Data Synchronization for Vehicular Digital Twin Network

Xiaoqing Yang[1], Jinkai Zheng[1], Tom H. Luan [1,†], Rui Li[2], Zhou Su[3] and Mianxiong Dong[4]
[1]School of Cyber Engineering, Xidian University, Shaanxi, China
[2]School of Computer Science and Technology, Xidian University, Xian, Shaanxi, China
[3] School of Cyber Science and Engineering, Xi'an Jiaotong University, Xian, Shaanxi, China
[4]Department of Sciences and Informatics, Muroran Institute of Technology, Muroran City, Japan
[†]Corresponding author: tom.luan@xidian.edu.cn

GLOBECOM 2022 - 2022 IEEE Global Communications Conference | 978-1-6654-3540-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/GLOBECOM48099.2022.10000831

***Abstract*—This paper considers the downlink data synchronization from the digital twin (DT) to the vehicle, in which a vehicle drives through consecutive roadside units (RSUs) along its trip, and the DT on the cloud transmits the data to the vehicle through the relay of RSUs. To this goal, the DT first chops the data into blocks and cache them in the RSUs along the driving path of the vehicle. The vehicle can then retrieve the blocks when driving into the RSU's coverage to recover the data. Since RSUs have different cache capacities and communication costs, the DT needs to determine how to optimally distribute the data blocks at RSUs so that vehicles can finish downloading all the data before the deadline yet with the minimal cost. To determine the optimal delivery strategy of DT, we model the problem as an optimization framework subject to the time-varying wireless channel of RSUs, their service load and the communication cost. We then resort to the Lyapunov optimization to derive a distributed solution. Using extensive simulation results, we demonstrate that our scheme can effectively reduce the cost of data synchronization and improve the network load performance.**

***Index Terms*—Data synchronization, Digital twin, Lyapunov optimization**

I. Introduction

The digital twin (DT) has attracted more and more attention from the industry and academia in recent years. It builds a bridge between the physical space and the digital system by establishing a high-correspondence digital model with a virtual representation of the physical world, which can significantly promote the operation of physical vehicles in the physical world [1].

Many researchers have conducted research based on the DT framework and have achieved remarkable results in researching the Internet of Vehicles (IoV). For example, Uhlemann *et al*. [2] propose to obtain, mine and analyze the state information of the entity vehicle from the DT to assist in driving. They also propose that the DT could select the optimal route from several candidate routes for the entity vehicle in time to alleviate traffic congestion by collecting the location information of each vehicle and constructing a physical network topology. At the same time, data processing and transmission have higher requirements for network resource allocation in the big data era. Applying DTs in intelligent vehicle networks is available to solve the problem of network resource allocations. Ning *et al.* [3] investigate a scenario in which the DT can predict the vehicle's driving track to assist the vehicle in performing computing task offloading and content caching, which could relieve the processing pressure on the core network and data centre. Huang *et al.* [4] utilize DTs cache files with high popularity in multiple Road Side Units (RSUs) for users' download in advance, which dramatically reduces the request-response time and maximizes network resources utilization.

Most research focuses on using the information already collected by the DTs to analyze the global information to propose some resource allocation schemes [5]. However, existing literature typically relies on accurate data transmitted through the Digital-To-Vehicle (DTV). The current DTV system still has the following challenges: 1) The traditional centralized one-time data transmission leads to heavy network load and data transmission delay when data is enormous. 2) Resource utilization maximization is taken into consideration, while other conditions such as network stability are ignored in many studies. Therefore, considering data synchronization between DT and the vehicle during driving, we propose a joint decision system between DT and vehicle to ensure network stability and the maximization of network resources utilization. The major contributions of this paper can be summarized as follows:

- We propose the DTV intelligent data synchronization system, which establishes an intermediate buffer and communication connection through RSUs. The vehicle which travels the coverage of RSUs would receive data collected by the DT. The DT can chop the data into blocks and store them in RSUs along the way. In this way, the vehicle can complete the data synchronization after the journey.
- We propose a system constrained by the capacity of RSUs, data transmission latency and network stability. To this end, we describe it as a finite time series decision problem and formulate an optimization problem that minimizes the cost of data synchronization.
- Since the problem is constrained by long-term network stability, we leverage Lyapunov optimization [6] to transform the original long-term problem into determinable sub-problems. In addition, the A3C algorithm in reinforcement learning is used to assist in solving the optimal solution of Lyapunov optimization in each time slot.

