Learning-based Incentive Mechanism for Task Freshness-aware Vehicular Twin Migration

Junhong Zhang^{*}, Jiangtian Nie[†], Jinbo Wen^{*}, Jiawen Kang^{*}, Minrui Xu[†], Xiaofeng Luo^{*}, Dusit Niyato[†], Fellow, IEEE *Guangdong University of Technology, China [†]Nanyang Technological University, Singapore

Abstract-Vehicular metaverses are an emerging paradigm that integrates extended reality technologies and real-time sensing data to bridge the physical space and digital spaces for intelligent transportation, providing immersive experiences for Vehicular Metaverse Users (VMUs). VMUs access the vehicular metaverse by continuously updating Vehicular Twins (VTs) deployed on nearby RoadSide Units (RSUs). Due to the limited RSU coverage, VTs need to be continuously online migrated between RSUs to ensure seamless immersion and interactions for VMUs with the nature of mobility. However, the VT migration process requires sufficient bandwidth resources from RSUs to enable online and fast migration, leading to a resource trading problem between RSUs and VMUs. To this end, we propose a learning-based incentive mechanism for migration task freshness-aware VT migration in vehicular metaverses. To quantify the freshness of the VT migration task, we first propose a new metric named Age of Twin Migration (AoTM), which measures the time elapsed of completing the VT migration task. Then, we propose an AoTMbased Stackelberg model, where RSUs act as the leader and VMUs act as followers. Due to incomplete information between RSUs and VMUs caused by privacy and security concerns, we utilize deep reinforcement learning to learn the equilibrium of the Stackelberg game. Numerical results demonstrate the effectiveness of our proposed learning-based incentive mechanism for vehicular metaverses.

Index Terms—Metaverse, vehicular twin, Stackelberg game, Age of Information, deep reinforcement learning.

I. INTRODUCTION

The rapid advancement of immersive communication, such as Virtual Reality (VR), Augmented Reality (AR), and ubiquitous Artificial Intelligence (AI) has given rise to the vehicular metaverse. Vehicular metaverses are expected to lead the revolution of intelligent transportation systems by seamlessly blending virtual and physical spaces, allowing for providing immersive services for Vehicular Metaverse Users (VMUs) (i.e., drivers and passengers within vehicles) [1]. Vehicular Twins (VTs) are highly accurate virtual hybrid replicas that cover the entire life cycle of vehicles and VMUs [2]. The VTs are updated by sensing data from the surrounding environment to achieve physical-virtual synchronization [3]. Through VTs, VMUs can access the vehicular metaverse to enjoy a wide range of metaverse applications, such as AR navigation, virtual education, and virtual games [2], [4].

To ensure seamless immersive experiences for VMUs in the vehicular metaverse, resource-limited vehicles offload latencysensitive and computation-intensive tasks of updating VTs to nearby edge servers in RoadSide Units (RSUs) [2]. However, due to the limited coverage of RSUs and the mobility of vehicles, each VT has to be migrated from the current RSU to another to provide uninterrupted immersive services for VMUs. Therefore, the task freshness of the VT migration, i.e., the time it takes to complete the VT migration, is critical to VMUs. To ensure VT migration efficiency, VMUs need to purchase sufficient resources from RSUs for facilitating VT migration, especially bandwidth resources. Without loss of generality, the Metaverse Service Provider (MSP) is set as the manager of RSUs, which is the sole provider of bandwidth resources during VT migration. The MSP aims to optimize its bandwidth selling price and maximize revenue from resource trading with incomplete information. Existing work has been conducted to optimize resource pricing and allocation based on the incentive mechanism in the metaverse [5]–[7]. The authors in [5] formulated a Stackelberg game joint user association and resource pricing. The authors in [6] proposed a hierarchical game-theoretic approach to study a reliable coded distributed computing scheme in vehicular metaverses. However, they ignore the VT migration issue caused by the mobility of vehicles. Therefore, it is still challenging in tackling the resource trading problem in VT migration.

