A Machine-Learning based Time Constrained Resource Allocation Scheme for Vehicular Fog Computing

Xiaosha Chen, Supeng Leng, Ke Zhang and Kai Xiong

***Abstract*—Through integrating advanced communication and data processing technologies into smart vehicles and roadside infrastructures, Intelligent Transportation System (ITS) has evolved as a promising paradigm for improving safety, efficiency of the transportation system. However, how to meet the strict delay constraints of the safety-related applications especially in dense traffic environment is still a great challenge for the ITS. In this paper, we propose the metric called Perception-Reaction Time (PRT), which reflects the time consumption of safety-related applications and is closely related to road efficiency and security. Through incorporating information-centric networking technology and fog virtualization approach into vehicular network management, we propose a novel fog resource scheduling mechanism to minimize the PRT. Furthermore, we adopt a deep reinforcement learning approach to design an on-line optimal resource allocation scheme. Numerical results demonstrate that our proposed schemes reduce about 70% of the RPT compared to the traditional architecture.**

***Index Terms*—Deep Reinforcement Learning, Information-Centric Networking, Intelligent Transport System, Perception-Reaction Time, Resource Allocation, Vehicular Fog**

I. Introduction

Being a powerful platform supplying versatile on-road applications, ITS has attracted extensive attentions from academia and industry recently [1]. ITS consists of a large number of smart vehicles equipped with advanced sensor devices, communication modules and information processors, which are capable to detect traffic environments and obtain intelligent traffic control strategies [2]. However, the vehicular applications that improve safety and efficiency of transportation networks always require intensive computation under strict delay constraints [3]. These requirements pose critical challenges on resource constrained individual smart vehicles [4].

Mobile Edge Computing (MEC) is a promising paradigm to address this challenge. Through providing computation services in proximity of smart vehicles, MEC copes with the intensive computation demands of traffic applications efficiently. Being nodes with powerful computing capabilities, MEC servers process offloaded tasks with low computational delay [5][6]. However, time costs for the implementation of the traffic applications consist of not only the computational delay but also communication delay generated in task offloading process. Thus, edge computation and offloading communication scheduling of vehicular networks needs to be collaboratively optimized [7]. Considering highly dynamic network topology caused by vehicle movement, constrained service area of an individual edge server and coupling between heterogeneous edge resources, jointly scheduling of heterogeneous resources of multiple edge servers is a significant challenge.

Furthermore, the design of vehicular edge service scheduling is closely related to both application characteristics and traffic states. For instance, in high-speed mobile and vehicle-intensive traffic environments, it is necessary to provide large amount of edge resources for traffic safety-related applications to ensure the timeliness of driving action decision and alarm information delivery. However, an efficient approach that reflects road traffic characteristics as edge resource requirements has not yet been developed.

To address the above challenges, in this paper, we introduce the concepts of the Information-Centric Network (ICN) [8] because in a vehicular fog, lots of packets contain the similar contents [9]. ICN is able to cache the contents of the different applications to the corresponding vehicles and merging the packets with the same contents to significantly reduce the delay of vehicular applications [10][11]. Moreover, the queue size of each vehicle will have an obvious upper bound, which will break through the bottleneck of the random-access Medium Access Control (MAC) protocols like Dedicated Short Range Communication (DSRC) in the traditional IP-based network. In addition, this paper focuses on minimizing the PRT of safety-related vehicular applications in the realm of ICN based vehicular fog computing architecture. Real-time IoV resource allocation is based on the current road traffic environment, demands of vehicular applications, and communication-

This is an early open access version, which will be submitted to China Communications. This work was partially supported by National Key R&D Program of China (No.2018YFC0807101), the Science and Technology Program of Sichuan Province, China (No.2019YFH0007), the joint fund of the Ministry of Education of China and China Mobile (MCM 20160304), and the EU H2020 Project COSAFE (MSCA-RISE-2018-824019).

Xiaosha Chen, Supeng Leng, Ke Zhang and Kai Xiong are with School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, 611731, China.

The corresponding author is Supeng Leng, email: spleng@uestc.edu.cn

computation resource distribution. Consequently, in this paper, the vehicle resource allocation process is modeled as a Markov Decision Process (MDP) [12] and a learning algorithm is adopted to allocate the vehicular resources based on the complex environment to minimize the PRT of the vehicular safety-related applications. The main contributions of this paper are listed as below:

- Different from the traditional computation delay of communication delay minimization, the optimization of the new metric called PRT which consists of the computation latency, task offloading delay and packet transfer delay is considered.
- We design a novel vehicular fog computing architecture to cope with the dynamic requirements and topology in the on-road scenario. Unlike the conventional fog, the content merging technique of ICN is adopted to form the vehicular fog.
- With the deep reinforcement learning approach, we propose a real-time computation and storage resource joint allocation scheme with fast converge speed and high stability.

The rest of this paper is organized as follows. Section II shows a brief overview of existing researches. Section III introduces the system model. We formulate the optimization problem in section IV. Section V designs the deep reinforcement learning based resource management algorithm. Section VI presents the experimental analysis. Finally, section VII concludes this paper.

II. Related Work

This section summarizes the most significant scientific contributions related to both fog computing and emerging information-centric network in IoV.

Fog computing was proposed by Cisco to maximally exploit the collaboration of local communication and computation resources [13], [14]. Peng et al. [15] presented the performance analysis of Fog computing where the model selection schemes were proposed to enhance the energy efficiency and the spectral efficiency of the fog computing. In addition, Park et al. [16] investigated the resource allocation in software defined fog vehicular networks. The optimization is modeled as a game formation. Feng et al. [17] proposed the Autonomous Vehicular Edge (AVE) framework to increase the computational capabilities by administering the resources of vehicles in a distributed way. K. Xiong et al. study how to allocate the storage and computation resources in the virtual vehicular network in [18].However, the effects of vehicular computing offloading have not been thoroughly investigated and the influence of the resource allocation schemes on PRT is not considered.