II. System Model

In this section, we describe the model of the proposed DTV data synchronization system and introduce the sub-models involved in the system in detail.

A. Digital Twin Decision System

Fig. 1 shows the proposed DTV synchronization system model. The DT network mainly provides data for the underlying physical vehicle network and provides intelligent decision-making in the data transmission process. We consider a one-way road with M RSUs with different buffer capacities. The RSUs deployed on the roadside can provide caching services for vehicles within its coverage. Meanwhile, a group of entity vehicles N pass through the road, and DTs collect the required data for them in advance. Data would chop into blocks through intelligent decision-making and delivered to RSUs by the DT. We assume that a vehicle is connected to at most one RSU at a time. In addition, the road segment is covered with multiple RSUs, and the coverage of each RSU does not overlap. Therefore, the data synchronization system needs to ensure that the transmission time of the entire data is less than the vehicle's driving time on the road.

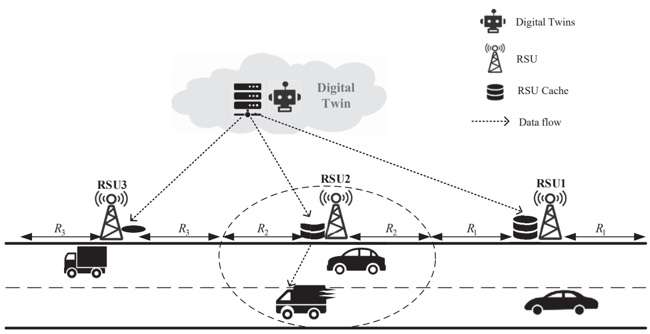

Fig. 1. System Model.

B. Communication Model

Since the DTs are deployed on cloud servers, we ignore the communication cost and time between the DT and RSU m in this paper. Furthermore, we consider communication between vehicle n and RSU m through the wireless network and that RSUs allocate channels by orthogonal multiplexing. The transmission rate between the vehicle n and RSU m is

$$r_{nm}^{t} = B\log_2\left(1 + \frac{p_n s_{nm}^{t}}{\sigma^2}\right), \tag{1}$$

where B is the available spectral bandwidth of the vehicle, p_n represents the transmission power of vehicle n, s_{nm}^{t} represents the channel gain between vehicle n and RSU m at the time slot t, σ^2 represents the noise power. s_{nm}^{t} is defined as

$$s_{nm}^{t} = \left|\vartheta_{nm}^{t}\right|^2 h_{nm}^{t},\ t = 1, 2, ..., \tag{2}$$

where ϑ_{nm}^{t} is the small-scale path loss and h_{nm}^{t} is the large-scale path loss and shadowing. We adopt a general block

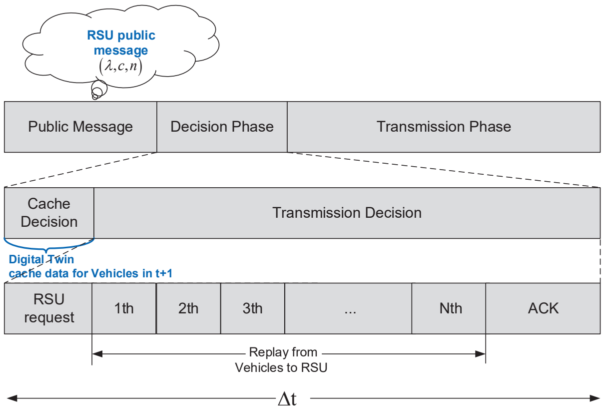

Fig. 2. Channel Contention Model.

fading channel model to our environment and use the first-order Gauss-Markov process to describe the small-scale fading component. The update of ϑ_{nm}^{t} can be defined as

$$\vartheta_{nm}^{t} = \kappa\vartheta_{nm}^{t-1} + \frac{1}{\left(\sqrt{\left(L_n^t - M_m^t\right)^2 + {H_m}^2}\right)^{\alpha}}. \tag{3}$$

Here, $\vartheta_{nm}^{t} \sim \mathcal{CN}(0,1)$ is a circularly symmetric complex Gaussian distribution unit variance, κ is the correlation coefficient. L_n^t and M_m^t are the position of vehicle n and RSU m, respectively. H_m is the height of RSU m and α is the path loss indicator.

