Chinese Journal of Electronics
Vol.32, No.6, Nov. 2023

Towards V2I Age-Aware Fairness Access: A DQN Based Intelligent Vehicular Node Training and Test Method

WU Qiong[1,2], SHI Shuai[1], WAN Ziyang[1], FAN Qiang[3], FAN Pingyi[4], and ZHANG Cui[5]

(1. *School of Internet of Things Engineering, Jiangnan University, Wuxi 214122, China*)

(2. *State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China*)

(3. *Qualcomm, San Jose CA 95110, USA*)

(4. *Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing 100084, China*)

(5. *Banma Network Technology Co., Ltd., Shanghai 200000, China*)

Abstract — Vehicles on the road exchange data with base station frequently through vehicle to infrastructure (V2I) communications to ensure the normal use of vehicular applications, where the IEEE 802.11 distributed coordination function is employed to allocate a minimum contention window (MCW) for channel access. Each vehicle may change its MCW to achieve more access opportunities at the expense of others, which results in unfair communication performance. Moreover, the key access parameter MCW is privacy information and each vehicle is not willing to share it with other vehicles. In this uncertain setting, age of information (AoI), which measures the freshness of data and is closely related with fairness, has become an important communication metric. On this basis, we design an intelligent vehicular node to learn the dynamic environment and predict the optimal MCW, which can make the intelligent node achieve age fairness. In order to allocate the optimal MCW for the vehicular node, we employ a learning algorithm to make a desirable decision by learning from replay history data. In particular, the algorithm is proposed by extending the traditional deep-Q-learning (DQN) training and testing method. Finally, by comparing with other methods, it is proved that the proposed DQN method can significantly improve the age fairness of the intelligent node.

Key words — Vehicles, Fair, Chennel access, Age of information (AoI), Deep-Q-learning.

I. Introduction

With the development of the economy and quality of life, people have an urgent demand for a more comfortable driving experience [1]. Therefore, more and more advanced networks and on-board sensing devices have been introduced to enable a large number of applications to meet the demand of users driving on the road. Vehicles on the road have to exchange data with the base station (BS) frequently through vehicle to infrastructure (V2I) communications to ensure the normal use of these applications [2], [3]. These data may include image streams, video streams, network instructions, and shared files. The vehicles adopt the traditional access mechanism IEEE 802.11 (802.11 for short) distributed coordination function (DCF) with a fixed minimum contention window (MCW) size to access channel at the media access control (MAC) layer [4], [5], but it has a drawback that the spectrum utilization is not efficient due to the dynamic change of vehicle density [6]–[9]. In this case, each vehicle may change its MCW to achieve more access opportunities at the expense of others, which results in unfair communication performance [10]. Moreover, the key access parameters MCW is the privacy information and each vehicle is not willing to share it with other vehicles. In this uncertain setting, we design an intelligent vehicular node to learn the dynamic environment and predict the optimal MCW to achieve fair communication performance.

Usually, communication performance is evaluated through wireless communication delay, throughput, and

Manuscript Received Apr. 20, 2022; Accepted June 21, 2022. This work was supported by the National Natural Science Foundation of China (61701197), the Open Research Fund of State Key Laboratory of Integrated Services Networks (ISN23-11), the National Key Research and Development Program of China (2021YFA1000500(4)), and the 111 Project (B12018).

 DOI:10.23919/cje.2022.00.093

quality of service (QoS) [11]. However, these performance metrics may not be able to reflect the freshness of the transmission data that is important to the vehicular applications. As a novel communication metric, the age of information (AoI) has received extensive research attention in recent years [12]–[14]. The AoI metric is different from the traditional performance metrics. It refers to the time interval between the current time and the generation time of the data to be transmitted. As compared to the transmission delay, the AoI can better measure the freshness of the transmitted data [15].

In this paper, we consider the dynamic scenario where the number and MCW of vehicles may change over time. In the scenario, when a new vehicle enters the coverage area of the BS, the new vehicle neither knows the MCW of the other vehicles nor how the other vehicles adjust their own MCW. We designed this newly entered vehicle as an intelligent vehicle node and proposed an extended DQN training and test method to predict the optimal MCW which can ensure its age fairness in the network[*1].

The main contributions are summarized as follows: 1) Considering the dynamic and unknown vehicular networks, we design an intelligent vehicular node to learn the environment and predict the optimal MCW which ensures the age fairness of the intelligent node's transmission data in the network; 2) The age fairness utility of the intelligent vehicular node is defined to measure the relationship between the number of vehicles in the network and the age fairness; 3) We establish a training model by defining the state space, action space, and reward mechanism, and propose an extended DQN training and test method to learn and predict the optimal action at each discrete observation time interval; 4) The performance of the proposed access mechanism is evaluated by simulations under different vehicle characteristics, as compared to other baseline methods.

The rest of this paper is organized as follows. Section II reviews the related work. Section III briefly describes the system model and average AoI of vehicles in the network. Section IV defines the age fairness metric and the deep reinforcement learning (DRL) framework is set up to formulate age fairness problem. Section V presents the extensions DQN algorithm on how to learn the optimal action based on the DRL framework. We present some simulation results in Section VI and conclude this paper in Section VII.

II. Related Work

In this section, we first review the related work on MAC protocol, then we review the existing work on the AoI.

1. MAC protocol

There has been a lot of work to improve the MAC protocol. Lv *et al.* [16] proposed a new function to adjust the contention window (CW) in 802.11 networks to extend DCF. Wu *et al.* [17] proposed a MAC layer protocol that uses the Q-learning algorithm to adjust the contention window in order to provide an effective channel access scheme for various network conditions. Jamali *et al.* [18] proposed a new type of MAC protocol based on 802.11 DCF, which is called adaptive back-off tuning MAC (ABTMAC). It also considers the appropriate transmission attempt rate for both cases where the request to send/clear to send (RTS/CTS) mechanism is not used. Zhou *et al.* [19] proposed a practical distributed back-off algorithm called adaptive contention window back-off (ACWB), which is suitable for 802.11 wireless local area network. It maximizes throughput and fairness based on idle back-off interval statistics. Pressas *et al.* [20] proposed a modified version of 802.11p MAC based on reinforcement learning (RL) to reduce the probability of packet collisions and bandwidth wastage. At the same time, the transmission delay is kept within an acceptable level. Wu *et al.* [21] proposed a routing scheme based on reinforcement learning, which can improve the contention-efficiency of the 802.11p MAC layer and achieve low latency.

2. Age of information

Many works also consider the AoI. To solve the problem that the frequent cache of updates of Internet of things (IoT) sensors may incur considerable energy costs, Wu *et al.* [22] proposed an online cache update scheme to obtain an effective trade-off between average AoI and energy costs. Chen *et al.* [13] investigated the AoI-aware radio resource management problem for an expected long-term performance optimization in a Manhattan grid V2V communication network and proposed a proactive algorithm based on the deep recurrent Q-network. Wu *et al.* [23] considered cellular Internet assisted by UAVs, studied the UAV's AoI minimization problem by designing the UAV's trajectory. Wang *et al.* [24] studied the problem of minimizing the weighted sum of the AoI and the total energy consumption of IoT devices. To minimize the weighted sum of AoI cost and energy consumption, the author proposed a distributed reinforcement learning method to optimize the sampling strategy. Han *et al.* [25] proposed a novel algorithm to solve the optimal blind radio resource scheduling problem in orthogonal frequency division multiplexing access (OFDMA) systems to minimize the

[*1]Simulation codes are provided to reproduce the results in this paper: *https://github.com/qiongwu86/Age-Fairness*

long-term average AoI. Leng *et al.* [26] considered the user scheduling problem on communication sessions, studied the AoI minimization problem in a network composed of energy harvesting transmitters, and formulated an infinite-state Markov decision problem to optimize AoI. Yates *et al.* [12] described the timeliness indicators of AoI and proposed general methods for AoI evaluation and analysis applicable to various sources and systems. Kadota *et al.* [27] considered a wireless network with a BS and derived the lower limit of AoI performance that can be achieved for any given network operating under any queuing rule.

As mentioned above, currently there is no work to design a scheme to jointly optimize the MAC protocol and the age fairness, which motivates us to conduct this work.

III. System Model

In this section, we will describe the system model in detail. It includes scenario model and AoI of the vehicle.

1. Scenario model

As illustrated in Fig.1, we consider a vehicular network with a BS deployed at the roadside of a one-way highway. The number of vehicles moving in the coverage area of the BS is N_v and each vehicle is moving towards the same direction. We divide the time domain into discrete time intervals $n = 1, 2, \ldots$, and the duration of each time interval is T. At the beginning of each observation time interval, the vehicles enter or leave the network according to the Poisson distribution process with the arrival rate λ_v and the departure rate μ_v, respectively.