ICN is the potential solution to conquer the limited support to mobility in IP networks. Amadeo et al. [19] confirmed the potential of ICN as a promising solution for future vehicular networks. Xu et al. [20] first proposed and extended the green information-centric multimedia streaming (GrIMS) framework is designed to drive the system toward optimal working points in practical settings. Zheng et al. [21] proposed a scheme where each node retrieves the requested content from other nodes or base stations and investigated. The optimal caching problem of ICN in the IoV environment is still an open question.

Deep Q-learning is a powerful tool that can be used in traffic field. The authors in [22] proposed learning approach was used in the video streaming. [23] applied deep Q-learning in traffic simulation study and vehicle pathway optimization. [24] utilizes a deep Q-learning approach for designing optimal offloading schemes, jointly considering selection of target server and determination of data transmission mode. However, the potential of learning based approaches have not been explored for vehicular fog computing resource management.

To complement the precious efforts and move a further step toward an intelligent learning algorithm in the ICN-IoV research field, this paper proposes the deep Q-learning to minimize the total delay of vehicular applications in ICN-IoV by leveraging novel ICN features.

III. ICN based Vehicular Fog

This section will introduce the assumptions and the framework in this paper. The essential purpose is to improve the traffic efficiency and safety with the ITS. We use the vehicular fog computing technology to offload the tasks to the resource-rich vehicles and the ICN technology to reduce the load of the fog by merging the packets with the same contents.

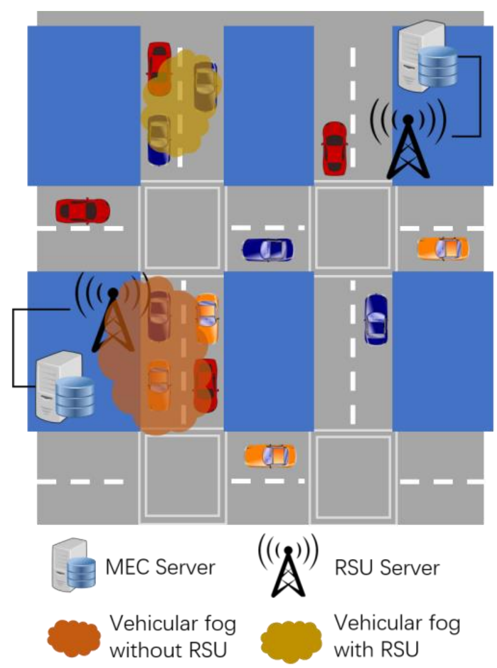


Fig. 1. Vehicular fog computing with and without RSU.

We consider a vehicular fog consists of N vehicles. All vehicles in the vehicular fog are equipped with the communication model. The authors of [25] suggest that the DSRC protocol is faster than the LTE-V protocol for V2V communication without RSU. Consequently, in this paper, we assume that vehicles communicate with each other with DSRC protocol.

We assume all the vehicles in a vehicular fog are in the same communicate scope. As a result, we do not consider the hidden node or the exposed node problem. Because of the mobility property of the vehicles, the vehicular fog computing should deal with the scenario with or without RSU. In the scenario with RSU, the vehicular fog can access the Mobile Edge Computing (MEC) server through RSU, which offers considerably extra computing resources. However, in the scenario without RSU, the vehicular fog need provide basic resource virtualization function. We assume that in the same vehicular fog, all vehicles have the same speed. Meanwhile, each vehicle can quit the vehicular fog and new vehicles can join the fog. Some of the road segments are in the RSU communication range and others are not. When a vehicular fog connects to an RSU, different data will need to be transmitted and processed in the fog with new communication and computation resources. The basic scenario of the system in this paper is illustrated in Fig.1.

In the on-road scenario, many typical applications will be maintained in a vehicular fog, such as beacon safety messages, accident warning messages, sensor information sharing, multi-media flow and so on. Each application requires some communication, computation and storage resources in the vehicular fog. The vehicles in the same fog are generally in a short distance to the others, which makes the same information is commonly required repeatedly. This feature can further improve the performance of the vehicular fog computing. As a result, ICN technology will be utilized to support the resource sharing in the fog.

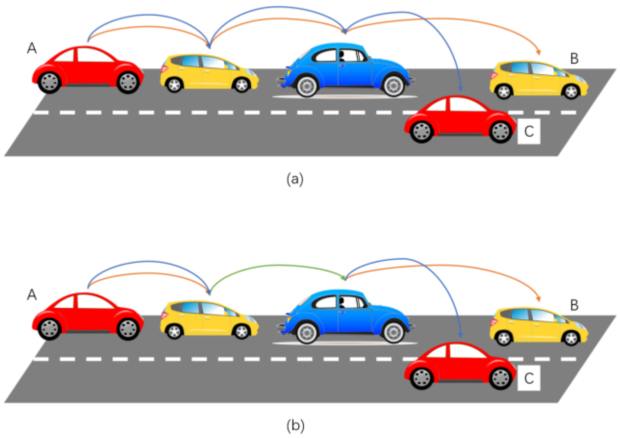


Fig. 2. Traffic flows in the vehicular fog:
(a) Conventional Network (b) ICN

The basic idea of the ICN-Fog is illustrated in Fig. 2. In Fig. 2, vehicle B and vehicle C require the same message from vehicle A. In the conventional network, vehicle A sends the message after getting the requirement from vehicle B. Then re-send the same message to the vehicle C after receiving the requirement from vehicle C. In the ICN-Fog, the last message will observe that there exists a message in queue of the first hop with identical contents. Then the last message will be dropped while the first message will be transmitted to both vehicle B and vehicle C. In this case, we can find that comparing to the conventional network, the ICN-Fog has lower traffic loads and end-to-end communication delay.

In the conventional IP-based network, the queue size of each vehicle can be extremely long. Nevertheless, in the ICN-Fog, the upper bound of the queue size can be easily obtained, which is the number of packets categories. Consequently, the queueing time can be easily approximated.