C. Channel Contention Model

The accuracy of DT intelligent decision-making requires an accurate understanding of vehicle channel conditions. We assume that data should be transmitted when the vehicle obtains the channel to utilize the channel resources fully. Otherwise, we consider that DT and vehicles waste the channel resources. At the same time, transmitting data to multiple mobile users causes excessive interference and heavy network load [7]. We assume that RSUs allocate their channels to vehicles at a time slot, which means that only one vehicle can get a chance to communicate with the RSU in this slot. For the ease of implementation, we propose that RSU can keep track of the number of vehicles traveling within its coverage area at time slot t.

The channel contention model is shown in Fig. 2. We divide a vehicle's driving time into multiple time slots and divide each time slot into three phases: the public information phase, the decision-making phase, and the transmission phase. At the beginning of each time slot, the RSU publishes $message = \{\lambda, c, n\}$ to physical vehicles within its coverage and all DTs, where λ represents the storage price of RSU at the time slot t, c represents the remaining capacity of RSU, and n represents the number of vehicles that access to RSU. After the vehicle receives the message, it enters the decision-making phase, which includes caching and transmitting data. The caching phase is mainly DT caching data for vehicles

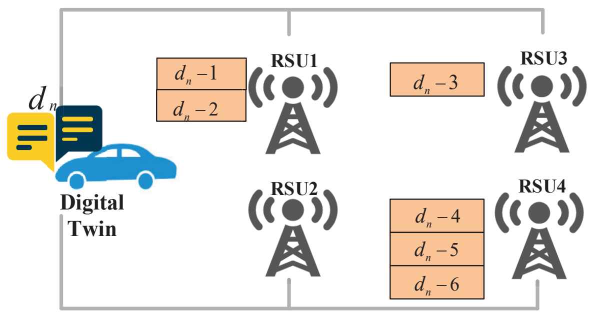

Fig. 3. Cache Model.

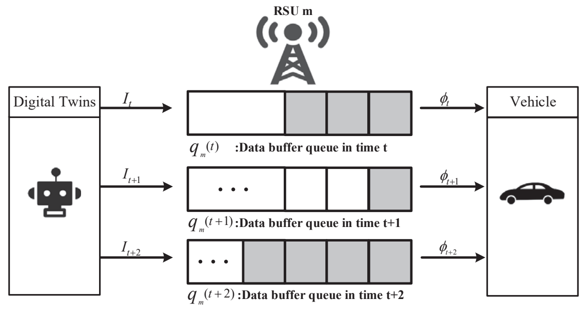

Fig. 4. Lyponuv Queue.

arriving at RSU at time slot $t+1$. Since DTs are deployed on cloud servers, their decision time and communication time are short. Therefore, we ignore them in this paper. In other words, while the DT is making cache decisions for the next time slot, the vehicle is also making transmission decisions. The process is specifically described as follows:

1) After the vehicle receives the broadcast message, the RSU would send a data transmission request to the vehicles in the cache list, and these vehicles will reply transmission decision $x_{nm}(t)$ to the RSU. When $x_{nm}(t)$ is equal to 1, which means that the vehicle n requests data transmission from the RSU m. Otherwise, $x_{nm}(t)$ is equal to 0, which means the vehicle doesn't request. Furthermore, each request would generate a cost of requesting the channel and we define the cost as q.

2) When the RSU receives these requests, it will randomly select one of the vehicles with these transmission requests to send ACK. When one of the vehicles receives the ACK, the RSU starts to transmit data to this vehicle until the end of this time slot.