To address the above challenges, in this paper, we propose a new metric named Age of Twin Migration (AoTM) according to the concept of Age of Information (AoI). Considering that VMUs may be reluctant to disclose their private information for privacy security during VT migration, we propose a learning-based incentive mechanism between the MSP and VMUs. The main contributions are summarized as follows:

- To quantify the freshness of the VT migration task, we propose a new metric named AoTM according to the concept of AoI for vehicular metaverses and apply it to evaluate the immersion of VMUs.
- To improve VT migration efficiency under information incompleteness, we formulate the Stackelberg game between the MSP and VMUs, in which the MSP acts as the leader and VMUs act as followers.
- We utilize Deep Reinforcement Learning (DRL) to solve

The work was supported by NSFC under grant No. 62102099, U22A2054, and the Pearl River Talent Recruitment Program under Grant 2021QN02S643, and also supported in part by National Key R&D Program of China (No. 2020YFB1807802), and the National Research Foundation (NRF), Singapore and Infocomm Media Development Authority under the Future Communications Research Development Programme (FCP). (Corresponding author: Jiawen Kang (e-mail: kavinkang@gdut.edu.cn)).



Fig. 1. A learning-based incentive mechanism framework for VT migration.

the Stackelberg game under incomplete information. Numerical results demonstrate that the proposed learningbased scheme can converge to the Stackelberg equilibrium and outperform baseline schemes.

II. SYSTEM MODEL

As shown in Fig. 1, edge-assisted remote rendering as a key technology is applied in vehicular metaverses [5]. To construct VTs for lower-latency and ultra-reliable metaverse services, such as AR navigation, e-commerce, and virtual games, the large-scale rendering tasks are offloaded to nearby edge servers in RSUs with abundant resources (i.e., storage, bandwidth, and computing) [2]. However, due to the dynamic mobility of vehicles and the limited service coverage of RSUs [1], VTs must be migrated from the source RSUs to the destination RSUs for realizing fully immersive metaverse services. We provide more details of the system model as follows:

- MSP: The MSP as the manager of RSUs can schedule resources of RSUs to provide necessary resources (e.g., computing and bandwidth) for VMUs [5]. After being authorized, the MSP can manage a number of communication channels between the source RSUs and the destination RSUs [5]. Besides, the MSP leverages sensing data (e.g., traffic conditions and vehicle locations) sent by VMUs to update VTs for providing ultra-reliable and real-time metaverse services for VMUs.
- VTs: VTs are the digital replicas deployed in RSUs. They cover the life cycle of vehicles and VMUs and act as intelligent assistants managing metaverse applications [2]. In addition, VTs can also analyze and predict their VMUs' behavior through a pre-trained machine learning model. Note that we consider that each VMU has a corresponding VT and the VT can be transmitted in the form of blocks during migration.
- VMUs: Without loss of generality, VMUs refer to drivers and passengers within vehicles. The widespread use of VR, AR, and spatial audio devices enables VMUs to enjoy metaverse services through Head-Mounted Displays (HMDs) as well as AR windshields and side windows [1]. Additionally, smart sensors on VMUs (e.g., cameras, Inertial Measurement Units (IMU) suits) collect and

send sensing data (e.g., driver fatigue level and vehicle locations) to the MSP for VT synchronization [2].

III. PROBLEM FORMULATION

In this section, to quantify the freshness of the VT migration task, we first propose a new metric named AoTM, which can evaluate the immersion of VMUs. Then, we design a Stackelberg game model between the MSP and VMUs for VT migration and analyze the game to prove the existence and the uniqueness of Stackelberg equilibrium among the MSP and VMUs [5], [8]. In this paper, we consider that one MSP and a set $\mathcal{N} = \{1, \ldots, n, \ldots, N\}$ of N VMUs participate in VT migration and all VTs of VMUs need to be migrated.

A. Age of Twin Migration

AoI has been widely utilized to quantify data freshness at the destination [9]. It is defined as the time elapsed since the latest received update was generated at its source, which is a promising metric to improve the performance of timecritical services [10]. Similarly, in vehicular metaverses, to quantify the freshness of the VT migration task, we propose a new metric named AoTM according to the concept of the AoI, which is defined as the time elapsed between the last successfully received VT block and the generation of the first VT block in the VT migration.

We consider that the Orthogonal Frequency Division Multiplexing Access (OFDMA) technology is applied in the system [5], which ensures that all communication channels occupied by the source RSU and the destination RSU are orthogonal. For VMU $n \in \mathcal{N}$, given the purchased bandwidth $b_n \in (0, +\infty)$ from the MSP, the achievable task transmission rate between the source RSU and the destination RSU is $\gamma_n = b_n \log_2 \left(1 + \frac{\rho h^0 d^{-\varepsilon}}{N_0}\right)$, where ρ , h^0 , d, ε , and N_0 represent the transmitter power of the source RSU, the unit channel power gain, the distance between the source RSU and the destination RSU, the path-loss coefficient, and the average noise power, respectively [5]. Therefore, for VMU n, the AoTM of the VT migration task is

$$A_n = \frac{D_n}{\gamma_n},\tag{1}$$

following the pre-copy live migration strategy in [11], the total migrated VT data D_n includes the information of system configuration (e.g., CPU and GPU), historical memory data, and real-time states of VMU n.

