Dynamic NOMA-Based Computation Offloading in Vehicular Platoons

Dongsheng Zheng , Yingyang Chen , Member, IEEE, Lai Wei , Bingli Jiao , Senior Member, IEEE, and Lajos Hanzo , Life Fellow, IEEE

Abstract—Both the Mobile edge computing (MEC)-based and fog computing (FC)-aided Internet of Vehicles (IoV) constitute promising paradigms of meeting the demands of low-latency pervasive computing. To this end, we construct a dynamic NOMA-based computation offloading scheme for vehicular platoons on highways, where the vehicles can offload their computing tasks to other platoon members. To cope with the rapidly fluctuating channel quality, we divide the timeline into successive time slots according to the channel’s coherence time. Robust computing and offloading decisions are made for each time slot after taking the channel estimation errors into account. Considering a certain time slot, we first analytically characterize both the locally computed source data and the offloaded source data as well as the energy consumption of every vehicle in the platoons. We then formulate the problem of minimizing the long-term maximum task queue by optimizing the allocation of both the communication and computing resources. To solve the problem formulated, we design an online algorithm based on the classic Lyapunov optimization method and successive convex approximation (SCA) method. Finally, the numerical simulation results characterize the performance of our algorithm and demonstrate its advantages both over the local computing scheme and the orthogonal multiple access (OMA)-based offloading scheme.

Index Terms—Computation offloading, non-orthogonal multiple access (NOMA), vehicular platoons, Lyapunov optimization, successive convex approximation (SCA).

I. INTRODUCTION

The Internet of Vehicles (IoV) has attracted substantial attention both in industry and academia with the objective of improving traffic efficiency and improving road safety [1], [2], [3], [4], by sharing information among vehicles and roadside units (RSUs) within their coverage area. Typical applications include safety-related, driving-assistance and passenger entertainment services [5]. The safety-critical tasks are mainly executed locally by the on-board unit for eliminating any queuing and communications delays as well as transmission impairments. By contrast, some applications rely on large amounts of data processing and/or require low latency. The specific services can include vehicular sensing [6], low-latency lip-synchronized video streaming [7], flawless interactive gaming [8], and other Internet-of-vehicles as well as social networking services [9], [10]. In a nutshell, the demands for computing resources have increased quite dramatically owing to the augmented security requirements and expectations of passengers, especially when aiming for taking into account the environmental awareness of all other vehicles in their decision-making.

To overcome these challenges, mobile edge computing (MEC) and fog computing (FC) constitute a pair of promising design paradigms for offloading computing-intensive tasks to the edge of mobile networks and fog nodes [8], [11], [12], [13]. As a benefit of closer proximity to computing servers, the data transmission delay can be significantly reduced in MEC-aided or FC-assisted vehicular networks, which substantially improves their efficiency. In urban scenarios, the computing nodes are usually part of the roadside units (RSUs) [14], [15]. However, it is costly to deploy lots of RSUs along all highways. As a design alternative, we consider vehicular platoons, which exhibit slowly varying formations [16], [17], where vehicles may offload their tasks to their platoon members having fewer or less demanding computing tasks. This also supports the option of formulating decisions based on the joint environmental awareness of the nodes in the platoon, taking into account the sensory information of all vehicles in it.

In vehicular communications, spectrum shortage is another issue when considering data transmission among vehicles requesting computation offloading and those assisting in offloading processing. In this context, non-orthogonal multiple access (NOMA) can serve several users within the same time and frequency resource block, which further improves the bandwidth efficiency [18], as evidenced in [19], [20], [21].
Motivated by the above-mentioned aspects, we construct a NOMA-based computation offloading scheme for targeted applications. Specifically, the vehicles in the platoon can offload computing tasks to the neighboring vehicles or retrieve results from other platoon members having less computing tasks by relying on the dedicated short-range communications (DSRC) protocol or on the cellular-vehicle-to-everything (C-V2X) protocol [22]. Given the potentially high speed of vehicles, we assume an uncorrelated Rayleigh fading channel model and conceive an online algorithm by making offloading decisions by allocating the communication resources according to the near-instantaneous channel conditions and additionally considering the task queues of the vehicles in each time slot. The channel estimation errors are also taken into account in the vehicular platoons.

The main contributions of our article are summarized as follows:

- We propose a dynamic NOMA-based computation offloading scheme for a vehicular platoon. The problem of minimizing the average maximum task queue length at vehicles is formulated for optimizing the offloading decisions and the communication resource allocation.

- We design the computing and offloading decisions on a time-slot by time-slot basis and take the channel estimation errors into account.

- In order to solve the problem formulated, we propose an online algorithm based on the classic Lyapunov optimization method [23] and on the successive convex approximation (SCA) method [24].

- Finally, the advantages of NOMA-based computation offloading over both local computing and orthogonal multiple access (OMA)-based offloading are quantified.

The remainder of this article is organized as follows. We discuss the related literature in Section II. Then, we introduce our system model and formulate the optimization problem considered in Section III. The algorithm scheduling the computation offloading and resource allocation decisions is designed in Section IV. Finally, our simulation results are given in Section V, and Section VI concludes this article.

II. RELATED CONTRIBUTIONS

NOMA-based computation offloading has been shown to enhance the performance of MEC and FC systems [25], [26], [27], [28], [29], [30], [31], [32], [33], [34]. The existing studies can be divided into two categories according to whether static or dynamic NOMA-based computation offloading is considered in the problem formulation.

For static NOMA-based computation offloading, diverse optimization metrics have been investigated without considering dynamic task arrivals and channel variations. To minimize the energy consumption, the authors of [25] considered a scenario where both the task offloading and result retrieval relied on NOMA. Moreover, the authors of [26] studied NOMA-based computation offloading among mobile devices, where beamforming was introduced for reducing the interference among different NOMA pairs. To minimize the time delay, Ding et al. [27] investigated three multiple access schemes – OMA, pure NOMA, and hybrid NOMA – for MEC offloading in a two-users scenario. Furthermore, the authors of [28] extended the NOMA-based computation offloading philosophy to a general multi-users scenario, where both task offloading and result retrieval relied on NOMA. Pei et al. [29] studied secure NOMA-based computation offloading in the presence of eavesdroppers. To maximize the total transmission rate, the authors of [30] investigated the scenario where a host device offloaded its latency-constrained computation task to the surrounding cooperative devices by combining both non-orthogonal downlink and uplink transmissions. However, these static NOMA-based computation offloading schemes are not suitable for the platoons considered in the face of dynamic task arrival.

