A Computation Offloading Strategy Based on Stackelberg Game for Vehicular Network

Yingdi Dai(✉), Zhanjun Liu, and Ya Kang

School of Communication and Information Engineering,
Chongqing University of Posts and Telecommunications, Chongqing, China
dongnan163@163.com

Abstract. Edge computing was proposed to offload the computing tasks of the vehicle to the vehicle fog nodes, which can achieve efficient vehicle services and higher utilization of computing resources. However, encouraging vehicles to share resources or execute applications for others remains a sensitive issue due to the selfishness of users. In this paper, we propose an incentive mechanism based on Stackelberg game and the problem of computation offloading of tasks on vehicles is modeled. In order to utilize the idle computing resources of nearby vehicles, for the computation offloading in vehicular networks. Specifically, we introduce multi-hop offloading into vehicle-to-vehicle (V2V) offloading, and model the probability of forwarding packets by assisting vehicles from multiple perspectives in order to ensure the stability and reliability of communication transmission. A distributed iterative algorithm is designed to solve the final Stackelberg equilibrium, so as to minimize the cost of requesting vehicles to execute tasks and maximize the service benefits of assisting vehicles. The simulation results show that the proposed computation offloading strategy has superiorities in improving utilization efficiency, reducing tasks execution latency and enhancing service quality.

Keywords: Vehicular network · Computation offloading · Stackelberg game

1 Introduction

With the advancement of the Internet of Things (IoT) and wireless communication technology, vehicles equipped with advanced sensors and communication devices can build large-scale interactive networks, which has promoted the development of the intelligent transportation systems (ITS) [1]. In the internet of vehicles, vehicles are usually equipped with on-board unit (OBD), which

This work is supported in part by Chongqing University of Posts and Telecommunications Doctoral Fund project, E010A2019017 and the other part by the Science and Technology Research Program of Chongqing Municipal Education Commission (Grant No. KJQN201900645) and PHD Initialted Fund Project (A2009-17).

H. Gao et al. (Eds.): ChinaCom 2020, LNICST 352, pp. 505–518, 2021.
https://doi.org/10.1007/978-3-030-67720-6_35

has a certain computing and storage capacity. However, some emerging vehicle applications require complex task computation and massive data storage. The resource constrained vehicles may be strained by these applications, thus resulting in bottlenecks and making it challenging for vehicles to ensure the required Quality of Service (QoS) [2].

Aiming at the above issue, mobile cloud computing offloading technology, which allows vehicles offload the computing tasks to a powerful cloud server or base station (BS) with mobile edge computing (MEC) servers for execution, and finally return the processing result to the corresponding vehicles [3,4]. However, in the wireless network environment without infrastructure, the deployment of a high performance server for a certain area is faced with problems of site selection and cost problems. Therefore, vehicular cloud computing (VCC) as a new computing paradigm is proposed by integrating the resources of vehicles [5]. A group of vehicles can share their computing resources and storage resources by using VCC technology. Compared with fixed-location mobile edge servers, the offloading scheme in VCC has the advantages of infrastructure independence and economic benefits. Therefore, the work of this paper is carried out on this basis.

Generally, vehicles belong to different private users or different organizations, forwarding data packets or executing computing tasks for others will occupy their computing and communication resources and consume their energy. Every vehicle is rational, they all want to maximize their utility, requesting vehicles will prefer to offload tasks to assisting vehicles with good link quality, low-priced and sufficient resources for execution, while assisting vehicles hope to rent computing resources to provide high-priced requesting vehicles to obtain as much revenue as possible. This interaction between them is a typical Stackelberg game problem. If the selfishness of the vehicle cannot be overcome, the connectivity of the network will become worse, and it will cause serious waste of computing resources. Therefore, the selfishness of vehicles cannot be ignored.

How to efficiently share idle resources is a key issue. In [6], the authors divided the computing task into two subtasks, and jointly optimizes the offloading ratio and computing resource allocation to minimize the latency of task completion. In [7], a federated offloading scheme was proposed that combined vehicle-to-infrastructure (V2I) and V2V communication in MEC-enabled vehicular networks to minimize the total latency. In [8], the authors studied the computation overhead (the weighted sum of the latency and the energy consumption) minimization problem by jointly optimizing communication and computation resources. However, the computation offloading schemes proposed by the above works are all to satisfy the optimization of system performance, but ignore the selfishness of resource providers in the network. Based on this, the authors in [9] established a computation offloading marketplace based on reverse auction mechanism, which provided sellers incentives to lease their idle computation resources and allowed buyers to express diverse preferences for different sellers by unilateral-matching. In [10], the authors proposed a contract-based incentive mechanism to motivate vehicles to share their resources and transform the task assignment problem into a two-sided matching problem between vehicles and user equipment to minimize the network delay. The strategies proposed in [9,10] used incentive mechanism to overcome the selfishness of vehicles, but the matching mode between vehicles is specified, which lead to the insufficient utilization of computing resources in the network. Therefore, in this paper, we analyze the probability of assisting vehicles forwarding request packets from multiple perspectives, and introduce it into multi-master and multi-slave computation offloading game to realize the adaptive transmission of multi-hop link. In this game, the pricing incentive mechanism is used to encourage the assisting vehicles to actively participate in the task execution, and a low-complexity algorithm is used to achieve equilibrium.

