (19), the proof of which is given in the online supplementary material.

Lemma 2. *Suppose that $\boldsymbol{u}_l^{\text{opt}}$ is the optimal feasible solution for (19). For any two nonzero optimal data partitions $u_l(k') > 0$ and $u_l(k) > 0$, $u_l(k'), u_l(k) \in \boldsymbol{u}_l^{\text{opt}}$ ($k \neq k'$), the partial derivative of the objective function $G(\boldsymbol{u}_l)$ with respect to them are identical, i.e., $\partial G(\boldsymbol{u}_l)/\partial u_l(k') = \partial G(\boldsymbol{u}_l)/\partial u_l(k)$. In contrast, for any zero data partition in $\boldsymbol{u}_l^{\text{opt}}$, the corresponding partial derivative of the objective function is not less than those associated with the nonzero partitions, i.e., $\partial G(\boldsymbol{u}_l)/\partial u_l(k^*) \geq \partial G(\boldsymbol{u}_l)/\partial u_l(k)$ holding for any $u_l(k^*) = 0$, $u_l(k) > 0$, $u_l(k^*), u_l(k) \in \boldsymbol{u}_l^{\text{opt}}$.*

TABLE 3
Cellular V2I communication configuration

B [Mbit]	200
Average noise strength [dBm]	-96
V2I transmission power [W]	1
V2I SNR gap	5
R_{V2I} [m]	1000
β	2

Following Lemma 2, we further obtain the closed-form optimal V2I data scheduling as well as the maximum probability of successful V2I data transmission in Theorem 2.

Theorem 2. *For the optimal V2I data scheduling solution $\boldsymbol{u}_l^{\text{opt}}$ of model (18), it always holds that for all $u_l(k) > 0$, $u_l(k) \in \boldsymbol{u}_l^{\text{opt}}$,*

$$u_l(k) = \frac{B\Delta\tau\left(\sum_{k=1}^{T_{\text{V2I},l}} \log_2 d_l^{\beta}(k) - T_{\text{V2I},l} \log_2 d_l^{\beta}(k)\right)}{MT_{\text{V2I},l}} + \frac{D_{\text{out},l}}{T_{\text{V2I},l}}. \tag{20}$$

Correspondingly, the maximum overall probability of successful V2I data transmission, $H^{\text{opt}}\left(D_{\text{out},l}, T_{\text{V2I},l}; M\right)$, is

$$\begin{aligned} & H^{\text{opt}}\left(D_{\text{out},l}, T_{\text{V2I},l}; M\right) = \\ & \exp\left(-\frac{T_{\text{V2I},l} 2^{\frac{MD_{\text{out},l}}{T_{\text{V2I},l}B\Delta\tau}} \left(\prod_{k=1}^{T_{\text{V2I},l}} d_l^{\beta}(k)\right)^{\frac{1}{T_{\text{V2I},l}}}}{w_l} + \frac{\sum_{k=1}^{T_{\text{V2I},l}} d_l^{\beta}(k)}{w_l}\right). \end{aligned} \tag{21}$$

The proof of Theorem 2 is given in the online supplementary material. Furthermore, it is noticed that in the optimal objective function, $H^{\text{opt}}\left(D_{\text{out},l}, T_{\text{V2I},l}; M\right)$, M is an independent random variable, which reflects the stochastic number of vehicles accessing the same infrastructure. At this point, we turn to calculate the expected success probability of V2I data transmission. Let the expected success probability of V2I data transmission be $p_{\text{V2I},l}\left(D_{\text{out},l}, T_{\text{V2I},l}\right)$. We use $p_{\text{V2I},l}\left(D_{\text{out},l}, T_{\text{V2I},l}\right)$ to quantify the *V2I communication reliability*. Specifically, also following the Poisson distribution assumption on M, we derive

$$\begin{aligned} & p_{\text{V2I},l}\left(D_{\text{out},l}, T_{\text{V2I},l}\right) \\ & = \sum_{m=1}^{M_{\max}} \frac{\left(S_{\text{V2I}}\rho_{\text{c}}\right)^m}{m!} \exp(-S_{\text{V2I}}\rho_{\text{c}}) H^{\text{opt}}\left(D_{\text{out},l}, T_{\text{V2I},l}; m\right) \end{aligned} \tag{22}$$

for $l = i, j$, where S_{V2I} is the coverage area of the infrastructure, i.e., $S_{\text{V2I}} = \pi R_{\text{V2I}}^2$, and ρ_{c} is the vehicle density within the coverage in vehicles per squared meter. $M_{\max}$ denotes the possible maximum number of vehicles simultaneously accessing the infrastructure via V2I communication.

We further conduct simulation experiments to investigate the V2I performance under different factors including the total number of vehicles accessing the infrastructure, the V2I relative distance, and the data volume required to be transmitted over the V2I link. The basic parameter settings

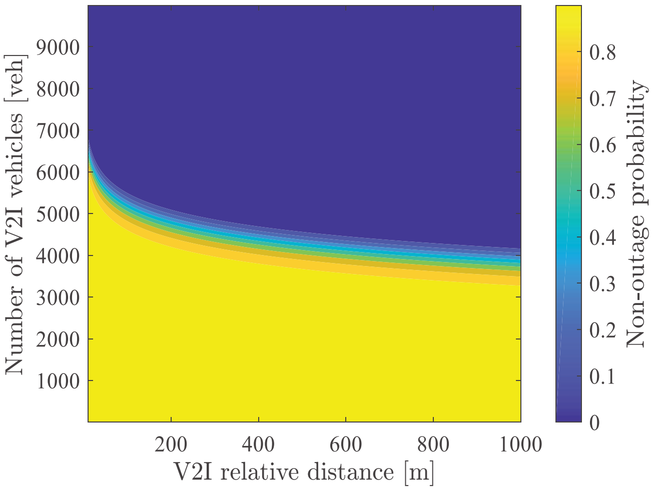


Fig. 5. The probability of success in the V2I transmission under different accessing vehicle amounts and V2I transmission distances.

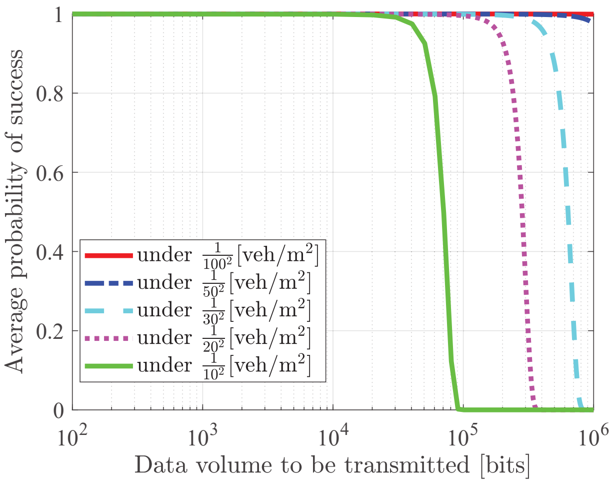


Fig. 6. The average probability of success in the V2I transmission with different data volumes, $D_{\mathrm{out},l}$, under different vehicle densities ρ_{c}.

for the V2I-based data transmission are given in Table 3. In Fig. 5 where the average data rate required is set to $u_l(k)/\Delta\tau = 1\,\mathrm{Mbit/s}$, the probability of success in the V2I transmission, i.e., the non-outage probability $h_l(u_l(k); M)$, is degraded by increasing the transmission distance $d_l(k)$. Compared to the effect of the geographic distance, the total amount of vehicles accessing to the infrastructure has a greater influence on the V2I transmission. Because more accessing vehicles can lead to less available bandwidth allocated to a single vehicle, the non-outage probability decreases with increasing the number of V2I communication vehicles. Interestingly, as Fig. 5 shows, the non-outage probability experiences a sharp decline when the total vehicle number within the infrastructure coverage increases from 4000 to 6000. With the total vehicle number exceeding 6000, the non-outage probability is at a low level (lower than 0.4). In Fig. 6, we evaluate the average probability of success in transmitting a certain volume of data over a V2I link based on the proposed model (22). The communication vehicle's kinematic parameters are given as $s_i(0) = 0\,\mathrm{m}$, $v_i(0) = 100\,\mathrm{km/h}$, and $a_i(k) = 0\,\mathrm{m/s^2}$ for all k. The relative distances are initialized as $L_0 = 500\,\mathrm{m}$ and $L_{\mathrm{V2I},i} = 0\,\mathrm{m}$. The application deadline is specified as $T_{\mathrm{V2I},i} = 1000\,\mathrm{ms}$, while the data volume required to be transmitted, $D_{\mathrm{out},i}$, is varied from 10^2 bit to 10^6 bit. The vehicle density within the coverage of the infrastructure, ρ_{c}, is set to $\frac{1}{10^2}$, $\frac{1}{20^2}$, $\frac{1}{30^2}$, $\frac{1}{50^2}$ and $\frac{1}{100^2}$ (veh/m^2), respectively. From Fig. 6, as the data volume to be transmitted increases, the probability of success in the V2I transmission decreases, and increasing the vehicle density can also reduce the V2I performance.

3.6 Modeling of Vehicular Computation Reliability

According to current literature [45], [46], the number of CPU clock cycles, W, required by processing an application can be estimated by a linear function of the input data size of the application, i.e., $W = D_{\mathrm{in},l}X$ where $D_{\mathrm{in},l}$ denotes the input data bits to be locally processed by vehicle $l \in \{i, j\}$ and X is a random variable depending on many factors, such as the nature of the application, the complexity of the processing algorithm, etc., which can be modeled by a certain empirical probabilistic distribution. Specifically, as shown in [45], [47], X can be estimated by a Gamma distribution, and in addition, the Gamma distribution-based modeling of stochastic CPU cycles per bit in mobile computing has been widely adopted in much literature such as [20], [21]. Accordingly, given a $D_{\mathrm{in},l}$, we can derive the CDF for W as follows

$$g_W(w; D_{\mathrm{in},l}) = \int_0^{\frac{w}{D_{\mathrm{in},l}}} \frac{e^{-\frac{x}{\mu_1}}}{\mu_1\Gamma(\mu_2)}\left(\frac{x}{\mu_1}\right)^{\mu_2-1} dx \tag{23}$$

where μ_1 and μ_2 are two fitting parameters of the Gamma distribution, and w is the observation of W. Based on (23), we can get the probability of requiring more than w CPU clock cycles to complete an application as $1 - g_W(w; D_{\mathrm{in},l})$.