B. Stackelberg Game

In VT migration, the MSP is the sole bandwidth resource holder and VMUs rely on bandwidth resources provided by the MSP to migrate VTs between RSUs. As a result, a monopoly market is formed, in which the MSP, as the monopolist, has the pricing power of bandwidth and VMUs need to respond to the price by deciding how much bandwidth to purchase. To be specific, when the selling price of bandwidth is low, VMUs may be willing to purchase more bandwidth for enhancing immersive experiences. Conversely, VMUs are reluctant to purchase when the selling price is high, resulting in poor task freshness. Therefore, the selling price of bandwidth has a significant impact on the immersion of VMUs.

To maximize the MSP's profit and maintain its monopoly power, the Stackelberg game can provide a powerful game theoretical model that has been widely used by the monopolist to strategically set the price. The Stackelberg game between the MSP and VMUs consists of two stages. In the first stage, the MSP as the leader decides the selling price of bandwidth for its maximum utility. In the second stage, each VMU as a follower determines the bandwidth demand to maximize its utility. Note that the second stage of the game can be formulated as a competitive game [6].

1) Utility formation in the VT migration: The utility of VMU n is the difference between the profit corresponding to its immersion and its cost of purchasing bandwidth. The higher AoTM impacts the immersive experiences of VMUs negatively, resulting in decreasing the immersion of VMUs [6]. Following [12], the immersion function of VMU n obtained from the MSP is defined as $G_n = \alpha_n \ln (1 + 1/A_n)$, where $\alpha_n > 0$ is the unit profit for the immersion of VMU n. Therefore, the utility function of VMU n is

$$U_n(b_n) = G_n - p \cdot b_n, \tag{2}$$

where p > 0 is the unit selling price of bandwidth. In the follower stage, each VMU *n* maximizes its revenue $U_n(b_n)$ by deciding the best bandwidth demand to purchase. Thus, the problem of maximizing the utility of VMU *n* is formulated as

Problem 1:
$$\max_{b_n} U_n(b_n)$$

s.t. $b_n > 0.$ (3)

For the MSP, its utility is the difference between the sum of bandwidth fees paid by all VMUs and the transmission cost for VT migration tasks, which is affected by the unit selling price of bandwidth and bandwidth demands of VMUs. Thus, the utility of the MSP is

$$U_s(p) = \sum_{n=1}^{N} (p \cdot b_n - C \cdot b_n), \qquad (4)$$

where C > 0 is the unit transmission cost of bandwidth for executing the VT migration task, which is proportional to the amount of bandwidth sold to the VMUs. In the first stage, considering that the bandwidth sold by the MSP has a maximum bandwidth B^{max} and the maximum bandwidth pricing p^{max} , the MSP maximizes its revenue by deciding a selling price that ensures the total bandwidth sales do not exceed B^{max} and the bandwidth price does not exceed p^{max} . Thus, the problem of maximizing the utility of the MSP is formulated as

Problem 2:
$$\max_{p} U_{s}(p)$$
s.t. $0 < \sum_{n=1}^{N} b_{n} \leq B^{max},$
 $b_{n} > 0, \forall n \in \{1, \dots, N\},$
 $0 < C \leq p \leq p^{max}.$
(5)

2) Stackelberg equilibrium analysis: The Stackelberg game is formulated by combining **Problem 2** and **Problem 1**. We seek the Stackelberg equilibrium to obtain the optimal solution to the formulated game. In the Stackelberg equilibrium, the MSP's utility is maximized considering that the VMUs make bandwidth demand strategies based on the best response, and neither the MSP nor any VMU can improve the individual utility by deviating from their strategies [5], [6]. The Stackelberg equilibrium is defined as follows:

Definition 1. (Stackelberg Equilibrium): We denote $\mathbf{b}^* = \{b_n^*\}_{n=1}^N$ and p^* as the optimal bandwidth demand strategy vector and the optimal unit bandwidth selling price, respectively. Then, the strategies ($\mathbf{b}^* = \{b_n^*\}_{n=1}^N, p^*$) can be Stackelberg equilibrium if and only if the following set of inequalities is strictly satisfied:

$$\begin{cases} U_{s}\left(\boldsymbol{b}^{*},p^{*}\right) \geq U_{s}\left(\boldsymbol{b}^{*},p\right),\\ U_{n}\left(\boldsymbol{b}^{*}_{n},\boldsymbol{b}^{*}_{-\boldsymbol{n}},p^{*}\right) \geq U_{n}\left(\boldsymbol{b}_{n},\boldsymbol{b}^{*}_{-\boldsymbol{n}},p^{*}\right), \ \forall n \in \mathcal{N}. \end{cases}$$
(6)

In the following, we adopt the backward induction method to prove the Stackelberg equilibrium [5].

Theorem 1. The sub-game perfect equilibrium in the VMUs' subgame is unique.

Proof. We derive the first-order derivative and the second-order derivative of $U_n(b_n)$ with respect to b_n as follows:

$$\frac{\partial U_n(b_n)}{\partial b_n} = \frac{\alpha_n \log_2 \left(1 + \frac{\rho h^0 d^{-\varepsilon}}{N_0}\right)}{D_n + b_n \log_2 \left(1 + \frac{\rho h^0 d^{-\varepsilon}}{N_0}\right)} - p,$$

$$\frac{\partial^2 U_n(b_n)}{\partial b_n^2} = -\frac{\alpha_n \left(\log_2 \left(1 + \frac{\rho h^0 d^{-\varepsilon}}{N_0}\right)\right)^2}{\left(D_n + b_n \log_2 \left(1 + \frac{\rho h^0 d^{-\varepsilon}}{N_0}\right)\right)^2} < 0.$$
(7)

As the first-order derivative of $U_n(b_n)$ has a unique zero point, and the second-order derivative of $U_n(b_n)$ is negative, the VMU's utility function $U_n(b_n)$ is strictly concave with respect to b_n . Then, based on the first-order optimality condition, i.e., $\frac{\partial U_n(b_n)}{\partial b_n} = 0$, we can obtain the best response function of VMU *n*, given by

$$b_n^* = \frac{\alpha_n}{p} - \frac{D_n}{\log_2\left(1 + \frac{\rho h^0 d^{-\varepsilon}}{N_0}\right)}.$$
(8)

Therefore, the sub-game perfect equilibrium in the VMUs' subgame is unique.

Theorem 2. There exists a unique Stackelberg equilibrium (b^*, p^*) in the formulated game.

Proof. Based on **Theorem 1**, the MSP as the leader in the Stackelberg game knows that there exists a unique Nash equilibrium among VMUs under any given value of p. Therefore, the MSP can maximize its utility by choosing the optimal p. By substituting (8) into (4), we have

$$U_{s} = \sum_{n=1}^{N} (p - C) \left(\frac{\alpha_{n}}{p} - \frac{D_{n}}{\log_{2} \left(1 + \frac{\rho h^{0} d^{-\varepsilon}}{N_{0}} \right)} \right).$$
(9)

Then, by taking the first-order derivative and the second-order derivative of $U_s(p)$ with respect to p, respectively, we have

$$\frac{\partial U_s(p)}{\partial p} = \sum_{n=1}^N \left(-\frac{D_n}{\log_2\left(1 + \frac{\rho h^0 d^{-\varepsilon}}{N_0}\right)} + \frac{\alpha_n C}{p^2} \right),$$

$$\frac{\partial^2 U_s(p)}{\partial^2 p} = \sum_{n=1}^N -\frac{2C \cdot \alpha_n}{p^3} < 0.$$
(10)

Since the first-order derivative of $U_s(p)$ has a unique zero point, i.e., $p^* = \sqrt{\frac{C \log_2 \left(1 + \frac{\rho h^0 d^{-\varepsilon}}{N_0}\right) \sum_{n=1}^N \alpha_n}{\sum_{n=1}^N D_n}}}$, and the second-order derivative of $U_s(p)$ is negative, $U_s(p)$ is strictly concave, indicating that the MSP has a unique optimal solution to the formulated game [8]. Based on the optimal strategy of the MSP, the VMUs' optimal strategies can be obtained [6]. Therefore, the Stackelberg equilibrium can be obtained uniquely in the formulated game.