This paper is organized as follows. The system model is introduced in Sect. 2. In Sect. 3, we model the computation offloading between requesting vehicles (resource requesters) and the assisting vehicles (resource providers) with a Stackelberg game approach. Numerical results are obtained and analyzed in Sect. 4. Finally, we conclude this paper in Sect. 5.

2 System Model

In this section, the system model include vehicular network model, data transmission model between vehicles, and partial offloading model, then these models will be described.

2.1 Vehicular Network Model

We consider a unidirectional road, where a RSU is located the road that enables vehicles to access the network and sends road information to vehicles, as depicted in Fig. 1. There are N requesting vehicles and M assisting vehicles arriving at the road. The i-th requesting vehicle is defined as $B_i, i \in [1, 2, \ldots, N]$, and the j-th assisting vehicles is defined as $S_j, j \in [1, 2, \ldots, M]$. Each requesting vehicle has a computing task, which can be described as $D_i = \{d_i, h_i, W_i, T_i^{res}, T_i^{max}\}$, where d_i denotes the size of input data (e.g. the program codes and input parameters), h_i is the mapping between data size and CPU cycles, W_i is the required computing resources for computing task. Referring to [11] and [12], we can obtain the number of CPU cycles, W_i for data d_i by $W_i = h_i \cdot d_i$. T_i^{res} denotes the maximum latency of receiving response in a period, T_i^{max} denotes the maximum latency allowed to accomplish the task.

We assume that each vehicle in the network is equipped with a positioning system and a wireless communication module, V2I communication can quickly obtain the required information, V2V communication can transmit information to the vehicles in the distance through multi-hop transmission [13], and can also cover all surrounding vehicles with broadcast technology [14]. The assisting vehicle S_j forwards the request packet to the l-th vehicle of k-th hop, which is expressed as $S_{j^{k_l}}$, If $k = 0$, then $l = 0$ indicates that the request message has not been forwarded by multi-hop, $S_j = S_{j^{k_l=0}}$.

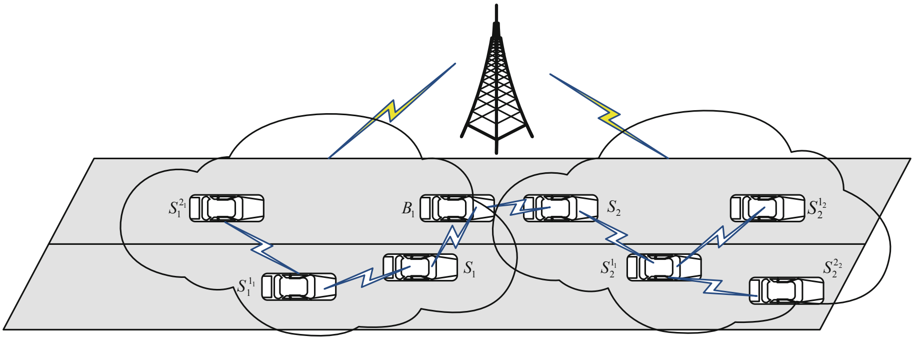


Fig. 1. A computation offloading case in vehicular network.

2.2 Data Transmission Model

The requesting vehicle B_i sends the request packet by broadcasting. If S_j within the communication range of B_i, S_j will compute the computing resources can provide and update the packet, then forward it with a certain probability and feedback the final results to B_i. The data transmission model is shown in Fig. 2.

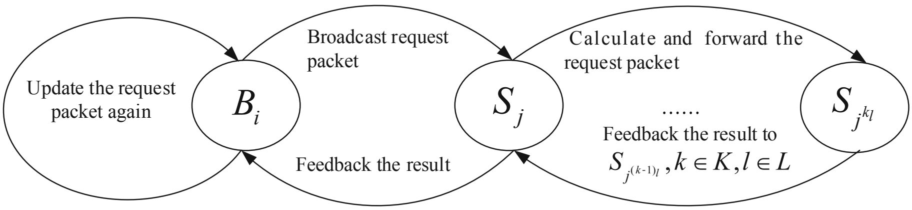


Fig. 2. Data transmission between vehicles.