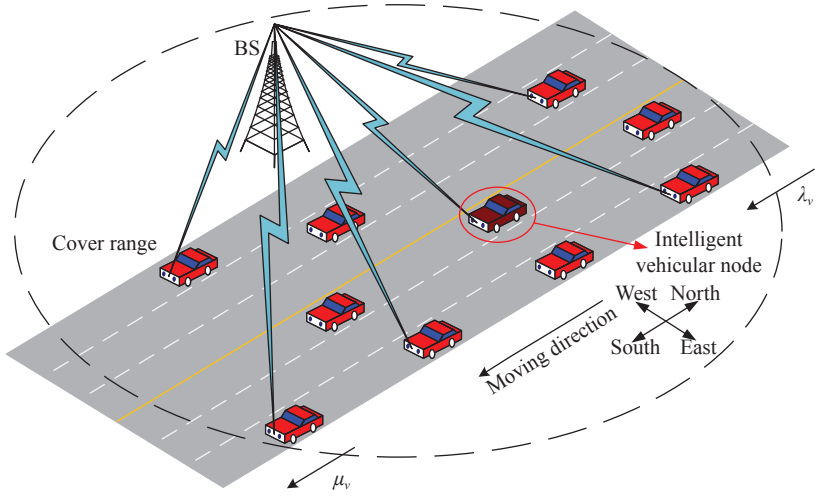

Fig. 1. System model.

Assume that the transceiver is installed on the headstock of each vehicle. Once the headstock of a vehicle enters the coverage of BS, it will transmit the data packet to the BS immediately. We consider the network to be operating in a high-load regime [28], i.e., each vehicle always has a packet to transmit, to explore the extreme performance of our scheme. Each vehicle employs the 802.11 DCF mechanism to transmit the packet. Specifically, a vehicle initializes a back-off process and randomly chooses an integer value from 0 to $W_0 - 1$ as the back-off counter, where W_0 is the minimum contention window. Then the value of the back-off counter is decreases by 1 after each time slot. When the value of the back-off counter decreases to 0, the vehicle captures the channel. Then the vehicle generates a packet and transmits it to the BS over the captured channel. If more than one vehicle is transmitting at the same time, a collision will occur and thus incurs the transmission failure; otherwise the transmission is successful. The above process will be repeated to transmit different packets.

We assume that the MCWs of all vehicles in the network keep constant in each discrete time interval n, i.e., $\boldsymbol{\omega}^n = [\omega_0^n, \omega_1^n, \ldots, \omega_{N_v}^n]$. The MCWs of these vehicles may be changed at different discrete time intervals. We design an intelligent vehicular node, which is referred to as $node_0$ in the following of this paper, and adjust its MCW adaptively to achieve the fair age. Note that $node_0$ cannot obtain the MCW of any vehicle in the network. To achieve the fairness of the AoI, $node_0$ communicates with the BS to obtain the observation data, including the number of vehicles, and learns from the replay history data to select the optimal ω_0^n at each time interval.

2. Age of information of the vehicle

Next, we describe the AoI metric to measure the freshness of data transmitted by a vehicle. The process of 802.11 DCF consists of discrete time slots, thus the

AoI of vehicle i's data at time slot t is defined as

$$a_i^t = \max\{t - v_i^t, 1\}, \forall i \in \mathcal{N} \tag{1}$$

where v_i^t is the time stamp of the generated packet of vehicle i, and $\mathcal{N}$ can be expressed as $\mathcal{N} = \{0, 1, \ldots, N_v\}$, here 0 indicates the intelligent vehicular $node_0$ and N_v is the number of the other vehicles in the network.

To further describe the AoI of vehicle, we assume that at most one packet is cached at each vehicle in the network. As shown in Fig.2, a vehicle starts to transmit a packet to the BS at time slot t_1, the BS successfully receives the packet transmitted by the vehicle at time slot t_2. The vehicle will immediately sample and generate the next packet to transmit at the same time. Note that the next packet is generated by the vehicle at time slot t_2 and arrives at time slot t_3 after a time slot. At this time, the age of the packet has passed a time slot, so the age at time slot t_3 becomes one. In addition, the other vehicles are transmitting at time slot t_4, thus a collision occurs and the AoI of the vehicle still increases by 1 after time slot t_5. The packet is successfully received at the BS at time slot t_6, so a new packet will be sampled immediately at the vehicle and the AoI of the vehicle is reset to 1 at time slot t_7.

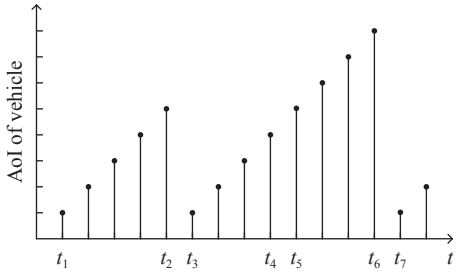

Fig. 2. Illustration of the AoI of vehicle.

Therefore, the AoI of vehicle i at slot $t+1$ can be expressed as

$$a_i^{t+1} = \begin{cases} a_i^t + 1, & \text{BS not receives a packet at slot } t \\ 1, & \text{BS receives a packet at slot } t \end{cases} \tag{2}$$

Therefore, we can get the average AoI of vehicle i in an observation time interval as

$$\overline{\Delta}_i = \frac{T_{\text{slot}}}{T} \sum_{t=1}^{T/T_{\text{slot}}} a_i^t, \forall i \in \{0, 1, \ldots, N_v\} \tag{3}$$

where T_{slot} is the duration time of a time slot, T is the duration of an observation time interval.

IV. Problem Formulation

In order to achieve the age fairness, we use the DRL framework with state, action and reward mechanisms to model the problem of the search for the optimal MCW for $node_0$. Next, state s_n, action a_n and reward r_n for $node_0$ at observation time interval n will be defined, respectively. The parameters used in this paper are listed in Table 1.

Table 1. Notations used in this paper

Notation	Description
a_i^t	The AoI of vehile i at time slot t
a_i^{t+1}	The AoI of vehicle i at time slot $t+1$
a_n	The action of $node_0$ in observation interval n
$\mathcal{A}$	The action set
F_{loss}	The loss of fairness
m	The maximum back-off stage
n	The discrete observation interval
N_d	The number of vehicular nodes in network
$N_{\max}$	The maximum number of vehicles in network
N_v	The number of vehicles in network
$Q_\theta(s_n, a_n)$	The state action value of prediction network
$Q_{\theta'}(s_n, a_n)$	The state action value of target network
r_n	The reward of $node_0$ in observation interval n
$\mathcal{R}$	The reward set
s_n	The state of $node_0$ in observation interval n
$\mathcal{S}$	The state set
T	The duration time of each observation interval
T_s	The average time of a successful transmission
T_c	The average time of a failed transmission
T_{slot}	The duration time of a time slot
v_i^t	The time stamp of the generated packet of vehicle i
W_0	The contention window of a vehicle
$\overline{\Delta}^n$	The observation vector of average AoI at observation intervals n
$\overline{\Delta}_0^n$	The average AoI for $node_0$ at observation interval n
$\overline{\Delta}_v^n$	The average AoI for all vehicles at observation interval n
λ_v	The arrival rate of vehicles
μ_v	The departure rate of vehicles
$\mu(\overline{\Delta}^n, N_d)$	The age fairness utility
$\boldsymbol{\omega}^n$	The MCW vector for all vehicles in the network at observation interval n
ω_0^n	The MCW of $node_0$ at observation interval n
ω_i^n	The MCW of vehicle i at observation interval n
α	The learning rate
β	The discount factor of future utility
γ	The discount factor of state action value
γ_r	The discount factor of immediate reward
π^*	The optimal policy
Ω	The discrete-time stochastic process state space

1. State

The intelligent $node_0$ communicates with the BS within each time interval to obtain the observation information. Since the BS can know the number of vehicles in the network within each time interval n, which is denoted as N_v^n, through communicating with vehicles, $node_0$ can know N_v^n from the BS. Moreover,

the BS calculates the average AoI of all vehicles in each time interval, thus $node_0$ can observe the average AoI of all vehicles for each time interval n, which is noted as $\overline{\Delta}_v^n$. In addition, $node_0$ can monitor the channel to observe its average AoI $\overline{\Delta}_0^n$ in time interval n. At this point, we define the observation vector $\overline{\Delta}^n = [\overline{\Delta}_0^n, \overline{\Delta}_v^n]$ of the AoI in observation interval n. Since the AoI of a vehicle is affected by the MCW of all vehicles in a time interval and each vehicle may change its MCW at each time interval, the age observation vector $\overline{\Delta}^n$ is a random vector for each time interval. Besides, $node_0$ can also know its own MCW ω_0^n in time interval n.

As described above, the state of $node_0$ in the nth time interval can be defined as

$$s_n = [\overline{\Delta}_0^n, \overline{\Delta}_v^n, \omega_0^n, N_v^n] \tag{4}$$

where $\overline{\Delta}_0^n$ and $\overline{\Delta}_v^n$ depends on ω_0^n and the MCW ω_i^n of all vehicles in the network. Since $\overline{\Delta}_0^n$, $\overline{\Delta}_v^n$, ω_0^n and N_v^n are the values within discrete space, the state space of $node_0$ is discrete.

2. Action

The intelligent $node_0$ changes MCW size for the next observation interval $n+1$ according to the s_n locally observed in the observation interval n, so the MCW action space of $node_0$ can be defined as

$$a_n = \{2^{j-1}32\}_{j=1}^{5} \cup \{2^{j-1}48\}_{j=1}^{2} \tag{5}$$

In order to make $node_0$ more selective, its action space not only covers the limited state space Ω of the MCW that can be used by the vehicle, but also can extend to additional MCW options: 48 and 96.

3. Reward

We first define the age fairness utility for $node_0$, then set the reward according to the age fairness utility.

