A Deep Reinforcement Learning-Based Content Updating Algorithm for High Definition Map Edge Caching

Haoru Li[1], Gaofeng Hong[1(✉)], Bin Yang[2], and Wei Su[1]

[1] School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing, China
{21120074,honggf,wsu}@bjtu.edu.cn
[2] School of Computer and Information Engineering, Chuzhou University, Chuzhou, China

Abstract. Edge caching is a promising technique to alleviate the communication cost during the content update and retrieving. Particularly, it is suitable for the High Definition Map (HDM) caching which needs frequent updates to avoid its contents becoming staleness. In this paper, we aim at minimizing the response latency while satisfying the content freshness of the vehicle's HDM request under the edge caching scenario. We first depict the change of the content freshness difference, in term of the Age of Information (AoI) difference value, of each request, which are determined by both the vehicular requirements and the content update decision of the Road Aide Unit (RSU). Then, we formulate the HDM content update optimization problem, which jointly considering the AoI difference and the extra responding latency of each request. On this basis, we transform the problem into a Markov Decision Process (MDP), and propose an optimization algorithm based on the deep reinforcement learning-based theory to obtain the optimal update decision by maximizing the long-term discounted reward. Finally, extensive simulations are presented to verify the effectiveness of the proposed algorithm by comparing it with various baseline policies.

Keywords: Vehicular Networks · High Definition Map · Edge Caching · Deep Reinforcement Learning · Content Update · Age of Information · Transmission Latency

1 Introduction

The High Definition Map (HDM) is an essential tool to help autonomous vehicles make path planning and relative driving decision [1]. Generally, the HDM can be roughly divided into two layer called the static layer and the dynamic layer [2]. The static layer contains the road topology information while the dynamic layer contains the real-time traffic condition of the specific road section which

Y. Cao and X. Shao (Eds.): MONAMI 2022, LNICST 474, pp. 169–181, 2023.
https://doi.org/10.1007/978-3-031-32443-7_12

needs to update frequently. To better support the time-critical and the location-dependent features of the autonomous driving, caching the dynamic layer contents of the HDM at the network edge corresponding to their geographical location is a promising solution [3–5]. However, how to ensure the freshness of the requested content is still a fundamental problem in mobile edge caching by considering the limited network resource [6,7].

To better characterize the freshness characteristic of the dynamic contents, a novel metrics named Age of Information (AoI) has been proposed [8]. The AoI of a cached content is defined as the time elapsed since the content was generated from the source [9]. Based on the concept of AoI, researches on the dynamic content caching strategies have been carried out [10–15]. Relevant studies can be divided into two categories: minimizing the average/(peak) AoI [10–12] of a local cache system and realizing a tradeoff between AoI and request latency [13–15]. The former aims at exploring a content update policy to minimize the average/(peak) AoI of a local cache system by considering other factors such as content popularity as well while the latter jointly optimizes content freshness and request latency during the content update process (Existing research points out that delay-optimal may not be AoI-optimal [16]). Notice that, both categories of the above studies have their defects when applied to the edge HDM dynamic layer caching: 1) The former focuses on the AoI of cached items on the edge network, while the freshness of contents received by the user which acts as a more important performance indicator in a practical situation has been ignored. 2) Both of which make the analysis with the queuing theory frameworks where the request patterns of users are regarded as a prior knowledge. However, the dynamic layer contents of the HDM requested by different vehicles are almost infeasible to estimate since they depend on the vehicle's autonomous driving level and target path planning.

Recently, learning-based methods such as Markov Decision Process (MDP) and Reinforcement Learning (RL) [17–21] have been applied to make AoI optimization in a variety of caching problems. In particular, most of these methods are efficient in solving the cache update problems under no prior request condition. Their objectives are to minimize the AoI when the energy of the information collector is limited [17,18] or minimize the average AoI of all the contents under a transmission resource limited scenario [19–21]. However, for autonomous vehicles, they are more concerned with obtaining HDM's dynamic layer contents which can meet their AoI requirements within the possible lowest request latency [2]. Therefore, the existing learning-based methods still have their limitations when apply to the HDM dynamic layer contents caching.