Since the network environment and the situation of vehicles are relatively complex, it is necessary to determine the probability of assisting vehicles forwarding request packet. We define the forward probability of assisting vehicles includes the following factors:

- *Vehicle Density:* Assume that all vehicles in the network follow a random poisson process with a rate of λ (vehicles per second) to arrive and drive on the road at a speed V, where $V \in [V_{min}, V_{max}]$ is a random variable that independent and uniformly distributed in the distribution interval. The probability density function of moving speed V can be expressed as follows:

$$f_V(x) = \begin{cases} \frac{1}{V_{max} - V_{min}} & , V_{min} \leq x \leq V_{max} \\ 0 & , others \end{cases} \tag{1}$$

Let ρ represents the vehicle density. According to the traffic flow theory [15] $\lambda = \rho v$, the expression of ρ can be obtained by (1):

$$\rho = \lambda E(\frac{1}{V}) = \frac{\lambda ln(V_{max}/V_{min})}{V_{max} - V_{min}} \tag{2}$$

Then the assisting vehicles can estimate the number of vehicles within the communication range of B_i in the current vehicular network environment as: $N_{pre} = \rho R$, where R is the communication radius of the vehicle.

• *Link Expiration Time:* Assuming that the initial position of B_i is (X_i, Y_i), and (X_j, Y_j) is the position of S_j. After moving $\triangle t$ time slot, the position of B_i is changed to $(X_i + V_i\triangle t, Y_i)$, and the position of S_j is changed to $(X_j + V_j\triangle t, Y_j)$. If the distance between vehicles is greater than R, the V2V communication link cannot be established. Therefore, the critical time to ensure the normal communication of the link needs to meet the following:

$$R^2 = [(X_i + V_iT_{LET}) - (X_j + V_jT_{LET})]^2 + (Y_i - Y_j)^2 \tag{3}$$

where T_{LET} is the maximum link expiration time of B_i and S_j. Solve formula (3), T_{LET} can be obtained (select the positive value):

$$T_{LET} = \frac{-(X_i - X_j) \pm \sqrt{R^2 - (Y_i - Yj)^2}}{V_i - V_j} \tag{4}$$

This method is also suitable for two-way driving roads. We can regard the road as a two-dimensional coordinate, and define the driving speed of B_i as the positive speed, and the vehicle with the opposite direction as the negative speed. The link lifetime between vehicles can also be calculated by using (4).

• *Willingness of Assisting Vehicles:* We define the willingness of S_j to participate in computing offloading services as $\theta_j = \alpha_j \frac{Q_j^{max}}{Q_j^{ex}}$, where α_j is the preset constant, Q_j^{max} and Q_j^{ex} respectively represent the largest available computing resources and the already occupied computing resources of S_j.

To sum up, the probability function of assisting vehicles to forward packets can be expressed as follows:

$$F_j = 1 - e^{-\frac{T_{LET} \cdot \widehat{N}}{T_{obs} \cdot N_{pre} \cdot \theta_j}} \tag{5}$$

where $\widehat{N}$ represents the number of request packets received by the assisting vehicle, and T_{obs} is the longest travel time on the observation road. In this paper, the assisting vehicle updates the request packet can reduce the unit resource bid in the original packet according to the forwarding probability. That is the unit resource bid is determined by $p_{i,j}$ and F_j.

2.3 Partial Offloading Model

In this paper, the task of the requesting vehicle can be divided into independent subtasks, which can be executed locally or offloaded to other vehicles.

• *Local Execution:*

$$T_{i,i}^{local} = \frac{W_i - \sum_{j=1}^{M} Q_{j^{k_l}}}{f_i} \tag{6}$$

where f_i is the computing capability of B_i. $T_{i,i}^{local}$ is the local execution time. $Q_{j^{k_l}} = \sum_{k=0}^{K} \sum_{l=0}^{L} q_{j^{k_l}}$ is the total computing resources that can be provided by S_j in combination with other vehicles. $q_{j^{k_l}}$ is the computing resources provided by $S_{j^{k_l}}$, $q_j = q_{j^{k_l=0}}$.