When the network is considered as a saturated state, there are N_v^n vehicles in the network in the time interval n. If the absolute fairness is provided, theoretically, the proportion of the AoI of nodes in the network should be $1/N_d^n$, where N_d^n is the number of nodes (both $node_0$ and the other vehicles) in the network, *i.e.*, $N_d^n = N_v^n + 1$. However, in the real scenario, the proportion of AoI for each node will be given by $\overline{\Delta}_0^n/(\overline{\Delta}_0^n + \overline{\Delta}_v^n)$, the absolute difference between $\overline{\Delta}_0^n/(\overline{\Delta}_0^n + \overline{\Delta}_v^n)$ and $1/N_d$ can be regarded as the fairness loss. According to the relationship between the average age observation vector $\overline{\Delta}^n$ and the number of nodes N_d^n in the network, the fairness loss at time interval n can be defined as

$$F_{\text{loss}}^n = \left| \frac{\overline{\Delta}_0^n}{\overline{\Delta}_0^n + \overline{\Delta}_v^n} - \frac{1}{N_d^n} \right| \tag{6}$$

According to (6), the age fairness utility at time interval n can be defined as

$$\begin{aligned} \mu(\overline{\Delta}^n, N_d^n) &= 1 - F_{\text{loss}}^n \\ &= 1 - \left| \frac{\overline{\Delta}_0^n}{\overline{\Delta}_0^n + \overline{\Delta}_v^n} - \frac{1}{N_d^n} \right| \end{aligned} \tag{7}$$

In this paper, as the number of vehicles and MCW changes, $node_0$ aims to maintain its age fairness utility in the network, so the reward of $node_0$ at the observation interval n is defined as its age fairness utility, i.e.,

$$r_n = \mu(\overline{\Delta}^n, N_d^n) \tag{8}$$

With the above definition, we will further introduce the MCW optimization problem for $node_0$ in the vehicle environment. Specifically, given the age observation vector for the observation interval n, i.e., $\overline{\Delta}^n = [\overline{\Delta}_0^n, \overline{\Delta}_v^n]$, the MCW of $node_0$ for observation interval n, i.e., ω_0^n, and the number of nodes N_d^n, $node_0$ predicts and allocates its MCW for the next observation interval $n+1$, i.e., $\omega_0^n + 1$, to maximize the long-term reward, i.e., the age fairness utility of the entire observation process $\mathbb{E}\left[\sum_{k=n}^{\infty} \beta^{k-n} \mu(\overline{\Delta}^k, N_d^n)\right]$, where $\beta \in (0,1)$ is used to establish the importance of future utility.

4. Policy

The action taken by $node_0$ in each state, i.e., $a_n = \pi(s_n)$ is determined by the policy π that maps from the state set $\mathcal{S}$ to the action set $\mathcal{A}$. The goal of Q-learning is to find the optimal policy π^* to maximize the long-term expected cumulative discount reward [29]. To do this, we define the optimal Q function $Q^* : \mathcal{S} \times \mathcal{A} \to \mathcal{R}$, where $Q^*(s_n, a_n)$ corresponds to the long-term reward when choosing action a_n in state s_n, i.e., following the optimal policy π^*. Therefore, the optimal policy is defined as

$$\pi^*(s_n) = \underset{a_n}{\text{argmax}}\ Q^*(s_n, a_n) \tag{9}$$

V. Solution

Since the state and action space are discrete, the DQN algorithm is more suitable to solve the DRL problems. Therefore, we propose an extended DQN algorithm to obtain the optimal action policy. In this section, we first introduce how we extend the DQN training and testing method to better solve the above problem, then describe the training procedures to obtain the optimal age fairness utility, finally introduce how to test the performance under the optimal policy.

1. Extensions to DQN

We extend the traditional DQN from five aspects,

i.e., double Q-learning, dueling networks, multi-step learning, distributional reinforcement learning, and noisy nets. The flow diagram of the extended DQN algorithm is shown in Fig.3.

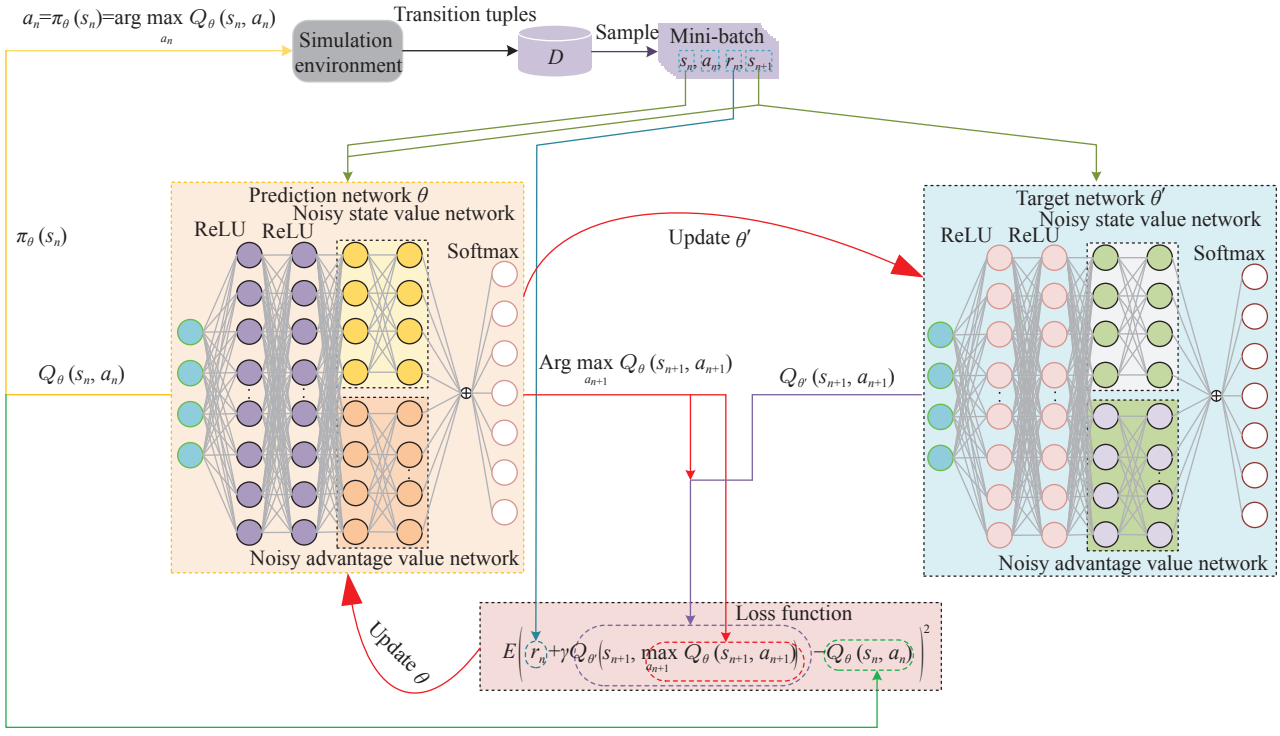

Fig. 3. Flow diagram of extended DQN.

1) Double Q-learning

The classical method of searching for the optimal Q value is based on the value iteration method of Bellman equation [30], i.e.,

$$Q_\theta(s_n, a_n) \leftarrow Q_\theta(s_n, a_n) + \alpha\left(r_n + \gamma \max_{a_{n+1}} Q_\theta(s_{n+1}, a_{n+1}) - Q_\theta(s_n, a_n)\right) \tag{10}$$

where $Q_\theta(s_n, a_n)$ is the state action value when action a_n is selected under state s_n, α is the learning rate, r_n is the immediate reward obtained by taking action a_n under the state s_n, γ is the discount factor, and $Q_\theta(s_n+1, a_n+1)$ is the state action value when action a_{n+1} is selected under state s_{n+1}. However, only one neural network parameter θ is used in (10) for Q value iteration, which results in the overestimation bias, thus it affects the Q value and further affects the performance of learning. Double Q-learning is designed in calculating the target Q value with two different neural network parameters including the prediction network parameter θ and the target network parameter θ' [31]. The particular formula is shown as follows,

$$Q_\theta(s_n, a_n) \leftarrow Q_\theta(s_n, a_n) + \alpha\left(r_n + \gamma Q_{\theta'}(s_{n+1}, \max_{a_{n+1}} Q_\theta(s_{n+1}, a_{n+1})) - Q_\theta(s_n, a_n)\right) \tag{11}$$

The prediction network is responsible to select the action and the target network is used to calculate the target Q value. By decoupling the selection of actions and the estimation of the value of state actions, the harmful overestimation of DQN is reduced, thereby improving the stability of the algorithm [32].

