Comprehensive Task Priority Queue for Resource Allocation in Vehicle Edge Computing Network Based on Deep Reinforcement Learning

Zhaonian Li[1](✉), Changxiang Chen[2], and ZhenJiang Zhang[1]

[1] Department of Electronic and Information Engineering, Key Laboratory of Communication and Information Systems, Beijing Municipal Commission of Education Beijing Jiaotong University, Beijing 100044, China
{20120083,zhangzhenjiang}@bjtu.edu.cn

[2] Westone Information Industry INC, Chengdu 610041, Sichuan, China

Abstract. The rapid increase in the number of vehicles and their intelligence have led to the lack of calculation resource of original network. However, the framework like vehicle-to-roadside infrastructure is still faced with the challenge of balancing the impact of time and energy consumption. To overcome these drawbacks, this paper establishes a comprehensive task priority queue on the basis of software defined network (SDN) based vehicular network instead of randomly offloading the tasks. According to the task type and vehicle speed, different tasks are graded and a joint optimization problem for minimizing the vehicles' time and energy consumption is formulated. Deep deterministic policy gradient (DDPG) algorithm is proposed to simulate the optimal resource allocation strategy of VEC model in the paper. Finally, this paper analyze the significance of the proposed model by giving numerical results.

Keywords: vehicle edge computing network · SDN · comprehensive task priority queue · task offloading · DDPG

1 Introduction

With the rapid development of technology and economic, everything becomes more intelligent and are connected to each other via Internet of Things. As for vehicles, it is important to build an efficient and practical task offloading model to guarantee the demands of different vehicles. In this model, new energy vehicles should be paid more attention, which is more sensitive to energy. As introduced in [1, 2], vehicle-to-vehicle, vehicle-to-roadside infrastructure and vehicle-to-cloud are the basic models of vehicular communication network. However, different types of tasks existed by vehicles have different demands. For example, vehicular application like autonomous driving is not suitable to be offloaded to remote cloud in order to save time instead of energy. So, how to assign tasks and ensure the service quality of the system becomes very important.

D.-J. Deng et al. (Eds.): SGIoT 2022, LNICST 497, pp. 125–132, 2023.
https://doi.org/10.1007/978-3-031-31275-5_13

So as to make overall decisions, SDN-based vehicle edge computing network [3, 4] can gather the information of the whole vehicular network. In this model, the SDN controller can obtain all of the task information and the servers' capacity information. Meanwhile, different types of tasks' demands [5] should be taken into consideration to make the model more realistic and meaningful. In the recent years, lots of researchers focus on the VEC and have done a series of signification contribution in the area of reducing time consumption, encouraging the collaboration between vehicles and so on. Whatever the starting point, researchers try to improve the quality of service of intelligent vehicular network, while saving more resources.

Previous studies have provided many research directions and methods for the problem. However, most of them belongs to the area of offline scenario vehicular network when discussing the optimization of task offloading [6]. Some environment parameters and task offloading conditions are over idealization or determined in a more fixed way. In the past few years, the tasks with different types or velocities are identified as different priorities [5, 7], which cannot objectively describe the real traffic situation. For example, a high grade task with a low vehicle velocity has a lower priority compared with a middle grade task with a high vehicle velocity. [8] introduces a method that the weight of energy consumption is redefined by the battery's remaining energy rate aiming at saving more energy for vehicles, while still in the range of adjusting the hyper parameters of the model. As for the tasking offloading direction and path, [9] proposes a novel communication model: the vehicle send the task to the server along its moving direction via vehicle-to-vehicle (V2V) multi hop mode. The authors of [10] chooses a binary offloading directions, executing the task locally or transmitting it to the edge node. Combining V2V and V2I [11], the task can be executed in the local, other vehicles or RSU. Apart from the research in the scenario of typical urban, the authors of [12] do researches in the unmanned-aerial-vehicle assisted edge computing environment. With the help of satellites and UAV network, it realizes the interaction in remote network scenarios.