Besides, in a mobile CPU circuit, the clock frequency of the chip is allowed to be dynamically configured by using the dynamic voltage scaling (DVS) technique [48]. Thus, we also consider the CPU clock-frequency configuration for the vehicular computation device. Let f_w denote the CPU clock-frequency to be scheduled in the next CPU cycle given that w CPU cycles have been completed. Then, the execution duration of such a CPU cycle is $1/f_w$. As stated in [20], [21], [46], [49], the total energy consumed during all the CPU cycles executed by vehicle l can be estimated by

$$E_l\left(D_{\mathrm{in},l}, W_l\right) = \kappa \sum_{w=1}^{W_l} \left(1 - g_W(w; D_{\mathrm{in},l})\right) f_w^2 \tag{24}$$

given that the required number of CPU cycles by l is W_l, where κ represents the chip architecture-dependent effective switched capacity. From the measurements reported in [46], we can set it to $\kappa = 10^{-11}$.

Now, instead of minimizing the computing energy efficiency, we aim to minimize the total vehicular computing time by optimally scheduling the CPU clock frequency with preallocated computation energy $E_l^{\max}$. Mathematically, we formulate the optimization problem as follows

$$\begin{aligned} &\min_{\{f_w\}} : T\left(D_{\mathrm{in},l}, W_l, E_l^{\max}\right) = \sum_{w=1}^{W_l} \frac{1}{f_w} \\ &s.t. \begin{cases} E_l\left(D_{\mathrm{in},l}, W_l\right) \leq E_l^{\max}; \\ f_w > 0, \forall w = 1, 2, \ldots, W_l. \end{cases} \end{aligned} \tag{25}$$

Based on model (25), we can establish the connection between the input data size of the vehicular execution $D_{\mathrm{in},l}$, the required CPU clock cycles W_l, the available computing energy $E_l^{\max}$, and the optimal vehicular computing efficiency in Theorem 3.

Theorem 3. *Given $D_{\mathrm{in},l}$, W_l and $E_l^{\max}$ for vehicle $l \in \{i, j\}$, the optimal CPU clock-frequency scheduling for model (25) is*

$$f_w = \sqrt{\frac{E_l^{\max}}{\kappa W_l \left(1 - g_W\left(w; D_{\mathrm{in},l}\right)\right)}} \tag{26}$$

for $w = 1, 2, \ldots, W_l$, and the minimum scheduling time is

$$\begin{aligned} &T^{\min}\left(D_{\mathrm{in},l}, W_l, E_l^{\max}\right) \\ &= \sqrt{\frac{W_l}{E_l^{\max}}} \left(\sum_{w=1}^{W_l} \sqrt{\kappa \left(1 - g_W\left(w; D_{\mathrm{in},l}\right)\right)} \right). \end{aligned} \tag{27}$$

The proof of Theorem 3 is available in the online supplementary material.

Denote by $T_{\mathrm{local},l}$, $T_{\mathrm{local},l} < T$, the actual deadline allocated for the local execution by vehicle $l \in i, j$. Considering the implementation of the optimal CPU clock-frequency scheduling, we can solve the corresponding required CPU clock cycle number W_l from (27). More specifically, we denote such a number of the CPU clock cycles under the optimal CPU scheduling policy by

$$\begin{aligned} &W_l\left(D_{\mathrm{in},l}, T_{\mathrm{local},l}, E_l^{\max}\right) \\ &= \operatorname{argmax}\left\{W_l \in \mathbb{Z}^+ \left| T^{\min}\left(D_{\mathrm{in},l}, W_l, E_l^{\max}\right) \leq T_{\mathrm{local},l} \right.\right\}. \end{aligned} \tag{28}$$

Therefore, with (28) above, we obtain the probability of successfully completing the local execution by using $W_l\left(D_{\mathrm{in},l}, T_{\mathrm{local},l}, E_l^{\max}\right)$ CPU clock cycles

$$\begin{aligned} &p_{\mathrm{local},l}\left(D_{\mathrm{in},l}, T_{\mathrm{local},l}\right) \\ &= g_W\left(W_l\left(D_{\mathrm{in},l}, T_{\mathrm{local},l}, E_l^{\max}\right); D_{\mathrm{in},l}\right) \end{aligned} \tag{29}$$

as a metric of l's computation reliability.

Compared with vehicular computation, in general, the computation capacity of an infrastructure (e.g., the cloud) is much more greater. At this point, it is reasonable to assume that the infrastructure-based computation reliability is 100%. That is, we assume that the infrastructure can always successfully complete the whole computation task within the given time window T.

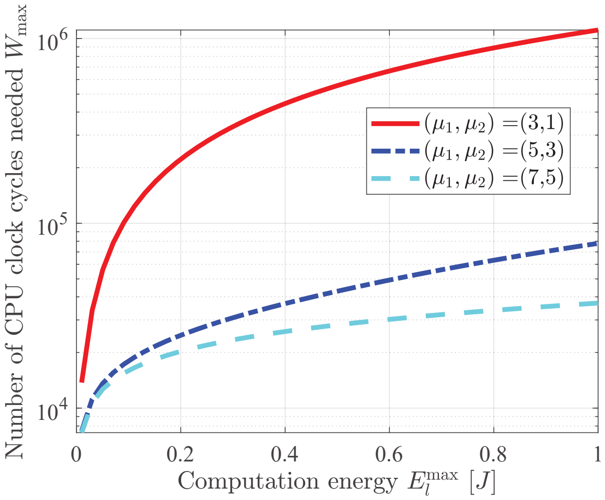


Fig. 7. The number of required CPU clock cycles to complete an specified application under different (μ_1, μ_2) and $E_l^{\max}$.

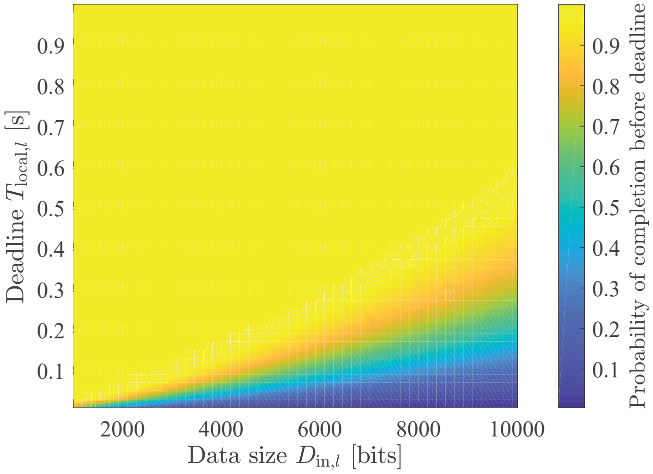


Fig. 8. The probability of success in the computation with different data volumes, $D_{\mathrm{in},l}$, and different deadlines, $T_{\mathrm{local},l}$.

To examine the impacts of some characteristic parameters related to the computation capacity of a vehicle l, such as (μ_1, μ_2) and $E_l^{\max}$, we calculate the number of CPU clock cycles required to complete a specified application under different situations, as shown in Fig. 7, where (μ_1, μ_2) is set to $(3, 1)$, $(5, 3)$ and $(7, 5)$, respectively, while the available computation energy $E_l^{\max}$ is varying from 0.01 J to 1 J. The total input data size of the application is assumed to be $D_{\mathrm{in},l} = 1000$ bit and its deadline is $T_{\mathrm{local},l} = 20$ ms. Fig. 7 shows that more computation energy can provide more power to increase the number of CPU cycles for the application within the deadline, which implies that the probability of application completion also increases. Besides, a larger μ_1 or a larger μ_2 can lead to a lower number of needed CPU cycles, which can also be observed according to the PDF model (23). In addition, we fix the scale and shape parameters (μ_1, μ_2) to $(5, 3)$ and the available computation power $E_l^{\max}$ to 0.5 J. We evaluate the probability of application completion by an imposed deadline in Fig. 8, where the size of the application input data, $D_{\mathrm{in},l}$, is varied from 1000 bit to 10 000 bit and the given deadline from 0.01 s to 1 s. As expected, the more the application data needed to be processed by a given deadline is, the smaller the completion probability becomes. Similarly, the computation is more likely not to be completed when the allowed execution time is short. For example, from Fig. 8, it is observed that the probability of successfully processing 8000-bit data within

200 ms is lower than 0.7. As discussed, an application with a relatively large volume of input data and a short deadline may hardly be accomplished by the fully local execution. Hence, cooperative computation solution with optimal design is needed to support such a situation.

4 Cooperative computation optimization

4.1 Computation Modes

As shown in Fig. 1, there are 7 (=$\mathrm{C}_3^1+\mathrm{C}_3^2+\mathrm{C}_3^3$) computation modes for executing a vehicular distributed application in a three-node cooperative vehicle-infrastructure system. It should be remarked that when considering multi-hop communications, the cooperation mode becomes much more complex. To implement a computation mode, we assume that the real-time location information and the kinematic parameters of a vehicle can be accessed by the on-board global position system (GPS) and other vehicular motion sensors. The location of the infrastructure (i.e., the cell base station) can also be known in advance. Messages carrying vehicular mobility information can be locally shared among the neighbors via a common V2V broadcasting channel (such as DSRC channel 178 for control signaling) [34]. Each computation mode is elaborated on as follows.

Mode 1: The host vehicle self-independently executes the whole computation task and then uploads the result to the infrastructure via V2I communication.