D. Cache Model

The object of the cache model is mainly to divide the data into blocks and cache them in RSUs. According to the historical channel allocation and capacity of RSU at the current slot, the DT can evaluate the data size that the vehicle can receive when the vehicle accesses the RSU. In this case, the DT can cache data blocks for its physical vehicle in this RSU. We assume that the data of vehicle n (denoted as d_n) is divided into b blocks(denoted as b_n) and the number of blocks placed in RSU m is k blocks (denoted as k_{nm}). We set the $y_{nm}^t = \frac{k_{nm}}{b_n}, y_{nm}^t \in [0,1]$ to represent the ratio of the number of blocks stored in RSU m to all blocks of the data. Specifically, as shown in Fig. 3, the DT divides d_n into six blocks and caches two blocks in RSU 1, so we can calculate $y_{n1}^t = \frac{1}{3}$. Meanwhile, the block caching strategy must satisfy the following definition:

Definition 1. The block caching strategy is subject to the data integrity, and we have

$$\sum_{m=1}^{M} y_{nm}^t = \frac{k_{nm}}{b_n} \geq 1. \tag{4}$$

E. Queue Model

We also consider the stability of the network, so we characterize the stability of the network by establishing RSU data buffer queue as shown in Fig. 4. The RSU has a task buffer queue, and Q_m represents the queue length of the data buffer of the RSU. Let I_m^t represent the total amount of data that DTs caches in RSUs, and ϕ_m^t represent the total amount of data that would be pushed to the physical vehicle at time t. $Q_m(0)$ denote the initial queue. We assumed that the queue is the mean rate stable if the long-term average output data rate from the queue is greater than or equal to the long-term average input data into the queue. Thus, the network is stable when the data queue is the mean rate stable. The updating of the data buffer queue of RSU in each time slot is formulated by

$$\begin{aligned} Q_m(t+1) &= \max\{Q_m(t) - \phi_m^t, 0\} + I_m^t \\ &= \max\{Q_m(t) - \phi_m^t, 0\} + \sum_{n=1}^{N} y_{nm}^t d_n. \end{aligned} \tag{5}$$

III. Problem Formulation and Solutions

A. Problem Formulation

We model the DTV transmission as a discrete-time slot system. In (6), we divide the timeline, where $\mathcal{T}_m$ denotes the travel time within the coverage of RSU m and $\tau(m,1)$ denotes the first slot in RSU m. Besides, the variables of transmission decision and block caching decision in time slot t is denoted as a 2-tuple, *i.e.*, $Dec_n(t) = \{x_{nm}(t), y_{nm}(t)\}$.

$$t \in T = \bigcup_{m \in M} \mathcal{T}_m = \bigcup_{m \in M} \{\tau(m,1), \ldots, \tau(m, T_m)\}. \tag{6}$$

Meanwhile, we consider that the transmission system cost is mainly composed of three essential parts:

- The cost of the DT for caching data.
- The connection costs between the vehicle and RSUs, including the cost of channel contention and data transmission.
- Penalty costs for DT decision.

When the DT makes the cache decision, it assumes that the vehicle would try to establish a request connection with RSU m at each time slot when the vehicle is within the coverage of RSU m. We assume that the probability of each successful connection is $p^c = \varphi(n)$, where n is the number of vehicles that access RSU m in the slot and $\varphi(n)$ is a monotonically decreasing function of n. The more vehicles there are, the lower the probability of vehicles successfully connected to RSU m. Therefore, the maximum amount of data d_{nm}^t that is placed in RSU m is calculated as

$$d_{nm}^t = y_{nm}^t d_n = \lceil p^c \cdot \Delta t \cdot r_{nm}^t \rceil, \tag{7}$$

where $\lceil \cdot \rceil$ is the upper-bound function and Δt denotes the size of the time slot. Let q denote the channel contention price per slot and λ_m present the transmission price of one bit. Therefore, the channel contention decision cost and transmission decision cost of the vehicle n in $\mathcal{T}_m$ can be expressed as

$$\sum_{t=0}^{\mathcal{T}_m} q x_{nm}^t + \sum_{t=0}^{\mathcal{T}_m} \lambda_m \cdot r_{nm}^t \Delta t. \tag{8}$$