In this paper, we investigate a dynamic HDM content update algorithm to satisfy the AoI requirements by autonomous driving vehicles and minimize the content request latency in an edge network, where the RSU doesn't know the vehicles' request patterns in advance. To balance the AoI requirement of vehicle and the content request latency with limited transmission resource, we consider the content update optimization as a discrete time slot decision problem to minimize the long-term discounted cost brought by the AoI difference (the difference

of the AoI requirement value and its actual value in a specific time slot) and the request latency. To solve the problem, we model the edge-cached HDM content update problem as an MDP and apply an RL-based algorithm [22] to obtain the optimal update decision. The proposed algorithm can well address the curse of dimensionality problem brought by the large state or action space, and doesn't need the prior information of the vehicular request. Extensive simulation results verify the efficiency of the proposed algorithm.

The rest of the paper is organized as follows. Section 2 introduces our concerned network model and formulates the problem. In Sect. 3, we transform the problem into an MDP and solve it with a model-free RL-based algorithm. Extensive simulation results are provided in Sect. 4. Finally, Sect. 5 concludes this paper.

2 System Model and Problem Formulation

2.1 Network Model

We consider a typical vehicular network scenario with a single Road Side Unit (RSU), F traffic information acquisition sensors and several vehicles under its communication range. Assuming that the RSU combined with the storage capability of the edge cloud server, and each sensor is responsible for refreshing the specific HDM content with the same size l cached on the RSU. The vehicular request and HDM dynamic content sets are denoted by $\mathcal{N} = \{1, 2, ..., N\}$ and $\mathcal{F} = \{1, 2, ..., F\}$ respectively. We focus on a discrete time slots system, where a time step $T(t)$ is defined to represent each decision epoch. $T(t)$ can be defined as the integral multiple of a constant time slot τ. During each time step, the RSU may receive the vehicular HDM content request, it then will decide whether to pull the up-to-date states of HDM contents from the relevant sensors due to the content AoI demands of vehicles, if any, the content update will be executed. On the other hand, the RSU will respond to the vehicular HDM content requests with its local cached contents.

Each vehicular request n includes its query details, which containing the requested HDM contents and the relevant content AoI demands based on the vehicle's driving path planning and autonomous driving level. Here, we use a query procontent $d_n(t) = \left\{d_n^1(t), d_n^2(t), ..., d_n^F(t)\right\}$ to represent the query details of the request n in time step t, where $d_n^f(t) \in \{d_{mid}, d_{high}, 0\}$ $(d_{mid} > d_{high})$, d_{mid} and d_{high} are two integers which depends on the autonomous driving level of each vehicle. We define that all the requested HDM contents in a same vehicular request n have the same AoI demand (d_{mid} or d_{high}). $d_n^f(t) = 0$ indicates that content f hasn't been requested in n at time step t. Meanwhile, the request indicator $\widetilde{d}_n(t)$ is set to represent whether request n exists in the time step t.

$$\widetilde{d}_n(t) = \begin{cases} 0, \sum_{f=1}^{F} d_n^f(t) = 0 \\ 1, Otherwise \end{cases} \tag{1}$$

The RSU can obtain the query procontents $D(t) = \{d_1(t), d_2(t), ..., d_N(t)\}$ of all the requests in time step t, but it has no prior knowledge of the vehicular request arrival rates and the popularity of each cached HDM content.

After the RSU received the query procontents $D(t)$, it will make the HDM dynamic content update decision based on the requested HDM contents and the relevant AoI demands. We use $U(t) = \{u_1(t), u_2(t), ..., u_F(t)\}$ to represent the HDM content update decision in time step t, where $u_f(t) \in \{0, 1\}$, $f \in \mathcal{F}$. $u_f(t) = 1$ represents that the RSU decides to refresh content f and pull the up-to-date states from the relevant sensor in time step t, otherwise $u_f(t) = 0$. The RSU will select content f to refresh in time step t based on the comparison of its real-time AoI on the RSU and the AoI demand in the query procontent $D(t)$. Notice that, when there is no query of content f in the query procontents $D(t)$, content f may also be updated to reduce the transmission delay caused by the temporary request update if there is available transport resources. Then, the RSU responds the vehicular requests with its cached HDM contents.