Mode 2: The host directly transfers all the input data to the infrastructure and then the infrastructure takes the responsibility to complete the whole data process.

Mode 3: The host can offload all the input data to a nearby vehicle via V2V communication, such that the cooperator can help processing the task. Finally, it also needs to upload the result to the infrastructure via V2I communication.

Mode 4: The host divide the input data into two sub-files of different sizes, one of which is offloaded to the infrastructure for the cloud execution, and the other is processed by the local execution. After completing the local execution, the local output is also uploaded to the infrastructure.

Mode 5: The application execution is partitioned into two local executions processed by the host vehicle and a cooperative vehicle. Then, both the outputs of the two vehicles are transferred to the infrastructure, respectively.

Mode 6: The host distributes two partitions of the application to the cooperative vehicle and the infrastructure, respectively. The application execution is operated by other two nodes. Besides, the resulting data obtained by the cooperative vehicle needs to be transmitted to the infrastructure.

Mode 7: The host partitions the entire input data into three files, two of which are offloaded to both the cooperative vehicle and the infrastructure. The host vehicle is also responsible to process the computation task. Thus, the computation can be completed by the three cooperative nodes in a parallel fashion.

As can be seen, V2V and V2I communications build fundamental and key blocks in all the computation modes. Thus, the overall reliability and efficiency in completing the application highly depend on the communication performance in addition to the computing capacity of each node. Based on modeling of vehicle mobility, vehicular communication and computation, we can propose an optimization formulation for each cooperation mode.

4.2 Optimization formulations

4.2.1 Time allocation for computation and communication

Generally, it needs to reserve time for both the computation and communication. The time needed is dependent of the data volume. At this point, we assume that the time allocated to a procedure is linearly proportional to the size of the data assigned to this procedure. Additionally, we introduce two parameters, denoted by $\omega_{\text{computation}}$ and $\omega_{\text{communication}}$, as two weights of time allocations between computation and communication, which are adjustable and reflect the number of time units needed to process per-unit data in the communication and in the computation, respectively. For example, suppose that the data rate of communication is 1 Mbit/s while that of computation is 3 Mbit/s, and suppose that the volume of data to be processed in communication and in computation is equal to 3 Mbit and the totally allowed deadline is 4 s. Logically, the time reserved for communication should be 3 s while that for computation should be 1 s. To realize such time allocations in the example, we can set $\omega_{\text{communication}}=1$ and $\omega_{\text{computation}}=\frac{1}{3}$, and use the following equations to calculate the time allocation

$$\begin{cases} T_1=\left(\frac{\omega_{\text{communication}}D_1}{\omega_{\text{communication}}D_1+\omega_{\text{computation}}D_2}\right)T; \\ T_2=\left(\frac{\omega_{\text{computation}}D_2}{\omega_{\text{communication}}D_1+\omega_{\text{computation}}D_2}\right)T, \end{cases} \tag{30}$$

with $T=4\,\text{s}$, $D_1=D_2=3\,\text{Mbit}$, where T_1 and T_2 denotes the time allocated for communication and computation, respectively, in this example. As we can see, these two weights enable control of time allocation for communication and computation. It is worth pointing out that in practice these two design parameters can be pre-specified according to an actual application situation.

Now, we can turn to Mode 1. We consider to divide the overall deadline T into two parts, $T_{\text{local},i}$ and $T_{\text{V2I},i}$, one of which is allocated for the local execution by vehicle i, and the other is allocated for V2I transmission. As the local execution is required to process the application with the total input data D_{in} while the output data of αD_{in} bits need to be uploaded to the infrastructure, the time allocations for the computation and the V2I communication can be

$$\begin{cases} T_{\text{local},i}=\frac{\omega_{\text{computation}}D_{\text{in}}}{\omega_{\text{computation}}D_{\text{in}}+\omega_{\text{communication}}\alpha D_{\text{in}}}T; \\ T_{\text{V2I},i}=\frac{\omega_{\text{communication}}\alpha D_{\text{in}}}{\omega_{\text{computation}}D_{\text{in}}+\omega_{\text{communication}}\alpha D_{\text{in}}}T. \end{cases} \tag{31}$$

Based on (31), the coupled reliability in Mode 1 is

$$MODE_1(D_{\text{in}})=p_{\text{local},i}\left(D_{\text{in}},T_{\text{local},i}\right)p_{\text{V2I},i}\left(\alpha D_{\text{in}},T_{\text{V2I},i}\right). \tag{32}$$

In Mode 2, vehicle i needs to offload the whole data to the infrastructure and then the infrastructure processes the input data. Indeed, such a situation is a typical offloading scheme. Here, to capture the computing efficiency of the cloud execution, we assume that the time consumed by the cloud to process D_{in}-bit data is characterized by a predefined parameter τ_{cloud}, and it is also reasonable to assume that τ_{cloud} is much smaller than T, $\tau_{\text{cloud}}<T$, due to the fact that the infrastructure has much greater computing power

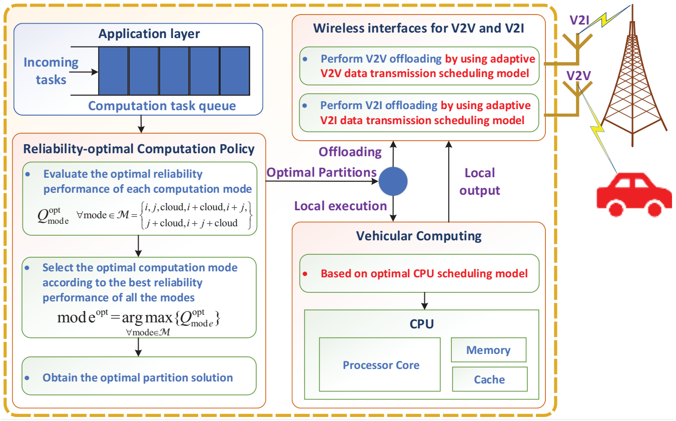

Application layer
Incoming tasks
Computation task queue
Reliability-optimal Computation Policy
Evaluate the optimal reliability performance of each computation mode
$Q_{\mathrm{mode}}^{\mathrm{opt}}\ \forall \mathrm{mode} \in \mathcal{M} = \left\{ \begin{matrix} i, j, \mathrm{cloud}, i+\mathrm{cloud}, i+j, \\ j+\mathrm{cloud}, i+j+\mathrm{cloud} \end{matrix} \right\}$
Select the optimal computation mode according to the best reliability performance of all the modes
$\mathrm{mode}^{\mathrm{opt}} = \arg\max_{\forall \mathrm{mode} \in \mathcal{M}} \{Q_{\mathrm{mode}}^{\mathrm{opt}}\}$
Obtain the optimal partition solution
Wireless interfaces for V2V and V2I
Perform V2V offloading by using adaptive V2V data transmission scheduling model
Perform V2I offloading by using adaptive V2I data transmission scheduling model
V2I
V2V
Offloading
Optimal Partitions
Local execution
Local output
Vehicular Computing
Based on optimal CPU scheduling model
CPU
Processor Core
Memory
Cache

Fig. 9. Structured view of an implementation framework based on the proposed reliability optimization models.

than a single vehicle (See Fig. 1). It is worth pointing out that modeling of an infrastructure's computing capacity depends on many factors, such as the number of computing servers in the pool and the computing resource of the embedded system in each server, etc., which is out of the scope of this paper. Additionally, it should be observed that the model is independent of the infrastructure computing technology, since the operation of the infrastructure computing is abstracted by a single parameter τ_{cloud}. This means that as long as an infrastructure computing situation of interest is well specified, our model can be exploited to facilitate the design and/or analysis of the targeted system. In our consideration, the V2I communication has a significant impact on the whole efficiency of the cloud execution. Thus, the computing reliability is mainly determined by the V2I communication performance. Formally, we express the reliability in Mode 2 as $MODE_2\left(D_{\mathrm{in}}\right) = p_{\mathrm{V2I},i}\left(D_{\mathrm{in}}, T - \tau_{\mathrm{cloud}}\right)$.

In Mode 3, vehicle i first transfers the whole data of D_{in} bits to vehicle j, and then vehicle j takes the responsibility to process the data and to upload the output result to the infrastructure. The durations allocated to the V2V, the local computation, and the V2I processes, $T_{i,j}$, $T_{\mathrm{local},j}$, and $T_{\mathrm{V2I},j}$, can be calculated by

$$\begin{cases} T_{i,j} = \frac{\omega_{communication} D_{in} T}{\omega_{computation} D_{in} + \omega_{communication}(D_{in} + \alpha D_{in})}; \\ T_{\mathrm{local},j} = \frac{\omega_{computation} D_{in} T}{\omega_{computation} D_{in} + \omega_{communication}(D_{in} + \alpha D_{in})}; \\ T_{\mathrm{V2I},j} = \frac{\omega_{communication} \alpha D_{in} T}{\omega_{computation} D_{in} + \omega_{communication}(D_{in} + \alpha D_{in})}. \end{cases} \tag{33}$$

With (33), we can derive the coupled reliability in Mode 3

$$\begin{aligned} MODE_3(D_{in}) =& p_{\mathrm{V2V}}\left(D_{\mathrm{in}}, T_{i,j}\right) \times p_{\mathrm{local},j}\left(D_{\mathrm{in}}, T_{\mathrm{local},j}\right) \\ & \times p_{\mathrm{V2I},j}\left(\alpha D_{\mathrm{in}}, T_{\mathrm{V2I},j}\right) \end{aligned} \tag{34}$$

In the following section, for notational simplicity, we denote by Q_i^{opt}, $Q_{\mathrm{cloud}}^{\mathrm{opt}}$ and Q_j^{opt}, respectively, the reliability performance of the independent computation modes aforementioned, i.e., $Q_i^{\mathrm{opt}} = MODE_1(D_{\mathrm{in}})$, $Q_{\mathrm{cloud}}^{\mathrm{opt}} = MODE_2(D_{\mathrm{in}})$, and $Q_j^{\mathrm{opt}} = MODE_3(D_{\mathrm{in}})$.