For dynamic NOMA-based computation offloading, both the channel fluctuations and dynamic task arrivals are considered. The authors of [31] and [32] investigated in the multi-user NOMA uplink computation offloading and multi-server aided downlink NOMA computation offloading, respectively. The classic Lyapunov optimization method was used for converting the long-term utility optimization problem into an online optimization problem in both these studies. As a further advance, the authors of [33] considered both the long-term and short-term system behavior, where machine learning based prediction and Lyapunov optimization were adopted, respectively. Qian et al. [34] investigated a multi-task multi-access MEC system, where deep reinforcement learning was used for finding near-optimal offloading solutions for time-varying channel realizations. However, these treatises have assumed perfect channel state information (CSI), which is impractical especially in vehicular scenarios, since the movement of wireless terminals will impact the CSI acquisition accuracy.

Specifically, we have summarized the related works and compared our contribution in Table I. By taking both the dynamic channel environment as well as dynamic task arrival and realistic imperfect channel estimation into account, we construct a dynamic cooperative NOMA-based computation offloading scheme for a vehicular platoon. Both the computation offloading and communication resource allocation are optimized for all platoon members according to the near-instantaneous channel state and task queues at the vehicles. NOMA-based computation offloading is considered among all platoon members, and no other edge computing servers are involved in the scenario considered.

III. SYSTEM MODEL AND PROBLEM FORMULATION

A. System Overview

As shown in Fig. 1, we consider a steadily moving vehicular platoon on the highway. Denote vehicles within the platoon as \( K = \{1, 2, \ldots, k, \ldots, K\} \) with \( K \) representing the number of vehicles, which can partially or fully offload their computation-intensive tasks to other platoon members. On the other hand, they can also serve as computing nodes processing the tasks offloaded by other platoon members to them. To differentiate the two roles of each vehicle, we denote the computing node set as \( N = \{1, 2, \ldots, n, \ldots, N\} \) with \( N = K \). The distance between
receives the superimposed Δ and computing node is the desired time headway, and g is the driving to computing node \( (4) \) and \( t - \tau \) remain time-invariant CN + \( \neq \) [37] \( \omega_h \) reflects the small-scale fading, which obeys a complex \( i, i = \) in TS \( is not offloaded to computing node \( d(1) + h \) and \( p - 0 \) indicates that the computing task at vehicle \( kn \) is the complex additive white \( k ) = \) in the platoon \( [38] \), the actual channel coefficient can be written as 
\[
\hat{h}_{kn}(t) = h_{kn}(t) + \Delta h_{kn}(t),
\]
with \( \Delta h_{kn} \sim \mathcal{CN}(0, \sigma^2_h) \), where \( \sigma^2_h \) is the variance of the channel estimation error. Under the assumption of uncorrelated Rayleigh fading, the estimated channel coefficient \( \hat{h}_{kn}(t) \) and the channel estimation error \( \Delta h_{kn}(t) \) remain time-invariant during a TS, but they vary randomly and independently between consecutive TSs. Moreover, the estimated channel coefficient \( \hat{h}_{kn} \) can be modeled by \( [38] \)
\[
\hat{h}_{kn}(t) = \sqrt{G}(d_{kn}^{-\hat{\phi}}) g_{kn}(t),
\]
where \( G \) is the path-loss factor, \( d_{kn} \) is the distance between vehicle \( k \) and computing node \( n, \phi \) is the path-loss exponent, and \( g_{kn} \) reflects the small-scale fading, which obeys a complex Gaussian distribution with zero mean and unit variance. The distance between vehicle \( k \) and computing node \( n \) can be further written as \( d_{kn} = |k - n|/d_v. \)

Let us consider the task offloading of vehicle \( k \). Assume that the computing node \( n (n \neq k) \) receives the superimposed NOMA signal from the requesting vehicle \( k \), which is formulated as 
\[
\hat{r}_{kn}(t) = (\hat{h}_{kn}(t) + \Delta h_{kn}(t)) \sum_{i \in N} \sqrt{p_{ki}(t)} x_{ki} + \omega_{kn}
\]
where \( p_{ki}(t) \) denotes the power allocated to transmit source data from vehicle \( k \) to computing node \( i \) in TS \( t \), \( x_{ki} \) with \( \mathbb{E}[|x_{ki}|^2] = 1 \) denotes the signal transmitted from vehicle \( k \) to computing node \( i \) and \( \omega_{kn}(t) \) is the complex additive white Gaussian noise process having a power spectral density \( N_0 \). It is noted that \( p_{ki}(t) = 0 \) indicates that the computing task at vehicle \( k \) is not offloaded to computing node \( i \) in TS \( t \), and \( p_{kk}(t) = 0 \) holds for all vehicles.

any two adjacent vehicles in the platoon is assumed to be equal. We note that this assumption is reasonable because the commercially available cooperative adaptive cruise control (CACC) of [35] is already capable of satisfying this. Furthermore, the distance is velocity-dependent [36], [37]
\[
d_v = \frac{d_0 + vt_0}{\sqrt{1 - (v/v_m)^4}},
\]
where \( d_0 \) is the minimum intra-platoon spacing, \( v \) is the driving velocity, \( t_0 \) is the desired time headway, and \( v_m \) is the maximum speed.