To analyze the performance of the ICN-Fog in the road traffic perspective, we will study the PRT of the vehicles in the fog. In traditional transportation studies, The PRT is the response time which consists of perception and decision time to be taken for applying the brake when the driver recognizes certain risky situation[26]. Nevertheless, in the intelligent transportation system, the safety related applications can find the accident and make decision fast and reliably. Therefore, in this paper, we consider the PRT as the length of the period from generating to arriving at the destination of a safety related packet.

In a vehicular fog, because of the same speed assumption, the related locations of the vehicles are fixed unless there are vehicles join or quit the vehicular fog. To guarantee the road safety, the speed of the vehicles must be limited to give the vehicles enough time to discover the accident and make decision. Consequently, shorter PRT will allow a higher speed of the vehicles and improve the road traffic efficiency.

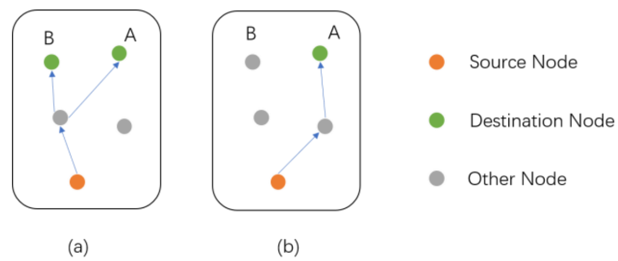


Fig. 3. Different number of destination nodes can influence the route path for the same task.

PRT consists of communication delay and computation delay. In this paper, we consider the communication and computation processes as the different stages of the safety-related application. The computation delay is determined by the data offloading scheme and the computation resources. If multiple vehicles calculate to decide whether the current environment is safe, the whole computation delay will be determined by the maximum total delay of offloading data transmission delay and the computation delay of each vehicle. The communication delay is determined by the communication bandwidth and the route path. The route path in the ICN-Fog is different from the IP-based network. Fig.3 shows the path selection problem in the ICN-Fog. In Fig.3, whether node B requires the application from the source node will change the route path for node A. Consequently, different number of destinations will influence the data route path of each individual vehicle.

IV. Problem Formulation

This section will formulate the system and present the optimization problem.

A vehicular fog can be formed by up to N vehicles, denoted by $\mathcal{N} = \{v_1, v_2, \dots, v_N\}$. Each vehicle can be represented by its location and resources. If the number of vehicles in a vehicular fog is less the N, the remaining element of $\mathcal{N}$ can be considered as the vehicles with no available resource. In this paper, we divide the road into short segments, and assign index number to them. The vehicles in the same segments can be considered in the same position. To make the vehicular fog computing resource management scheme feasible, we number the candidate positions in a fog as $\mathcal{L} = \{l_1, l_2, \ldots, l_L\}$, where $L = |\mathcal{L}|$ is the number of elements of positions set $\mathcal{L}$.

Each vehicle can be denoted by a three ordered tuple $v_i = \; p_i, c_i, s_i \;$, where p_i, c_i and s_i represent the index of the relative position in the vehicular fog, available computation resource and available storage resource, respectively. The distance of each pair of candidate location in a vehicular fog is denoted as a matrix $\boldsymbol{D} = \{d_{ij}\}_{L\times L}$, where d_{ij} is the distance from l_i to l_j in $\mathcal{L}$. Therefore, the distance from vehicle i to vehicle j can be represented as d_{p_i,p_j}. It is obvious that the matrix $\boldsymbol{D}$ is a symmetric matrix.

All the vehicles in the vehicular fog transmit data in the same power, denoted as P. With the exponential backoff algorithm in the DSRC, there are no interference during the transmission. Based on this fact, the link capacity between vehicle i and vehicle j is:

$$C_{ij} = \frac{1}{2}\log\left(1 + \frac{PL_0 d_{p_i,p_j}^{\alpha'}}{N_0}\right) \tag{1}$$

where L_0 and α' are the path loss at a reference unit distance and path loss exponent, respectively. N_0 is the power of additive white Gaussian noise.

The applications in a vehicular fog are considered as two categories: a) safety related and b) non-safety related applications. Like the definition of vehicle set $\mathcal{N}$, we denote the sets of safety related applications and non-safety related applications as $\mathcal{A}^{\mathrm{s}}$ and $\mathcal{A}^n$, respectively. And each element in the sets, represent as $a_i^s, i = 1 \ldots A_s$ and $a_i^n, i = 1 \ldots A_n$, can be considered as a two tuple $\; c_i^{\mathrm{s}}, s_i^s \;$ and $\; c_i^n, s_i^n \;$, respectively. Where $A^s = |\mathcal{A}^s|$, $A^n = |\mathcal{A}^n|$, c_i^s and c_i^n are the computation resource requirement of safety related application i and non-safety related application i, respectively, s_i^s and s_i^n are the storage resource requirement of safety related application i and non-safety related application i, respectively.

For application a_i, one vehicle will cache its data and several vehicles will join the computation process of the application a_i. $\Omega_i = \{\omega_{i,1}, \omega_{i,2}, \ldots, \omega_{i,N}\}$ and $\Gamma_i = \{\gamma_{i,1}, \gamma_{i,2} \ldots, \gamma_{i,N}\}$ represents the amount of data corresponding vehicles cache and compute for application $\boldsymbol{a_i}$, respectively. In addition, $\omega_{i,j} = 0$ or $\gamma_{i,j} = 0$ means vehicle j does not cache data or compute for application a_i. Moreover, we do not consider the case where different parts of data caches in various vehicles, so $\omega_{i,j} \in \{c_i, 0\}$.

The computation delay of application $\boldsymbol{a_i}$ is decided by the maximum transmission delay plus computation delay, which is shown in Fig. 4. In Fig. 4, source vehicle asks vehicle A and vehicle B to compute together. The transfer delay is overlapping because the delay has already included the queueing delay. To simplify the problem, in this paper, we assume that the amount of required computation resource and the compute result is proportional to the data size, which can be represented as $c_i = k_c \cdot \gamma_i$ and $r_i = k_r \cdot \gamma_i$, respectively.