In our network model, we consider that the RSU is assigned with limited transmission resource blocks which are orthogonal to each other, the number of its whole available resource blocks is H_b. The maximum number of requests which can be served by the in a time step is N. Each resource block is allocated to a different communication point for the wireless data transmission. The actual number of requests in each time step $|\mathcal{N}(t)|$ is defined as:

$$|\mathcal{N}(t)| = \sum_{n=1}^{N} \widetilde{d}_n(t) \tag{2}$$

Typically, $(H_b - |\mathcal{N}(t)|)$ is the number of available resource blocks which can be divided for the HDM content updating. Therefore, at most $(H_b - |\mathcal{N}(t)|)$ sensors can execute the HDM content refreshing simultaneously in time step t. We use the content transmission occupancy rate $\beta(t)$ to denote the proportion of resource blocks occupied for transferring contents from RSU to the vehicles. As for the HDM content update decision $U(t)$ made by the RSU in the time step t, some contents whose $u_f(t) = 1$ may not be refreshed due to transmission resource constraints. We set an update success indicator $y_f(t) \in \{0, 1\}$ $(y_f(t) \leq u_f(t))$ to represent whether a content f whose $u_f(t) = 1$ has been refreshed successfully or not, which satisfies: $\sum_{f=1}^{F} y_f(t) \leq H_b - |\mathcal{N}(t)|$.

Without loss of generality, we assume that the content update time consumption of each HDM content remains the same in different time steps due to the identical content update size and transmission time, which can be abstracted as $\mathbb{T}_r = \left\{\mathcal{T}_r^1, \mathcal{T}_r^2, ..., \mathcal{T}_r^F\right\}$ $(\mathcal{T}_r^f < T(t), f \in \{1, 2, ..., F\})$.

2.2 AoI Analysis

Based on the proposed network model, we analyse the real-time change in the value of cached HDM contents' AoI on the RSU and the influence of content response latency on AoI when the requested content received by the vehicle.

In order to prevent the impact on the vehicle caused by the staleness of the requested HDM contents, we define a metric α_{max} called the maximum allowable AoI, which represents the maximum AoI a content cached on the RSU can reach. Specifically, the value of α_{max} can be set to d_{mid}. For the RSU, the real-time AoI value of its cached HDM content f can be expressed as:

$$\alpha_0^f(t) = \begin{cases} T(t-1), & y_f(t) = 1 \\ min\left\{\alpha_0^f(t-1) + T(t-1), \alpha_{max}\right\}, & Otherwise \end{cases} \tag{3}$$

Figure 1 illustrates the AoI variation for the RSU cached HDM content due to the vehicular requests and the pre-defined maximum AoI threshold.

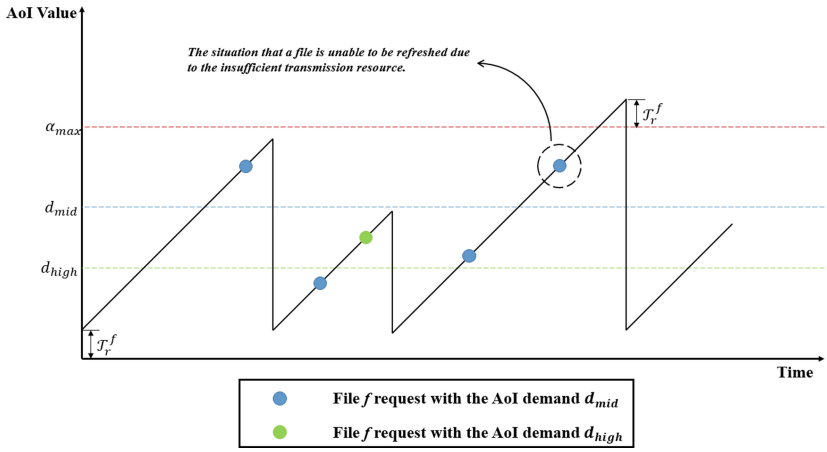

Fig. 1. The AoI variation for the cached HDM content on the RSU.

For the vehicle that makes the request n, the transmission latency brought by the content respond process also increases the staleness of the information. To ensure a requested content f which can be directly responded without updating still meets the vehicular AoI demand when it received by the vehicle, we redefine the actual AoI demand of each request as $d_{mid} - T_{lat}^{max}$ and $d_{high} - T_{lat}^{max}$, where T_{lat}^{max} represents the maximum data transmission latency of a request n that can be generated under the proposed network model in this paper.

$$L_n(t) = \begin{cases} 0, & \sum_{f=1}^{F} d_n^f(t) y_f(t) = 0 \\ max\left\{\mathcal{T}_r^f \mid \mathcal{T}_r^f \in \mathbb{T}_r, d_n^f(t) y_f(t) = 1\right\}, & Otherwise \end{cases} \tag{4}$$

2.3 Problem Formulation

In this paper, our objective is to meet the AoI requirements of vehicles with limited transmission resources and reduce the extra request latency by designing

a dynamic content update mechanism. The extra request latency cost $L_n(t)$ of the vehicular request n in time step t is depended on whether the requested HDM contents need to be updated or transmit to the vehicle directly, which is expressed in the Eq. (4).