We divide the timeline into successive time slot (TSs) according to the channel’s coherence time, which are denoted by \( T = \{1, 2, \ldots, t, \ldots\} \). The duration of a TS is denoted by \( \tau \). Furthermore, the channel quality at TSs is assumed to remain time-invariant and be uncorrelated with that of other TSs. Each vehicle has some computation-intensive tasks. We assume that the arrival of such tasks follows Poisson distribution with an arrival rate of \( \lambda, \) and the data size of the tasks follows a uniform distribution \( \theta_k(t) \sim U(\theta_{k_{\min}}, \theta_{k_{\max}}) \). Usually, the time interval between two successive computation-intensive tasks tends to be long, which indicates that \( \theta_k(t) \) equals to 0 in most TSs.

In each TS, the specific vehicles having heavy computational burden can offload parts of their tasks to other vehicles by relying on the non-orthogonal downlink and can execute parts of their tasks locally. Meanwhile, some vehicles receive the offloaded tasks and push it into their task buffer. These tasks are then executed in the following TS.

**Fig. 1.** The considered vehicular platoon system.
The successive interference cancellation decoding process [39] is adopted to retrieve the transmitted data at the computing nodes. To be specific, the computing node first decodes the signal transmitted to vehicles having better channel quality, and then remodulates the decoded signal followed by subtracting it from the received composite signal. Therefore, the signal to interference plus noise ratio (SINR) at computing nodes is formulated as

$$\Gamma_{kn}(t) = \frac{p_{kn}(t)|\hat{h}_{kn}(t)|^2}{H_n[p_k(t), w_k(t)] + |\Delta h_{kn}(t)|^2 \sum_{i \in N} p_{ki}(t)},$$  

(5)

where the function $H_n[p_k(t), w_k(t)]$ is defined as

$$H_n[p_k(t), w_k(t)] = |\hat{h}_{kn}(t)|^2 \sum_{i \in N} p_{ki}(t)[|\hat{h}_{ki}(t)|^2 > |\hat{h}_{kn}(t)|^2] + w_k(t)N_0$$

with the indicator function $\mathbb{I}(\cdot)$ given by

$$\mathbb{I}[|\hat{h}_{ki}(t)|^2 > |\hat{h}_{kn}(t)|^2] = \begin{cases} 1, & \text{if } |\hat{h}_{ki}(t)|^2 > |\hat{h}_{kn}(t)|^2, \\ 0, & \text{otherwise.} \end{cases}$$  

(6)

Moreover, the frequency band occupied by different vehicles for offloading their task is assumed to be orthogonal, which indicates that the constraint of $\sum_{k \in K} w_k(t) \leq W_0$ with $W_0$ denoting the total available bandwidth holds. Then the achievable data rate of offloading data from vehicle $k$ to computing node $n$ in TS $t$ is

$$R_{kn}^{\text{achi}}(t) = w_k(t) \cdot \log_2[1 + \Gamma_{kn}(t)].$$  

(7)

We denote the transmission data rate of vehicle $k$ to computing node $n$ in TS $t$ by $R_{kn}(t)$. Then, the data offloading $D_k^{\text{off}}(t)$ in TS $t$ can be characterized by

$$D_k^{\text{off}}(t) = \sum_{n \in N \setminus \{k\}} \pi_{kn}(t)R_{kn}(t)\tau,$$  

(8)

where $\pi_{kn}(t)$ is a binary variable with $\pi_{kn}(t) = 1$ representing successful data offloading from vehicle $k$ to vehicle $n$, and $\pi_{kn}(t) = 0$, otherwise. Moreover, we have the following equations

$$\begin{align*}
\Pr \{\pi_{kn}(t) = 0\} &= \eta_0, \\
\Pr \{\pi_{kn}(t) = 1\} &= 1 - \eta_0,
\end{align*}$$  

(9)

where $\Pr(\cdot)$ denotes the probability of an event, and $\eta_0$ represents the outage probability, which is formulated as

$$\eta_0 = \Pr \{R_{kn}(t) > R_{kn}^{\text{achi}}(t)\} = \Pr \{R_{kn}(t) > w_k(t) \log_2(1 + \Gamma_{kn}(t))\}$$

$$= \Pr \{|\Delta h_{kn}(t)|^2 > \frac{p_{kn}(t)|\hat{h}_{kn}(t)|^2}{2R_{kn}(t)/w_k(t) - 1 - H_n[p_k(t), w_k(t)]} \sum_{i \in N} p_{ki}(t)\}.$$  

(10)

Alternatively, we can express the transmission rate $R_{kn}(t)$ in terms of the outage probability $\eta_0$ as

$$R_{kn}(t) = w_k(t) \log_2\left\{1 + \frac{p_{kn}(t)|\hat{h}_{kn}(t)|^2}{H_n[p_k(t), w_k(t)]} - \ln(\eta_0)\sum_{i \in N} p_{ki}(t)\right\}.$$  

(11)

Considering the relatively small size of computation results, we ignore the data rate requirement of retrieving the results from the corresponding computing nodes. Then, the energy consumption during this offloading process can be formulated as

$$E_k^{\text{off}}(t) = \sum_{n \in N} p_{kn}(t)\tau.$$  

(12)

C. Local Computation Model

Here, we briefly introduce the computational model of [40] used for characterizing the computing capability of vehicles. The maximum operating frequency (in CPU cycles per second) of all vehicles is assumed to be equal and it is denoted by $f_m$. Leveraging dynamic voltage and frequency scaling techniques, the operating frequency can be dynamically adjusted according to the current computing burden. We denote the clock-frequency (in CPU cycles per second) at vehicle $k$ in TS $t$ as $f_k(t)$. Then the locally computed data during TS $t$ can be written as

$$D_k^{\text{comp}}(t) = \frac{f_k(t)}{\epsilon_k}\tau,$$  

(13)

where $\epsilon_k$ is the computational workload reflecting the number of CPU cycles required for processing a single input bit.

Furthermore, according to the well-established computational model of [12], the energy consumption of the local computing at vehicle $k$ in TS $t$ is formulated as

$$E_k^{\text{comp}}(t) = \xi_k f_k(t)\tau,$$  

(14)

where $\xi_k$ is the energy efficiency coefficient determined by the architecture of the processor used at vehicle $k$.