Meanwhile, with the popular of machine learning, it is conveniently to use it to forecast future arrival data and make decisions. The authors of [14] uses the LSTM algorithm to make a dynamic prediction of edge communication and computing resources using mobile users' space-time. Besides, reinforcement learning is used to make task offloading decisions. DDPG algorithm is available to solve the problem of continues decision-making and be applied to high dimensional inputs [15].

Inspired by the previous studies, we propose a latency and energy aware deep deterministic policy gradient algorithm, which allows the agent to make intelligent and dynamic decisions on observation state based on reward and punishment mechanism. So, in the following part of this paper, the resource allocation problems are discussed in the SDN-based VEC model. When building the model of data transmission, we consider the interaction among edge computing devices and their operation modes. Specially, this paper adopts a comprehensive task priority queue and use the available computing resources in the network as far as possible to obtain the optimal resource allocation strategy, balancing the impact of the consumption of time and energy. In the paper, DDPG algorithm is deployed in the SDN controller, which uses the mode of centralized learning and distributed application in the roadside units (RSUs) or vehicle's processor to seek the optimal strategy via learning.

2 System Model

2.1 System Architecture

As illustrated in Fig. 1, we construct a SDN-based vehicular network model (SDN-VN), taking vehicles and multiple roadside units (RSU) into consideration. In the controller plane, it contains several RSU-controllers and a center SDN controller [3]. The RSU-controller is in charge of collecting the data from the data plane, including the data provided by vehicles and RSUs, as well as the resources like computing, communication, remaining energy. After gathering the information, the center SDN controller can make a global resource management rule, which is forwarded by the RSU-controllers. In this way, the model achieves the optimal overall strategy of task offloading in the vehicular network.

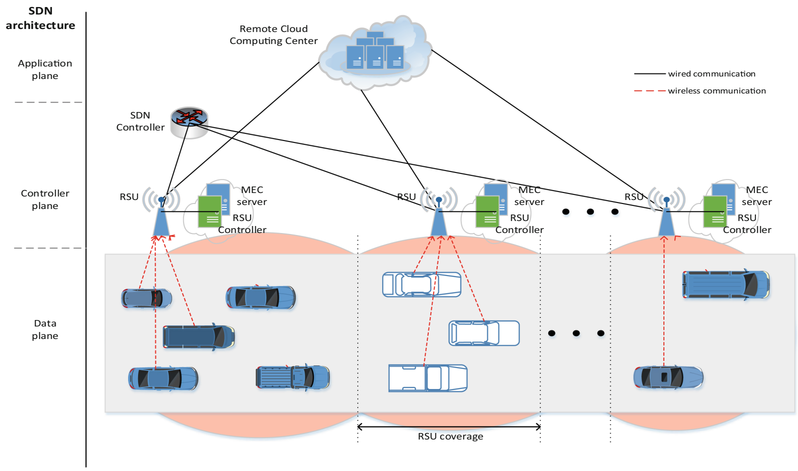

Fig. 1. The architecture of SDN-based vehicular network

We set that there is an one-direction road covered by RSUs and they can communicate with their neighbours via backbone network. The set of RSUs is given by $M = \{1, 2, ..., m\}$. Located along the roadside, RSUs can obtain ample energy through various paths. In this paper, we assume that the coverage of each RSU is the same and the adjacent ones do not overlap. Besides, each RSU is equipped with a MEC server, providing huge calculation resource to handle the tasks in vehicular edge network. In order to simplify the problem, using the diameter of the coverage area to represent the working range of RSU, recorded as L_{RSU}. We set the set of vehicles in a cell as $N = \{1, 2, ..., n\}$. Vehicles can only establish the connection with the RSU in its cell via wireless network. However, we still assume that the vehicle can receive the computed result from the previous MEC server when it enter the next cell because the latency of computing task is quite small.

2.2 Vehicular Velocity Model

Similar to [11, 16], we take a free traffic model into consideration in order to describe the velocity information. In a cell, all of vehicles drive at a constant speed, which follows the Gaussian speed distribution. Thus, the relationship between traffic density and vehicle's average velocity is described as $\overline{v} = v_{\max}\left(1 - \frac{\rho}{\rho_{\max}}\right)$, where ρ represents the traffic density of a cell, $v_{\max}$ and $\rho_{\max}$ are the maximum value of velocity and traffic density. The vehicles initialize its speed independently and randomly in each time slot, recorded as $v \sim F\left(\overline{v}, \sigma_v^2\right)$. In the reality traffic situation, the difference among vehicles' velocity should be lower when the traffic density becomes higher so as to reduce the probability of vehicle impact. So, similar to [16], we set the variance of vehicle speed is positive correlation to the average velocity, given as $\sigma_v \propto \overline{v}$.