To better characterize the satisfaction with the AoI of the requested HDM content, we define a new metric called AoI difference cost $\Delta_n^f(t)$ to represent the overhead caused by the request HDM content f not meeting the corresponding AoI requirements when it received by the vehicle, which can be expressed as:

$$\Delta_n^f(t) = \begin{cases} 0, & \alpha_n^f(t) - d_n^f(t) \leq 0 \quad or \quad d_n^f(t) = 0 \\ \alpha_n^f(t) - d_n^f(t), & Otherwise \end{cases} \tag{5}$$

As for the vehicle who sent the request n, we use the average AoI difference cost $\overline{\Delta}_n(t)$ of all the HDM contents it requested as the representative of its AoI satisfaction within time step t, which can be expressed as:

$$\overline{\Delta}_n(t) = \frac{1}{\sum_{f=1}^{F} u_f(t)} \sum_{f=1}^{F} \Delta_n^f(t) \tag{6}$$

According to the above analysis, the AoI related cost during each time step can be expressed as the weighted sum of each vehicle's average AoI difference cost, that is:

$$\Delta_{AoI}(t) = \sum_{n=1}^{N} \beta_n \overline{\Delta}_n(t) \tag{7}$$

where $\sum_{n=1}^{N} \beta_n = 1 \quad \beta_n \in [0, 1]$, and the value of each β_n depends on the automatic driving level of the vehicle. Vehicle with higher automatic driving level possesses a higher value β_n.

The overall system cost in each time step t can be expressed as:

$$C_{tot}(t) = \omega_{AoI} \Delta_{AoI}(t) + \omega_L \frac{1}{N} \sum_{n=1}^{N} L_n(t) \tag{8}$$

where $\omega_{AoI} + \omega_L = 1$, ω_{AoI} and ω_L can realize a tradeoff update decision between the AoI difference cost of vehicles and the extra request latency cost in each time step. We adopt a larger ω_{AoI} than ω_L in this paper, for the AoI requirements of the requested HDM content is more important than the content request latency for an automatic driving vehicle.

Based on the cost function (8), the average future cost of the HDM content requests can be defined as:

$$C_{ave} = \lim_{T_{max} \to \infty} \frac{1}{T_{max}} \mathbb{E}\left(\sum_{t=0}^{T_{max}} C_{tot}(t) \right) \tag{9}$$

The RSU should make optimal HDM contents refreshing decisions by interacting with the environment to minimize C_{ave}.

3 Deep Reinforcement Learning-Based HDM Updating Algorithm

To achieve the expected performance, we formulate the HDM content update process on the RSU as an MDP, for which we build a DRL-based algorithm to obtain the optimal content update strategy.

3.1 MDP Model

Our MDP is modeled as a 4-tuple $\langle S, A, P, R \rangle$, relevant details are described as below:

- ***Modeling of System State Space S:*** $s(t) = (s_0(t), s_1(t), ..., s_N(t))$ is defined as the system state at time step t, which is composed of the real-time HDM content AoI value on the RSU $s_0(t) = (\alpha_0^1(t), \alpha_0^2(t), ..., \alpha_0^F(t))$ and the vehicular content AoI demands $s_n(t) = (d_n^1(t), d_n^2(t), ..., d_n^F(t))$ $n \in \mathcal{N}$. The whole state space S can be regarded as a combination of communication node states ($S = s_0 \times s_1 \times s_2 \times ... \times s_N$) in the proposed network, which is finite due to the maximum AoI value restriction.
- ***Modeling of System Action Space A:*** $a(t) = (u_1(t), u_2(t), ..., u_F(t))$ is defined as the system action at time step t, which represents the HDM content update decision of the RSU. The action space A of the system can be expressed as:

$$A = \{U \mid u_n \geq y_n, u_n \in \{0, 1\}, y_n \in \{0, 1\}, \forall n \in \mathcal{N}\} \tag{10}$$