4.2.2 Optimization under cooperation modes

Notice that the computation modes from 4 to 7 in Fig. 1 correspond to different cooperations among vehicles i, j and the infrastructure, the optimization formulation of which can be established on the basis of Modes 1, 2 and 3 above. Our goal is to design an optimal data partition to maximize the overall reliability under each cooperative computing mode given the deadline constraint T.

For Mode 4, denoting by $D_{\mathrm{in},i}$ and $D_{\mathrm{in},c}$ the partitioned input data for i's local execution and the cloud execution, respectively, we have

$$\begin{aligned} Q_{i+\mathrm{cloud}}^{\mathrm{opt}} = & \max_{\{D_{\mathrm{in},i}, D_{\mathrm{in},c}\}} : MODE_1(D_{\mathrm{in},i}) MODE_2(D_{\mathrm{in},c}) \\ & s.t. \begin{cases} D_{\mathrm{in},i} + D_{\mathrm{in},c} = D_{\mathrm{in}}; \\ D_{\mathrm{in},i}, D_{\mathrm{in},c} > 0. \end{cases} \end{aligned} \tag{35}$$

For Mode 5, we denote the input data for j's local execution by $D_{\mathrm{in},j}$. Thus, we can get

$$\begin{aligned} Q_{i+j}^{\mathrm{opt}} = & \max_{\{D_{\mathrm{in},i}, D_{\mathrm{in},j}\}} : MODE_1(D_{\mathrm{in},i}) MODE_3(D_{\mathrm{in},j}) \\ & s.t. \begin{cases} D_{\mathrm{in},i} + D_{\mathrm{in},j} = D_{\mathrm{in}}; \\ D_{\mathrm{in},i}, D_{\mathrm{in},j} > 0. \end{cases} \end{aligned} \tag{36}$$

For Mode 6, the optimization model of the input data partition is expressed as

$$\begin{aligned} Q_{j+\mathrm{cloud}}^{\mathrm{opt}} = & \max_{\{D_{\mathrm{in},c}, D_{\mathrm{in},j}\}} : MODE_2(D_{\mathrm{in},c}) MODE_3(D_{\mathrm{in},j}) \\ & s.t. \begin{cases} D_{\mathrm{in},c} + D_{\mathrm{in},j} = D_{\mathrm{in}}; \\ D_{\mathrm{in},c}, D_{\mathrm{in},j} > 0. \end{cases} \end{aligned} \tag{37}$$

Similarly, considering the cooperation among the three computing nodes, we present the optimization model for Mode 7 as follows

$$\begin{aligned} Q_{i+j+\mathrm{cloud}}^{\mathrm{opt}} = & \max_{\{D_{\mathrm{in},i}, D_{\mathrm{in},c}, D_{\mathrm{in},j}\}} \left\{ \begin{aligned} & MODE_1(D_{\mathrm{in},i}) \\ & \times MODE_2(D_{\mathrm{in},c}) \\ & \times MODE_3(D_{\mathrm{in},j}) \end{aligned} \right\} \\ & s.t. \begin{cases} D_{\mathrm{in},i} + D_{\mathrm{in},c} + D_{\mathrm{in},j} = D_{\mathrm{in}}; \\ D_{\mathrm{in},i}, D_{\mathrm{in},c}, D_{\mathrm{in},j} > 0. \end{cases} \end{aligned} \tag{38}$$

4.2.3 Reliability-optimal computing policy

Fig. 9 shows how to implement the reliability-optimal mobile computing in the CVIS based on our proposed optimization models above. We also remark that additional information interaction between vehicles and roadside infrastructure is needed to exchange the vehicle mobility (e.g., the real-time location, speed and acceleration) or channel state information in order to support the evaluation of the optimization models. In practice, the information interaction can be performed by employing a common broadcasting channel such as the DSRC channel 178 for control signaling [34]. In Fig. 9, to obtain the numerical results of the optimal reliability performances under different computing modes, i.e., Q_{mode}^{opt}, $mode \in \mathcal{M} = \{i, j, \mathrm{cloud}, i + \mathrm{cloud}, i + j, j + \mathrm{cloud}, i + j + \mathrm{cloud}\}$, we can resort to many existing numerical constrained optimization algorithms, such as the projected gradient method, the sequential quadratic programming (SQP), etc. In particular, for the cooperative computing modes, we can use a one-dimensional searching

TABLE 4
Parameter settings

α	0.001
$E_i^{\max}, E_j^{\max}$ [J]	0.5
μ_1, μ_2	5, 3
$a_i(0), a_j(0)$, [m/s^2]	0
$v_i(0), v_j(0)$ [km/h]	60,90
$s_i(0), s_j(0)$ [m]	0
$L_{\text{V2V},ij}, L_0, L_{\text{V2I},i}$ [m]	150, 300, 400
$\frac{\tau_{\text{cloud}}}{T}$	0.5
$\omega_{\text{communication}}, \omega_{\text{computation}}$	1

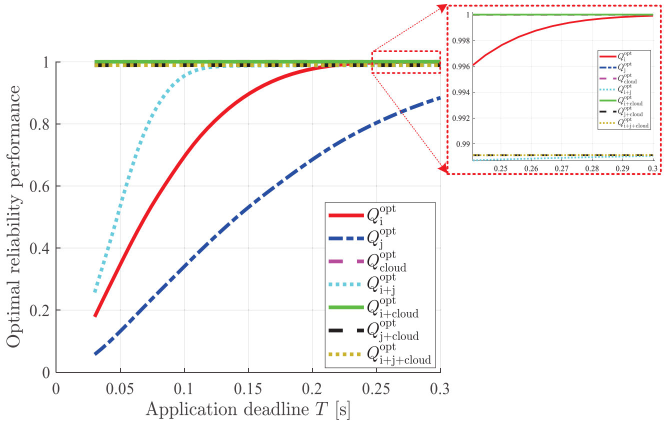


Fig. 10. The optimal reliability performance of each computation mode under different deadline requirements.

technique to solve (35), (36), and (37), and use a two-dimensional searching to solve (38), since the equality constraints in (35), (36), and (37) involve only two decision variables, and the equality constraint in (38) contains tree variables. After that, we can select the best computing mode to implement the computation partition and offloading, i.e., choosing $\text{mode}^{\text{opt}} = \text{argmax}_{\text{mode}\in\mathcal{M}}\{Q_{\text{mode}}^{\text{opt}}\}$. The vehicle can perform the local execution by using the optimal CPU clock-frequency scheduling proposed in Theorem 3. Besides, the vehicle adopts the adaptive transmission mechanism given in the subsection 3.4.1 to implement the V2V-based computation offloading, while it can use the optimal V2I transmission scheduling mechanism in the subsection 3.5.3 to support the V2I-based computation offloading.

5 Performance Evaluation

In this section, we demonstrate how the key factors related to the application profile, the mobility environment, and the time allocation for communication and computation affect the performance of different computation modes.

5.1 Parameter settings

The parameter settings given in Table 4 are employed for configuration of local execution and remote execution as well as for initializing vehicles' kinematic parameters and locations in our experiments. Additionally, with respect to the V2V and V2I communication configurations, we refer to the parameter settings given in Tables 1, 2 and 3. If not specially stated, the parameter settings in Tables 1 to 4 are adopted throughout the simulation experiments.

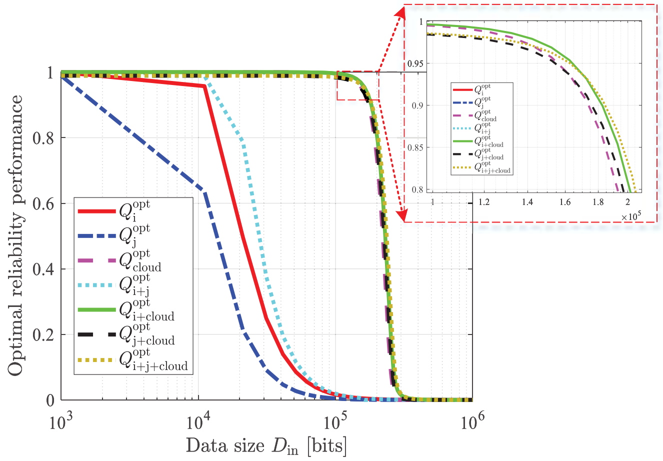


Fig. 11. The optimal reliability performance of each computation mode under different input data sizes.

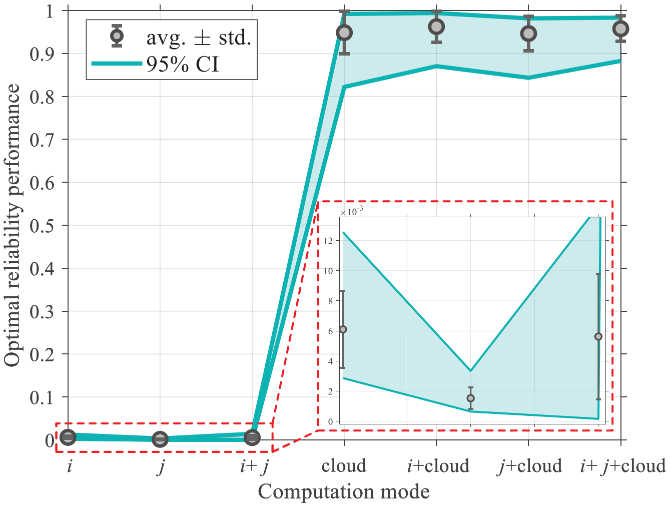


Fig. 12. Optimal reliability performance of each computation mode in a stochastic application situation where the computation demand D_{in} is generated by following a normal distribution with the mean at 1.5×10^5 bits and the standard deviation at 2.0×10^4 bits, i.e., $D_{\text{in}} \sim \mathcal{N}\left(1.5 \times 10^5, 4.0 \times 10^8\right)$, and $T = 0.6$s.