• *V2V Offloading:* Considering the vehicles and vehicles' computational capacity distribute randomly, some vehicles with powerful computation capacity may be located out of 1-hop range. Therefore, we introduce multi-hop offloading into V2V offloading. The total latency for V2V offloading consists of two part: the latency for offloading the data of subtasks and the latency to execute subtasks on assisting vehicles [10]. The latency for offloading the data from B_i to S_j can be expressed as:

$$T_{i,j}^{trans} = max\left\{\frac{Q_{j^{k_l}}/h_i}{R_i} \mid i \in N, j \in M\right\} \tag{7}$$

where R_i is the transmission rate of B_i, $Q_{j^{k_l}}/h_i$ represents the data size of B_i offloading subtasks to S_j. Since the V2V offloading in this paper is a multi-hop offloading, the latency to execute subtasks on assisting vehicles should be included the maximum latency of subtasks execution and the maximum transmission latency of offloading subtasks on each hop, which can be expressed as:

$$T_{j^{k_l}}^{ex} = max\left\{\frac{Q_{j^{k_l}}}{f_{j^{k_l}}} \mid k \in K, l \in L\right\} \tag{8}$$

$$T_{j^{k_l}}^{trans} = max\left\{\frac{q_{j^{(k+1)_l}}/h_i}{R_{j^{k_l}}} \mid k \in K, l \in L\right\} \tag{9}$$

where $R_{j^{k_l}}$ is the transmission rate of the $S_{j^{k_l}}$. Since the amount of computation results data is very little, we are not concerned about it in this paper. Then the latency to execute subtasks on S_j can be expressed as : $T_{i,j}^{comp} = T_{i,j}^{trans} + (T_{j^{k_l}}^{trans} + T_{j^{k_l}}^{ex})$. Finally, the total latency for B_i to execute the whole task in our scheme can be given by :

$$T_i = max\{T_{i,i}^{local}, T_{i,j}^{comp}\} \tag{10}$$

3 Computation Offloading Based on Stackelberg Game

In this section, computation offloading the workload of the requesting vehicle to the assisting vehicle is modeled as a Stackelberg game. The computing resources provided by assisting vehicles are equivalent to the workload that can undertake.

3.1 Requesting Vehicle Side Analysis

The requesting vehicle with computation-intensive application waiting to be executed as the game buyer. Due to unsatisfactory computational capability and limited on-board resources, a buyer is inclined to offload part or entire of the computation workload to an applicable nearby vehicles to reduce the application completion overhead, improving the quality of experience (QoE). We define the cost function for the buyer of computing resources as follows:

$$\begin{aligned} \min_{p_{i,j}} \quad & U_i = p_{ci}(W_i - \sum_{j=1}^{M} Q_{j^{k_l}}) + p_{i,j} \sum_{j=1}^{M} Q_{j^{k_l}} + \chi_i \sum_{j=1}^{M} Q_{j^{k_l}}/h_i \\ \text{S.t} \quad & p_{i,j} \le p_{ci}, \qquad \forall i \in N \\ & 0 \le \sum_{j=1}^{M} Q_{j^{k_l}} \le W_i, \quad \forall j \in M, k \in K, l \in L \\ & T_i \le T_i^{max}, \qquad \forall i \in N \end{aligned} \tag{11}$$

where p_{ci} is the unit workload execution cost of B_i in local, which is a piecewise function:

$$p_{ci} = \begin{cases} p_{base}, & \frac{W_i}{f_i} \le T_i^{max} \\ p_{base} \times \left(\frac{W_i/f_i}{T_i^{max}}\right)^{\eta}, & \frac{W_i}{f_i} > T_i^{max} \end{cases} \tag{12}$$

p_{base} is the basic cost of vehicle to execute unit workload, and η ($\eta \ge 1$) is the price growth coefficient. $p_{i,j}$ is the unit computing resource bid by B_i. χ_i is the unit transmission cost over the V2V link. The unit resource bid of B_i cannot exceed the unit workload execution cost in local. The computing resources purchased by B_i cannot exceed that required to execute the whole task and B_i needs to complete the execution of the task within T_i^{max}.