2) Dueling networks

Different from the traditional DQN where the neural network directly outputs the Q value of each action, the dueling DQN first divides the fully connected layer of the network into an output state value $V(s_n)$ and an output action advantage value $A(s_n, a_n)$. The state value function $V(s_n)$ has nothing to do with actions, but represents the value of the static state environment itself. Meanwhile, action advantage function $A(s_n, a_n)$ is related to the action and represents the additional value of choosing an action. In addition, the action advantage value function can reflect the difference between the value obtained by taking an action and the average value obtained by this state. Finally, the two features are aggregated in a fully-linked manner and nonlinear activation is performed through the Softmax activation function to obtain the Q value of each action. In practical applications, the action advantage function is generally set as the single action advantage function minus the average value of all the action ad-

vantage functions in a certain state, i.e.,

$$Q_\theta(s_n, a_n, \alpha^n, \beta^n) = V_\theta(s_n, \beta^n) + A_\theta(s_n, a_n, \alpha^n) - \frac{\sum_{a'_n} A_\theta(s_n, a_n', \alpha^n)}{N_{\text{actions}}} \tag{12}$$

where α^n and β^n represent the number of two fully linked layers of the neural network, respectively, $V_\theta(s_n, \beta^n)$ is the state value of state s_n, $A_\theta(s_n, a_n, \alpha^n)$ is the action advantage value when action a_n is selected under state s_n, N_{actions} is the number of actions provided by the neural network, a'_n is all available actions. Based on this, $node_0$ can finally achieve a more accurate $V(s_n, \beta^n)$ in the environment without the influence of actions, i.e., it can narrow the range of Q value, remove redundant degrees of freedom and further improve the stability of the algorithm [33].

3) Multi-step learning

The traditional DQN uses the immediate reward r_n and the value estimate at the next step as the target value. However, with a large number of deviations in the network parameters at the early stage, the target value obtained by this method will also require a large number of deviations, and thus incur a relatively slow learning rate [34], [35]. A more accurate estimate can be obtained by changing single-step learning to multi-step learning, because immediate rewards can be accurately obtained through interaction with the environment, so the immediate reward can be rewritten as

$$r_n^{(k)} \equiv \sum_{j=0}^{k-1} \gamma_r \; r_{n+j} \tag{13}$$

where γ_r is the discount factor of immediate reward. We can replace r_n in (11) with $r_n^{(k)}$ in (13) to minimize the loss of action value. Note that, proper adjustment of the number of steps in multi-step learning can achieve faster learning.

4) Distributional RL

In the traditional DQN, the outputs of the network are always the expected estimated value of the state-action value Q. However, the expectations for different groups of state-action values may be the same, we should choose a more stable action if we want to reduce the risk of taking actions. Distributed DQN based deep reinforcement learning models are designed from a distributed perspective. It uses a histogram to represent the estimate of the value distribution, limits the value among $[v_{\min}, v_{\max}]$ and selects N equidistant value sampling points in $[v_{\min}, v_{\max}]$. Thus the output of the network is the probability of these N value sampling points, where the sampling point with the largest probability is the optimal action to be selected. After projecting the state-action value output by the neural network onto the vector $\boldsymbol{z}$, the action with the highest probability becomes the stable action we expect. Among them, the calculation method of each element in $\boldsymbol{z}$ is

$$z^i = v_{\min} + (i-1)\frac{v_{\max} - v_{\min}}{N-1}, i \in \{1, \ldots, N\} \tag{14}$$

5) Noisy nets

In the learning process, $node_0$ needs to perform a lot of action explorations, but the ε-greedy algorithm has limitations in performing exploration [36]. In order to improve the search ability of the agent, by combining the determinism and the noise linear hidden layer of the noise flow, i.e., the weight and the bias are interfered by some parameter zero-mean noise, the existing expression $y = wx + b$ of the neural network hidden layer can be modified as

$$y = (\mu^w + \sigma^w \odot \varepsilon^w)x + (\mu^b + \sigma^b \odot \varepsilon^b) \tag{15}$$

Here, $\odot$ denotes the element-wise product, μ^w and μ^b obey a uniform distribution ranging of $[-\frac{1}{\sqrt{N_n}}, \frac{1}{\sqrt{N_n}}]$, N_n is the number of neural network nodes, σ^w and σ^b can be initialized via $\frac{S_{td}}{\sqrt{N_n}}$, and S_{td} is the neural network parameter. In addition, ε^b and ε^w are the constant randomly generated noise variables in each exploration, which can be expressed as

$$\begin{cases} \varepsilon^w = f(\varepsilon_i) \\ \varepsilon^b = f(\varepsilon_j) \end{cases} \tag{16}$$

where ε_i and ε_j follow the standard normal distribution with mean 0 and variance 1, $f(x)$ can be expressed as $f(x) = \text{sgn}(x)\sqrt{|x|}$. Over time, the network can overcome different rates of noise flow through self-learning, thereby completing the exploration of actions in the form of self-annealing.

2. Training stage

The pseudocode of the proposed algorithm is described in Algorithm 1. For ease of understanding, we will further introduce the proposed DQN algorithm in detail below in conjunction with the pseudocode.

Algorithm 1 Training stage for extended DQN

1: Create simulation environment, initialize N_v according to the λ_v and μ_v, initialize ω_i^n;
2: Create predict network and target network;
3: Initialize replay experience buffer $\mathcal{D}$ to capacity $|\mathcal{D}|$;
4: Reset simulation environment;
5: FOR episode = 1 to $E_{\max}$
6: Initialize observation sequence $s_0 = [\overline{\Delta}_0^0, \overline{\Delta}_v^0, \omega_0^0, N_v]$

and normalized sequenced $\phi_0 = \phi(s_0)$;
7: FOR time interval $n = 1$ to $T_{\max}$
8: Generate the action a_n according to (9);
9: Execute next state s_{n+1} and observe reward r_n from the system model;
10: Normalized sequenced $\phi_{n+1} = \phi(s_{n+1})$;
11: Store transition $(\phi_n, a_n, r_n, \phi_{n+1})$ in $\mathcal{D}$;
12: IF number of tuples in $\mathcal{D} \geq I$ THEN
13: Randomly sample a mini-batch of I transitions tuples from $\mathcal{D}$;
14: Update the predict network by minimizing the loss function according to (17);
15: IF time interval $n = T_{\max}$ THEN
16: Update target networks $\theta' \leftarrow \theta$.

First, we create the simulation environment. In this step, we declare the available set of MCWs of both $node_0$ and vehicles, initialize the MCW value of $node_0$ and vehicles at the beginning and initialize the number of vehicles in the network according to the vehicle arrival rate λ_v and departure rate μ_v in the network.

Then we create neural networks for $node_0$. We use two neural networks to create our extended DQN network, i.e., the prediction network and the target network. The prediction neural network consists of four layers. The first two layers are ordinary fully connected layers and the latter two layers are noise nets, where each layer is composed of N_n neural network nodes. Each layer of the neural network is fully connected and the output of each node uses ReLu activation function for non-linear activation. In addition, the input of the neural network is the state, while the output is the state action value of all available actions. We also introduce the dueling network to build a neural network. The prediction network starts from the third layer and is divided into a state value network and an action advantage value network. The state value network and the action advantage value network share the first two layers of fully connected neural networks. The two-way features are aggregated together in a fully-linked manner and the Softmax activation function is used for nonlinear activation before outputting the Q value. Then according to our description of distributed RL, the state action value is limited to $[v_{\min}, v_{\max}]$. By selecting N equidistant value sampling points in $[v_{\min}, v_{\max}]$, the network outputs the projection vector $\boldsymbol{z}$ of these N value sampling points and finally outputs the action index value with the largest probability value, i.e., the most stable action. The target network has the same structure as the prediction network and will not be described in detail here.

Then we initialize the experience buffer $\mathcal{D}$ to store the state, action, reward and next state set during the training process. The experience buffer has the storage capacity of $|\mathcal{D}|$ sets, which corresponds to line 3 of the pseudocode.

Before starting the training, we need to reset the environment, which corresponds to line 4 of the pseudocode.

At the beginning of the loop, we initialize the state set for $node_0$ and execute it at the beginning of each episode. In addition, to facilitate the training process of neural networks and reduce the magnitude of data processing, we have carried out data normalization processing on the state set of $node_0$. This corresponds to line 6 of the pseudocode.

For each observation interval n, we input the state into the neural network above to decide the action. At the initial stage of training, due to the influence of the randomly initialized parameters, the action will be randomly given by the neural network. This corresponds to line 8 of the pseudocode.

In the training process, the MCW of vehicles in the environment will change according to a Markov process and the number of vehicles will change momentarily in each time interval n. After $node_0$ selects the MCW, it obtains the next state s_{n+1} by interacting with the environment and can calculate the immediate reward for each state transition according to (8). This corresponds to line 9 of the pseudocode.

After getting the next state s_{n+1}, we also need to execute normalization for the convenience of processing. Then we cache the current state ϕ_n, the action a_n, the next state ϕ_{n+1} and the immediate reward r_n into the experience buffer $\mathcal{D}$ for subsequent parameter training.

If the number of samples in the experience buffer is less than the mini-batch size, the interaction with the environment will continue and the sample will be cached into the experience buffer. On the contrary, a sample is randomly picked up from the experience buffer to train neural network parameters according to the defined loss function [37]. The loss function is defined as

$$\mathcal{L}(\theta) = \mathbb{E}\left(r_n + \gamma Q_{\theta'}(s_{n+1}, \max_{a_{n+1}} Q_\theta(s_{n+1}, a_{n+1})) - Q_\theta(s_n, a_n)\right)^2 \tag{17}$$

The neural network parameters are gradually updated through gradient descent by backpropagation, i.e., $\theta \leftarrow \theta - \alpha \nabla \theta$, where $\alpha = 0.0001$ is the learning rate of gradient descent. This corresponds to lines 13 to 14 of the pseudocode.

Finally, at the end of each episode, we will update the parameters θ' of the target network through the neural network parameters θ of the prediction network, i.e., $\theta' \leftarrow \theta$. This corresponds to the end of the pseudo-

code.