5.2 Effects of Application Profile

To show how a vehicular user can benefit from the cooperative computation, we compare the optimal reliability performance (i.e., the maximum success probability of application completion) under different cooperative computation models, i.e., Q_{i+j}^{opt}, $Q_{i+\text{cloud}}^{\text{opt}}$, $Q_{j+\text{cloud}}^{\text{opt}}$ and $Q_{i+j+\text{cloud}}^{\text{opt}}$, with those of the independent computation models, Q_i^{opt}, Q_j^{opt} and $Q_{\text{cloud}}^{\text{opt}}$, under different application profiles, (D_{in}, T). First, we fix the input data size of the application at $D_{\text{in}} = 5000$ bit and vary the deadline T from 0.03 s to 0.3 s. The optimal reliability performance of each computation mode obtained at each point of T is illustrated in Fig. 10. It can be seen that the reliability of every computation mode increases as the allowed deadline becomes longer. However, when given a relatively short deadline, for instance, $T < 0.1$ s, the local execution independently by vehicle i or j and the cooperative computation by both the vehicles are more

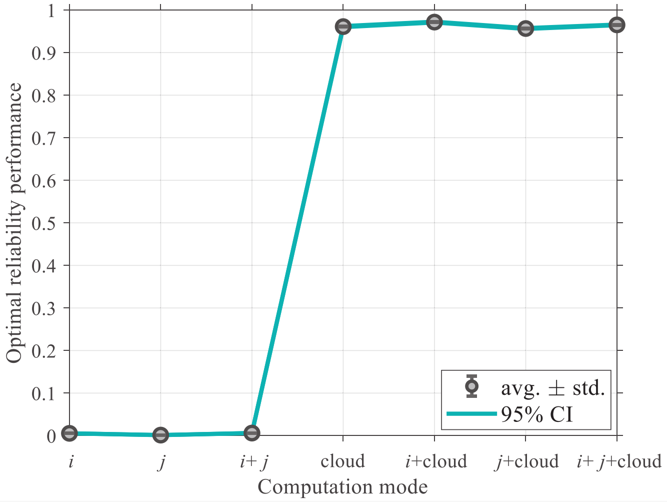
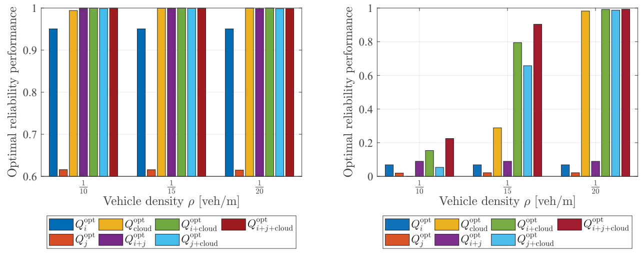


Fig. 13. Optimal reliability performance of each computation mode in a stochastic application situation where the computation demand D_{in} is generated by following a Poisson distribution with the mean at 1.5×10^5 bits, i.e., $D_{\text{in}} \sim \text{Pois}\left(1.5 \times 10^5\right)$, and $T = 0.6\text{s}$.



(a) With $D_{\text{in}} = 1 \times 10^4$ bit

(b) With $D_{\text{in}} = 5 \times 10^4$ bit

Fig. 14. Optimal reliability performance of each computation mode under different vehicle densities.

likely to fail to complete processing the application by the deadline. The main reason is that the constrained computation capacity of each vehicle cannot fully support the execution of the application within a short period. Besides, even though deploying the optimal cooperation between i and j, the reliability performance Q_{i+j}^{opt} is not satisfactory, which is due to the fact that a part of time has to be spent to achieve the data transmission from i to j through V2V communication. After V2V communication, less time is left for the other cooperator, j, to complete its assigned computation task. But, the $i+j$-based cooperative computation can still perform better than the independent computation mode by i or by j when $T < 0.3\,\text{s}$. By contrast, using the cooperative computation modes between the vehicles and the infrastructure can achieve a higher reliability. In particular, the cooperation mode of $i + \text{cloud}$ can better benefit the application execution under most of the deadline requirements as shown in Fig. 10, the reliability of which is almost 100%. Besides, the independent cloud computation mode, cloud, also stays at a relatively high reliability level, since the V2I communication can always guarantee the offloading of 5000-bit data with a high reliability even when the allowed deadline is short, and the V2I communication reliability dominates the whole reliability performance in the cloud computation situation.

In Fig. 11, we fix the total deadline at $T = 0.6\,\text{s}$ while varying the input data size D_{in} from 1×10^3 bit to 1×10^6 bit. As expected, when the input data size increases, the probability of application completion decreases. Nevertheless, if D_{in} ranges between 1×10^4 bit and 1×10^5 bit, the cooperation computation modes, $i+\text{cloud}$, $j+\text{cloud}$ and $i+j+\text{cloud}$, and the cloud computation mode cloud can complete the application with a sufficiently high reliability (e.g., above 95% even if $D_{\text{in}} = 1 \times 10^5$ bit as shown in the sub-figure in Fig. 11), while the others including i, j and $i+j$ can no longer reliably complete the application. In addition, the cooperation mode $i + \text{cloud}$ can perform best among the computation modes under most of D_{in} settings. It is also interesting to observe that the reliability performance of the cooperation mode $i + j + \text{cloud}$ degrades slightly slower than that of $i + \text{cloud}$ when $D_{\text{in}} > 1.7 \times 10^5$ bit. This is because the degradation in the V2V communication reliability is smaller than that in the computation reliability as D_{in} increases.

Furthermore, to show the effectiveness of the cooperative vehicle-infrastructure computing approach, we have also conducted extensive Monte Carlo simulations based on the different computing modes. Specifically, we simulate two typical types of stochastic application situations, where the application input size, D_{in}, is generated by following a normal distribution and a Poisson distribution, respectively. Each of the computing modes has been performed with 1000 replications per stochastic application situation, and the results are shown with the average plus and minus the standard deviation (i.e., avg.±std.) and with the 95% confidence intervals (CI) as in Figs. 12 and 13. It can be seen that the cooperative computing modes with assistance of the cloud computing perform better than those independent modes on average in both the stochastic situations. From Fig. 12, the lower bound of the 95% CI of the cloud-assisted cooperative computing modes is higher than 0.82, and their average performance is around 0.95. By comparison, the performance of the other modes only based on vehicular computing is close to zero. These results indicate that the cooperation between cloud and vehicular computing can better benefit the system in terms of the reliability performance even in a stochastic application situation. The similar fact can also be confirmed by Fig. 13 where the lower bound of the 95% CI of the cloud-assisted cooperative computing modes is higher than 0.95. Interestingly, when comparing Fig. 12 with Fig. 13, we can find that the standard deviations in Fig. 13 are much smaller than those in Fig. 12. The main reason is that the variance of the random application input following the Poisson distribution is smaller than that following the normal distribution. The 95% CIs of the cloud-assisted cooperative computing modes shown in Fig. 13 are also much narrower than those shown in Fig. 12, indicating that these cooperative computing modes can perform in a more deterministic manner even when the computation demand follows the Poisson distribution pattern.

5.3 Effects of Mobility Environment

We examine the effects of the vehicle density in the mobility environment on the optimal reliability performance of each

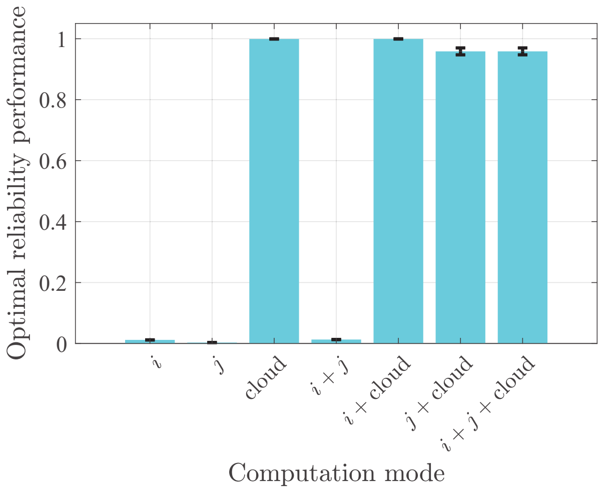
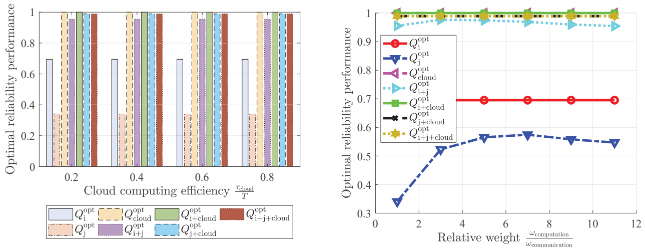


Fig. 15. Optimal reliability performance of each computation mode with different initial inter-vehicle distances and infrastructure positions.

computation mode. Two specific situations are considered, in one of which the application input data size is set to $D_{\text{in}} = 1 \times 10^4$ bit and in the other $D_{\text{in}} = 5 \times 10^4$ bit. In both the situations, we fix the deadline at $T = 0.5$ s, and test all the computation modes under three vehicle density settings, i.e., $\rho = \frac{1}{10}, \frac{1}{15}, \frac{1}{20}$ veh/m. In addition, with respect to the vehicle density within the coverage of the infrastructure, we set $\rho_c = \rho^2$. From Fig. 14a, except for the mode j, almost computation modes can achieve a high probability of application completion by the deadline with these vehicle density settings. This is because vehicle j has less time to execute the whole application after receiving the data via the V2V communication. In addition, compared to the independent computation modes, the cooperative computation modes can perform better especially when the vehicle density is high, e.g., $\rho = \frac{1}{10}$ veh/m.

From Fig. 14b, it can be seen that the reliability performance of each computation mode increases with reducing the vehicle density. This is because reducing the vehicle density alleviates the contention for V2V and V2I channels, which further improves the V2V and V2I communication reliability. In this figure, the performance enhancement by the cooperation among the vehicles and the infrastructure is more obvious. With $\rho = \frac{1}{15}$ veh/m, the cooperative computation mode $i + j + \text{cloud}$ can still guarantee a probability of about 90% to complete the application, while the reliability of the other modes is below 80%.