To model the application and resources on RSU, we model the RSU as a vehicle v_0 with much more resources than the other vehicles. Vehicles communicate with the RSU in different frequency, therefore, the uplink can be regarded as a fixed capacity link during the data transmission, which is denoted by C_{RSU}. After time T_{RSU}, the RSU will quit the vehicular fog. The application on RSU is a_0 with computation and storage resource requirement c_0 and s_0, respectively. And the computation and storage resource on RSU is c_0^{RSU} and s_0^{RSU}, respectively.

For application a_i, the probability that vehicle j requires application a_i is denoted by $P_{i,j}$. Because of the ICN network, once several vehicles require an application, the route path will be a tree. Furthermore, the queue size will not exceed $A_s + A_n$. To simplify the problem, we consider the queue delay for each vehicle as a constant delay, which is denoted as T_q and is equal to the upper bound of the queue delay. We can get the average transmission delay for the application $a_i, i > 0$:

$$\begin{aligned} \mathrm{E}[T_i, v_s] = & \sum_{\mathcal{V}\in\mathcal{P}\;\mathcal{N}} \prod_{j=1}^{N} (I\;\; j,\mathcal{V}\;\; P_{i,j} \\ & + (1 - I\;\; j,\mathcal{V}\;\;)(1 - P_{i,j})) \\ & \times \left(\sum_{k,l\;\in\mathcal{E}\;\mathcal{V},v_s} \frac{C_{k,l}}{s_i} + |\mathcal{E}\;\mathcal{V}, v_s\;| \cdot T_q \right) \end{aligned} \tag{2}$$

where v_s is the vehicle caches the data of application a_i and $\mathcal{P}\;\mathcal{N}$ is the power set of $\mathcal{N}$. The nodes in $\mathcal{V}$ is the nodes that require the application a_i. $I\;\; j,\mathcal{V}$ is an indicator function which is defined as following,

$$I\;\; j,\mathcal{V}\;\; = \begin{cases} 1 & \text{if } v_j \in \mathcal{V} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Furthermore, $\mathcal{E}\;\mathcal{V}, v_s$ is the edge set of the Minimum Spanning Tree (MST) with root node v_s. And $|\mathcal{E}\;\mathcal{V}, v_s\;|$ is the number of edges of the MST.

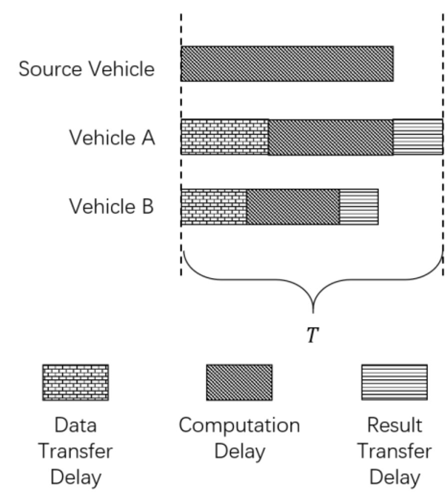


Fig. 4. Composition of the computation delay

The average sum of the computation delay and transmission delay, which is called PRT in this paper, of application a_i can be calculated by the following equation,

$$E[T^i_{\mathrm{PRT}}] = E[T_i, v_s] + \max_{1\le j\le N}\left\{\frac{\gamma_{i,j}}{k_c\gamma_{i,j}} + 2E[T_{\mathrm{trans}}] + \frac{\gamma_{i,j}\ 1+k_r}{C_{v_s,j}}\right\} \tag{4}$$

where $E[T_{\mathrm{trans}}]$ is the average queueing delay of the HOL packet for the exponential backoff algorithm, which can be calculated according [27]:

$$E[T_{\mathrm{trans}}] = \sum_{j=0}^{R} p^j E(b_j) t_B \tag{5}$$

where p is collision probability, R is the stage number of the exponential backoff algorithm, $E(b_j)$ represents the number of backoff slots at stage j, t_B is the average length of a backoff slot.

For the application a_0 on RSU, the sum of average computation delay and transmission delay will be the following,

$$E[T^{\mathrm{RSU}}_{\mathrm{PRT}}] = \frac{E[\beta]c_0^{\mathrm{RSU}}}{c_0} + \frac{s_0^{\mathrm{RSU}}}{C_{\mathrm{RSU}}} \tag{6}$$

The optimization problem now can be written as following:

$$\begin{aligned}
&\min_{\mathbf{\Omega},\mathbf{\Gamma}} \max\{E^s[T^i_{\mathrm{PRT}}]\}\\
\text{s.t.}\quad & C1\text{: } \sum_{i=1}^{|\mathcal{A}_s|}\omega^s_{i,j} + \sum_{i=1}^{|\mathcal{A}_n|}\omega^n_{i,j} \le s_j, \quad 1\le j\le N\\
& C2\text{: } \sum_{i=1}^{|\mathcal{A}_s|}\gamma^s_{i,j} + \sum_{i=1}^{|\mathcal{A}_n|}\gamma^n_{i,j} \le c_j, \quad 1\le j\le N\\
& C3\text{: } E^n[T^i_{\mathrm{PRT}}] \le T_n, \quad 1\le i\le |\mathcal{A}^n|\\
& C4\text{: } \omega^s_{i,j}\in\{0, s^s_i\}, \quad \sum_j \omega^s_{i,j} = s^s_i, 1\le i\le N\\
& C5\text{: } \omega^n_{i,j}\in\{0, s^n_i\}, \quad \sum_j \omega^n_{i,j} = s^n_i, 1\le i\le N\\
& C6\text{: } , E[T^{\mathrm{RSU}}_{\mathrm{PRT}}] \le T_{\mathrm{RSU}}
\end{aligned} \tag{7}$$

In (7), the superscripts s and n are used to distinguish the safety and non-safety related applications for the corresponding variables. C1 and C2 means storage and compute resources on each vehicle is limited by their capacity. C3 give an upper bound for the non-safety related applications. C4 and C5 means each application only can have one source vehicle. C6 gives a delay upper bound for the applications on RSU with the RSU existing time T_{RSU} The optimization problem is to minimize the maximum PRT for application related applications, and for the non-safety related applications, we just need to satisfy their time constrained. Each time the resources or applications change, the system will re-calculate the optimization problem.