IV. LEARNING-BASED INCENTIVE MECHANISM WITH INCOMPLETE INFORMATION

In this section, we first introduce the DRL algorithm. Then, we describe how to transform the Stackelberg game into a learning task. Specifically, we model the Stackelberg game between the MSP and VMUs as a Partially Observable Markov Decision Process (POMDP) and design a DRL-based learning algorithm to explore the optimal solution to the Stackelberg model, where the MSP is the learning agent.

A. Deep Reinforcement Learning for Stackelberg Game

Due to the competitive effect, each VMU only has its local information which is incomplete in the game and determines the bandwidth strategies in a fully non-cooperative manner [5]. DRL can be utilized to learn an optimal policy from past experiences based on the current state and the given reward without knowing any prior information. Here are the details of the DRL formulation.

1) State space: At the current game round $k \in \mathcal{K} = \{0, \dots, k, \dots, K\}$, the state space is defined as a union of the current MSP's pricing strategy and VMUs' bandwidth demand strategies, which is denoted as $S_k \triangleq \{p_k, \mathbf{b}_k\}$.

2) Partially observable policy: To tackle the non-stationary problem in the DRL system for facilitating VT migration, we formulate the partially observable space for VT migration. The MSP agent can only make decisions according to its local observation of the environment. We define the observation space o_k of the MSP at the current game round k as a union of its historical pricing strategies and VMUs' bandwidth demand strategies for past L rounds, given by

$$o_k \triangleq \{p_{k-L}, \boldsymbol{b}_{k-L}, p_{k-L+1}, \boldsymbol{b}_{k-L+1}, \dots, p_{k-1}, \boldsymbol{b}_{k-1}\}.$$
 (11)

Note that p_{k-L} and b_{k-L} can be generated randomly during the initial stage when k < L. We consider historical information because it enables the MSP agent to learn how its strategy changes impact the game result of the current time slot. When receiving an observation o_k , the MSP agent needs to take a pricing action p_k to maximize its utility. Given the lower bound cost C and the upper bound price p^{max} for the pricing action, the action space can be represented as $p_k \in [C, p^{max}]$, and the MSP's policy can be represented as $\pi_{\theta} (p_k \mid o_k) \to [C, p^{max}]$. Note that we use a neural network to represent the policy π_{θ} and the value function $V_{\pi_{\theta}}(\cdot)$, where θ is the neural network parameter.

3) Reward: After the state transition, the MSP would gain a reward based on the current state S_k and the corresponding action p_k . The reward function of the MSP can be defined as

$$R(S_k, p_k) = \begin{cases} 1, \ U_s^k \ge U_{best}^k, \\ 0, \ U_s^k < U_{best}^k, \end{cases}$$
(12)

where U_s^k is the current utility of the MSP in (4) and U_{best}^k is the highest utility that the MSP has obtained until round k.

4) Value function: Given a policy π_{θ} , the value function $V_{\pi_{\theta}}(S)$ can measure the expected return when starting in S and following π_{θ} thereafter [13], which is defined as

$$V_{\pi_{\theta}}(S) \triangleq \hat{\mathbb{E}}_{\pi_{\theta}} \left[\sum_{k=0}^{K} \gamma^{k} R\left(S_{k}, p_{k}\right) \mid S_{0} = S \right], \quad (13)$$

where $\mathbb{E}_{\pi_{\theta}}(\cdot)$ is the expected value of a random variable given that the MSP agent follows the policy π_{θ} , and $\gamma \in [0, 1]$ is the reward discounting factor to reduce the weights as the time step increases.

5) Actor-critic framework design: We leverage the popular actor-critic framework and the Proximal Policy Optimization method for policy iteration [8]. Following [13], at each training iteration, we randomly sample experiences from the replay buffer to update the network parameter. Then, Generalized Advantage Estimation [14] is used to compute variance-reduced advantage function estimator A(S, p) that utilizes a learning state-value function $V_{\pi_{\theta}}(S)$. Since the policy and the value function share the same parameter θ of the neural network, the loss function consists of the policy surrogate $L^{CLIP}(\theta)$ and the value function error term $L^{VF}(\theta)$. Finally, to update the policy and the value function, we utilize stochastic gradient ascent to maximize the objective function as follows:

$$\boldsymbol{\theta}_{e+1} = \arg\max_{\boldsymbol{\theta}_{e}} \frac{1}{|I|} \sum_{|I|} \hat{\mathbb{E}}_{k} \Big[L_{k}^{CLIP} \left(\boldsymbol{\theta}_{e}\right) - c L_{k}^{VF} \left(\boldsymbol{\theta}_{e}\right) \Big],$$
(14)