To further demonstrate the effects of the initial inter-vehicle distance and the infrastructure position, we set $D_{\text{in}} = 1 \times 10^5$ bit and $T = 0.5$ s. The vehicle density is set to $\rho = \frac{1}{50}$ veh/m to simulate a low-density mobility environment. The initial inter-vehicle distance, $L_{\text{V2V},ij}$, ranges from 10 m to 300 m, and the relative vertical distance between the infrastructure and the road, L_0, ranges from 100 m to 1000 m. 50 samples on $L_{\text{V2V},ij}$ and 20 samples on L_0 are generated. The computation modes are tested with each combination of (D_{in}, L_0). The average result of each computation mode is shown with the associated standard deviation in Fig. 15. As can be seen, most of cooperative computation modes with the support of cloud computation, except for $i + j$, can provide a high reliability performance for the application execution, whose results are above 90% on average. The high reliability achieved by the cloud computation mode implies that L_0, even ranging within $[100, 1000]$m, has a negligible impairment on the V2I data transmission. This is due to the fact that the V2I link achieves a good throughput in the low-density situation. As indicated by the error bars in Fig. 15, the cooperative computation modes, $j + \text{cloud}$ and $i + j + \text{cloud}$, experience a larger standard deviation, since their processes involve the V2V communication. Compared with the V2I communication, the V2V communication is more significantly affected by varying the inter-vehicle distance.



(a) Impact of τ_{cloud}

(b) Impact of $\frac{\omega_{\text{computation}}}{\omega_{\text{communication}}}$

Fig. 16. Optimal reliability performance of each computation mode with different settings on the cloud computation time and on the weights for computation and communication.

5.4 Effects of Time Allocation for Communication and Computation

In our model, the efficiency of the cloud computation is characterized by the parameter τ_{cloud}. Namely, the larger τ_{cloud} is, the larger time consumption the cloud computation accounts for. Indeed, given a τ_{cloud}, the time reserved for V2I transmission is $T - \tau_{\text{cloud}}$. To show the impact of τ_{cloud}, we set τ_{cloud} to $0.2T$, $0.4T$, $0.6T$ and $0.8T$ in the respective experiments. In the experiments, D_{in} and T are fixed as $D_{\text{in}} = 5000$ bit and $T = 0.1$ s. The vehicle density is $\rho = \frac{1}{30}$ veh/m ($\rho_c = \frac{1}{30^2}$ veh/m^2). The optimal reliability performance of each computation mode is shown in Fig. 16a. Logically, those modes without involving the cloud computation, i, j and $i + j$, are not affected by the variation of τ_{cloud}. As Fig. 16a shows, the influence of the variance of τ_{cloud} on the cooperative computation modes, $i + \text{cloud}$, $j + \text{cloud}$ and $i + j + \text{cloud}$, and on the cloud computation mode, cloud, are also negligible. The main reason is that even though given a short V2I transmission period, for instance, $0.2T$ when $\tau_{\text{cloud}} = 0.8T$, the V2I link can guarantee transferring 5000-bit data from the vehicle to the infrastructure with almost 100% non-outage probability.

Additionally, we fix τ_{cloud} at $0.5T$. The impacts of the weights, $\omega_{\text{computation}}$ and $\omega_{\text{communication}}$, which are used to determine the time allocation for computation and communication processes, are shown in Fig. 16b. When the ratio $\omega_{\text{computation}}/\omega_{\text{computation}}$ increases, the computation modes, j and $i + j$, experience more obvious variance in their reliability performance than the other modes. It is because increasing $\omega_{\text{computation}}$ can lead to an increase of time period reserved for the vehicular computation by j. When there is no support of the infrastructure's cooperation, the overall performance of j or $i + j$ heavily relies on the computation reliability in j's execution. It can also be found that there is a performance gap between the computation mode i and j even when their computation capacity is identical. This is because the implementation of the computation mode j involves the V2V transmission, a process that requires certain time consumption. As a consequence, the time reserved for vehicle j-based local execution is always shorter than that for vehicle i-based local execution. By comparison, the cooperative computation modes and the cloud computation, with above 95% probability of application completion, can provide higher reliability than the independent local computation modes i and j even when the weight ratio increases from 1 to 11.

6 Conclusion and Future Work

We have proposed a theoretically analytical framework for cooperative computation optimization in a cooperative vehicle-infrastructure system, which takes into account modeling and optimization of vehicular communication and computation. With respect to modeling, we have presented some stochastic analysis to evaluate the reliability performance of data transmissions over the dynamic V2V and V2I links with consideration of the vehicle mobility, channel contentions and channel fading. Moreover, we have modeled the vehicular computation and obtained a closed-form optimal solution for CPU scheduling in terms of minimizing the application execution time. The computation reliability has been formulated by taking into consideration both the computation capacity and the application requirements. With respect to optimization, we consider that a vehicular distributed application can be partitioned into multiple parallel computation tasks and transmitted to different computing nodes for cooperative execution. Several optimization models have been developed to maximize the coupled reliability of communication and computation processes by optimizing the data partitions. Numerical results are also provided to demonstrate the impacts of many factors including communication, computation and application profile on the optimal reliability performance of different cooperative computation modes. The comparative results confirm the strength of reliability-optimal cooperative computation in support of distributed application execution.

Notably, our theoretical development and numerical results can motivate how to leverage the power of vehicular networking and communication to benefit the deployment of a reliable CVIS. The results in this paper can also be utilized to develop optimal admission control and optimal cooperator selection strategies for efficient and reliable cooperative computation applications. Another research direction is to extend the cooperative computation optimization paradigm to more complex cooperation situations where multi-hop routing should be carefully modeled and integrated into the communication and computation processes.

Acknowledgments

This research was supported in part by the National Natural Science Foundation of China under Grant Nos. 61672082 and 61822101, Beijing Municipal Natural Science Foundation No.4181002, the scholarship under the State Scholarship Fund (No. 201706020112), Jiangsu Province Collaborative Innovation Center of Modern Urban Traffic Technologies, Asa Briggs Visiting Fellowship from University of Sussex, Royal Society-Newton Mobility Grant (IE160920) and The Engineering, and Physical Sciences Research Council (EPSRC) (EP/P025862/1).

References

[1] M. Whaiduzzaman, M. Sookhak, A. Gani, and R. Buyya, "A survey on vehicular cloud computing," *Journal of Network and Computer Applications*, vol. 40, pp. 325 – 344, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804513001793

[2] D. Tian, J. Zhou, Y. Wang, Y. Lu, H. Xia, and Z. Yi, "A dynamic and self-adaptive network selection method for multimode communications in heterogeneous vehicular telematics," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 3033–3049, Dec 2015.

[3] T. Zahn, G. O'Shea, and A. Rowstron, "Feasibility of content dissemination between devices in moving vehicles," in *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies*, ser. CoNEXT '09. New York, NY, USA: ACM, 2009, pp. 97–108. [Online]. Available: http://doi.acm.org/10.1145/1658939.1658951

[4] R. He, A. F. Molisch, F. Tufvesson, Z. Zhong, B. Ai, and T. Zhang, "Vehicle-to-vehicle propagation models with large vehicle obstructions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2237–2248, Oct 2014.

[5] Y. Cao, H. Zhang, D. Wu, and D. Yuan, "Ogcmac: A novel ofdm based group contention mac for vanet control channel," *IEEE Transactions on Wireless Communications*, vol. 16, no. 9, pp. 5796–5809, Sept 2017.

[6] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for vm-based cloudlets in mobile computing," *IEEE Pervasive Computing*, vol. 8, no. 4, pp. 14–23, Oct 2009.

[7] X. Zhang, A. Kunjithapatham, S. Jeong, and S. Gibbs, "Towards an elastic application model for augmenting the computing capabilities of mobile devices with cloud computing," *Mobile Networks and Applications*, vol. 16, no. 3, pp. 270–284, Jun 2011. [Online]. Available: https://doi.org/10.1007/s11036-011-0305-7

[8] B.-G. Chun and P. Maniatis, "Augmented smartphone applications through clone cloud execution." in *HotOS*, vol. 9, 2009, pp. 8–11.

[9] X. Sun and N. Ansari, "Edgeiot: Mobile edge computing for the internet of things," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 22–29, December 2016.

[10] Y. Sahni, J. Cao, S. Zhang, and L. Yang, "Edge mesh: A new paradigm to enable distributed intelligence in internet of things," *IEEE Access*, vol. 5, pp. 16 441–16 458, 2017.

[11] W. Yu, F. Liang, X. He, W. G. Hatcher, C. Lu, J. Lin, and X. Yang, "A survey on the edge computing for the internet of things," *IEEE Access*, vol. PP, no. 99, pp. 1–1, 2017.

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

[13] A. Alnoman, G. H. S. Carvalho, A. Anpalagan, and I. Woungang, "Energy efficiency on fully cloudified mobile networks: Survey, challenges, and open issues," *IEEE Communications Surveys Tutorials*, vol. PP, no. 99, pp. 1–1, 2017.

[14] Y. Xu and S. Mao, "A survey of mobile cloud computing for rich media applications," *IEEE Wireless Communications*, vol. 20, no. 3, pp. 46–53, June 2013.

[15] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, pp. 1628–1656, thirdquarter 2017.

[16] C. Xian, Y.-H. Lu, and Z. Li, "Adaptive computation offloading for energy conservation on battery-powered systems," in *2007 International Conference on Parallel and Distributed Systems*, vol. 2, Dec 2007, pp. 1–8.

[17] J. Liu, B. Priyantha, T. Hart, Y. Jin, W. Lee, V. Raghunathan, H. S. Ramos, and Q. Wang, "Co-gps: Energy efficient gps sensing with cloud offloading," *IEEE Transactions on Mobile Computing*, vol. 15, no. 6, pp. 1348–1361, June 2016.