V. Resources Management Scheme

Because of the infeasibility of solving the optimization problem (7), in this section, we will develop a deep reinforcement learning based resource management algorithm to improve the efficiency for the communication, computation and storage resources in a vehicular fog.

To develop the resource management algorithm, the system model needs to be formulated as a Markov Decision Process. The state of the system should present the current distribution and usage of the resources. Therefore, the system state should be defined as following,

$$S = \ \mathcal{N}, \mathbf{\Omega^s}, \mathbf{\Omega^n}, \mathbf{\Gamma^s}, \mathbf{\Gamma^n} \tag{8}$$

where $\mathbf{\Omega^s} = \{\Omega^s_1, \ldots, \Omega^s_{A^s}\}$, and $\mathbf{\Omega^n}$, $\mathbf{\Gamma^s}$, $\mathbf{\Gamma^n}$ are defined in the similar manner. Notice in (8), $\mathcal{N}$ include v_0, which represents the RSU in this paper.

As for the action in the Markov Decision Process, in order to reduce the number of actions, in each action, only one pair of elements in $\mathbf{\Omega^s}$ or $\mathbf{\Omega^n}$ can be swapped. Moreover, each element in Γ^s_i or Γ^n_i can only increase or decrease by 1. Following this definition, the action can be defined as

$$a = \{(\omega^s_{s,1}, \omega^s_{s,2}), (\omega^n_{s,1}, \omega^n_{s,2}), \Delta\Gamma^s, \Delta\Gamma^n\} \tag{9}$$

where $(\omega_{s,1}, \omega_{s,2})$ is the indexes swapped source vehicles, if both are zero, the system will not swap the source vehicles. $\Delta\Gamma$ is a matrix whose size is the same as $\mathbf{\Gamma}$, and each element of $\Delta\Gamma$ is -1, +1 or 0. As a result, in general, for a certain state, the number of possible actions is

$$|\mathcal{S}\ S\ | = \left(\binom{A^s}{2}+1\right)\left(\binom{A^n}{2}+1\right)\cdot 3^{N\ A^s+A^n} \tag{10}$$

where $\mathcal{S}\ S$ is the set of possible actions in state S. Adding 1 in (9) is because of the "doing nothing" action. From (10) we can find that even we already try to control the number of actions, it is also a relative huge number. Therefore, the traditional Q-Learning algorithm is hard to deal with this situation. We need the Deep Learning algorithm to predict the reward value of different actions in the current state.

After taking an action in the current state, the system will observe the new state and get a reward. The accurate reward needs to calculate the average delays in (2), but as we are developing an online algorithm, we need to simplify the average calculation as follows,

$$E_i\ v_s\ = \sum_{j\in\mathcal{N}, j\ne v_s} P_{i,j}\cdot\left(s_i\cdot d^{\mathrm{MST}}_{j,v_s} + n_{j,v_s}\cdot T_q\right) \tag{11}$$

where $E_i\ v_s$ is the transmission cost for putting the data of application i on vehicle v_s, d^{MST}_{j,v_s} is the distance between vehicle j and vehicle v_s in the MST on the whole vehicle set. The lengths of each edge are defined as (1). n_{j,v_s} represents the number of edges from vehicle j to vehicle v_s in the MST. The computation cost of application a_i can be considered as

$$E_i\ v_s, \Gamma_i\ = \max_{j\in\mathcal{N}, j\ne v_s}\left\{\frac{\gamma_{i,j}\ 1+k_r}{C_{j,v_s}} + \frac{1}{k_c}\right\} \tag{12}$$

Therefore, we define the reword of state S as:

$$\begin{aligned}R\ S\ = & -\{\textstyle\sum[E^s_i\ v_s\ + E^s_i\ v_s, \Gamma_i\ ]\\ & +k_n\textstyle\sum I_{T_n}[E^s_i\ v_s\ + E^s_i\ v_s, \Gamma_i\ ]\}\end{aligned} \tag{13}$$

where $I_{T_n}[x]$ equals 0 when $x \le T_n$, otherwise, it equals 1. k_n is a constant coefficient.

From [28], the maximum action-value function approximation in deep reinforcement learning is

$$Q^*\ S, a; w\ \approx E_{s'}\left[R\ S'\ + \lambda\max_{a'} Q^*\ S', a'\ \right] \tag{14}$$

where w is the parameter of the deep neural network. And the update function for the neural network is

$$L(w) = E\left[\left(R + \lambda \max_{a'} Q^*(S', a'; w) - Q(S, a; w)\right)^2\right] \quad (15)$$

And the action-value function update formula is

$$Q(S,a;w) \leftarrow Q(S,a;w) + \alpha\left(R(S') + \lambda \max_{a} Q(S',a;w) - Q(S,a;w)\right) \quad (16)$$

With the deep reinforcement learning framework and the state, action, reward we defined in (8), (9) and (13), respectively, we can develop the following resource management algorithm.

ALGORITHM I
DEEP REINFORCEMENT LEARNING BASED RESOURCE MANAGEMENT ALGORITHM

1. Initialize $Q(S,a;w)$ with random number.
2. Set $\varepsilon = 1$.
3. **REPEAT**
4. $r = \mathrm{random}(0,1)$.
5. **IF** $r < \varepsilon$ **THEN**
6. $a = \mathrm{random\ action}$
7. **ELSE**
8. $a = \arg\max_{a} Q(S,a;w)$
9. **END IF**
10. Take the action a to get a new state S'.
11. Calculate the reward $R(S')$ with (13)
12. Update $Q(S,a;w)$ with (16)
13. Update *w* by minimizing $L(w)$ in (15)
14. $S \leftarrow S'$
15. $\varepsilon \leftarrow \max\{\varepsilon - \Delta\varepsilon, 0\}$
16. **END REPEAT**

VI. PERFORMANCE EVALUATION

This section presents the simulation results of the proposed algorithm for the performance analyzing. For comparison, we also implement a Q-Leaning version of the proposed algorithm. The key simulation parameters are shown in TABLE I.