$$L_{k}^{CLIP}(\boldsymbol{\theta}_{e}) = \hat{\mathbb{E}}_{k} \bigg[\min \bigg(r_{k}(\boldsymbol{\theta}_{e}) A(S_{k}, p_{k}), f_{clip}\left(r_{k}(\boldsymbol{\theta}_{e}) \right) A(S_{k}, p_{k}) \bigg) \bigg],$$
(15)

$$L_k^{VF}(\boldsymbol{\theta}_e) = \left(V_{\pi_{\boldsymbol{\theta}_e}}(S_k) - V_k^{targ} \right)^2, \tag{16}$$

where

$$\tau_k(\boldsymbol{\theta}_e) = \frac{\pi_{\boldsymbol{\theta}_e}(p_k|o_k)}{\pi_{\boldsymbol{\theta}^{old}}(p_k|o_k)},\tag{17}$$

$$A(S_k, p_k) = -V_{\pi_{\boldsymbol{\theta}_e}}(S_k) + \sum_{l=k}^{K-1} \gamma^{l-k} R(S_l, p_l) + \gamma^{K-k} V_{\pi_{\boldsymbol{\theta}_e}}(S_K),$$
(18)

r

Algorithm 1: Proposed DRL-based Solution for VT Migration

1 Initialize max round in an episode K, number of episodes
E, batch size I and network parameter θ ;
2 for Episode $e \in 1, \ldots, E$ do
3 Reset environment state S_0 and replay buffer \mathcal{BF} ;
4 for Round $k \in 0, \ldots, K$ do
5 MSP observes a state S_k and updates its observation
o_{k-1} into o_k ;
6 Input o_k into MSP's actor policy π_{θ_e} and determine
the current price strategy p_k ;
7 VMUs determine bandwidth demands through (8);
8 Update S_k into S_{k+1} and calculate reward R_k for
the MSP through (12). Then, update U_{best}^k when a
higher reward is obtained;
9 Store transition (o_k, p_k, R_k, o_{k+1}) into \mathcal{BF} ;
10 if $k\% I == 0$ then
11 for $m \in 1, \ldots, M$ do
12 Sample a random mini-batch of data with a
size $ I $ from \mathcal{BF} to update the actor and
critic through (14);
13 end
14 end
15 end
16 end

and

$$f_{clip}(r_k(\boldsymbol{\theta}_e)) = \begin{cases} 1 - \epsilon, \ r_k(\boldsymbol{\theta}_e) < 1 - \epsilon, \\ 1 + \epsilon, \ r_k(\boldsymbol{\theta}_e) > 1 + \epsilon, \\ r_k(\boldsymbol{\theta}_e), \ 1 - \epsilon \le r_k(\boldsymbol{\theta}_e) \le 1 + \epsilon. \end{cases}$$
(19)

Here, V_k^{targ} is the total discount reward from time step k until the end of the episode, θ_e and θ_{e+1} are the policy parameter in episode e and e + 1, θ_e^{old} represents the policy parameter for sampling in episode e, c is a loss coefficient of the value function, r_k is the importance ratio, and I is the batch size of sampled experiences for calculating policy gradients.

B. Algorithm Details

Motivated by the above analysis, the proposed DRL algorithm details are illustrated in **Algorithm 1**. The time complexity of the proposed DRL algorithm is determined by the multiplication operations in a fully connected deep neural network [8], which can be expressed as $\mathcal{O}\left(\sum_{f=1}^{F} \epsilon_{f} \epsilon_{f-1}\right)$, where ϵ_{f} is the number of neural units in layer f and F is the number of hidden layers.

V. NUMERICAL RESULTS

In this section, we evaluate the performance of the VT migration system for vehicular metaverses and the proposed DRL-based incentive mechanism through simulation experiments. We first describe the experimental settings, followed by the experimental results and analysis.

A. Experiment Settings

We consider that there is one MSP and the number of VMUs $N \in [1, 6]$. Each VT has the data size $D_n \in [100, 300]$ (MB) and the immersion coefficient $\alpha_n \in [5, 20]$. The MSP's



Fig. 2. Convergence of DRL-based incentive mechanism.

maximum bandwidth, transmission cost, and maximum selling price are set to 50MHz, 5, and 50, respectively. As for the RSU parameters, the transmitter power of the source RSU ρ is 40dBm, the unit channel power gain h_0 is -20dB, the distance between the RSUs d is 500m, the path-loss coefficient ϵ is 2, and the average noise power N_0 is -150dBm. The parameters of the DRL are selected through fine-tuning. Specially, we set L = 4, D = 20, E = 500, K = 100, M = 10, and lr = 0.00001 during experiments. Both the two hidden layers of the neural network have 64 nodes.