Through continuous iterative calculations, the neural network parameters of double DQN will finally approach the optimal θ^*, and the training process is completed at this time.

3. Testing stage

To evaluate the performance of the proposed algorithm, we load the optimal neural network parameters θ^* obtained in the training stage and start to test its performance. The pseudocode of the testing stage is shown in Algorithm 2.

Algorithm 2 Testing stage for extended DQN

1: Create simulation environment;
2: Import the trained neural network parameters;
3: FOR episode = 1 to $E_{\max}$
4: Initialize N_v according to the λ_v and μ_v;
5: Initialize ω_i^n;
6: Initialize initial observation state $s_0 = [\overline{\Delta}_0^0, \overline{\Delta}_v^0, \omega_0^0, N_v]$ and normalized sequenced $\phi_0 = \phi(s_0)$;
7: FOR time interval $n = 1$ to $T_{\max}$
8: Generate the action according to $\pi^*(s_n) = \underset{a_n}{\operatorname{argmax}}\, Q_{\theta^*}^*(s_n, a_n)$;
9: Execute action a_n, observe reward r_n and new state s_{n+1} from the system model;

VI. Numerical Simulation

In this section, we evaluate the performance of the algorithm by extensive simulation experiments and discuss the results in detail. The simulation parameters are shown in Table 2. The simulation is based on Python 3.8. The system scenario is a one-way highway network scenario as described in Section III. We have considered two scenarios with dynamic MCW and number of vehicles, i.e., the simple vehicle scenario and the complex vehicle scenario mentioned in the below simulation analysis. In these two scenarios, the simulation analysis is carried out when the MCW transition probability ps of all vehicles is 1.0 and 0.75 respectively.

Table 2. Related parameters

Parameters of system model			
Parameter	Value	Parameter	Value
N_v	0–9	N_d	1–10
λ_v	1–6	μ_v	1–6
T	1 s	T_{slot}	50 μs
T_s	179.64 μs	T_c	174.26 μs
m	3	$N_{\max}$	0–9
Parameters of extended DQN			
Parameter	Value	Parameter	Value
$E_{\max}$	200/1000	$T_{\max}$	200/400
N_n	64/480	S_{td}	0.4
α^n	64/480	β^n	64/480
$v_{\min}$	45/43	$v_{\max}$	50
γ	0.99	α	0.0001
γ_r	0.99	β	0.99
$\lvert\mathcal{D}\rvert$	10000/100000	I	32

In the simple vehicle scenario, the MCWs of all vehicles follow a Markov process with only two states $[s_1, s_2]$. In our simulation experiment, these two states are $[32, 128]$, and we set $\omega_i^n \in [32, 128]$, $\forall i \neq 0$. In the second complex vehicle scenario, the MCWs of all vehicles follow a Markov process with five states $[s_1, s_2, s_3, s_4, s_5]$. In our simulation experiment, these five states are [32, 64, 128, 256, 512], and we set $\omega_i^n \in$ [32, 64, 128, 256, 512], $\forall i \neq 0$. At each discrete time interval, the MCWs of all vehicles in the network transit to the next state with probability ps, and the state changes follow an increasing order from s_1 to s_5, and then a decreasing order from s_5 to s_1, so on and so forth.

1. Training stage

We first obtain the age dataset based on real-time protocol simulation. We will train the model for $node_0$ based on this dataset. To make our results more general, we train the model 10 times in different scenarios and averaged the convergence curves.

Fig.4 shows the learning curve of the training process in the simple vehicle scenario, which reflects the average age fairness utility in each step, i.e., discrete time interval, under different episodes. It can be seen that the average age fairness utility gradually increases from episode 0 to episode 60 when $ps = 1.0$. Then the age fairness utility turns to be stable from episode 60 to 200, which means that the optimal MCW adjustment policy for this simple vehicle scenario has been learned. For the convergent curve, we can see that it is not smooth, this is mainly because in the training process of the model, we use the real simulation value of the real-time protocol to calculate the age fairness utility. In addition, due to the randomness of the integer selection of the back-off counter, and the impact of back-off freezing, collision transmission, and successful transmission during device communication, there will be a slight difference in the average age statistics for each time interval. In addition, for the scenario where the state transition probability $ps = 0.75$, it can be seen that the curve has the same convergence trend as the scenario of $ps = 1.0$, but the age fairness utility after convergence is lower than that when $ps = 1.0$, and the amplitude of its fluctuation is higher than that in $ps = 1.0$ scenario. This is mainly because when the transition probability is not 1.0, the changes of MCW states of other vehicles will become complicated and no longer have a regularity to follow. At the same time, the number of vehicles changes dynamically at each time interval, which makes

the MCW prediction and adjustment of $node_0$ challenging. Therefore, the prediction may be undesirable in some steps, resulting in a decrease in the average age fairness utility.

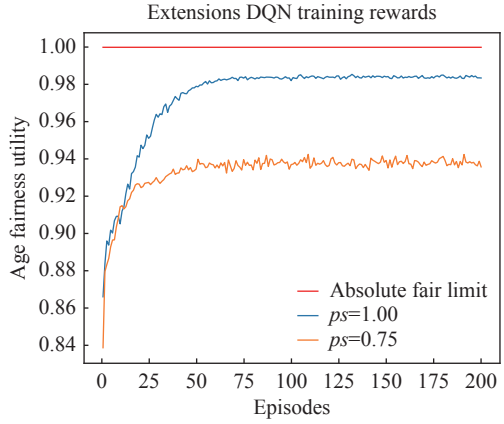

Fig. 4. Learning cuve of simple vehicle scenario.

In Fig.4, the red straight line represents the absolute fairness limit, because according to equation (4), when the age is completely fair, the fairness loss should be 0. Therefore, according to (5), for each step (i.e., one time interval) of each episode, the age fairness utility is always 1. However, in the real environment, even if different vehicles use the same MCW, their ages cannot be completely equal because of the randomness of the integer selection of the back-off counter in the exponential back-off mechanism. Therefore, we only use it to express the absolute fairness upper limit. In the simulation analysis of the following training stage and testing stage, we will no longer describe the absolute fairness limit.

Fig.5 shows the learning curve of the training process in the complex vehicle scenario. It can be seen that the tendency of the curve is similar to that in the simple vehicle scenario, which means that the proposed scheme is suitable for different dynamic scenarios, except that the average age fairness utility surged from episode 0 to 100. The curve increases slowly from episode 100 to 500 and reaches relative stability from 500 to 1000. The difference is that when $ps = 1.0$, the age fairness utility after convergence is lower than that in the simple vehicle scenario. This is mainly because the MCWs of all vehicles in complex vehicle scenario will be more difficult to predict than that in simple vehicle scenario. However, when $ps = 0.75$, the age fairness utility decreases slightly compared with $ps = 1.0$, which means that the proposed scheme has similar adaptability to some complex scenarios. In addition, compared with the simple vehicle scenario, the age fairness utility of complex vehicle scenario converges at around 500 episodes, which also indicates that more model training

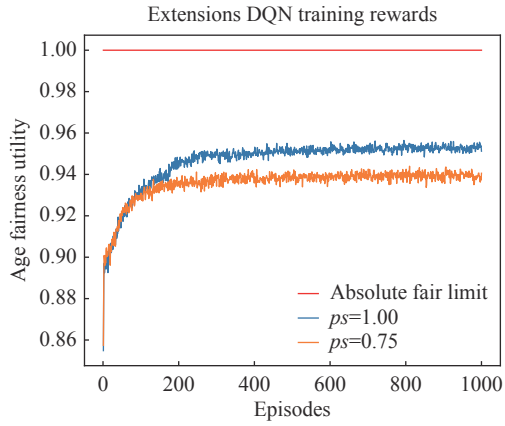

Fig. 5. Learning cuve of complex vehicle scenario.

time is required for the complex vehicle scenario.

2. Testing stage

In the testing stage, $node_0$ adopts the trained model to achieve the age fairness.

Figs.6(a) and (b) show the test results of age fairness utility in simple vehicle scenarios. To restore the real scenario, we only execute the simulation in different situations for one time, without averaging the age fairness utility. Fig.6(a) is the test result of 200 episodes (each episode contains 200 steps, i.e., 200 time intervals). It can be seen that during the test, for both $ps = 1.0$ or $ps = 0.75$, the age fairness utility converges similar to that in Fig.4, which proves the reliability of our model. In addition, Fig.6(b) shows the test result of the age fairness utility in an episode for the simple vehicle scenario with $ps = 1.0$. The yellow line is the average age fairness utility of 200 steps. Since we are using the training dataset of the real protocol simulation, we can see that the utility of most steps fluctuate around the average value. It is worth noting that the age fairness utility of some steps is 1, which means that only $node_0$ exists in the network at this time; thus there is no fairness problem or we can also say that it is absolutely fair for the network. For individual points where the age fairness utility is less than 0.95, such as 90, 119, 175, 199, $node_0$ may make an error in predicting the MCW, which causes a sudden decrease in the age fairness utility, but we can see that we still get a relatively high average age fairness utility.