2.3 Task Model

In the vehicular network (VN), we define three different priorities of computing tasks to simulate the real VN model, $\Phi = \{\phi_1, \phi_2, \phi_3\}$. Firstly, security information is defined as ϕ_1, which is the highest priority application. Related to the auto driving and road safety tasks, we need limit the computation latency so as to ensure the safety of vehicle driving. Thus, task ϕ_1 is executed locally with a short delay threshold, set to $\tau_{\max}^1$. As for some auxiliary driving tasks, like navigation and optional security applications, they are defined as ϕ_2, with a delay threshold set to $\tau_{\max}^2$. Tasks ϕ_3 with the low priority, is defined to represent vehicular entertainment applications with delay $\tau_{\max}^3$.

2.4 Priority Task Queue Model

Different from the queue model introduced by [7] and FIFO model, we consider both the importance of the task and the vehicle velocity. Compared with the previous model, not only can it describe the task offloading arisen in reality vehicle network more comprehensively, but also it can reflect the emergency of different tasks form different vehicles.

In this model, we should normalize the parameters first so as to balance the impacts of different parameters. So we use normalization of arctangent function and a method similar to min-max normalization to normalize the velocity and task type, where.

Then, we set $I_m = \alpha'\phi' + \beta' v'$ to show the priority of task, where α' and β' are the constants. The RSU controller obtain the information of vehicles' tasks and determine the offloading policy based on the priority task queue model and the resource consumption. So, the tasks in a slot are remarked as $K = \{1, 2, ..., k\}$, where k is the processing sequence number. We set that the MEC can only handle a task at a time, so the total execution time delay should include the waiting time. Let $t_w = \{t_{local}, t_{RSU}\}$ represent the set of waiting time of different processors.

2.5 Task Execution Latency and Energy Analysis

In this model, if the task is executed locally, we just consider the process of task calculation. If the task is executed in the edge side, we should consider the transmission process, including upload and download. However, we just take the transmission energy consumption into consideration. The resource offloading policy is influenced by the time delay and energy execution during the whole progress. Thus, we set a formula with two constants α and β to describe the relationship between them, shaped like $P_k = \alpha\frac{\bar{t}-T_k}{\bar{t}} + \beta\frac{\bar{e}-E_k}{\bar{e}}$. The constants show the relation between latency and energy and their sum is one. The total latency T_k includes task execution latency and waiting time. The minuends on numerator are the standard values for unifying the unit of calculation. Thus, the problem can be formulated as the following function (P):

$$\begin{aligned} P &: \text{maxmize R} \\ &\text{subject to X} = \{0, 1\} \\ &\qquad p_n \in (0,\ P_n] \\ &\qquad f_n^{loc} \in (0,\ F_n^{loc}], f^{RSU} \in (0,\ F^{RSU}] \\ &\qquad T_k \leq \tau_{\max,k} \end{aligned}$$

$$R = \sum\nolimits_{n=1}^{N} \sum\nolimits_{k=1}^{K} \iota_n \times P_k \times X = \sum\nolimits_{n=0}^{N} \iota_n \times \begin{bmatrix} (1-X_k)P_1 \\ X_k P_2 \\ X_k P_3 \end{bmatrix},$$

where X represents the offloading direction and ι represents the existence of tasks.

3 Resource Allocation Algorithm Based on Deep Reinforcement Learning and Simulation Results

Because the task is a multi-processor cooperated computing problem, which is a NP hard problem. So, we use the deep reinforcement learning algorithm to solve the optimal resource allocation and task offloading question. In the following, the DDPG algorithm is introduced in detail.