[18] D. Huang, P. Wang, and D. Niyato, “A dynamic offloading algorithm for mobile computing,” *IEEE Transactions on Wireless Communications*, vol. 11, no. 6, pp. 1991–1995, June 2012.

[19] Y. Wen, W. Zhang, and H. Luo, “Energy-optimal mobile application execution: Taming resource-poor mobile devices with cloud clones,” in *2012 Proceedings IEEE INFOCOM*, March 2012, pp. 2716–2720.

[20] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. O. Wu, “Energy-optimal mobile cloud computing under stochastic wireless channel,” *IEEE Transactions on Wireless Communications*, vol. 12, no. 9, pp. 4569–4581, September 2013.

[21] Z. Sheng, C. Mahapatra, V. Leung, M. Chen, and P. Sahu, “Energy efficient cooperative computing in mobile wireless sensor networks,” *IEEE Transactions on Cloud Computing*, vol. PP, no. 99, pp. 1–1, 2015.

[22] X. Chen, L. Jiao, W. Li, and X. Fu, “Efficient multi-user computation offloading for mobile-edge cloud computing,” *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795–2808, October 2016.

[23] S. K. Datta, J. Haerri, C. Bonnet, and R. F. D. Costa, “Vehicles as connected resources: Opportunities and challenges for the future,” *IEEE Vehicular Technology Magazine*, vol. 12, no. 2, pp. 26–35, June 2017.

[24] C. Wang, Y. Li, D. Jin, and S. Chen, “On the serviceability of mobile vehicular cloudlets in a large-scale urban environment,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 10, pp. 2960–2970, Oct 2016.

[25] 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 Communications Magazine*, vol. 55, no. 12, pp. 60–67, DECEMBER 2017.

[26] M. Sookhak, F. R. Yu, Y. He, H. Talebian, N. S. Safa, N. Zhao, M. K. Khan, and N. Kumar, “Fog vehicular computing: Augmentation of fog computing using vehicular cloud computing,” *IEEE Vehicular Technology Magazine*, vol. 12, no. 3, pp. 55–64, Sept 2017.

[27] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, “Vehicular fog computing: A viewpoint of vehicles as the infrastructures,” *IEEE Transactions on Vehicular Technology*, vol. 65, no. 6, pp. 3860–3873, June 2016.

[28] Y. Zhang, D. Niyato, and P. Wang, “Offloading in mobile cloudlet systems with intermittent connectivity,” *IEEE Transactions on Mobile Computing*, vol. 14, no. 12, pp. 2516–2529, Dec 2015.

[29] A. Ashok, P. Steenkiste, and F. Bai, “Vehicular cloud computing through dynamic computation offloading,” *Computer Communications*, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140366417304140

[30] O. Trullols-Cruces, M. Fiore, and J. Barcelo-Ordinas, “Cooperative download in vehicular environments,” *IEEE Transactions on Mobile Computing*, vol. 11, no. 4, pp. 663–678, April 2012.

[31] K. Zhang, Y. Mao, S. Leng, A. Vinel, and Y. Zhang, “Delay constrained offloading for mobile edge computing in cloud-enabled vehicular networks,” in *2016 8th International Workshop on Resilient Networks Design and Modeling (RNDM)*, Sept 2016, pp. 288–294.

[32] K. Zhang, Y. Mao, S. Leng, Y. He, and Y. ZHANG, “Mobile-edge computing for vehicular networks: A promising network paradigm with predictive off-loading,” *IEEE Vehicular Technology Magazine*, vol. 12, no. 2, pp. 36–44, June 2017.

[33] Z. Jiang, S. Zhou, X. Guo, and Z. Niu, “Task replication for deadline-constrained vehicular cloud computing: Optimal policy, performance analysis and implications on road traffic,” *IEEE Internet of Things Journal*, vol. PP, no. 99, pp. 1–1, 2017.

[34] D. Niyato, E. Hossain, and P. Wang, “Optimal channel access management with qos support for cognitive vehicular networks,” *IEEE Transactions on Mobile Computing*, vol. 10, no. 4, pp. 573–591, April 2011.

[35] B. Heintz, A. Chandra, R. K. Sitaraman, and J. Weissman, “End-to-end optimization for geo-distributed mapreduce,” *IEEE Transactions on Cloud Computing*, vol. 4, no. 3, pp. 293–306, July 2016.

[36] D. Jiang and L. Delgrossi, “Ieee 802.11p: Towards an international standard for wireless access in vehicular environments,” in *VTC Spring 2008 - IEEE Vehicular Technology Conference*, May 2008, pp. 2036–2040.

[37] L. Cheng, B. E. Henty, D. D. Stancil, F. Bai, and P. Mudalige, “Mobile vehicle-to-vehicle narrow-band channel measurement and characterization of the 5.9 ghz dedicated short range communication (dsrc) frequency band,” *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 8, pp. 1501–1516, Oct 2007.

[38] Q. Liu, S. Zhou, and G. B. Giannakis, “Cross-layer combining of adaptive modulation and coding with truncated arq over wireless links,” *IEEE Transactions on Wireless Communications*, vol. 3, no. 5, pp. 1746–1755, Sept 2004.

[39] F. Bai and B. Krishnamachari, “Spatio-temporal variations of vehicle traffic in vanets: Facts and implications,” in *Proceedings of the Sixth ACM International Workshop on VehicuLAr InterNETworking*, ser. VANET ’09. New York, NY, USA: ACM, 2009, pp. 43–52. [Online]. Available: http://doi.acm.org/10.1145/1614269.1614278

[40] D. Jiang, Q. Chen, and L. Delgrossi, “Optimal data rate selection for vehicle safety communications,” in *Proceedings of the Fifth ACM International Workshop on VehicularAr Inter-NETworking*, ser. VANET ’08. New York, NY, USA: ACM, 2008, pp. 30–38. [Online]. Available: http://doi.acm.org/10.1145/1410043.1410050

[41] J. J. Haas, Y.-C. Hu, and K. P. Laberteaux, “Real-world vanet security protocol performance,” in *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*. IEEE, 2009, pp. 1–7.

[42] T. H. Luan, X. S. Shen, and F. Bai, “Integrity-oriented content transmission in highway vehicular ad hoc networks,” in *INFOCOM, 2013 Proceedings IEEE*. IEEE, 2013, pp. 2562–2570.

[43] B. Sklar, “Rayleigh fading channels in mobile digital communication systems. i. characterization,” *IEEE Communications Magazine*, vol. 35, no. 9, pp. 136–146, Sep 1997.

[44] ——, *Digital communications*. Prentice Hall Upper Saddle River, 2001, vol. 2.

[45] J. R. Lorch and A. J. Smith, “Improving dynamic voltage scaling algorithms with pace,” in *Proceedings of the 2001 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, ser. SIGMETRICS ’01. New York, NY, USA: ACM, 2001, pp. 50–61. [Online]. Available: http://doi.acm.org/10.1145/378420.378429

[46] A. P. Miettinen and J. K. Nurminen, “Energy efficiency of mobile clients in cloud computing,” in *Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing*, ser. HotCloud’10. Berkeley, CA, USA: USENIX Association, 2010, pp. 4–4. [Online]. Available: http://dl.acm.org/citation.cfm?id=1863103.1863107

[47] W. Yuan and K. Nahrstedt, “Energy-efficient cpu scheduling for multimedia applications,” *ACM Trans. Comput. Syst.*, vol. 24, no. 3, pp. 292–331, Aug. 2006. [Online]. Available: http://doi.acm.org/10.1145/1151690.1151693

[48] J. M. Rabaey, A. P. Chandrakasan, and B. Nikolic, *Digital integrated circuits*. Prentice hall Englewood Cliffs, 2002, vol. 2.

[49] T. D. Burd and R. W. Brodersen, “Processor design for portable systems,” *Journal of VLSI signal processing systems for signal, image and video technology*, vol. 13, no. 2-3, pp. 203–221, 1996.

Jianshan Zhou received the B.Sc. and M.Sc. degrees in traffic information engineering and control in 2013 and 2016, respectively. He is currently working towards the Ph.D. degree with the School of Transportation Science and Engineering, Beihang University, Beijing, China. His current research interests are focused on wireless communication, artificial intelligent system, and intelligent transportation systems.

Daxin Tian [M’13, SM’16] is an associate professor in 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 intelligent.

Yunpeng Wang is a professor in the School of Transportation Science and Engineering, Beihang University, Beijing, China. His current research interests include intelligent transportation systems, traffic safety, and vehicle infrastructure integration.

Zhengguo Sheng is currently a Lecturer with the Department of Engineering and Design, University of Sussex, U.K. He has authored over 50 international conference and journal papers. His current research interests cover IoT/M2M, vehicular communications, and edge/cloud computing.

Xuting Duan is currently an assistant professor with the School of Transportation Science and Engineering, Beihang University, Beijing, China. His current research interests are focused on vehicular ad hoc networks.

Victor C. M. Leung [S'75, M'89, SM'97, F'03] received the B.A.Sc. (Hons.) degree in electrical engineering from the University of British Columbia (UBC) in 1977, and was awarded the APEBC Gold Medal as the head of the graduating class in the Faculty of Applied Science. He attended graduate school at UBC on a Canadian Natural Sciences and Engineering Research Council Postgraduate Scholarship and received the Ph.D. degree in electrical engineering in 1982.

From 1981 to 1987, Dr. Leung was a Senior Member of Technical Staff and satellite system specialist at MPR Teltech Ltd., Canada. In 1988, he was a Lecturer in the Department of Electronics at the Chinese University of Hong Kong. He returned to UBC as a faculty member in 1989, and currently holds the positions of Professor and TELUS Mobility Research Chair in Advanced Telecommunications Engineering in the Department of Electrical and Computer Engineering. Dr. Leung has co-authored more than 1000 journal/conference papers, 37 book chapters, and co-edited 12 book titles. Several of his papers had been selected for best paper awards. His research interests are in the broad areas of wireless networks and mobile systems.