TABLE I
SIMULATION PARAMETERS

Parameter	Value
k_c	2
k_r	0.5
T_q	Half of application number
P	100W
N_0	20dBm
α'	-2
d	Distance between adjacent vehicle, 7m
$P_{i,j}$	Random (0, 1)
k_n	50
$\Delta\varepsilon$	0.05
α	0.2
λ	0.1

Fig.5 presents the rewards of each iteration in different traffic loads. Before the eleventh iteration, although the trends of the rewards are increasing, the graph shows a relative huge fluctuation. This is because in this early period, the deep reinforcement learning algorithm has little knowledge of the environment. It is in the exploration manner in this state. Moreover, the high value of ε make the algorithm tends to make random choice to get more information of the environment. After eleventh iteration, the rewards converge to a small value fast compare to the initial state. Because the rewards in different traffic loads vary dramatically, we show the normalized rewards in Fig.5, which is obtained by dividing each reward by the absolute value of the minimum reward.

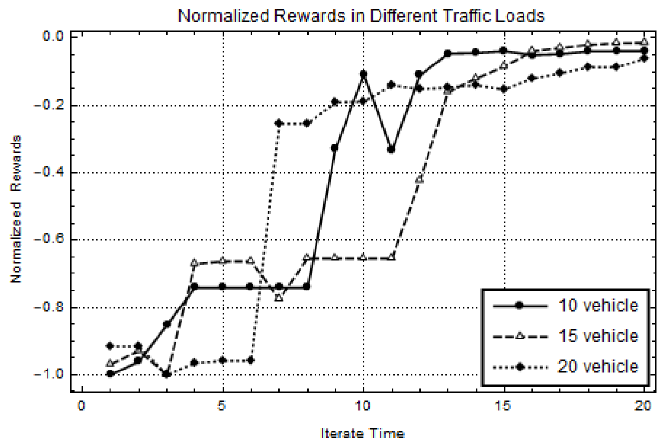


Fig. 5. Rewards in different traffic loads

We illustrate the PRT in different traffic loads for each iteration in Fig. 6. Before the tenth iteration, the PRT is relatively high, especially in the heavy traffic loads scenario. This is because the search space of the heavy traffic is considerably huge, and the transmission delay can be very high for the long-range communication. However, the PRTs converge to the value less than 50m in each scenario. This is caused by the rich computation and storage resources of the vehicular fog in the heavy traffic.

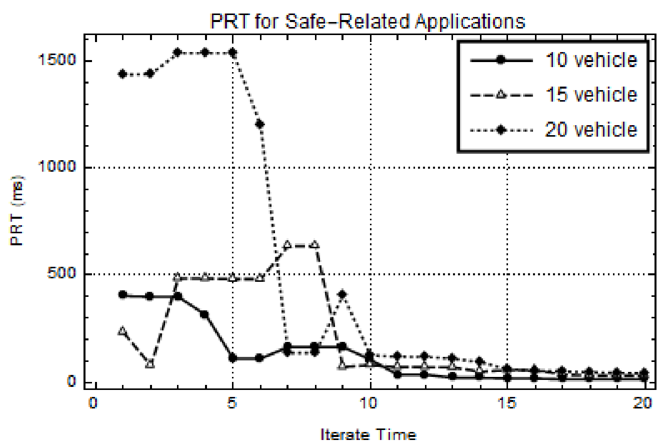


Fig. 6. PRT in different traffic loads

Fig. 7 shows the average delay for the nonsafety related applications. The average delay for the nonsafety related applications has no obvious trend, which is caused by the different reward evaluate method in (13). In the proposed algorithm, we only care about the number of applications that exceed the delay upper bound without minimizing them. As a result, nonsafety applications only need to satisfy the time constraint, which lead to the inapparent trends in Fig. 7.

We compare the deep reinforcement learning and the Q-Learning in Fig. 8. The convergence of the deep reinforcement learning on takes about 10 iterations, while the convergence of the Q-Learning takes about 16 iterations. With the help of the neural network, the deep reinforcement learning will predict the Q-value of unknown states and actions. As a result, the deep reinforcement learning will search more possible and outperforms the Q-Learning in the converge time and the stability.

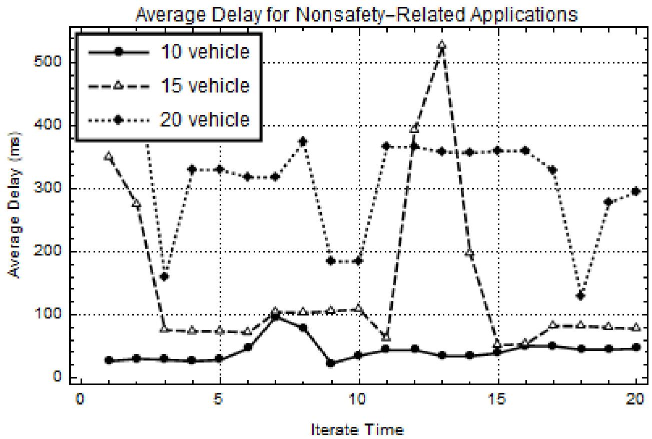


Fig. 7. Average Delay for Non-safety Related Applications

Fig. 9 compares the PRT of different algorithms. The location greedy algorithm first picks the center locations to minimize the communication delay. The resource greedy algorithm chooses the vehicles with rich resources in high priority to minimize the computation delay. The PRT of the greedy algorithms is much higher than the learning-based algorithms. This is because that they only minimize one part but not the entire of the PRT.

Different architectures are compared in Fig. 10. We can find that without task offloading and resource sharing of the Fog, the PRT will increase about 40%. Without the content merging of the ICN, the PRT will be doubled. This result implies that the ICN contributes more in the PRT reducing than the Fog technology. If we remove both resource sharing method and content merging scheme, the PRT will almost tripled. Therefore, the proposed architecture will decrease the PRT about 70%.