3.2 Assisting Vehicle Side Analysis

The assisting vehicle has idle computing resources that can lease to buyers, helping to relieve the heavy on-board workload while receiving a reasonable revenue as the game seller. The revenue of S_j can be divided into two parts, the first part is that S_j provids the computing resources of itself under the unit resource bid of B_i:

$$U_j^1 = \theta_j ln\left(1 + p_{i,j} q_j\right) - c_j q_j \tag{13}$$

And the second part is that S_j reduces the bid price in the request packet, and cooperates with other assisting vehicles to execute the task to obtain the derivative income:

$$U_j^2 = \theta_j ln\left(1 + p_{i,j}(1 - F_j) Q_{j^{(k+1)_l}}\right) - \chi_j \frac{Q_{j^{(k+1)_l}}}{h_i} \tag{14}$$

In summary, the revenue maximization problem of S_j can be expressed as:

$$\begin{aligned} \min_{q_j} \quad & U_j = U_j^1 + U_j^2 \\ \text{S.t} \quad & q_j \le Q_j^{max} - Q_j^{ex}, \qquad \forall j \in M \\ & \frac{q_j}{f_j} \le min(T_{LET}, T_i^{max}), \quad \forall i \in N, j \in M \end{aligned} \tag{15}$$

The computing resources provided by each assisting vehicle cannot exceed its own available resources. The latency for processing computing tasks in the assisting vehicle cannot exceed the link expiration time with the requesting vehicle and the requesting vehicle task processing latency threshold.

3.3 Game Equilibrium Analysis

In this subsection, the inverse derivation method is used to solve the above problems. The optimal workload of the seller under different unit resource bidding is analyzed. Finding the first and second partial derivatives of U_j with respect to q_j can be obtained:

$$\frac{\partial U_j}{\partial q_j} = \frac{\theta_j p_{i,j}}{1 + p_{i,j} q_j} \tag{16}$$

$$\frac{\partial^2 U_j}{\partial q_j^2} = -\frac{\theta_j p_{i,j}^2}{(1 + p_{i,j} q_j)^2} \tag{17}$$

The revenue function of U_j is convex and has a maximum value. Using $\frac{\partial U_j}{\partial q_j} = 0$, we can get:

$$q_{j^{k_l}} = \begin{cases} \frac{\theta_j}{c_j} - \frac{1}{p_{i,j}} & p_{i,j} \ge \frac{c_j}{\theta_j} \\ 0 & p_{i,j} < \frac{c_j}{\theta_j} \end{cases} \tag{18}$$

As the buyer, each requesting vehicle needs to find an optimal quotation to minimize the cost of executing the computing task. When the optimal unit resource bid is determined, the buyer have the minimum cost to execute the task, so the strategy will not be changed. At the same time, each seller have the maximum revenue. At this time, both the buyers and the sellers are in a state of maximizing benefits, and neither side has the motivation and demand to change the strategy again, and the game finally reaches equilibrium.

3.4 Algorithm to Reach Stackelberg Equilibrium

In this subsection, we propose a distributed algorithm for the buyers to select their optimal offloading strategies. For reasonable consideration, we assume that the unit resource bid of each buyer has a range $p_{i,j}^{min} \le p_{i,j} \le p_{ci}, i \in N, j \in M$, and the iteration step is $\triangle\ \delta = 0.01$. The details of the proposed distributed algorithm are illustrated in Algorithm 1:

Algorithm 1. Distributed algorithm for obtaining optimal offloading strategy

Input: D_i^{in}, h_i, T_i^{res}, T_i^{max}, f_i, f_j, R_i, R_j, p_{base}, χ_j, η, c_j , R, $\triangle\delta$, $Q_{j^{k_l}}^{max}$, $Q_{j^{k_l}}^{ex}$
Output: $p_{i,j}^*$, $Q_{j^{k_l}}^*$, U_j^*

1: Compute the demand response of the vehicles, set $p_{i,j} = p_{i,j}^{min}$, $\triangle\delta = 0.01$;
2: **for** $i = 1 : M$ **do**
3:     **for** $j = 1 : N$ **do**
4:         Send request packet to $S_{j^{k_l}}$ within the coverage;
5:         compute $q_{j^{k_l}}$ using formula (18) and the transmission time $T_{j^{k_l}}$ of forwarding request message.;
6:         **if** $q_{j^{k_l}} \geq Q_{j^{k_l}}^{max} - Q_{j^{k_l}}^{ex}$ **then**
7:             $q_{j^{k_l}} = Q_{j^{k_l}}^{max} - Q_{j^{k_l}}^{ex}$
8:         **end if**
9:         **if** $T_{j^{k_l}} \geq T_i^{res}$ **then**
10:             $F_{j^{k_l}} = 0$
11:         **else**
12:             $T_{j^{k_l}}^* = T_i^{res} - T_{j^{k_l}}$ back to steps 5 to calculate the workload $q_{j^{k_l}}$ that other vehicles can undertake, until $T_{j^{k_l}}^* \leq 0$
13:         **end if**
14:     **end for**
15: **end for**
16: Calculate $Q_{j^{k_l}}^*$ and U_i^*, update $p_{i,j}^* = p_{i,j}$ then back to steps 5
17: **if** $U_i^* < U_i$ and $p_{i,j}^* < p_{ci}$ **then**
18:     back to steps 16
19: **else**
20:     $P_{i,j}^* = p_{i,j}, q_{j^{k_l}}^* = q_{j^{k_l}}, U_i^* = U_i$
21:     break
22: **end if**

4 Simulation Results and Discussions

In this section, we evaluate the performance of the proposed algorithm by simulations. For simulation, a unidirectional 2-lane highway segment is considered. The length of the segment is 1000 m, the vehicle flow rate is 0.25 (vehicles/sec). The idle computing resources of the assisting vehicles are random numbers that meet the normal distribution. The specific simulation parameters are shown in Table 1.