Therefore, the total energy consumption of vehicle $k$ in TS $t$ is formulated as

$$E_k^{\text{total}}(t) = \sum_{n \in N} p_{kn}(t)\tau + \xi_k f_k(t)\tau.$$  

(15)

D. Problem Formulation

Considering an individual vehicle $k$, the data that departed from the task queue includes the offloading data $D_k^{\text{off}}(t)$ and the locally executed data $D_k^{\text{comp}}(t)$ in TS $t$. Meanwhile, for the computing node $k$, the data that arrived at the input of the task
queue is comprised of the data received from other vehicles
\[ A_{k}^{\text{rec}}(t) = \sum_{i \in \mathcal{K}(k)} \pi_{ik}(t) R_{ik}(t) \tau, \]  
and the newly generated task \( \theta_{k}(t) \) in TS \( t \).

Assume that each vehicle is equipped with sufficient storage to store the source data to be processed and the length of the task queue at vehicle \( k \) in TS \( t \) is denoted by \( Q_{k}(t) \). Then the associated computing task queue is formulated as
\[ Q_{k}(t + 1) = \{ Q_{k}(t) - D_{k}^{\text{eff}}(t) - D_{k}^{\text{comp}}(t) \}^+ + A_{k}^{\text{rec}}(t) + \theta_{k}(t) \]
where \([x]^+\) equals the larger one between 0 and \( x \), and \( Q_{k}^{\text{eff}}(t) \) denotes the outstanding task queue not including the newly generated task at the end of TS \( t \).

Given that we aim for lowest possible delay, the problem to be solved is to minimize the average maximum outstanding task queue length \( Q_{k}^{\text{left}}(t) \), which can be written as

\[ \mathcal{P}1 : \min_{\mathbf{P}(t), \mathbf{w}(t), \mathbf{f}(t)} \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \max_{k \in \mathcal{K}} \{ \mathbb{E} [Q_{k}^{\text{left}}(t)] \} \quad (18) \]

s.t. \( f_{k}(t) \leq f_{m} \), \( \forall k \in \mathcal{K}, \forall t \in \mathcal{T} \), \( (18a) \)
\[ \sum_{n \in \mathcal{N}} p_{kn}(t) \leq P_{0}, \quad \forall k \in \mathcal{K}, \forall t \in \mathcal{T}, \quad (18b) \]
\[ \sum_{k \in \mathcal{K}} w_{k}(t) \leq W_{0}, \quad \forall t \in \mathcal{T}, \quad (18c) \]
\[ \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E} \left[ \frac{E_{k}^{\text{total}}(t)}{T} \right] \leq E_{0}, \quad \forall k \in \mathcal{K}, \quad (18d) \]

where the optimization parameters \( \mathbf{P}(t) \), \( \mathbf{w}(t) \) and \( \mathbf{f}(t) \) are represented by a matrix or vector holding the transmit power, the frequency band and the CPU clock-frequency of all vehicles in TS \( t \), which are given by
\[ \mathbf{P}(t) = [p_{1}(t), p_{2}(t), \ldots, p_{k}(t), \ldots, p_{K}(t)]^T, \]
\[ \mathbf{w}(t) = [w_{1}(t), w_{2}(t), \ldots, w_{k}(t), \ldots, w_{K}(t)]^T, \]
\[ \mathbf{f}(t) = [f_{1}(t), f_{2}(t), \ldots, f_{k}(t), \ldots, f_{K}(t)]^T. \]

The vector \( \mathbf{p}_{k}(t) \) can be further written as \( \mathbf{p}_{k}(t) = [p_{k1}(t), p_{k2}(t), \ldots, p_{kn}(t), \ldots, p_{kn}(t)]^T. \)

Constraints (18a), (18b) and (18d) reflect that the processing frequency, transmit power and average energy consumption of each vehicle must not exceed their corresponding maximum value \( f_{m} \), \( P_{0} \), and \( E_{0} \), respectively. Constraint (18c) guarantees that the sum of occupied bandwidth for all vehicles is less than or equal to the total available frequency bandwidth.

IV. ALGORITHM DESIGN

The problem \( \mathcal{P}1 \) of (18) satisfies the long-term average energy consumption, but reliable decisions have to be made also for each TS. The objective function (OF) and its parameters consider different timescales. Therefore, the long-term OF and long-term constraints have to be projected into each TS for tackling the problem in (18). In this context, the Lyapunov optimization method of [23] constitutes a powerful tool of constructing a new OF, while ensuring that all long-term objectives are reflected. As a benefit, the transformed problem can be solved by only considering the current network state in each TS. Then, the classic SCA method has been adopted for solving the non-convex transformed problem. We elaborate the algorithm design as follows.

A. Lyapunov Optimization

We now proceed by first reformulating the original optimization task of (18) into a new online optimization problem based on the Lyapunov optimization method, where the system decisions \( \mathbf{P}(t), \mathbf{w}(t), \) and \( \mathbf{f}(t) \) are determined solely by the system state of the current TS. The general procedures of converting this kind of dynamic optimization problem into an online optimization problem by leveraging Lyapunov’s method are summarized as follows:

- **Step 1**: Define the Lyapunov function that involves all actual or virtual queues constructed to reflect the current state of them, such as the average energy consumption of all vehicles in our model considered.

- **Step 2**: The Lyapunov drift is appropriately adjusted for guaranteeing the stability of the queues constructed in Step 1. For example, the Lyapunov drift will be increased when the energy consumption of vehicles becomes large.

- **Step 3**: The Lyapunov drift-plus-penalty function regarded as the new optimization objective is formulated by appropriately weighting the conditional Lyapunov drift and the primary optimization OF.

Based on the above steps, the long-term utility and long-term constraints can be removed from the problem formulated, which simplifies the algorithmic design. Furthermore, the performance gap between the optimal solution of the primary problem and the solution of the reformulated problem can be rendered arbitrarily small, as proven in [23]. Here, we adopt the Lyapunov-based method for designing an online algorithm to approximately apportion the computing and communication resources for all vehicles in the vehicular platoon.