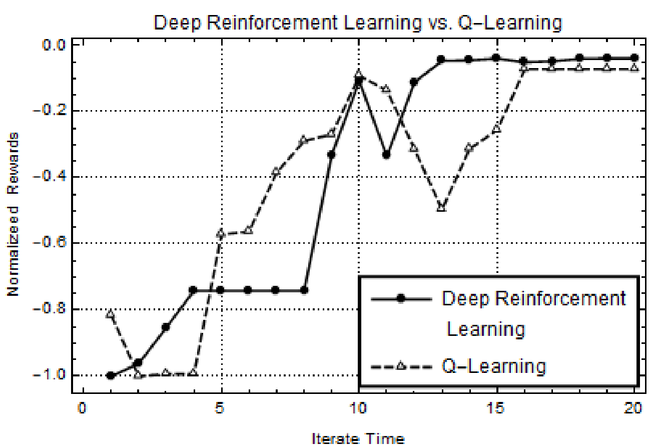


Fig. 8. Proposed algorithm vs. Q-Learning algorithm

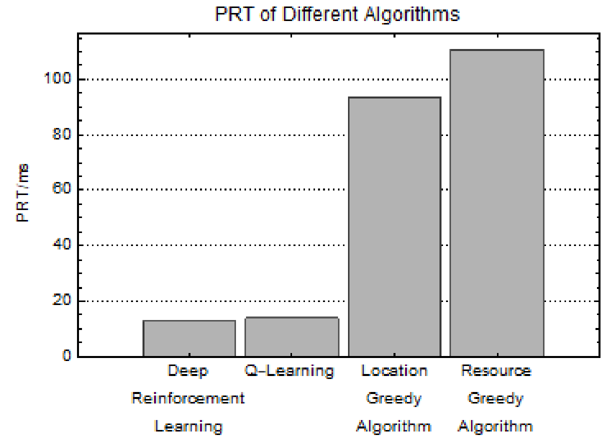


Fig. 9. Comparison of different algorithms

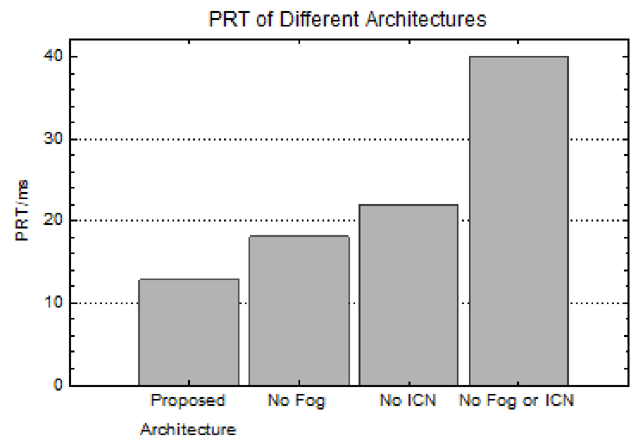


Fig. 10. Comparison of different architectures

VII. Conclusion

In this paper, we reduce the PRT with the vehicular fog computing, ICN and deep reinforcement learning technologies. We formulate the joint computation and storage resources allocation optimization problem with the features of the fog architecture and properties of ICN. A Markov Decision model is developed to describe the resource allocation scheme. Based on this model, we propose a deep reinforcement learning based algorithm to optimize the resource allocation problem. From the simulation results, the proposed algorithm behaves differently on the safety-related applications and nonsafety-related applications. Furthermore, we can find that the deep reinforcement learning converges faster than the Q-Learning algorithm based on the simulation results.

References

[1] E. Lee, E.-K. Lee, M. Gerla, and S. Oh, "Vehicular cloud networking: architecture and design principles," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 148–155, Feb. 2014.

[2] M. Zhu, J. Cao, Z. Cai, Z. He, and M. Xu, "Providing flexible services for heterogeneous vehicles: an NFV-based approach," *IEEE Netw.*, vol. 30, no. 3, pp. 64–71, May 2016.

[3] Susumu Tachi, Masahiko Inami, and Yuji Uema, "Augmented Reality Helps Drivers See Around Blind Spots," *IEEE Spectr.*, vol. 31, 2014.

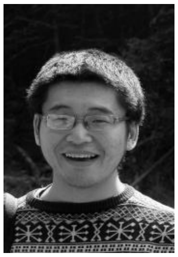
[4] C. Campolo, A. Molinaro, A. Iera, and F. Menichella, "5G Network Slicing for Vehicle-to-Everything Services," *IEEE Wirel. Commun.*, vol. 24, no. 6, pp. 38–45, Dec. 2017.