State space: we define the state space of vehicle n as $s_n(t)$, including reduced task information and waiting time information, which is depicted as $s_n(t) = [\tau_{\max,k}, E_{r,n}, E_n(t), t_{w,n}]$. In this set, $\tau_{\max,k}$ is the maximum latency of task k of vehicle n. $E_{r,n}$ is the remaining energy of vehicle n. $E_n(t)$ is the virtual energy queue of vehicle n. And in the part of waiting information, it just contains that of task vehicle n apart from RSUs. Therefore, the state space of the system can be defined as: $S_t = (s_1(t), s_2(t), \cdots, s_n(t))$.

The specific simulation parameters are shown in Table 1 and Table 2.

In this paper, the algorithms compared are as follows:

All Local Consumption (ALC): All tasks are executed locally.
Random Offloading (RD): The tasks of type 1 are executed locally and the others are calculated randomly.
First-In-First-Out Greedy (FIFO): The task queue is without priority, but obey latency and energy greedy algorithm (Fig. 3).

In Fig. 2, we show the convergence situation of our algorithm. This model becomes stable after 300 iterations of training. And in the next figure, we present the comparison

Table 1. Simulation parameters of model

Parameter	Value	Parameter	Value
Number of vehicles	20	Size of task	0.1 Mb
Average speed of vehicle	60 km/h	Average energy of vehicle	60%
Channel model	Typical Urban	Output/Input ratio	0.1
Delay Threshold	2, 20, 40 ms	Power of vehicle	0.25 W
Computation capacity of RSU	10 G cycles/s	Computation capacity of vehicle	1 G cycles/s

Table 2. Parameters of the neural network

Parameter	Value	Parameter	Value
Layers	3	Layer Type	Fully Connected
Learning Rate of Actor	10^{-6}	Learning Rate of Critic	10^{-6}
Episode	500	Batch	128

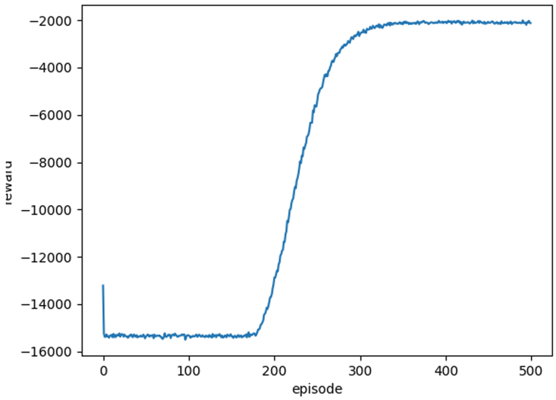

Fig. 2. Algorithm convergence diagram

of reward, execution time and consumption energy with different algorithms. Compared with ALC, RD and FIFO algorithm, our comprehensive task priority queue model can always obtain the maximize reward.

Meanwhile, it can balance the importance of time and energy consumption for different vehicle conditions, appropriately spending more time to save energy and obtain larger reward.

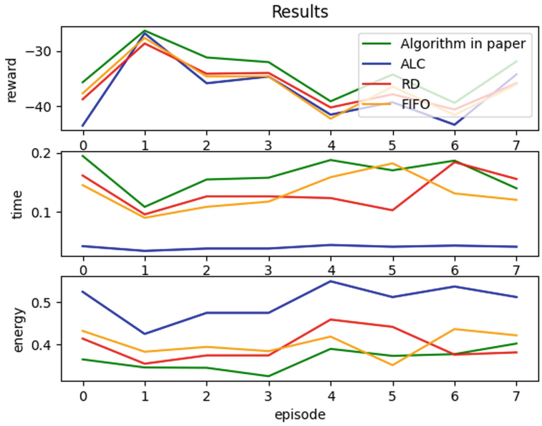

Fig. 3. Simulation results

4 Conclusion

In this paper, we propose a comprehensive task priority queue for resource allocation to achieve the goal of energy-efficiency and time-save for all vehicles in the model within task latency threshold constraint. First, we establish the model of task queue and communication model. Then, the DDPG method is used to obtain the offloading strategy. Simulation results shows that the our algorithm can improve the performance of model while saving energy.

Acknowledgement. The authors gratefully acknowledge the support and financial assistance provided by the National Natural Science Foundation of China (NO. 62173026).