Firstly, similar to [32, Lemma 1], the constraint (18d) can be replaced by
\[ \lim_{T \to \infty} \frac{\mathbb{E} [U_{k}(t)]}{T} = 0, \quad \forall k \in \mathcal{K}, \quad (19) \]
where \( U_{k}(t) \) is a virtual energy consumption queue defined as
\[ U_{k}(t + 1) = [U_{k}(t) + E_{k}^{\text{total}}(t) - E_{0}]^{+}. \]

Denote all virtual energy consumption queues as \( \Theta(t) = [U_{1}(t), U_{2}(t), \ldots, U_{K}(t)] \). Then the Lyapunov function is defined by
\[ L[\Theta(t)] = \frac{1}{2} \sum_{k \in \mathcal{K}} U_{k}^{2}(t), \quad (21) \]
while the Lyapunov drift can be expressed as
\[ \Delta[\Theta(t)] = L[\Theta(t + 1)] - L[\Theta(t)]. \]
Then, the Lyapunov drift-plus-penalty function is formulated as
\[
\Delta_V(\Theta(t)) \leq \Phi + \mathbb{E} \left[ \sum_{k \in K} U_k(t) \left( E_{\text{total}}^k(t) - E_0 \right) \right] + \mathbb{E} \left[ \max_{k \in K} \left\{ Q_{\text{eff}}^k(t) \right\} | \Theta(t) \right],
\]
(23)
where \( V \) is a non-negative weight reflecting the tradeoff between the system’s delay vs. energy consumption and its queue stability. A larger value of \( V \) indicates that reducing the outstanding task queue \( ^{\text{Q}}_k(t) \) is more important than the energy consumption of the system. Therefore, the choice of the weight \( V \) relies on the specific requirements of the vehicular platoons considered.

Lemma 1: For arbitrary feasible \( P(t), \ w(t), \ f(t) \), the Lyapunov drift-plus-penalty function is upper bounded by
\[
\Delta_{V}(\Theta(t)) \leq \Phi + \mathbb{E} \left[ \sum_{k \in K} U_k(t) \left( E_{\text{total}}^k(t) - E_0 \right) \right] + \mathbb{E} \left[ \max_{k \in K} \left\{ Q_{\text{eff}}^k(t) \right\} | \Theta(t) \right],
\]
(24)
where \( \Phi \) is a constant.

The proof is relegated to Appendix A for retaining the flow of our arguments.

Then, the primary optimization problem \( P1 \) of (18) can be transformed into minimizing the upper bound of the Lyapunov drift-plus-penalty function, which is formulated as

\[
P2 : \min_{P(t), w(t), f(t)} \sum_{k \in K} U_k(t) \left( E_{\text{total}}^k(t) - E_0 \right) + VZ(t)
\]
(25)
\[
s.t. \quad Z(t) \geq \mathbb{E} \left[ Q_{\text{eff}}^k(t) \right], \quad \forall k \in K,
\]
(25a)
(18a), (18b) and (18c)

where we introduce the optimization parameter \( Z(t) \) to deal with the maximization that appeared in the OF.

Upon combining (8), (13), (16) and (17), we obtain
\[
\mathbb{E} \left\{ Q_{\text{eff}}^k(t) \right\} = \mathbb{E} \left\{ \left( [Q_k(t) - D_k^\text{eff}(t) - D_k^\text{comp}(t)]^+ + A_k^\text{rece}(t) \right) \right\} 
\]
\[
\geq Q_k(t) + \mathbb{E} \left\{ A_k^\text{rece}(t) - D_k^\text{eff}(t) - D_k^\text{comp}(t) \right\} = Q_k(t) + \sum_{i \in K(k)} \left[ 1 - \eta_0 \right] R_{ik}(t) \tau
\]
\[
- \sum_{n \in N(k)} \left[ 1 - \eta_0 \right] R_{kn}(t) \tau - \frac{f_k(t)}{\epsilon_k} \tau \leq Q_k^\text{m}(t).
\]
(26)
Here, we define a function given by
\[
g_k(P(t), w(t), f(t), Z(t)) = Q_k^\text{m}(t) - Z(t),
\]
which is neither convex nor concave. Then, the constraint (25a) can be replaced by
\[
g_k(P(t), w(t), f(t), Z(t)) \leq 0, \quad \forall k \in K.
\]
(28)

B. Successive Convex Approximation (SCA)

On the one hand, upon substituting (15) into the OF of problem \( P2 \), the detailed OF expression can be written as:
\[
O[\text{P}(t), w(t), f(t), Z(t)] = \sum_{k \in K} U_k(t) \left( \sum_{i \in \Theta(k)} p_{ik}(t) \tau + \xi_k f_k(t) \tau - E_0 \right) + VZ(t),
\]
(29)
which is a convex function.