- ***System State Transition Probability P:*** $P = S \times A \times S \rightarrow [0, 1]$ represents the distribution of the transition probability $P(s' \mid s, a)$ from the system state s to a new system state s' $(s, s' \in S)$ when an action $a \in A$ is chosen, which is largely effected by the real environment conditions, such as the HDM content request rate, the HDM content transmission occupancy rate $\beta(t)$ of the RSU resource blocks etc.
- ***Modeling of Reward Function R:*** $S \times A \rightarrow R$ maps a state-action pair to a value $R(s(t), A(t))$. Our objective in this paper is to minimize the average future cost $C_{ave}(t)$ given in Eq. (9), so that we define the reward function as $R(s(t), a(t)) = -C_{ave}(t)$.

We define the policy π as an action $a \in A$ that the RSU will execute by given a specific system state $s \in S$. Policy π is uncorrelated to the time step length. The difficulty here is to find an optimal policy π^* to maximize the long-term average reward, that is:

$$\arg\max_{\pi^*} \lim_{T_{max} \rightarrow \infty} \frac{1}{T_{max}} \mathbb{E}\left[\sum_{t=0}^{T_{max}} R(s(t), a(t)) \mid s(0)\right] \tag{11}$$

3.2 HDM Updating Algorithm Design

With the MDP model aforementioned, we need to design an adaptive and efficient HDM dynamic layer update strategy, which can proactively make content update decision in each state, so as to earn a higher reward by considering the long-term system performance.

Deep Reinforcement Learning (DRL) is a model-free method to solve MDP problems with large state or action space [22]. The goal of DRL is to maximize the long-term discounted reward by utilizing the deep neural network (DNN) as an approximation function to learn policy and state value. The agent can obtain enough experience by interacting with the environment, and train its policy network model. The well-trained model can quickly perform the optimal actions for executing content update. The state value function $V_\pi(s)$ and the state-action value function $Q_\pi(s,a)$ of the DDQN can be expressed as follows:

$$V'_\pi(s) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k R(s(t+k), \pi(t+k)) \mid s(t) = s\right] \tag{12}$$

$$Q'_\pi(s,a) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k R(s(t+k), a(t+k)) \mid s(t) = s, a(t) = a\right] \tag{13}$$

where γ is the discount factor.

The optimal policy π^* can be obtained by utilizing the Bellman Optimality Equation:

$$V_{\pi^*}(s) = max_{a \in A} Q_{\pi^*}(s,a) \tag{14}$$

The architecture of our DRL-based HDM dynamic layer update mechanism is presented in Fig. 2. θ and θ^- are the DNN parameters of the main network and the target network respectively. The agent interacts with the environment and observes the real-time system state. Based on the current state $s(t)$, the agent selects an action by utilizing the ϵ-greedy strategy (select the action $max_a R(s,a,\theta)$ with probability $(1-\epsilon)$, and randomly select action $a \in A$ with probability ϵ, where $\epsilon \in [0,1]$). After the agent performs an action $a(t)$, the corresponding reward $R(s(t),a(t))$ can be obtained from environment, and the system state $s(t)$ transfers to $s(t+1)$. So that a new experience tuple $\mathcal{E}(t) = (s(t), a(t), R(s(t),a(t)), s(t+1))$ is generated and will be cached in the experience replay buffer $\mathbb{M}$. Then, the former steps go into a loop to obtain enough experience in the replay buffer for the future training. The oldest experience tuple will be discarded when the experience buffer $\mathbb{M}$ is full.

As for the training procedure, a mini-batch of the cached experience tuples $\mathbb{W} = \{\mathcal{E}_1, \mathcal{E}_2, ..., \mathcal{E}_{W_m}\}$ will be sampled randomly from the experience replay buffer. The goal of the training procedure is to minimize the loss function $L(\theta)$, which can be expressed as:

$$L(\theta) = \mathbb{E}\left[(R(s_j,a_j) - \gamma \max_{a'_j} Q'(s'_j, a'_j; \theta^-) - Q(s_j,a_j;\theta))^2\right] \tag{15}$$