References

1. Liu, C., Liu, K., Guo, S., Xie, R., Lee, V.C., Son, S.H.: Adaptive offloading for time-critical tasks in heterogeneous internet of vehicles. IEEE Internet Things J. **7**(9), 7999–8011 (2020)
2. Liu, K., Xu, X., Chen, M., Liu, B., Wu, L., Lee, V.C.S.: A hierarchical architecture for the future internet of vehicles. IEEE Commun. Mag. **57**(7), 41–47 (2019)
3. Zhang, J., Guo, H., Liu, J., Zhang, Y.: Task offloading in vehicular edge computing networks: a load-balancing solution. IEEE Trans. Veh. Technol. **69**(2), 2092–2104 (2019)
4. Zhang, J., Guo, H., Liu, J.: Adaptive task offloading vehicular edge computing networks: a reinforcement learning based scheme. Mob. Netw. Appl. **25**(5), 1736–1745 (2020)
5. Huang, X., He, L., Zhang, W.: Vehicle speed aware computing task offloading and resource allocation based on multi-agent reinforcement learning in a vehicular edge computing network. In: 2020 IEEE International Conference on Edge Computing (EDGE), pp. 1–8. IEEE (2020)
6. Cho, H., Cui, Y., Lee, J.: Energy-efficient cooperative offloading for edge computing-enabled vehicular networks. IEEE Trans. Wireless Commun. **21**(12), 10709–10723 (2022)
7. Shi, J., Du, J., Wang, J., Wang, J., Yuan, J.: Priority-aware task offloading in vehicular fog computing based on deep reinforcement learning. IEEE Trans. Veh. Technol. **69**(12), 16067–16081 (2020)

8. Li, X.: A computing offloading resource allocation scheme using deep reinforcement learning in mobile edge computing systems. J. Grid Comput. **19**(3), 1–12 (2021)
9. Raza, S., Mirza, M.A., Ahmad, S., Asif, M., Rasheed, M.B., Ghadi, Y.: A vehicle to vehicle relay-based task offloading scheme in vehicular communication networks. PeerJ Comput. Sci. **7**, e486 (2021)
10. Khayyat, M., Elgendy, I.A., Muthanna, A., Alshahrani, A.S., Alharbi, S., Koucheryavy, A.: Advanced deep learning-based computational offloading for multilevel vehicular edge-cloud computing networks. IEEE Access **8**, 137052–137062 (2020)
11. Zhao, J., Kong, M., Li, Q., Sun, X.: Contract-based computing resource management via deep reinforcement learning in vehicular fog computing. IEEE Access **8**, 3319–3329 (2019)
12. Seid, A.M., Boateng, G.O., Mareri, B., Sun, G., Jiang, W.: Multi-agent drl for task offloading and resource allocation in multi-uav enabled iot edge network. IEEE Trans. Netw. Serv. Manage. **18**(4), 4531–4547 (2021)
13. Wu, J., et al.: Resource allocation for delay-sensitive vehicleto-multi-edges (v2es) communications in vehicular networks: a multi-agent deep reinforcement learning approach. IEEE Trans. Netw. Sci. Eng. **8**(2), 1873–1886 (2021)
14. Zhang, K., Cao, J., Zhang, Y.: Adaptive digital twin and multiagent deep reinforcement learning for vehicular edge computing and networks. IEEE Trans. Industr. Inf. **18**(2), 1405–1413 (2021)
15. Zheng, C., Liu, S., Huang, Y., Yang, L.: Hybrid policy learning for energy-latency tradeoff in MEC-assisted VR video service. IEEE Trans. Veh. Technol. **70**(9), 9006–9021 (2021)
16. Tan, W.L., Lau, W.C., Yue, O., Hui, T.H.: Analytical models and performance evaluation of drive-thru internet systems. IEEE J. Sel. Areas Commun. **29**(1), 207–222 (2011)
17. Su, Z., Hui, Y., Luan, T.H.: Distributed task allocation to enable collaborative autonomous driving with network softwarization. IEEE J. Sel. Areas Commun. **36**(10), 2175–2189 (2018)