On the other hand, upon substituting (11) and (26) into (27), the detailed expression is given in (31), which is shown at the bottom of the page. Here, we represent the function
\[
g_k(P(t), w(t), f(t), Z(t)) = Q_k(t) + \sum_{i \in \Theta(k)} (1 - \eta_0) w_i(t) \log_2 \left\{ 1 + \frac{p_{ik}(t) \hat{h}_k(t)^2}{H_k[p_k(t), w_i(t)] + \ln(1/\eta_0)\sigma_h^2 \sum_{n \in N} P_{in}(t)} \right\} \tau
\]
\[
- \sum_{n \in N(k)} (1 - \eta_0) w_n(t) \log_2 \left\{ 1 + \frac{p_k(t) \hat{h}_n(t)^2}{H_n[p_k(t), w_n(t)] + \ln(1/\eta_0)\sigma_h^2 \sum_{i \in \Theta(k)} p_{ki}(t)} \right\} \tau - \frac{f_k(t)}{\epsilon_k} \tau - Z(t)
\]
\[
= Q_k(t) - \frac{f_k(t)}{\epsilon_k} \tau - Z(t) - \sum_{i \in \Theta(k)} (1 - \eta_0) w_i(t) \log_2 \left\{ \frac{H_k[p_k(t), w_i(t)] + \ln(1/\eta_0)\sigma_h^2 \sum_{n \in N} P_{in}(t)}{w_i(t)N_0} \right\} \tau
\]
\[
- \sum_{n \in N(k)} (1 - \eta_0) w_n(t) \log_2 \left\{ \frac{p_k(t) \hat{h}_n(t)^2 + H_n[p_k(t), w_n(t)]}{w_n(t)N_0} \right\} \tau
\]
\[
+ \sum_{i \in \Theta(k)} (1 - \eta_0) w_i(t) \log_2 \left\{ \frac{H_n[p_k(t), w_i(t)] + \ln(1/\eta_0)\sigma_h^2 \sum_{i \in \Theta(k)} p_{ki}(t)}{w_i(t)N_0} \right\} \tau
\]
\[
+ \sum_{n \in N(k)} (1 - \eta_0) w_n(t) \log_2 \left\{ \frac{H_n[p_k(t), w_n(t)]}{w_n(t)N_0} \right\} \tau
\]
\[
G_+[P(t), w(t), f(t), Z(t)]
\]
\[
G_-[P(t), w(t), f(t), Z(t)]
\]
(31)
where $x = [p_1^T(t), p_2^T(t), \ldots, p_K^T(t), w(t), f(t), Z(t)]$. Moreover, it can be shown that $G_+(x)$ is a convex function and $G_-(x)$ is a concave function. In the following, we adopt the SCA approach [24] to solve the problem. The main operation of the SCA approach is to successively find a convex restrictive approximation of the non-convex constraint.

Specifically, the updated function $\hat{g}_k(x)$ can be obtained by

$$\hat{g}_k(x) = G_+(x) + \nabla G_-(x^*)(x - x^*)^T,$$

where $x^*$ is the optimal solution with the last updated constraint $\hat{g}_k(x) \leq 0$. Given this approximation, the optimization problem $P2$ converts into

$$P3: \quad \min_{x} \mathcal{O}(\mathbf{P}(t), w(t), f(t), Z(t)) \quad (33)$$

subject to

$$\hat{g}_k(x) \leq 0, \quad \forall k \in \mathcal{K}, \quad (33a)$$

which is a convex optimization problem. However, it still remains a challenge to obtain the analytical solutions of the problem stated in $P3$. Here, we adopt the off-the-shelf convex optimization toolbox CVX [41] to solve this convex problem. Fig. 2 illustrates the design flow of the proposed algorithm.

\section{Complexity Analysis}

The SCA algorithm conceived for solving the problem $P3$ is summarized in Algorithm 1. Here, we analyze the complexity of the proposed algorithm. According to [42], the complexity of the SCA approach is given by $O((\vartheta/\delta_0)u(K^2 + 2K + 1, 3K + 1))$, where $\vartheta = O(x^*_0)/O(x^*)$, $\delta_0$ is the termination threshold as shown in Algorithm 1, and $u(K^2 + 2K + 1, 3K + 1)$ denotes the complexity of the problem $P3$ with $K^2 + 2K + 1$ and $3K + 1$ representing the number of variables and constraints, respectively. We note that the optimization problem $P3$ is a convex one, which can be solved at polynomial complexity. Therefore, the function $u(K^2 + 2K + 1, 3K + 1)$ is a polynomially increasing function with respect to $K^2 + 2K + 1$ and $3K + 1$.

\section{Numerical Results}

In this section, we evaluate the performance of the proposed algorithm in the dynamic NOMA-based computation offloading aided platooning system. Unless otherwise stated, the default simulation parameters are set as shown in Table II. The average maximum queue length in TS $t$ is defined as $1/t \sum_{s=1}^{t} \max_{k \in \mathcal{K}} Q_k(i)$. For comparison, we choose the OMA-based offloading system and the local computing system as the benchmarks. In the OMA-based counterpart, the bandwidth allocated to each communication mode pair is orthogonal. Furthermore, to ensure the fairness of comparisons, we have appropriately modified the algorithm designed for NOMA-based computation offloading to make it suitable for its OMA-based counterpart. All simulation results are obtained by averaging over 10000 independent channel realizations. The number of vehicles $K$ is set to 8.

Fig. 3 shows the average maximum queue length in different TSs under different offloading schemes. The velocity $v$, the channel estimation error variance $\sigma^2_k$, and the weighting factor $V$ are set to 60 km/h, $1 \times 10^{-6}$, and $1 \times 10^{-8}$, respectively. Over time, the average queue length converges to a specific value. It can be inferred from Fig. 3 that cooperative computing
offloading among platoon members through NOMA or OMA reduces the average computing burden of the individual vehicles. Furthermore, the NOMA-based computation offloading scheme outperforms its OMA-based counterpart.

Table II: Simulation Parameters

| Parameter                  | Value       |
|---------------------------|-------------|
| Minimum intra-platoon spacing $d_0$ | 3.0 m       |
| Time headway $t_0$          | 1.5 s       |
| Maximum driving speed $v_m$  | 120 km/h    |
| Computation workload $\epsilon_k$ | 40          |
| Energy coefficient $\xi_k$   | $1 \times 10^{-27}$ |
| Time interval $\tau$        | 1 ms        |
| Maximum operating frequency $f_m$ | 1 GHz     |
| Constant path-loss factor $G$ | $-31.5$ dB |
| Path-loss exponent $\varphi$ | 2           |
| Noise power spectral density $N_0$ | $-174$ dBm/Hz |
| Maximum transmit power of vehicles $P_0$ | 35 dBm |
| Bandwidth $W_0$             | 20 MHz      |
| Arrival rate $\lambda$     | 1/120       |
| Lower bound of generated data $\theta_{min}^k$ | $5 \times 10^5$ bits |
| Upper bound of generated data $\theta_{max}^k$ | $6 \times 10^5$ bits |
| Outage probability $\eta_k$ | 0.1         |
| Termination threshold $\delta_0$ | $1 \times 10^{-6}$ |

To study the effect of driving velocity on NOMA-based computation offloading, Fig. 4 plots the average maximum queue length over 10000 TSs at vehicles versus the velocity $v$ under different offloading schemes. The weighting factor $V$ and the channel estimation error variance $\sigma_h^2$ are set to $1 \times 10^{-8}$ and $1 \times 10^{-16}$ for all velocities, respectively. Intuitively, the average queue length is expected to increase with the velocity $v$ for both NOMA-based and OMA-based offloading. This is because the spacing among vehicles becomes larger for a higher velocity, which reduces the offloading probability at the same weighting factor $V$. Moreover, the performance gap between NOMA-based and OMA-based offloading is reduced upon increasing the velocity $v$, which would tend to zero for a sufficiently large velocity $v$. Explicitly, computation offloading will seldom be used over long distances.