Figs.7(a) and (b) show the test results of age fairness utility in complex vehicle scenarios. Fig.7(a) is the test result of 1000 episodes (each episode contains 200 steps, i.e., 200 time intervals). It can be seen that during the test, for both $ps = 1.0$ and $ps = 0.75$, the age fairness utility converges similar to that in Fig.5, which proves the reliability of our model. Different from Fig.6(a), the age fairness utility in Fig.7(a) fluctuates more greatly, and when $ps = 1.0$, the age fairness utility is lower than that in Fig.6(a). This is mainly be-

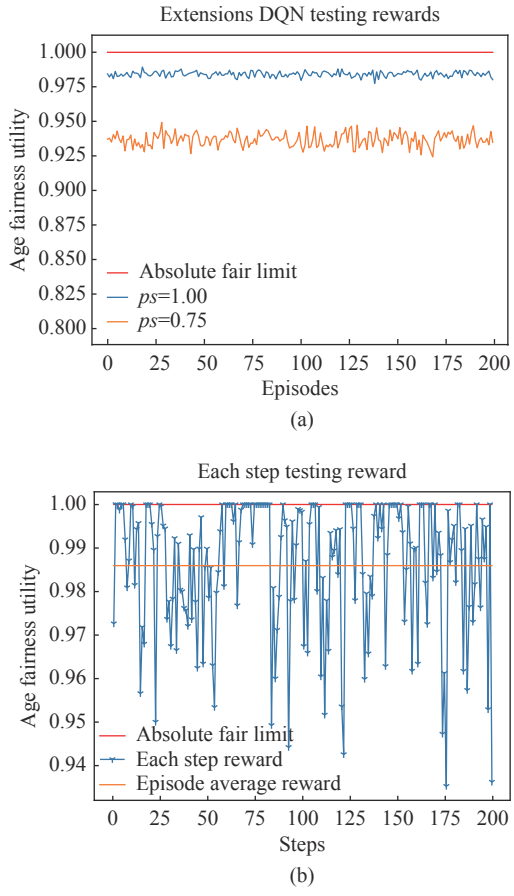

Fig. 6. Testing cuve of simple vehicle scenario.

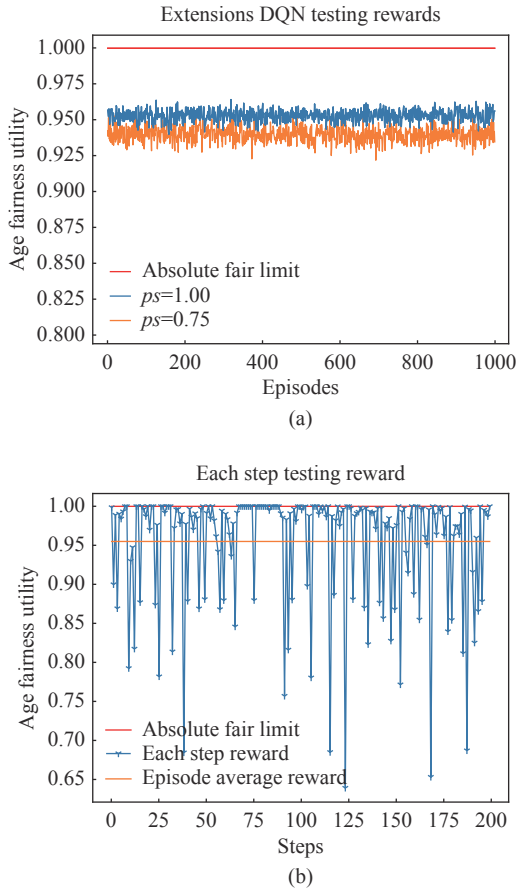

Fig. 7. Testing cuve of complex vehicle scenario.

cause for scenarios where MCW changes are more complex, the accuracy of MCW prediction will be degraded, but this does not affect the superiority of our method. Fig.7(b) shows the test result of the age fairness utility of 200 steps in an episode of the complex vehicle scenario with $ps = 1.0$. In addition, our description of the curve in Fig.7(b) is the same as that in Fig.6(b). It can be seen that Fig.7(b) has a lower average age fairness utility than Fig.6(b) (i.e., yellow straight line in Fig.6(b)). It can be seen that it has gone through more steps before the age fairness utility decreases significantly. It is mainly because the MCW changes for the more complex vehicle scenario are more complicated, which will degrade the accuracy of $node_0$ predictions of the MCW, thus reducing the age fairness utility. However, it can be seen that the age fairness utility in the steps is still around the average value. It indicates that although the performance has been slightly reduced, $node_0$ can still adapt to the complex scenario.

3. Vehicle characteristic analysis

We simulate a complex vehicle scenario and average the test results of 1000 episodes. Fig.8 shows the age fairness utility when the maximum number of vehicles changes.

The vehicle arrival rate and departure rate equal to 3. When the number of vehicles in the network is greater than 1, i.e., there are vehicles in the network, for both $ps = 1.0$ and $ps = 0.75$, we can see that different numbers of vehicles may incur similar age fairness utility. Ideally, they should have exactly the same age fairness utility. But as we described earlier, we calculate the age fairness utility based on the age dataset obtained by the actual protocol simulation, there will be slight errors in the case of different numbers of vehicles, so the complete equality cannot be achieved.

Fig.9 shows the age fairness utility under different

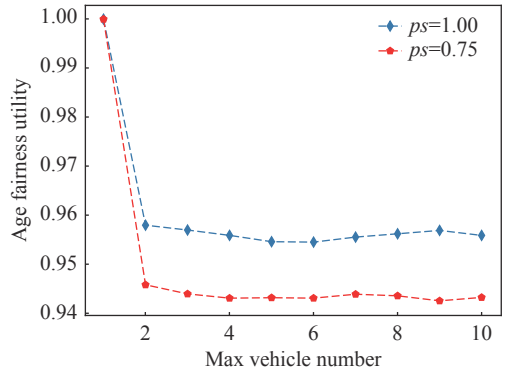

Fig. 8. Different max vehicle number rewards cuve of complex vehicle scenario.

arrival rates when the maximum number of vehicles is 6 and the departure rate is 3. We can see that in each observation interval n, due to constant departure rate, the number of vehicles in the network is very small when the arrival rate is very low. As there is only $node_0$, it is absolutely fair for the network. However, as the arrival rate increases, the number of vehicles in each time interval n in the network gradually increases, thus leading to degradation of the age fairness utility. But the curve will eventually stabilize. The jitter ranging from 3 to 6 is attributed to the fact that age data is derived from the real protocol simulation.

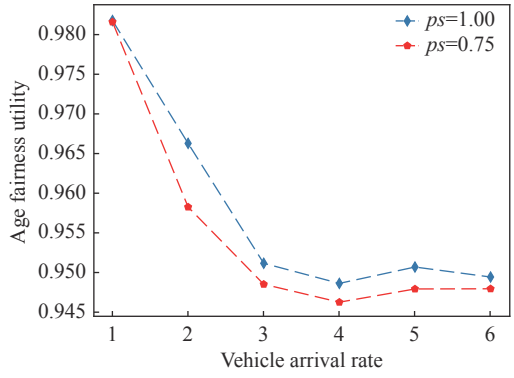

Fig. 9. Different arrival rate rewards cuve of complex vehicle scenario.

Fig.10 shows the age fairness utility under different departure rates when the maximum number of vehicles is 6 and the arrival rate is 3. We can see that in each observation interval n, due to the constant arrival rate, there are many vehicles in the network when the departure rate is very low. However, as the departure rate increases, the number of vehicles in each time interval n in the network gradually decreases, thus leading to an increase in the age fairness utility. As the departure rate reaches 6, there will be only $node_0$ in the network for a long time, which is absolutely fair for the network, and thus the age fairness utility reaches the maximum.

4. Performance evaluation

We set the maximum number of vehicles to 6, the arrival rate, and the departure rate to 3, and evaluate the performance of our RL method and the following baseline methods in two scenarios.

Optimal (OPT) This method is for $node_0$ which is fully aware of changes in the network environment including the MCW changes of other vehicles. However, it should be noted that this is impractical in reality, so we only use this method as the upper limit of our performance evaluation.

Random forest (RF) Random forest is a supervised machine learning algorithm. The classifier takes local observations $[\overline{\Delta}^n, \omega_0^n]$ as input and ω_0^n as the target label for training. The training label $\omega_0^{(n+1)}$ is used as the optimal action for the next step. We fixed the number of trees to 20 and the depth of each tree to 15.

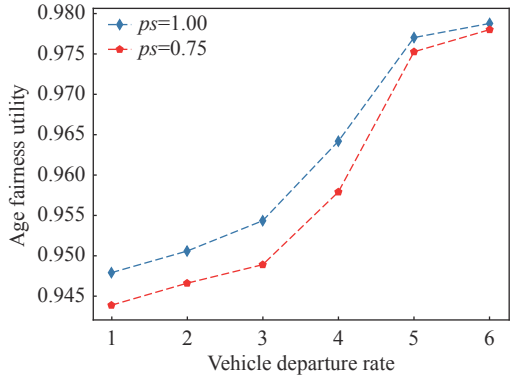

Fig. 10. Different departure rate rewards cuve of complex vehicle scenario.

Decision tree (DT) Decision tree is also a supervised machine learning algorithm. Random forest algorithm just uses multiple randomly generated decision trees to generate the final output result. This classifier also takes local observations $[\overline{\Delta}^n, \omega_0^n]$ as input and ω_0^n as the target label for training. We set the depth of the tree to 20.