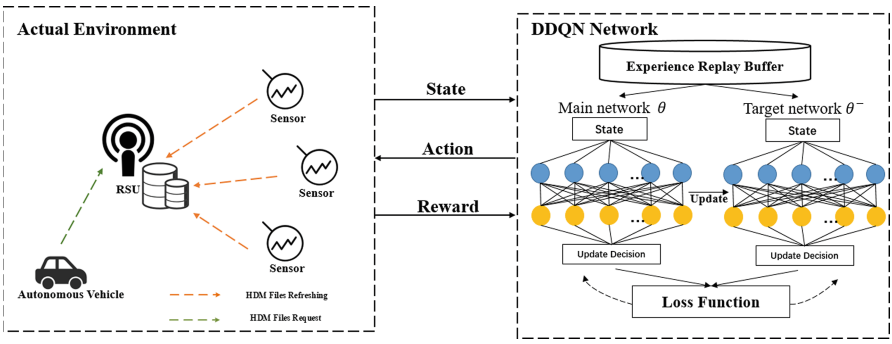

Fig. 2. The architecture of the proposed mechanism

The DNN parameter θ updates iteratively as Eq. (16):

$$\theta' = \theta + \xi \nabla_\theta L(\theta) \tag{16}$$

where ξ is the learning rate. The parameter θ of the main network is updated every step while the parameter θ^- of the target network will be updated every i steps, that is $\theta_t^- = \theta_{t-i}$.

4 Simulation Results and Discussions

In this section, we evaluate the performance of the proposed HDM updating algorithm. Firstly, we describe our simulation settings and the present the baseline algorithms used for the performance comparison. Then, we show the performance comparison of the proposed algorithm with the baseline policies in different environments and give the relevant analysis. The whole experiment is implemented by the Tensorflow frame and runs on a PC with an Intel Core i7-6700 CPU @2.6 GHz, Memory 16G.

4.1 Simulation Settings

We build a simulation scenario with one RSU (integrated with an MEC server), N connected vehicles and 10 traffic information acquisition sensors. The value of N ranges from 10 to 40. We set the available number of the orthogonal transmission resource blocks as $H_b = 50$. In each time step, we set each vehicular request for each edge-cached HDM content subjecting to a random distribution. For each sensor, the relevant content update latency is randomly selected from the value set $\{0.8\tau, 0.9\tau, 1.0\tau, 1.1\tau, 1.2\tau\}$, where $\tau = 1$ is the length of the unit time slot. Once the content update latency of each sensor has been determined, their values will remain unchanged during the whole simulation process. Based on this, we set the extra request latency of a specific content to be the same as its update latency. Without loss of generality, we set the value of maximum allowable AoI

$\alpha_{max} = d_{mid} = 20$ and $d_{high} = 10$. The value of the HDM content transmission occupancy rate $\beta(t)$ is set to be 0.3 to 0.8 randomly. The mini-batch size is set to be 64. The value ω_{AoI} is set to be 0.6. The exploration rate increases linearly from 0 to 1 and keeps fixed.

The baseline algorithms which used to be compared with the proposed algorithm are described as follows:

Random Policy: During each time step, the RSU randomly selects an update action for the current state if there are available transmission resources.
Greedy Policy: During each time step, the RSU will execute the update action which can maximize the immediate reward when there are available transmission resources.

4.2 Simulation Results

Convergence Performance. To ensure the reliability of our proposed method, we first verify its convergence performance.

Figure 3 shows the convergence comparison of the proposed algorithm and the baseline policies when $N = 30$. Here, we also evaluate the performance of the proposed algorithm under different discount factor γ. It can be seen from Fig. 3 that the long-term reward becomes higher with a bigger γ. However, the network model became hard to converge when the value of γ is too big (0.98). Meanwhile, compared with the greedy policy and the random policy, our proposed algorithm obtained a significant high reward.

Based on the above analysis, we find that when the value of the discount factor is 0.95, the proposed policy shows its best performance. So in the subsequent simulation, we set $\gamma = 0.95$ in the proposed policy.

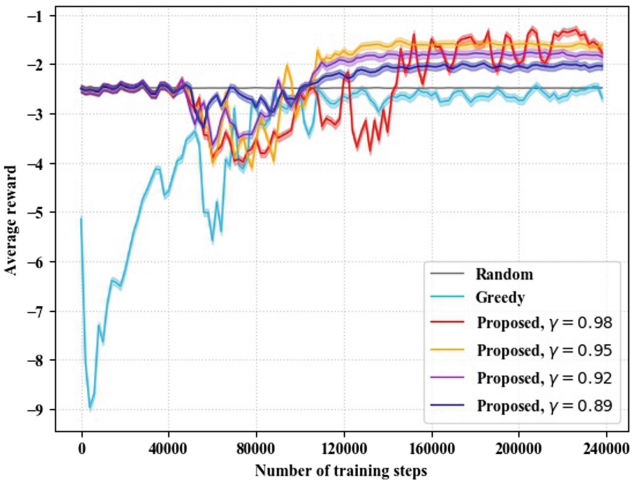

Fig. 3. The training rewards comparison under different policy.