Furthermore, the average maximum queue length at the vehicles versus number of vehicles $K$ under different offloading schemes is plotted in Fig. 5. We conclude from Fig. 5 that the average maximum queue length will increase, when the number of vehicles $K$ is increased. Nevertheless, compared to local computing, NOMA-based computation offloading and its OMA-based counterpart is less sensitive to the number of vehicles. Moreover, NOMA-based computation offloading always shows its superiority over the other two schemes. We note that the performance advantage of NOMA-based computation offloading over its OMA-based counterpart is beneficial, but it is gradually reduced upon increasing the total number $K$ of vehicles in a platoon. Intuitively, the average number of vehicles in overloaded status increases, as the cardinality $K$ of the vehicular
In Fig. 7, we investigate the impact of the weighting factor $V$ both on the average maximum queue length and on the average energy consumption of the vehicular platoons considered. The number of vehicles $K$, the driving velocity $v$, and the channel estimation error variance $\sigma_h^2$ are set to 8, 60 km/h, and $1 \times 10^{-16}$, respectively. According to Fig. 7, it can be observed that the average maximum queue length decreases, when the weighting factor $V$ is increased. In contrast to the average maximum queue length, the average energy consumption increases with the increase of the weighting factor $V$. These phenomena indicate that a flexible tradeoff between the system’s delay and energy consumption can be achieved by tuning the weighting factor $V$.

VI. CONCLUSION

A dynamic NOMA-based robust computation offloading scheme was conceived and analyzed for a vehicular platoon. Then the problem of minimizing the average maximum task queue has been formulated to optimize the offloading decisions and the associated resource allocation. By converting the problem formulated into an online optimization problem, we can solve the primary problem by solely taking into account the current queue state and channel state. Then the classic SCA method has been adopted for solving the non-convex transformed problem. Our numerical results have shown that NOMA-based computation offloading outperforms both the OMA-based computation offloading and the local computing scheme. Moreover, the effect of driving velocity and the channel estimation error of the NOMA-based computation offloading scheme have been investigated. Our simulation results have revealed that a lower velocity and lower channel estimation error will shorten the task queues at the vehicles. In our future research, we may conceive a similar scheme for drones flying in formation.

APPENDIX A

PROOF OF LEMMA 1

Proof: According to (20), we can proceed as follows:

$$U_k^2(t + 1) = \left\{ U_k(t) + E_k^{\text{total}}(t) - E_0 \right\}^2$$

$$\leq U_k^2(t) + E_k^{\text{total}}(t) - E_0$$

$$+ 2E_k(t) \left\{ E_k^{\text{total}}(t) - E_0 \right\}.$$

Then, the Lyapunov drift may be rewritten as

$$\Delta(\Theta(t)) = \frac{1}{2} \sum_{k \in \mathcal{K}} U_k^2(t + 1) - \frac{1}{2} \sum_{k \in \mathcal{K}} U_k^2(t)$$

$$= \frac{1}{2} \sum_{k \in \mathcal{K}} \{ U_k^2(t + 1) - U_k^2(t) \}$$

$$\leq \frac{1}{2} \sum_{k \in \mathcal{K}} \left\{ U_k^2(t) + E_k^{\text{total}}(t) - E_0 \right\}^2$$

$$+ \sum_{k \in \mathcal{K}} U_k(t) \left\{ E_k^{\text{total}}(t) - E_0 \right\}.$$
where $\Phi$ is a finite constant denoting the upper bound of 
\[
\frac{1}{2} \sum_{k \in K} \{U_k(t) + E_k(t)\}.
\]
Accordingly, the upper bound of the Lyapunov drift-plus-penalty function is formulated as
\[
\Delta_V(\Theta(t)) = E[\Delta(\Theta(t)) + V^E_{\text{total}}(t)|\Theta(t)]
\]
\[
\leq \Phi + \sum_{k \in K} U_k(t) \left[E_k(t) - E_0\right] |\Theta(t)| + \sum_{k \in K} E_k(t) |\Theta(t)| \leq \Phi + \sum_{k \in K} U_k(t) \left[E_k(t) - E_0\right] |\Theta(t)| + \sum_{k \in K} E_k(t) |\Theta(t)| \leq \Phi + \sum_{k \in K} U_k(t) \left[E_k(t) - E_0\right] |\Theta(t)| + \sum_{k \in K} E_k(t) |\Theta(t)| .
\]
which completes the proof.