[5] J. Kennedy and R. Eberhart, “Particle swarm optimization,” *Neural Networks, 1995. Proceedings., IEEE Int. Conf.*, vol. 4, pp. 1942–1948 vol.4, 1995.
[6] X. Wang, S. Leng, and K. Yang, “Social-Aware Edge Caching in Fog Radio Access Networks,” *IEEE Access*, vol. 5, pp. 8492–8501, 2017.
[7] G. Qiao, S. Leng, K. Zhang, and Y. He, “Collaborative Task Offloading in Vehicular Edge Multi-Access Networks,” *IEEE Commun. Mag.*, vol. 56, no. 8, pp. 48–54, Aug. 2018.
[8] F. M. Modesto and A. Boukerche, “SEVeN: A novel service-based architecture for information-centric vehicular network,” *Comput. Commun.*, vol. 117, pp. 133–146, Feb. 2018.
[9] M. Wang, J. Wu, G. Li, J. Li, Q. Li, and S. Wang, “Toward mobility support for information-centric IoV in smart city using fog computing,” in *2017 IEEE International Conference on Smart Energy Grid Engineering (SEGE)*, 2017, pp. 357–361.
[10] F. M. Modesto and A. Boukerche, “An analysis of caching in information-centric vehicular networks,” in *2017 IEEE International Conference on Communications (ICC)*, 2017, pp. 1–6.
[11] S. Arshad, M. A. Azam, M. H. Rehmani, and J. Loo, “Recent Advances in Information-Centric Networking based Internet of Things (ICN-IoT),” *IEEE Internet Things J.*, pp. 1–1, 2018.
[12] H. Baier, H. Baier, and M. H. M. Win, “Nested Monte-Carlo Tree Search for Online Planning in Large MDPs.”
[13] M. Peng, S. Yan, K. Zhang, and C. Wang, “Fog-computing-based radio access networks: issues and challenges,” *IEEE Netw.*, vol. 30, no. 4, pp. 46–53, Jul. 2016.
[14] K. Zhang, Y. Mao, S. Leng, S. Maharjan, and Y. Zhang, “Optimal delay constrained offloading for vehicular edge computing networks,” *IEEE Int. Conf. Commun.*, pp. 1–6, 2017.
[15] K. Wang, H. Yin, W. Quan, and G. Min, “Enabling Collaborative Edge Computing for Software Defined Vehicular Networks,” pp. 1–6, 2018.
[16] J. Zhang, W. Xia, F. Yan, and L. Shen, “Joint Computation Offloading and Resource Allocation Optimization in Heterogeneous Networks with Mobile Edge Computing,” *IEEE Access*, vol. 3536, no. c, pp. 1–1, 2018.
[17] M. Bouet and V. Conan, “Mobile Edge Computing Resources Optimization: a Geo-clustering Approach,” *IEEE Trans. Netw. Serv. Manag.*, vol. 4537, no. c, pp. 1–1, 2018.
[18] K. Xiong, S. Leng, J. Hu, X. Chen, and K. Yang, “Smart Network Slicing for Vehicular Fog-RANs,” *IEEE Trans. Veh. Technol.*, pp. 1–1, 2019.
[19] M. Amadeo, C. Campolo, and A. Molinaro, “Content-centric networking: Is that a solution for upcoming vehicular networks?,” in *Proceedings of the ninth ACM international workshop on Vehicular inter-networking, systems, and applications - VANET ’12*, 2012, p. 99.
[20] C. Xu, W. Quan, H. Zhang, and L. A. Grieco, “GrIMS: Green Information-Centric Multimedia Streaming Framework in Vehicular Ad Hoc Networks,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 2, pp. 483–498, Feb. 2018.
[21] K. Zheng, Y. Cui, X. Liu, X. Wang, X. Jiang, and J. Tian, “Asymptotic Analysis of Inhomogeneous Information-Centric Wireless Networks With Infrastructure Support,” *IEEE Trans. Veh. Technol.*, vol. 67, no. 6, pp. 5245–5259, Jun. 2018.
[22] M. Gadaleta, F. Chiariotti, M. Rossi, and A. Zanella, “D-DASH: A Deep Q-Learning Framework for DASH Video Streaming,” *IEEE Trans. Cogn. Commun. Netw.*, vol. 3, no. 4, pp. 703–718, Dec. 2017.
[23] K. Zhang, Y. Mao, S. Leng, Y. He, and Y. Zhang, “Mobile-edge computing for vehicular networks: A promising network paradigm with predictive off-loading,” *IEEE Veh. Technol. Mag.*, vol. 12, no. June, pp. 2–10, 2017.
[24] K. Zhang, Y. Zhu, S. Leng, Y. He, S. Maharjan, and Y. Zhang, “Deep Learning Empowered Task Offloading for Mobile Edge Computing in Urban Informatics,” *IEEE Internet Things J.*, pp. 1–1, 2019.
[25] K. Katsaros, M. Dianati, R. Tafazolli, and X. Guo, “End-to-End Delay Bound Analysis for Location-Based Routing in Hybrid Vehicular Networks,” *IEEE Trans. Veh. Technol.*, vol. 65, no. 9, pp. 7462–7475, Sep. 2016.
[26] D. Lee and H. Yeo, “A study on the rear-end collision warning system by considering different perception-reaction time using multi-layer perceptron neural network,” in *2015 IEEE Intelligent Vehicles Symposium (IV)*, 2015, pp. 24–30.
[27] X. Zhang, “A new method for analyzing nonsaturated IEEE 802.11 DCF networks,” *IEEE Wirel. Commun. Lett.*, vol. 2, no. 2, pp. 243–246, 2013.
[28] V. Mnih *et al.*, “Playing Atari with Deep Reinforcement Learning,” *Computer Science*, 2013.

Xiaosha Chen received the B.Sc. degree in information and communication engineering from University of Electronic Science and Technology of China, Chengdu, China, in 2016. He is currently working toward the Ph.D. degree in University of Electronic Science and Technology of China. His research interests include handling routing protocols, performance analysis and congestion control in vehicular network.

Supeng Leng is a Full Professor and a Vice Dean in the School of Information & Communication Engineering, University of Electronic Science and Technology of China (UESTC). He is also the leader of the research group of Ubiquitous Wireless Networks. He received his Ph.D. degree from Nanyang Technological University (NTU), Singapore. He has been working as a Research Fellow in the Network Technology Research Center, NTU. His research focuses on resource, spectrum, energy, routing and networking in Internet of Things, vehicular networks, broadband wireless access networks, smart grid, and the next generation mobile networks. He published over 180 research papers in recent years. He serves as an organizing committee chair and TPC member for many international conferences, as well as a reviewer for over 10 international research journals.

Ke Zhang received the Ph.D. degree from University of Electronic Science and Technology of China in 2017. He is currently a Lecturer with the School of Information and Communication Engineering, University of Electronic Science and Technology of China. His research interests include scheduling of mobile edge computing, design and optimization of next-generation wireless networks, smart grid, and the Internet of Things.



Kai Xiong is a Ph.D. degree candidate in the School of Communication and Information Engineering, University of Electronic Science and Technology of China, Chengdu. His research interests include design and optimization of next-generation wireless networks, Internet of Vehicles, Machine Learning, Mobile Edge Computing, and Communication Theory.