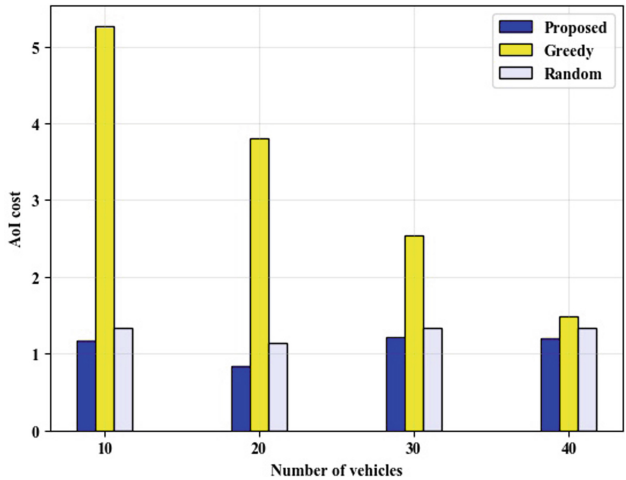

Fig. 4. Performance comparison in AoI cost.

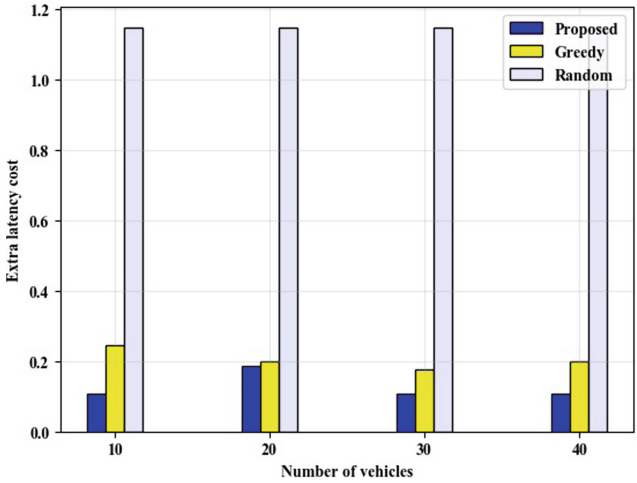

Fig. 5. Performance comparison in extra latency cost.

Efficiency Analysis. To verify the efficiency of our proposed method, we make performance comparison with the mentioned baseline policies.

Figure 4 shows the AoI cost which brought by the AoI difference when vehicle requests the cached HDM contents under different number of vehicles. It can be observed from Fig. 4 that while the RSU adopts the proposed policy, the AoI cost maintains a relatively low and stable value compared with the baseline policies. The proposed policy considers to maximize the long-term discounted reward, it can execute optimal update actions in response to the vehicular requests. So

that proposed policy can ensure the stability of the AoI cost performance and realize a reasonable utilization of the network resources.

Figure 5 shows the extra latency cost which brought by the instant content updating when the content is requested by the vehicle. It can be seen from Fig. 5 that the proposed algorithm keeps a relatively stable latency cost with the number of vehicle increasing. Notice that, even though we have already emphasized the effect of the AoI cost in our previous parameter settings ($\omega_{AoI} = 0.6$), the performance of the proposed algorithm on the extra latency cost is much better than the baseline policies.

5 Conclusion

This paper focused on the HDM update problem in the edge caching system. We first formulated the HDM update optimization problem as how to minimize the AoI difference and the extra request latency in the scenario where transmission resources are limited. Then, we modeled the problem as an MDP and utilized a DRL-based algorithm to obtain the optimal update strategy. We have verified the performance of the proposed algorithm through the simulations, The results shown that, compared with the baseline policy, our proposed algorithm could achieve higher long-term reward with suitable discounted factor, and realized relative low AoI and request latency.

Acknowledgment. This paper is supported by the Fundamental Research Funds for the Central Universities (No. 2022JBGP005).