Figure 3 shows the relationship between the maximum transmission hops of V2V link and the input data-size and the load rate of network (the existing workload of all vehicles on this highway segment accounts for the maximum bearable workload). In this simulation, Fig. 3(a) shows the transmission hops of the request link are not consistent although the data-size of each requesting vehicle is the same, what caused this is the randomness distribution of vehicles. Another observation shows that the size of the input data increases leads to an increase in transmission hops, this is because the idle computing resources of assisting vehicles are limited, in order to satisfy the task offloading requirement and obtain more benefits, the assisting vehicles will forward the request packet

Table 1. Simulation parameter settings

Parameter	Value
The size of input data, d_i	$100 \sim 600$ KB
Vehicle communication radius, R	200 m
The mapping between data size and CPU cycles, h_i	$1.8 * 10^4$ cycles/B
Data transmission rate of vehicle, $R_i = R_j$	$5 \sim 6$ MHz
The velocity of B_i and S_j, respectively, V_i, V_j	$80 \sim 100$ Km/h
Price growth coefficient, η	2
Maximum workload of the vehicles, Q_j^{max}	$N(7.0, 1.0)$ Gcycles
Existing workload of vehicles, Q_j^{ex}	$N(4.0, 2.0)$ Gcycles
Cost of executing unit load c_j	0.2
Willingness constant, α	$0.1 \sim 1$
The computational capability of B_i, f_i	$N(2.5, 0.2)$ GHz
The computational capability of S_i, f_j	$N(3.0, 0.2)$ GHz

spontaneously. Figure 3 (b) shows the relationship between the transmission hops and the load rate of network. We assume the input data size of the three requesting vehicles is 500 KB and change the load rate of network. It can be seen that the transmission hop increases with the load rate. This is because the higher the load rate of network, the fewer computing resources available to the assisting vehicles, and the greater the probability that assisting vehicles forward request packet, thus increasing the transmission hops of the request link. Since each vehicle is independent, the intersection in the picture is meaningless.

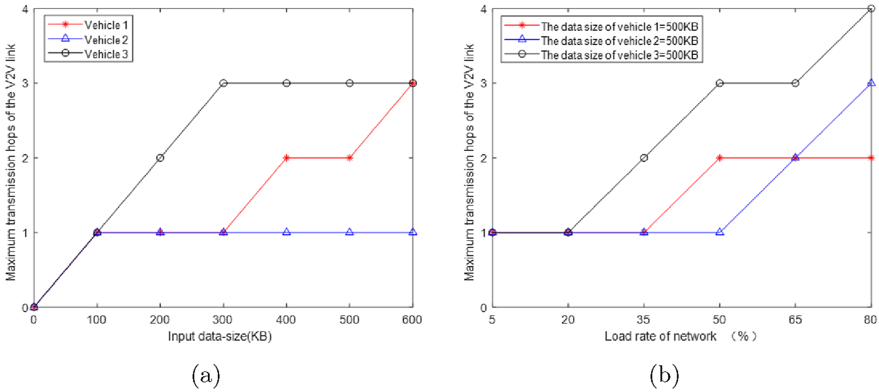


Fig. 3. The transmission hops of V2V link

We evaluate our proposed computation offloading with multi-hop execution (MHCO) strategy with two conventional computing offloading strategies. One