The vehicle is still within the coverage of RSU m if the vehicle has received all the data blocks. In this case, we believe that the vehicle can still receive some data on the rest of the road, but the DT does not cache adequate data for the vehicle. Therefore, the communication resources obtained by the vehicle are not fully utilized, which means the vehicle does not make a connection request every time slot. So the DT would be punished. Let $\mathcal{T}_m - \sum_{t=0}^{\mathcal{T}_m} x_{nm}^t$ denote the number of time slots that the vehicle without the connection request, $Z(t)$ is an increasing function of t, and the penalty function can be defined as

$$Z\left(\mathcal{T}_m - \sum_{t=0}^{\mathcal{T}_m} x_{nm}^t\right). \tag{9}$$

Therefore, the total cost of the DT transmitting data to the vehicle n through RSU m can be expressed as

$$\begin{aligned} E_n = &\sum_{t=0}^{\mathcal{T}_m} \left(q x_{nm}^t + \lambda_m \cdot r_{nm}^t \Delta t + \mu_m d_{nm}^t\right) \\ &+ Z\left(\mathcal{T}_m - \sum_{t=0}^{\mathcal{T}_m} x_{nm}^t\right), \; q, \mu_m > 0. \end{aligned} \tag{10}$$

Where μ_m represents the cost of caching one bit of data in RSU m, and the data cache price of each RSU is different. Our goal is to minimize the long average cost of the system,

$$\begin{aligned} E = \; & \tfrac{1}{T} \sum_{n=0}^{N} \sum_{m=0}^{M} \left\{ \left(\sum_{t=0}^{\mathcal{T}_m} q x_{nm}^t + \lambda_m \cdot r_{nm}^t \Delta t + \mu_m d_{nm}^t \right) \right. \\ & \left. + Z\left(\mathcal{T}_m - \sum_{t=0}^{\mathcal{T}_m} x_{nm}^t\right) \right\}, \; q, \mu_m > 0, \end{aligned}$$

$$\mathbf{P0} : \min_{X,Y} E, \tag{11a}$$

$$\text{s.t. } E_n^t \le E_{\max}, \forall n \in N, \forall t \in T, \tag{11b}$$

$$\sum_{n \in N} t_n^t \le T_n, \forall t \in T, \tag{11c}$$

$$\sum_{n \in N} y_{nm}^t d_n \le C_m^t, \forall t \in T, \tag{11d}$$

$$\text{Queue } Q_m(t) \textit{ is mean rate stable}, \forall m \in M, \tag{11e}$$

$$x_{nm}^t \in \{0,1\}, \forall n \in N, \forall t \in T, \tag{11f}$$

$$y_{nm}^t \in [0,1], \forall m \in M, \forall t \in T. \tag{11g}$$

Where (11b), (11c) and (11d) are used to limit that the total cost of the vehicle in time slot t, the data transmission delay, the total number of data blocks cached from DTs to the RSU does not exceed the maximum cost, the maximum data transmission delay and the total number of data blocks respectively. (11e) is used to constrain the stability of the data buffer to achieve system network stability. (11f) is a binary variable constraint with a value within $\{0,1\}$ and (11g) is the value range constrain within $[0,1]$.

Algorithm 1 Online Joint-Decision Making

Input: input parameters Trade-off parameter V, deficit queue backlog $Q_m = 0$;

1: **for** each $t = 0, 1, ..., T-1$ **do**
2: Solve the optimization problem P1 to get optimal $(X_t, Y_t)^*$:
3: $P1 : \Delta(\Theta(t)) + V \sum_{m=1}^{M} \mathbb{E}\{\mathrm{E}_t \mid \Theta(t)\}$
4: Update the data queue for all RSUs:
5: $Q_m(t+1) = \max\{Q_m(t) - \phi_m^t, 0\} + I_m^t$
6: **end for**

Output: Digital Twin Caching decision Y, Vehicle transmission decision X, Min cost

B. Problem Transformation

Due to the change in network and communication condition of the system in each time slot, $P0$ is a stochastic optimization problem, which is challenging to solve using classical optimization algorithms. Therefore, in this section, we first use Lyapunov optimization to transform the above stochastic optimization problem into a real-time optimization problem.