REFERENCES

[1] K. M. Alam, M. Saini, and A. E. Sadik, “Toward social internet of Vehicles: Concept, architecture, and applications,” IEEE Access, vol. 3, pp. 343–337, 2015.
[2] J. Contreras-Castillo, S. Zeadally, and J. A. Guerrero-Ibaez, “Internet of Vehicles: Architecture, protocols, and security,” IEEE Internet Things J., vol. 5, no. 5, pp. 3701–3709, Oct. 2018.
[3] W. Zhang, Z. Zhang, and H. Chao, “Cooperative fog computing for dealing with Big Data in the Internet of Vehicles: Architecture and hierarchical resource management,” IEEE Commun. Mag., vol. 55, no. 12, pp. 60–67, 2017.
[4] Z. Ning, J. Huang, X. Wang, J. J. C. Rodrigues, and L. Guo, “Mobile edge computing-enabled Internet of Vehicles: Towards energy-efficient scheduling,” IEEE Netw., vol. 33, no. 5, pp. 198–205, Sep./Oct. 2019.
[5] J. Feng, Z. Liu, C. Wu, and Y. Ji, “AVE: Autonomous vehicular edge computing framework with ACO-Based scheduling,” IEEE Trans. Veh. Technol., vol. 66, no. 12, pp. 10680–10695, Dec. 2019.
[6] J. Wang, C. Jiang, K. Zhang, T. Q. S. Quek, Y. Ren, and L. Hanzo, “Vehicular sensing networks in a smart city: Principles, technologies and applications,” IEEE Wireless Commun., vol. 25, no. 1, pp. 122–132, Feb. 2018.
[7] G. Zhou, L. Zhao, Y. Wang, G. Zheng, and L. Hanzo, “Energy efficiency and delay optimization for edge caching aided video streaming,” IEEE Trans. Veh. Technol., vol. 69, no. 11, pp. 14146–14161, Nov. 2020.
[8] K. Zhang, Y. Yao, M. Leng, L. Hanzo, “Energy-aware mobile edge computing for vehicular networks: A promising network paradigm with predictive off-loading,” IEEE Veh. Technol. Mag., vol. 12, no. 2, pp. 36–44, Jun. 2017.
[9] Y. Ai, L. Wang, Z. Han, P. Zhang, and L. Hanzo, “Social networking and caching aided collaborative computing for the Internet of Things,” IEEE Commun. Mag., vol. 56, no. 12, pp. 149–155, Dec. 2018.
[10] J. Hu, L.-L. Yang, H. V. Poor, and L. Hanzo, “Branding the social and wireless networking divide: Information dissemination in integrated cellular and opportunistic networks,” IEEE Access, vol. 3, pp. 1809–1848, 2015.
[11] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, “Vehicular fog computing: A viewpoint of vehicles as the infrastructures,” IEEE Trans. Veh. Technol., vol. 65, no. 6, pp. 3860–3873, Jun. 2016.
[12] C. You, K. Huang, H. Chae, and B.-H. Kim, “Energy-efficient resource allocation for mobile-edge computation offloading,” IEEE Trans. Wireless Commun., vol. 16, no. 3, pp. 1397–1411, Mar. 2017.
[13] C. Huang, R. Lu, and K. R. Choo, “Vehicular fog computing: Architecture, use case, and security and forensic challenges,” IEEE Commun. Mag., vol. 55, no. 11, pp. 105–111, Nov. 2017.
[14] J. Du, F. R. Yu, X. Chui, J. Feng, and G. Lu, “Computation offloading and resource allocation in vehicular networks based on dual-side cost minimization,” IEEE Trans. Veh. Technol., vol. 68, no. 2, pp. 1079–1092, Feb. 2019.
[15] M. Javidi, D. Ebrahimi, R. Atallah, and C. Assi, “Workload scheduling in vehicular networks with edge cloud capabilities,” IEEE Trans. Veh. Technol., vol. 68, no. 9, pp. 8472–8486, Sep. 2019.
[39] Y. Liu, Z. Qin, M. Elkashlan, Z. Ding, A. Nallanathan, and L. Hanzo, “Nonorthogonal multiple access for 5G and beyond,” Proc. IEEE, vol. 105, no. 12, pp. 2347–2381, Dec. 2017.

[40] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, “A survey on mobile edge computing: The communication perspective,” IEEE Commun. Surv. Tut., vol. 19, no. 4, pp. 2322–2358, Fourthquarter 2017.

[41] M. Grant and S. Boyd, “CVX: MATLAB software for disciplined convex programming,” 2020. [Online]. Available: http://cvxr.com/cvx/

[42] C. N. Efrem and A. D. Panagopoulos, “Dynamic energy-efficient power allocation in multibeam satellite systems,” IEEE Wireless Commun. Lett., vol. 9, no. 2, pp. 228–231, Feb. 2020.

Dongsheng Zheng received the B.E. degree in microelectronic science and engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2018. He is currently working toward the Ph.D. degree in signal and information processing with the School of Electronics, Peking University, Beijing. His research interests include ultra-reliable low-latency communications, signal processing in wireless communications, and information theory.

Yingyang Chen (Member, IEEE) received the B.Eng. degree from the Yingcai Honors College, University of Electronic Science and Technology of China, Chengdu, China, in 2014, and the Ph.D. degree from Peking University, Beijing, China, in 2019. In 2018, she was with the University of Southampton as a Visiting Student. She is currently an Associate Professor with Jinan University, Guangzhou, China. Her research interests spread over wireless communications, cross-layer resource management, and array signal processing. Dr. Chen was the recipient of the Award for Scientific Research of Peking University, and First prize of Science and Technology Progress Award from China Work Safety Association. She is the Editor of the IEEE COMMUNICATIONS LETTERS.

Bingli Jiao (Senior Member, IEEE) received the B.S. and M.S. degrees from Peking University, Beijing, China, in 1983 and 1988, respectively, and the Ph.D. degree from Saarland University, Saarbrucken, Germany, in 1995. He became an Associate Professor in 1995 and a Professor with Peking University in 2000. He is currently the Director of the Wireless Communication and Signal Processing Research Center, Peking University. He is also the Director of the Joint Laboratory for Advanced Communication Research between Peking University and Princeton University, Princeton, NJ, USA. He is a pioneer of co-frequency and co-time full-duplex as found in his early patent in 2006. His research interests include full-duplex communications, information theory, and signal processing.

Lai Wei received the B.S. degree in electronic engineering in 2018 from Peking University, Beijing, China, where he is currently working toward the Ph.D. degree in signal and information processing with the School of Electronics. His research interests include wireless communication, signal processing, and information theory.

Lajos Hanzo (Life Fellow, IEEE) received the Honorary Doctorates degree from the Technical University of Budapest and Edinburgh University, Budapest, Hungary. He is a Foreign Member of the Hungarian Science-Academy, and a Fellow of the Royal Academy of Engineering, IET, EURASIP. He was the recipient of the IEEE Eric Sumner Technical Field Award.