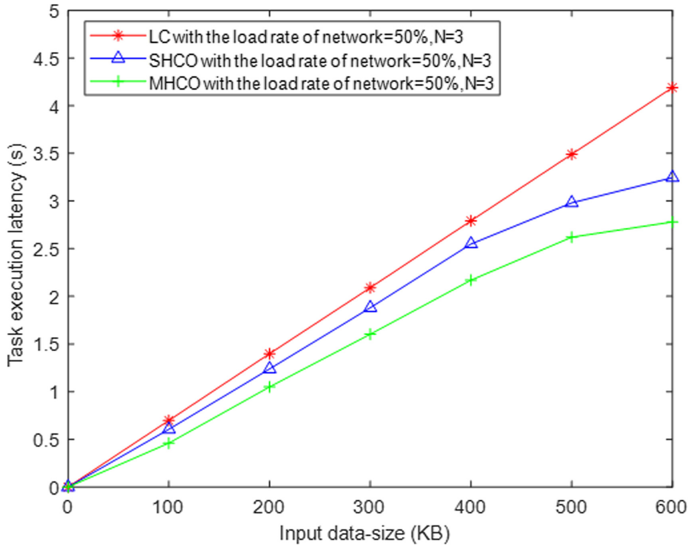
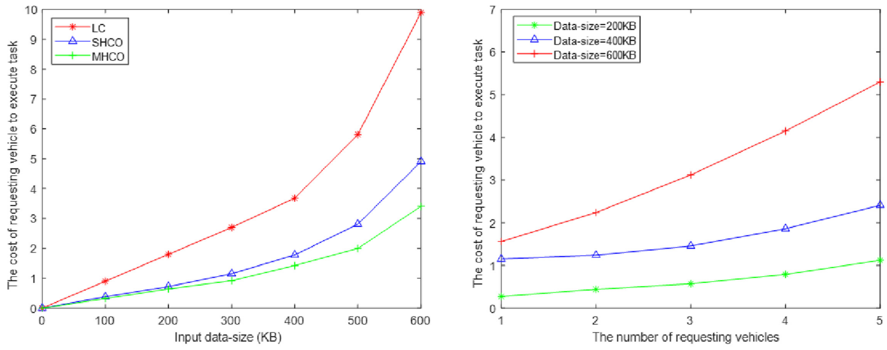


Fig. 4. The task executing latency.

is the task is completely calculated locally (LC), and the other is computation offloading with 1-hop execution (SHCO). Figure 4 presents the task execution latency versus the size of the input data. In this figure, the latency of LC is the largest. Since the computing capability of three requesting vehicle is fixed, they are 2.57 GHz, 2.72 GHz, 2.45 GHZ, the task executing latency using the LC increases linearly. It can be obtained from the simulation data, among the three computing offloading strategies, the task executing latency of SHCO and MHCO is reduced by 15% and 27% respectively compared with LC.

Figure 5 shows the effect of input data-size and the number of requesting vehicles on the cost of requesting vehicles to execute task. We assume that the maximum latency allowed for the three requesting vehicles to accomplish the task is 3.28 s, 2.63 s, 3.76 s. Figure 5(a) shows that when the latency for executing the task is greater than the maximum latency allowed to accomplish the task, the cost increases exponentially. With the increase of data-size, the requesting vehicle needs to increase the bid of unit resources to encourage assisting vehicles to provide computing resources. Compared with LC and SHCO, the MHCO offloading strategy proposed in this paper can forward request packet spontaneously, which can effectively reduce the cost of the requesting vehicles. Set the data-size of the requesting vehicles is 200 KB, 400 KB and 600 KB respectively. In Fig. 5(b), we can observe that the task executing cost is positively correlated with the number of requesting vehicles. This is because when the number of requesting vehicles increases, there is competition between them. The requesting vehicle needs to increase the bid of unit resources to meet the offloading demand.



(a) (b)

Fig. 5. The cost of requesting vehicles to execute task

In Fig. 6, set the utilization rate of computing resources in the network is 50% and the number of requesting vehicles is three. If the task is completely executed locally, the utilization ratio of computing resources in the network will not change when the computing resources required to execute the task exceed the local remaining resources. Besides, the figure shows that the utilization rate of computing resources in the network using MHCO computing offloading strategy is greater than that of SHCO.

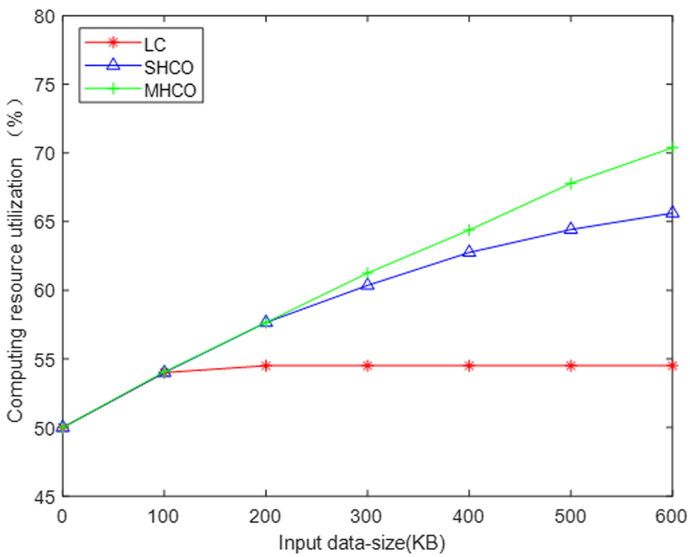


Fig. 6. The computing resource utilization.