Definition 2. According to the definition of stability, a queue is considered stable if all of the queues satisfy the following constraint: A network is stable if all individual queues in the network are stable. Let $\mathbb{E}$ represent the expected value. We formulate this premise as

$$\lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{\mathrm{m} \in M} \mathbb{E}\{Q_m(t)\} < \infty. \tag{12}$$

Therefore, we could optimize the cost function without damaging the stability of the control queue in the current slot.

We define a quadratic Lyapunov function, a scalar and non-negative description of all queues in the current time t.

$$L(\Theta(t)) = \frac{1}{2}\left(\sum_{m\in M} Q_m(t)^2\right). \tag{13}$$

Accordingly, the Lyapunov drift function represents the growth of the entire queue from time t to time $t+1$, which is a data buffer queue in our system. The Lyapunov drift function can be defined as

$$\begin{aligned}
\Delta(\boldsymbol{\Theta}(t)) &= L(\boldsymbol{\Theta}(t+1)) - L(\boldsymbol{\Theta}(t)) \\
&= \frac{1}{2}\sum_{m\in M} Q_m(t+1)^2 - \frac{1}{2}\sum_{m\in M} Q_m(t)^2 \\
&= \frac{1}{2}\sum_{m\in M}\left[\max\left(Q_m(t)-\phi_m^t, 0\right) + \sum_{n=1}^{N} y_{nm}^t d_n\right]^2 \\
&- \sum_{m\in M}\left(Q_m(t)\right)^2 \\
&\le \sum_{m\in M}\frac{\left[\phi_m^t + \sum_{n=1}^{N} y_{nm}^t d_n\right]}{2} \\
&+ \sum_{m\in M} Q_m(t)\left[\sum_{n=1}^{N} y_{nm}^t d_n - \phi_m^t\right].
\end{aligned} \tag{14}$$

Therefore, we only need to solve the minimum value of P1 in each time slot. The parameter V is used to adjust the weight between cost and the stability of the data buffer queue.

$$\mathbf{P}1: \Delta(\Theta(t)) + V\sum_{m=1}^{M}\mathbb{E}\left\{\mathrm{E}_t \mid \Theta(t)\right\} \tag{15}$$

$$\begin{aligned}
&\triangleq \mathbb{E}\{L(\Theta(t+1)) - L(\Theta(t)) \mid \Theta(t)\} + V\sum_{m=1}^{M}\mathbb{E}\left\{E_t \mid \Theta(t)\right\} \\
&\le B + \sum_{m=1}^{M} Q_m(t)\mathbb{E}\left\{\sum_{n\in N} y_{nm}^t d_n(t)\left|\Theta(t) - \phi_m(t)\right|\Theta(t)\right\} \\
&+ V\sum_{m=1}^{M}\mathbb{E}\left\{E_t \mid \Theta(t)\right\},
\end{aligned}$$

where $B = \frac{1}{2}\sum_{m\in M}\left[\phi_{m,\max}^2 + \left(\sum_{n\in N} y_{nm}^t d_n(t)\right)_{\max}^2\right]$ is a constant. $\phi_{m,\max}$ and $\left(\sum_{n\in N} y_{nm}^t d_n(t)\right)_{\max}$ are the bound of ϕ_m and $\sum_{n\in N} y_{nm}^t d_n(t)$, respectively.

C. Solution

1) Online Joint-Decision Making: We transform the original stochastic optimization problem into a real-time optimization problem. Hence, to minimize the system cost, we could only solve optimization problem P1. We propose an online multi-decision approach to make optimal transmission decisions, as shown in Algorithm 1.

2) Training Algorithm of Data Caching and Transmission Based on A3C: The solution of P0 obtained by substituting into P1 is still NP-hard. We observe that the variables sought are all discrete values, so we propose using the A3C algorithm of reinforcement learning [8] to solve P1. The algorithm uses an asynchronous update mechanism to solve the correlation among complicated states, making training more efficient.