B. Experiment Results

Figure 2 shows the convergence of the proposed DRLbased incentive mechanism when there are two VMUs. We set $\alpha_1 = \alpha_2 = 5$, $D_1 = 200$ MB, $D_2 = 100$ MB, and cost C = 5. As shown in Fig. 2(a), the game return of each episode converges to the maximum round, which indicates that the MSP can always choose the optimal strategy in each round. In Fig. 2(b), the utility of the MSP converges to the Stackelberg equilibrium. Therefore, the DRL-based incentive mechanism under incomplete information is as strong as the Stackelberg game with complete information.

Figure 3 shows the performance of the proposed DRL-based incentive mechanism. In Fig. 3(a) and Fig. 3(b), we study the influence of the unit transmission cost. Specifically, we study the unit transmission cost by changing it from 5 to 9 and consider that there are two VMUs whose VT data sizes are 200MB and 100MB, and whose immersion coefficients are both 5. From Fig. 3(a) and Fig. 3(b), we can see that both the utilities and strategies of the MSP and VMUs in the optimal solutions of the proposed scheme are approaching the Stackelberg equilibrium, which demonstrates that the proposed scheme can find the optimal solution under incomplete information. As the unit transmission cost increases, the pricing of the MSP also increases in Fig. 3(a). For example, when the unit transmission cost is 5, the MSP sets the price at 25 to incentive VMUs to perform VT migration. However, when the unit transmission cost is 9, a higher price of 34 will be set. In Fig. 3(b), we can observe that the total bandwidth strategy of VMUs decreases when the unit transmission cost increases. For example, when the unit transmission cost is 6, VMUs purchase bandwidth resources of 27.9. While VMUs only purchase bandwidth resources of 23.4 when the unit transmission cost is 8. Both the utilities of the MSP and VMUs significantly decrease due to the high cost of transmission in Fig. 3(a) and



(a) The utility and price strategy of the (b) Total utility and bandwidth strat- (c) The utility and price strategy of the (d) Average utility and bandwidth MSP vs. Transmission cost. MSP vs. Number of VMUs. strategy of VMUs vs. Number of VMUs.

Fig. 3. The performance of the proposed DRL-based incentive mechanism.

Fig. 3(b). The reason is that when the transmission cost is high, the MSP would increase the bandwidth price due to the cost consideration, leading to a decrease in bandwidth purchased by VMUs because of the high price. Furthermore, we compare the proposed DRL-based scheme with random and greedy schemes. In the random scheme, the MSP determines the price randomly in each game round, while in the greedy scheme, the MSP determines the best price by selecting from past game rounds. In Fig. 3(a), we can find that our proposed scheme outperforms the baseline schemes.

Next, we study the impacts of the number of VMUs in Fig. 3(c) and Fig. 3(d). We set the data size of the VT as 100MB, and the immersion coefficient α_n is 5. As shown in Fig. 3(c), the utility of the MSP increases when the number of VMUs increases. For example, the utility of the MSP is 7.03 when there are only two VMUs. When the number of VMUs increases to 6, the MSP can obtain a higher utility of 20.35. Note that the price of the MSP remains unchanged initially and increases later. The reason is that when there are fewer VMUs, the bandwidth resources of the MSP are sufficient, but when the number of VMUs is too large, the bandwidth of the MSP becomes insufficient. Therefore, the MSP needs to increase the price of bandwidth to limit the purchase of excessive bandwidth by VMUs. As shown in Fig. 3(d), the average bandwidth purchased by VMUs remains unchanged at first and decreases later. Due to the competition among VMUs, the average utility of VMUs decreased by 12.8% as the number of VMUs increases from 2 to 6.

VI. CONCLUSION

In this paper, we proposed a learning-based incentive mechanism for task freshness-aware VT migration in vehicular metaverses. To quantify the task freshness of the VT migration, we proposed a new metric called AoTM according to the concept of the AoI. Then, we formulated the resource trading problem between the MSP and VMUs as a Stackelberg game. Furthermore, we utilized DRL to solve the game under incomplete information. Finally, numerical results demonstrate the effectiveness of the proposed mechanism. In the future, we will adopt more effective immersive metrics in conjunction with AoTM to better evaluate the immersion of VMUs and may develop a prototype system to evaluate our framework. Besides, we aim to extend our model to scenarios with multiple MSPs and VMUs.