Standard protocol (SP) The intelligent $node_0$ follows the fixed MCW protocol mentioned in the 802.11 DCF protocol, and its MCW is fixed to 64 and 128 respectively in simple vehicle scenarios. In complex vehicle scenes, MCW is fixed to 64, 128, 256, and 512, respectively.

Fig.11(a) shows the comparison of the age fairness utility between our RL method and the other baseline methods in the simple vehicle scenario. Among them, the red and blue box-whisker plots represent the cases of $ps = 1.0$ and $ps = 0.75$, respectively. We can see that no matter what the situation is, the age fairness utility achieved by our RL method is always the highest, and the fluctuation range of the box-whisker plot is also the smallest, (except for the OPT method), which means that the average age fairness utility of each episode is very stable. When $ps = 1.0$, the performance of RF is slightly worse than that of RL, but it also achieves a relatively ideal age fairness utility. But as compared to RF, the DT's performance will be slightly reduced, when $ps = 0.75$; however, RL can still learn a better policy. In addition, as compared to OPT, RL only slightly extends the fluctuation range, while RF and DT not only decrease the performance in terms of average age fairness utility, but also extend the fluctuation range of the box-whisker plot. Generally speaking, the age fairness utility achieved by the RL method is slightly better than that of DT method. Nevertheless, the above three methods are better than the SP meth-

od. For the SP method, we can see that the change of ps has little impact on the age fairness utility, because in either case, $node_0$ uses a fixed MCW and the box-whisker plots of $ps = 1.0$ and $ps = 0.75$ are almost equal. For MCW = 64, the age fairness utility is higher than that when MCW = 128. This is because for simple vehicle scenario, the MCW state space of all vehicles is $[32, 128]$. When the MCW of other vehicles is 32, the opportunity for $node_0$ to access the channel will be reduced, and the age fairness utility of $node_0$ will be reduced. When the MCW of all vehicles becomes 128, the opportunity of $node_0$ to access the channel will increase, and thus the age fairness utility of $node_0$ will increase while the age fairness utility of the whole simulation process will not be too low. When the MCW of $node_0$ is 128, $node_0$ will have less opportunity to access the channel for a long time, thus incurring a lower age fairness utility of the whole simulation process than that when MCW = 64.

Fig.11(b) shows the comparison of the age fairness utility between our RL method and other baseline methods in the complex vehicle scenario. In complex vehicle scenario, the MCW of all vehicles has a larger state space $[32, 64, 128, 256, 512]$, and the changes are more complex, as described at the beginning of this section.

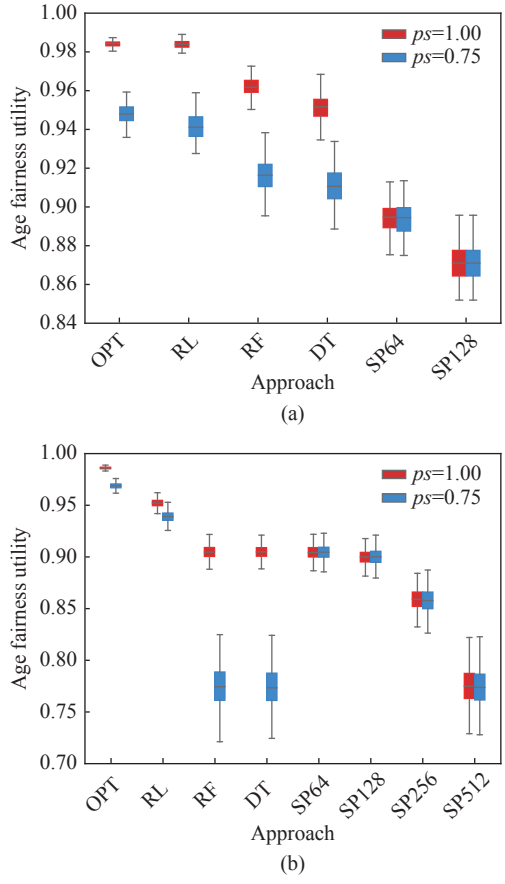

Fig. 11. Performance evaluation of the proposed RL method (a) simple vehicle scenario; (b) complex vehicle scenario.

Fig.11(b) shows that RL achieves the result closest to the optimal utility among all the methods considering the two values of ps. As expected, the case of $ps = 0.75$ is more challenging. Also note that the performance gap between OPT and RL is reasonable. Because the former has complete knowledge of the environment, while the latter must rely entirely on local observations to gather knowledge of the environment. In addition, the performance of RL and DT methods has been greatly reduced. When $ps = 1.0$, it can only be the same as the performance when the MCW is 64 and 128, and when $ps = 0.75$, its performance is significantly degraded. This is because for more complex situations, it is difficult for the classifier to give the best action to the current state of the network. Sometimes, the action given by the classifier may be the worst action, which will cause the average age fairness utility to be significantly degraded or even worse than the standard protocol. For the SP method, when the MCW used by $node_0$ increases, from the perspective of $node_0$, the age fairness utility will gradually decrease. This is because the opportunities for $node_0$ to access the channel will decrease, leading to a sharp increase in terms of the age of $node_0$. So far it also further reflects the effectiveness of the proposed method.

VII. Conclusions and Future Work

In this paper, we considered a dynamic and uncertain V2I communication scenario, where each vehicle may change its MCW to achieve more access opportunities at the expense of others and each vehicle is not willing to share its MCW with other vehicles. In this scenario, we designed an intelligent vehicular node that is used to learn the dynamics and predict the optimal MCW from local observations to ensure its age fairness. In order to allocate the optimal MCW for the vehicular node, we proposed a learning algorithm by extending the traditional DQN training and testing method to make a desirable decision by learning from replay history data. In addition, we obtain an age dataset for model training through real-time protocol simulation, and the superiority of our proposed RL method is proved through experiment simulations. According to theoretical analysis and simulation experiment, we can get the following conclusions:

- For different state transition probabilities ps, the model has excellent adaptability and can achieve relatively high age fairness utility.
- For the maximum number of vehicles in different networks, the model can also achieve approxim-

ately the same age fairness utility.

• In order to maximize long-term discount rewards, given different vehicle arrival rates and departure rates, this method can also achieve approximately the same age fairness utility.

If each vehicle is allowed to become an intelligent vehicle, it has the ability to independently learn and adjust its own MCW, so that it can realize the adaptive compatibility problem of various network environments. This may lead to huge training difficulties. In the future, we will further study the multi-agent training problem for the adaptive MCW in the network.

References

[1] Q. Wu, H. X. Liu, C. Zhang, *et al.*, "Trajectory protection schemes based on a gravity mobility model in IoT," *Electronics*, vol.8, no.2, article no.148, 2019.

[2] H. B. Zhu, Q. Wu, X. J. Wu, *et al.*, "Decentralized power allocation for MIMO-NOMA vehicular edge computing based on deep reinforcement learning," *IEEE Internet of Things Journal*, vol.9, no.14, pp.12770–12782, 2022.

[3] J. H. Zhao, Q. P. Li, Y. Gong, *et al.*, "Computation offloading and resource allocation for cloud assisted mobile edge computing in vehicular networks," *IEEE Transactions on Vehicular Technology*, vol.68, no.8, pp.7944–7956, 2019.

[4] Q. Wu and J. Zheng, "Performance modeling and analysis of the ADHOC MAC protocol for VANETs," in *Proceedings of IEEE International Conference on Communication (ICC)*, London, UK, pp.3646–3652, 2015.

[5] Q. Wu, Y. Zhao, and Q. Fan, "Time-dependent performance modeling for platooning communications at intersection," *IEEE Internet of Things Journal*, vol.9, no.19, pp.18500–18513, 2022.

[6] Defense Advanced Research Projects Agency, "Spectrum collaboration challenge (SC2)," Available at: https://www.darpa.mil/program/spectrum-collaboration-challenge, 2023.

[7] J. H. Zhao, L. H. Yang, M. H. Xia, *et al.*, "Unified analysis of coordinated multipoint transmissions in mmWave cellular networks," *IEEE Internet of Things Journal*, vol.9, no.14, pp.12166–12180, 2022.

[8] Q. Wu and J. Zheng, "Performance modeling and analysis of the ADHOC MAC protocol for vehicular networks," *Wireless Networks*, vol.22, no.3, pp.799–812, 2016.

[9] J. H. Zhao, X. K. Sun, Q. P. Li, *et al.*, "Edge caching and computation management for real-time Internet of vehicles: An online and distributed approach," *IEEE Transactions on Intelligent Transportation Systems*, vol.22, no.4, pp.2183–2197, 2021.

[10] Q. Wu, H. M. Ge, P. Y. Fan, *et al.*, "Time-dependent performance analysis of the 802.11p-based platooning communications under disturbance," *IEEE Transactions on Vehicular Technology*, vol.69, no.12, pp.15760–15773, 2020.

[11] W. W. Lu, S. L. Gong, Y. H. Zhu, "Timely data delivery for energy-harvesting IoT devices," *Chinese Journal of Electronics*, vol.31, no.2, pp.322–336, 2022.

[12] R. D. Yates, Y. Sun, D. R. Brown, *et al.*, "Age of information: An introduction and survey," *IEEE Journal on Selected Areas in Communications*, vol.39, no.5, pp.1183–1210, 2021.