Dr. Leung is a registered Professional Engineer in the Province of British Columbia, Canada. He is a Fellow of IEEE, the Royal Society of Canada, the Engineering Institute of Canada, and the Canadian Academy of Engineering. He was a Distinguished Lecturer of the IEEE Communications Society. He is serving on the editorial boards of the IEEE Wireless Communications Letters, IEEE Transactions on Green Communications and Networking, IEEE Access, Computer Communications, and several other journals, and has previously served on the editorial boards of the IEEE Journal on Selected Areas in Communications Wireless Communications Series and Series on Green Communications and Networking, IEEE Transactions on Wireless Communications, IEEE Transactions on Vehicular Technology, IEEE Transactions on Computers, and Journal of Communications and Networks. He has guest-edited many journal special issues, and provided leadership to the organizing committees and technical program committees of numerous conferences and workshops. He received the IEEE Vancouver Section Centennial Award and 2011 UBC Killam Research Prize. He is the recipient of the 2017 Canadian Award for Telecommunications Research. He is a co-author of the paper that has won the 2017 IEEE ComSoc Fred W. Ellersick Prize.

Supplementary Material for Reliability-Optimal Cooperative Communication and Computing in Connected Vehicle Systems

Jianshan Zhou, Daxin Tian, *Senior Member, IEEE,* Yunpeng Wang, Zhengguo Sheng, Xuting Duan, and Victor C.M. Leung, *Fellow, IEEE*

1 Proof of Theorem 1

Based on the definition of the variance, we have $\mathbb{V}\left[\theta(T_{i,j})\right] = \mathbb{E}\left[\theta(T_{i,j}) - \mathbb{E}[\theta(T_{i,j})]\right]^2$, which can be rearranged as follows

$$\mathbb{V}\left[\theta(T_{i,j})\right] = \mathbb{E}\left[\theta^2(T_{i,j})\right] - \left(\mathbb{E}[\theta(T_{i,j})]\right)^2. \tag{S.1}$$

Using the Cauchy-Schwarz inequality, we can further obtain

$$\begin{aligned}\theta^2(T_{i,j}) &= \left(\sum_{k=1}^{T_{i,j}} \Omega(C,N)\Delta\tau\right)^2 \\ &\leq \left(\sum_{k=1}^{T_{i,j}} (\sqrt{\Delta\tau})^2\right)\left(\sum_{k=1}^{T_{i,j}} \Omega^2(C,N)(\sqrt{\Delta\tau})^2\right) \\ &= T_{i,j}\Delta\tau \sum_{k=1}^{T_{i,j}} \Omega^2(C,N)\Delta\tau.\end{aligned} \tag{S.2}$$

Combining (S.1) and (S.2) immediately results in (13).

2 Proof of Lemma 2

Letting $\Lambda = [\lambda_k \geq 0, k = 1, 2, \ldots, T_{\text{V2I},l}]^{\text{T}}$ and $\delta \in \mathbb{R}$ denote Lagrange multipliers, we construct a generalized Lagrange function for model (19) as follows

$$\begin{aligned}\mathcal{L}(\boldsymbol{u}_l, \Lambda, \delta) =& G\left(\boldsymbol{u}_l\right) - \sum_{k=1}^{T_{\text{V2I},l}} \lambda_k u_l(k) \\ &- \delta\left(\sum_{k=1}^{T_{\text{V2I},l}} u_l(k) - D_{\text{out},l}\right).\end{aligned} \tag{S.3}$$

Based on the the Karush-Kuhn-Tucker (KKT) theorem, the optimal feasible solution satisfies the complementary slackness. That is, we can have $\lambda_k u_l(k) = 0$ for all $u_l(k) \in \boldsymbol{u}_l^{\text{opt}}$. Therefore, it can be observed that for any nonzero $u_l(k) \in \boldsymbol{u}_l^{\text{opt}}$, $u_l(k) > 0$, it must hold that $\lambda_k = 0$, while for any zero $u_l(k^*) \in \boldsymbol{u}_l^{\text{opt}}$, $u_l(k^*) = 0$, $\lambda_{k^*} \geq 0$ always holds as the given condition.

Moreover, the KKT theorem also guarantees that the optimal feasible solution $\boldsymbol{u}_l^{\text{opt}}$ satisfies the gradient condition, i.e., $\nabla_{\boldsymbol{u}_l}\mathcal{L}(\boldsymbol{u}_l^{\text{opt}}, \Lambda, \delta) = \nabla_{\boldsymbol{u}_l}G(\boldsymbol{u}_l^{\text{opt}}) - \Lambda - \delta\mathbf{1} = \mathbf{0}$, where $\mathbf{1}$ and $\mathbf{0}$ are column vectors whose entries are all equal to 1 and 0, respectively. Accordingly, for each $u_k(k) \in \boldsymbol{u}_l^{\text{opt}}$, we have $\partial G(\boldsymbol{u}_l^{\text{opt}})/\partial u_l(k) = \lambda_k + \delta$. For any nonzero $u_l(k) > 0$, $\partial G(\boldsymbol{u}_l^{\text{opt}})/\partial u_l(k) = \delta$ since $\lambda_k = 0$. For any zero $u_l(k^*) = 0$, $\partial G(\boldsymbol{u}_l^{\text{opt}})/\partial u_l(k^*) = \lambda_{k^*} + \delta \geq \delta$ since $\lambda_{k^*} \geq 0$. Thus, the lemma has been proven, which indicates that all the partial derivatives associated with the nonzero $u_l(k)$ are identical and not larger than those with zero $u_l(k^*)$.

3 Proof of Theorem 2

Let $q = M/(B\Delta\tau)$ for national simplicity. Based on Lemma 2, we can further get

$$\delta = \frac{\partial G(\boldsymbol{u}_l^{\text{opt}})}{\partial u_l(k)} = d_l^\beta(k) 2^{u_l(k)q} q \ln 2 \tag{S.4}$$

for all $u_l(k) > 0$, which indicates

$$u_l(k) = \frac{1}{q}\left(\log_2(\delta) - \log_2\left(d_l^\beta(k)\right) - \log_2\left(q\ln 2\right)\right). \tag{S.5}$$

and the summation $\sum_{k=1}^{T_{\text{V2I},l}} d_l^\beta(k) 2^{u_l(k)q} = \sum_{k=1}^{T_{\text{V2I},l}} \left(\delta/(q\ln 2)\right) = T_{\text{V2I},l}\delta/(q\ln 2)$. Hence, we can further rearrange the overall probability of successful V2I data transmission as

$$H^{\text{opt}}\left(D_{\text{out},l}, T_{\text{V2I},l}; M\right) = \exp\left(-\frac{T_{\text{V2I},l}\delta}{q\ln 2 w_l} + \frac{\sum_{k=1}^{T_{\text{V2I},l}} d_l^\beta(k)}{w_l}\right) \tag{S.6}$$

On the other hand, applying the multiple products to (S.4) can yield

$$\delta^{T_{\text{V2I},l}} = \left(\prod_{k=1}^{T_{\text{V2I},l}} d_l^\beta(k)\right) 2^{q\sum_{k=1}^{T_{\text{V2I},l}} u_l(k)} \left(q\ln 2\right)^{T_{\text{V2I},l}}. \tag{S.7}$$

Noting $\sum_{k=1}^{T_{\text{V2I},l}} u_l(k) = D_{\text{out},l}$, we get

$$\delta = \left(\prod_{k=1}^{T_{\text{V2I},l}} d_l^\beta(k)\right)^{\frac{1}{T_{\text{V2I},l}}} 2^{\frac{qD_{\text{out},l}}{T_{\text{V2I},l}}} (q\ln 2). \tag{S.8}$$

Substituting (S.8) into (S.5) and (S.6), respectively, can immediately achieve the proof.

4 Proof of Theorem 3

Applying the Cauchy-Schwarz inequality can yield

$$\sum_{w=1}^{W_l} \frac{1}{f_w} \geq \frac{\left(\sum\limits_{w=1}^{W_l} \sqrt{\kappa \left(1 - g_W(w; D_{\mathrm{in},l})\right)} \right)^2}{\left(\sum\limits_{w=1}^{W_l} \kappa \left(1 - g_W(w; D_{\mathrm{in},l})\right) f_w \right)} \tag{S.9}$$

where the equality is satisfied if and only if $\exists \nu \in \mathbb{R}$, $\nu = \sqrt{\kappa \left(1 - g_W(w; D_{\mathrm{in},l})\right)} f_w$ holds for all w. For national simplicity, we denote the right term of (S.9) by $T^{\mathrm{lb}}\left(D_{\mathrm{in},l}, W_l, E_l^{\max}\right)$, which is indeed a lower bound of $T\left(D_{\mathrm{in},l}, W_l, E_l^{\max}\right)$ and can be further rearrange as

$$T^{\mathrm{lb}}\left(D_{\mathrm{in},l}, W_l, E_l^{\max}\right) = \frac{1}{\nu} \left(\sum_{w=1}^{W_l} \sqrt{\kappa \left(1 - g_W(w; D_{\mathrm{in},l})\right)} \right). \tag{S.10}$$

Additionally, recalling the inequality constraint given in (25), we see $\sum_{w=1}^{W_l} \nu^2 = W_l \nu^2 \leq E_l^{\max}$, i.e., $\nu^2 \leq E_l^{\max}/W_l$. This implies that the lower bound $T^{\mathrm{lb}}\left(D_{\mathrm{in},l}, W_l, E_l^{\max}\right)$ attains its minimum point only when $\nu = \sqrt{E_l^{\max}/W_l}$. Thus, substituting this result into (S.10) can obtain (27), and for each w, the optimal f_w can also be solved from $\sqrt{\kappa \left(1 - g_W(w; D_{\mathrm{in},l})\right)} f_w = \sqrt{E_l^{\max}/W_l}$.