References

1. Liu, R., Wang, J., Zhang, B.: High definition map for automated driving: overview and analysis. J. Navig. **73**(2), 324–341 (2020)
2. Xu, X., Gao, S., Tao, M.: Distributed online caching for high-definition maps in autonomous driving systems. IEEE Wirel. Commun. Lett. **10**(7), 1390–1394 (2021)
3. Bastug, E., Bennis, M., Debbah, M.: Living on the edge: the role of proactive caching in 5G wireless networks. IEEE Commun. Mag. **52**(8), 82–89 (2014)
4. Vu, T.X., Chatzinotas, S., Ottersten, B.: Edge-caching wireless networks: performance analysis and optimization. IEEE Trans. Wirel. Commun. **17**(4), 2827–2839 (2018)
5. Yuan, Q., Zhou, H., Li, J., Liu, Z., Yang, F., Shen, X.S.: Toward efficient content delivery for automated driving services: an edge computing solution. IEEE Netw. **32**(1), 80–86 (2018)
6. Qiao, G., Leng, S., Maharjan, S., Zhang, Y., Ansari, N.: Deep reinforcement learning for cooperative content caching in vehicular edge computing and networks. IEEE Internet Things J. **7**(1), 247–257 (2020)
7. Yao, J., Han, T., Ansari, N.: On mobile edge caching. IEEE Commun. Surv. Tutor. **21**(3), 2525–2553 (2019)
8. Kaul, S., Yates, R., Gruteser, M.: Real-time status: how often should one update? In: 2012 Proceedings IEEE INFOCOM, pp. 2731–2735 (2012)

9. Kosta, A., Pappas, N., Angelakis, V.: Age of Information: A New Concept, Metric, and Tool (2017). https://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=8187436
10. Tang, H., Ciblat, P., Wang, J., Wigger, M., Yates, R.: Age of information aware cache updating with content- and age-dependent update durations. In: 2020 18th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOPT), pp. 1–6 (2020)
11. Yang, L., Zhong, Y., Zheng, F.-C., Jin, S.: Edge caching with real-time guarantees (2019). arXiv:1912.11847
12. Kam, C., Kompella, S., Nguyen, G.D., Wieselthier, J.E., Ephremides, A.: Information freshness and popularity in mobile caching. In: 2017 IEEE International Symposium on Information Theory (ISIT), pp. 136–140 (2017)
13. Zhang, S., Wang, L., Luo, H., Ma, X., Zhou, S.: AoI-delay tradeoff in mobile edge caching with freshness-aware content refreshing. IEEE Trans. Wirel. Commun. **20**(8), 5329–5342 (2021)
14. Cao, J., Zhu, X., Jiang, Y., Wei, Z.: Can AoI and delay be minimized simultaneously with short-packet transmission? In: IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), IEEE INFOCOM 2021, pp. 1–6 (2021)
15. Zhang, S., Li, J., Luo, H., Gao, J., Zhao, L., Shen, X.S.: Low-latency and fresh content provision in information-centric vehicular networks. IEEE Trans. Mob. Comput. (Early Access) (2020)
16. Najm, E., Nasser, R.: Age of information: the gamma awakening. In: 2016 IEEE International Symposium on Information Theory (ISIT), pp. 2574–2578 (2016)
17. Hatami, M., Leinonen, M., Codreanu, M.: AoI minimization in status update control with energy harvesting sensors. IEEE Trans. Commun. **69**(12), 8335–8351 (2021)
18. Wang, S., et al.: Distributed reinforcement learning for age of information minimization in real-time IoT systems. IEEE J. Sel. Top. Signal Process. (Early Access) (2022)
19. Ceran, E.T., Gündüz, D., György, A.: A reinforcement learning approach to age of information in multi-user networks with HARQ. IEEE J. Sel. Areas Commun. **39**(5), 1412–1426 (2021)
20. Kam, C., Kompella, S., Ephremides, A.: Learning to sample a signal through an unknown system for minimum AoI. In: IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), IEEE INFOCOM 2019 (2019)
21. Sert, E., Sönmez, C., Baghaee, S., Uysal-Biyikoglu, E.: Optimizing age of information on real-life TCP/IP connections through reinforcement learning. In: 2018 26th Signal Processing and Communications Applications Conference (SIU), pp. 1–4 (2018)
22. van Hasselt, H., Guez, A., Silver, D.: Deep reinforcement learning with double Q-learning. In: AAAI, vol. 30, no. 1 (2016)