REFERENCES

- P. Zhou, J. Zhu, Y. Wang, Y. Lu, Z. Wei, H. Shi, Y. Ding, Y. Gao, Q. Huang, Y. Shi *et al.*, "Vetaverse: Technologies, applications, and visions toward the intersection of metaverse, vehicles, and transportation systems," *arXiv preprint arXiv:2210.15109*, 2022. I, II, II
- [2] J. Yu, A. Alhilal, P. Hui, and D. H. Tsang, "Bi-directional digital twin and edge computing in the metaverse," *arXiv preprint arXiv:2211.08700*, 2022. I, II, II
- [3] M. Xu, D. Niyato, B. Wright, H. Zhang, J. Kang, Z. Xiong, S. Mao, and Z. Han, "Epvisa: Efficient auction design for real-time physical-virtual synchronization in the metaverse," *arXiv preprint arXiv:2211.06838*, 2022. I
- [4] L. U. Khan, M. Guizani, D. Niyato, A. Al-Fuqaha, and M. Debbah, "Metaverse for wireless systems: Architecture, advances, standardization, and open challenges," arXiv preprint arXiv:2301.11441, 2023. I
- [5] X. Huang, W. Zhong, J. Nie, Q. Hu, Z. Xiong, J. Kang, and T. Q. Quek, "Joint user association and resource pricing for metaverse: Distributed and centralized approaches," in 2022 IEEE 19th International Conference on Mobile Ad Hoc and Smart Systems (MASS). IEEE, 2022, pp. 505–513. I, II, III, III, III-A, III-A, III-B2, III-B2, IV-A
- [6] Y. Jiang, J. Kang, D. Niyato, X. Ge, Z. Xiong, C. Miao, and X. Shen, "Reliable distributed computing for metaverse: A hierarchical gametheoretic approach," *IEEE Transactions on Vehicular Technology*, 2022. I, III-B, III-B1, III-B2, III-B2
- [7] C. T. Nguyen, D. T. Hoang, D. N. Nguyen, and E. Dutkiewicz, "Metachain: A novel blockchain-based framework for metaverse applications," in 2022 IEEE 95th Vehicular Technology Conference:(VTC2022-Spring). IEEE, 2022, pp. 1–5. I
- [8] Y. Zhan, P. Li, Z. Qu, D. Zeng, and S. Guo, "A learning-based incentive mechanism for federated learning," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6360–6368, 2020. III, III-B2, IV-A, IV-B
- [9] R. D. Yates, Y. Sun, D. R. Brown, S. K. Kaul, E. Modiano, and S. Ulukus, "Age of information: An introduction and survey," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 5, pp. 1183–1210, 2021. III-A
- [10] J. Kang, D. Ye, J. Nie, J. Xiao, X. Deng, S. Wang, Z. Xiong, R. Yu, and D. Niyato, "Blockchain-based federated learning for industrial metaverses: Incentive scheme with optimal aoi," in 2022 IEEE International Conference on Blockchain (Blockchain). IEEE, 2022, pp. 71–78. III-A
- [11] M. Imran, M. Ibrahim, M. S. U. Din, M. A. U. Rehman, and B. S. Kim, "Live virtual machine migration: A survey, research challenges, and future directions," *Computers and Electrical Engineering*, vol. 103, p. 108297, 2022. III-A
- [12] Y. Fan, L. Wang, W. Wu, and D. Du, "Cloud/edge computing resource allocation and pricing for mobile blockchain: an iterative greedy and search approach," *IEEE Transactions on Computational Social Systems*, vol. 8, no. 2, pp. 451–463, 2021. III-B1
- [13] M. Xu, J. Peng, B. Gupta, J. Kang, Z. Xiong, Z. Li, and A. A. Abd El-Latif, "Multiagent federated reinforcement learning for secure incentive mechanism in intelligent cyber–physical systems," *IEEE Internet of Things Journal*, vol. 9, no. 22, pp. 22095–22108, 2021. IV-A, IV-A
- [14] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "Highdimensional continuous control using generalized advantage estimation," *arXiv preprint arXiv:1506.02438*, 2015. IV-A