[13] X. F. Chen, C. Wu, T. Chen, *et al.*, "Age of information aware radio resource management in vehicular networks: A proactive deep reinforcement learning perspective," *IEEE Transactions on Wireless Communications*, vol.19, no.4, pp.2268–2281, 2020.

[14] M. K. Abdel-Aziz, S. Samarakoon, C. F. Liu, *et al.*, "Optimized age of information tail for ultra-reliable low-latency communications in vehicular networks," *IEEE Transactions on Communications*, vol.68, no.3, pp.1911–1924, 2020.

[15] Q. Wu, Z. Y. Wan, Q. Fan, *et al.*, "Velocity-adaptive access scheme for MEC-assisted platooning networks: Access fairness via data freshness," *IEEE Internet of Things Journal*, vol.9, no.6, pp.4229–4244, 2022.

[16] J. Lv, X. M. Zhang, X. J. Han, *et al.*, "A novel adaptively dynamic tuning of the contention window (CW) for distributed coordination function in IEEE 802.11 Ad hoc networks," in *Proceedings of 2007 International Conference on Convergence Information Technology (ICCIT 2007)*, Gwangju, Korea (South), pp.290–294, 2007.

[17] C. Wu, S. Ohzahata, Y. S. Ji, *et al.*, "A MAC protocol for delay-sensitive VANET applications with self-learning contention scheme," in *Proceedings of 2014 IEEE 11th Consumer Communications and Networking Conference (CCNC)*, Las Vegas, NV, USA, pp.438–443, 2014.

[18] A. Jamali, S. M. S. Hemami, M. Berenjkoub, *et al.*, "An adaptive MAC protocol for wireless LANs," *Journal of Communications and Networks*, vol.16, no.3, pp.311–321, 2014.

[19] X. Zhou, C. W. Zheng, and X. X. He, "Adaptive contention window tuning for IEEE 802.11," in *Proceedings of 2015 22nd International Conference on Telecommunications (ICT)*, Sydney, NSW, Australia, pp.74–79, 2015.

[20] A. Pressas, Z. G. Sheng, F. Ali, *et al.*, "Contention-based learning MAC protocol for broadcast vehicle-to-vehicle communication," in *Proceedings of 2017 IEEE Vehicular Networking Conference (VNC)*, Torino, Italy, pp.263–270, 2017.

[21] C. Wu, Z. Liu, F. Q. Liu, *et al.*, "Collaborative learning of communication routes in edge-enabled multi-access vehicular environment," *IEEE Transactions on Cognitive Communications and Networking*, vol.6, no.4, pp.1155–1165, 2020.

[22] X. W. Wu, X. H. Li, J. Li, *et al.*, "Deep reinforcement learning for IoT networks: Age of information and energy cost tradeoff," in *Proceedings of GLOBECOM 2020–2020 IEEE Global Communications Conference*, Taipei, China, pp.1–6, 2020.

[23] F. Y. Wu, H. L. Zhang, J. J. Wu, *et al.*, "UAV-to-device underlay communications: Age of information minimization by multi-agent deep reinforcement learning," *IEEE Transactions on Communications*, vol.69, no.7, pp.4461–4475, 2021.

[24] S. H. Wang, M. Z. Chen, W. Saad, *et al.*, "Reinforcement learning for minimizing age of information under realistic physical dynamics," in *Proceedings of GLOBECOM 2020–2020 IEEE Global Communications Conference*, Taipei, China, pp.1–6, 2020.

[25] B. Han, Y. Zhu, Z. Y. Jiang, *et al.*, "Fairness for freshness: Optimal age of information based OFDMA scheduling with minimal knowledge," *IEEE Transactions on Wireless Communications*, vol.20, no.12, pp.7903–7919, 2021.

[26] S. Y. Leng and A. Yener, "An actor-critic reinforcement learning approach to minimum age of information scheduling in energy harvesting networks," in *Proceedings of ICASSP 2021–2021 IEEE International Conference on Acoustics, Speech and Signal Processing*, Toronto, ON, Canada, pp.8128–8132, 2021.

[27] I. Kadota and E. Modiano, "Minimizing the age of informa-

tion in wireless networks with stochastic arrivals," *IEEE Transactions on Mobile Computing*, vol.20, no.3, pp.1173–1185, 2021.
[28] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE Journal on Selected Areas in Communications*, vol.18, no.3, pp.535–547, 2000.
[29] P. Peng, F. Zhu, Q. Liu, *et al.*, "Achieving safe deep reinforcement learning via environment comprehension mechanism," *Chinese Journal of Electronics*, vol.30, no.6, pp.1049–1058, 2021.
[30] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, USA, 1998.
[31] H. van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, Phoenix, AZ, USA, pp.2094–2100, 2016.
[32] Y. Zhang, P. Sun, Y. H. Yin, *et al.*, "Human-like autonomous vehicle speed control by deep reinforcement learning with double Q-learning," in *Proceedings of 2018 IEEE Intelligent Vehicles Symposium (IV)*, Changshu, China, pp.1251–1256, 2018.
[33] Z. Y. Wang, T. Schaul, M. Hessel, *et al.*, "Dueling network architectures for deep reinforcement learning," in *Proceedings of the 33rd International Conference on Machine Learning*, New York, NY, USA, pp.1995–2003, 2016.
[34] M. Hessel, J. Modayil, H. van Hasselt, *et al.*, "Rainbow: Combining improvements in deep reinforcement learning," in *Proceedings of the 32nd AAAI conference on Artificial Intelligence*, New Orleans, LA, USA, pp.3215–3222, 2018.
[35] B. Li, S. Y. Liang, D. Q. Chen, *et al.*, "A decision-making method for air combat maneuver based on hybrid deep learning network," *Chinese Journal of Electronics*, vol.31, no.1, pp.107–115, 2022.
[36] M. Fortunato, M. G. Azar, B. Piot, *et al.*, Noisy networks for exploration, *arXiv preprint*, arXiv: 1706.10295, 2017.
[37] H. van Hasselt, A. Guez, M. Hessel, *et al.*, "Learning values across many orders of magnitude," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, Barcelona, Spain, pp.4294–4302, 2016.

WU Qiong received the Ph.D. degree in information and communication engineering from the National Mobile Communications Research Laboratory, Southeast University, Nanjing, China, in 2016. From 2018 to 2020, he was a Postdoctoral Researcher with the Department of Electronic Engineering, Tsinghua University. He is currently an Associate Professor with the School of Internet of Things Engineering, Jiangnan University, Wuxi, China. His current research interest focuses autonomous driving communication technology.
(Email: qiongwu@jiangnan.edu.cn)

SHI Shuai was born in Shandong Province, China. He received the B.S. degree from Dezhou University, Dezhou, China, in 2019. He is currently working toward the M.S. degree at Jiangnan University. His current research interests include data freshness and fairness in autonomous driving communications and federated learning for edge network. (Email: shuaishi@stu.jiangnan.edu.cn)

WAN Ziyang was born in Shandong Province, China. He received the B.S. degree from Ludong University, China, in 2019. He is currently working toward the M.S. degree at Jiangnan University. His current research interests include data freshness and fairness in autonomous driving communications.
(Email: ziyangwan@stu.jiangnan.edu.cn)

FAN Qiang received the Ph.D. degree in electrical and computer engineering from New Jersey Institute of Technology in 2019, and the M.S. degree in electrical engineering from Yunnan University of Nationalities, China, in 2013. He was a Postdoctor Researcher in the Department of Electrical and Computer Engineering, Virginia Tech. Currently, he is a Staff Engineer in Qualcomm, USA. His current research interests include wireless communications and networking, mobile edge computing, machine learning and drone assisted networking.
(Email: qiangfan29@gmail.com)

FAN Pingyi (corresponding author) received the B.S. and M.S. degrees from the Department of Mathematics of Hebei University in 1985 and Nankai University in 1990, respectively, received the Ph.D. degree from the Department of Electronic Engineering, Tsinghua University, Beijing, China, in 1994. He is a Professor of Department of Electronic Engineering of Tsinghua University currently. Dr. Fan is a Senior Member of IEEE and an Oversea Member of IEICE. He has attended to organize many international conferences including as TPC Co-chair of Chinacom 2020, IEEE WCNIS 2010 and TPC Member of IEEE ICC, Globecom, WCNC, VTC, Infocom, etc. He has served as an Editor of several journals for IEEE, Inderscience, Wiley, MDPI, etc. He is also a reviewer of more than 30 international journals including 24 IEEE journals and 8 EURASIP journals. He has received some academic awards, including the IEEE ICC20, TAOS20, Globecom14, WCNC08 Best Paper Awards, ACM IWCMC10 Best Paper Award and IEEE ComSoc Excellent Editor Award for *IEEE Transactions on Wireless Communications* in 2009. His main research interests include B5G technology in wireless communications, machine learning and AI in wireless communications, information theory in big data analysis, network coding and network information theory, etc.
(Email: fpy@tsinghua.edu.cn)

ZHANG Cui received the Ph.D. degree from Institute of Information Engineering, CAS, in 2017. From 2017 to 2021, she worked in Huawei 2012 Laboratory for Trustworthy System and Formal Verification Research. Currently, she is an Expert at security technique of real-time operating system (RTOS) for intelligent driving in Banma Network Technology Co., Ltd. Her main research interests include mobile computing, RTOS, security and formal verification.
(Email: faircas85@163.com)