5 Conclusion

In this paper, we proposed a computation offloading strategy based on Stackelberg game in vehicular network. The problem of computing task offloading of requesting vehicles was described as a pricing and quantitative problem, and the cost of requesting vehicles to execute task was minimized by game theory. Based on the framework, in order to improve the utilization efficiency of the idle computing resources in the vehicular network, we introduced multi-hop offloading into V2V offloading, and modeled the probability of forwarding request packets by assisting vehicles from multiple perspectives to ensure the stability and reliability of communication transmission. Then, we designed a simple and feasible distributed iterative algorithm to solve the final Stackelberg equilibrium state. Finally, the simulation results showed the effectiveness of our proposed computation offloading strategy can reduce the cost of requesting vehicles to execute tasks, and improve the utilization of computing resources in vehicular network significantly.

References

1. Jia, D., Lu, K., et al.: A survey on platoon-based vehicular cyber-physical systems. IEEE Commun. Surv. **18**(1), 263–284 (2016)
2. Popli, S., Jha, R.K., Jain, S.: A survey on energy efficient narrowband internet of things (NBIoT): architecture, application and challenges. IEEE Access **7**, 16739–16776 (2019)
3. Kaiwartya, O., et al.: Internet of vehicles: motivation, layered architecture, network model, challenges, and future aspects. IEEE Access **4**, 5356–5373 (2016)
4. Mach, P., Becvar, Z.: Mobile edge computing: a survey on architecture and computation offloading. IEEE Commun. Surv. **19**(3), 1628–1656 (2017)
5. Iyer, V., Kumari, R., et al.: Deterministic approach for performance efficiency in vehicular cloud computing. In: 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, pp. 78–82 (2015)
6. Zhou, J., Wu, F., Zhang, K., Mao, Y., Leng, S.: Joint optimization of offloading and resource allocation in vehicular networks with mobile edge computing. In: 2018 10th International Conference on Wireless Communications and Signal Processing (WCSP), Hangzhou, pp. 1–6 (2018)
7. Wang, H., Li, X., Ji, H., Zhang, H.: Federated offloading scheme to minimize latency in MEC-enabled vehicular networks. In: IEEE Globecom Workshops (GC Wkshps), Abu Dhabi, United Arab Emirates 2018, pp. 1–6 (2018)
8. Wang, J., Feng, D., Zhang, S.: Computation offloading for mobile edge computing enabled vehicular networks. IEEE Access **7**, 62624–62632 (2019)
9. Liwang, M., Dai, S., Gao, Z., Tang, Y., Dai, H.: A truthful reverse-auction mechanism for computation offloading in cloud-enabled vehicular network. IEEE Internet Things J. **6**(3), 4214–4227 (2019)
10. Zhou, Z., Liu, P., Feng, J., Zhang, Y., et al.: Computation resource allocation and task assignment optimization in vehicular fog computing: a contract-matching approach. IEEE Trans. Veh. Technol. **68**(4), 3113–3125 (2019)
11. Muñoz, O., Pascual-Iserte, A., Vidal, J.: Optimization of radio and computational resources for energy efficiency in latency-constrained application offloading. IEEE Trans. Veh. Technol. **64**(10), 4738–4755 (2015)

12. Wang, Y., et al.: Energy-optimal partial computation offloading using dynamic voltage scaling. In: 2015 IEEE International Conference on Communication Workshop (ICCW), London, pp. 2695–2700 (2015)
13. Sharshembiev, K., Yoo, S., Elmahdi, E.: Broadcast storm mitigation from unintentional misbehavior in vehicular ad hoc networks. In: 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, pp. 925–930 (2018)
14. Cheng, L., Henty, B.E., Stancil, D.D., Bai, F., Mudalige, P.: Mobile vehicle-to-vehicle narrow-band channel measurement and characterization of the 5.9 GHz dedicated short range communication (DSRC) frequency band. IEEE J. Sel. Areas Commun. **25**(8), 1501–1516 (2007)
15. Wei-Dong, Y., Ji-Zhao, L., Hong-Song, Z.: A mobility model based on traffic flow theory for vehicular delay tolerant network. In: 2010 IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA), Changsha, pp. 1071–1075 (2010)