IV. Simulation

A. Environment Setup

In this section, we carry out numerical simulations to illustrate the performance of our scheme for data synchronization from the DT to the vehicle. Our algorithm and network architecture are implemented using TensorFlow 2.0 and Keras. In the simulation setup, the length of each time slot is set as 0.2s. The distance between two RSUs is set as 100m, and the vehicle speeds are sampled from uniform distributions within $[5, 35]$ m/s. The transmission power of each vehicle is set as 1.25W, and the noise power is set as -114 dBm. In addition, the cache capacity of each RSU is set as 1GB.

To compare the performance of different schemes, we consider three different baseline strategies to compare with our strategy. The baseline strategies are introduced as follows:

- Random Put (RP): For each time slot, the DT chops the data that needs to be synchronized into data blocks and then randomly transmits them to RSUs.
- All Put (AP): In this scheme, each DT first caches as many data blocks as possible in the RSU. In this case, we only consider the capacity of the RSU without considering the receiving capability and communication conditions of the vehicle.
- Without Lyponuv (No-L): To evaluate the performance of the proposed stability of Lyponuv optimization, we set the No-L scheme without considering the stability of the buffer queue of the RSU.

B. Performance Evaluation

In this part, we analyze the convergence of our proposed algorithm and compare the performance of our proposal with other baselines. We set 10 workers to train the actor-network and critic network 700 times to train our neural network model. We calculate the average cost of all time slots and take the normalized reward. Fig. 5 shows the learning process of our algorithm. We can see from the figure that the normalized reward increases as the number of iterations until it converges to a stable number, which illustrates the convergence of our algorithm.

We implement all the baselines with different vehicle speeds and data sizes, respectively. The average cost comparison of different vehicle speeds is shown in Fig. 6. We run each algorithm 10 times at different speeds and get the averaged cost. The figure shows that RP, AP, and No-L always have a high cost at different speeds. This is because the three baselines cannot adaptively adjust their strategies in a highly dynamic environment. Especially, the cost of No-L has a large variance. The reason for this condition is that this scheme

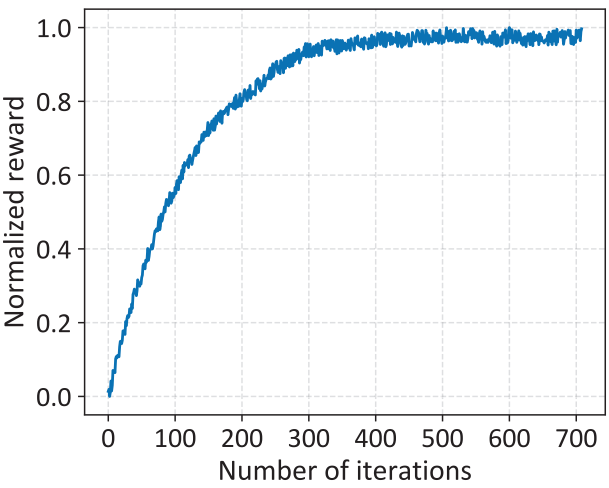

Fig. 5. Learning process of our algorithm.

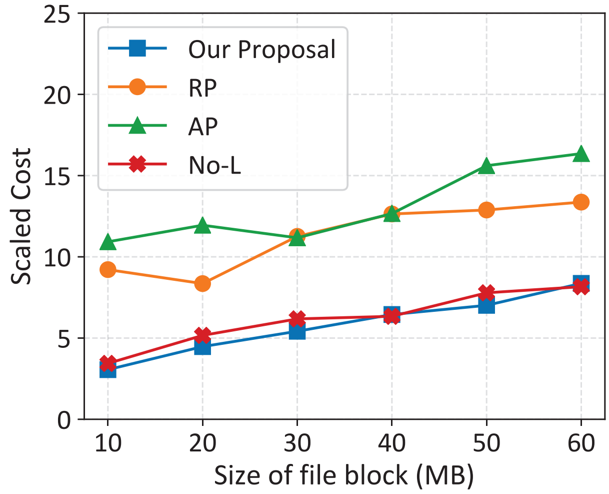

Fig. 7. Average cost comparison for different block size of data

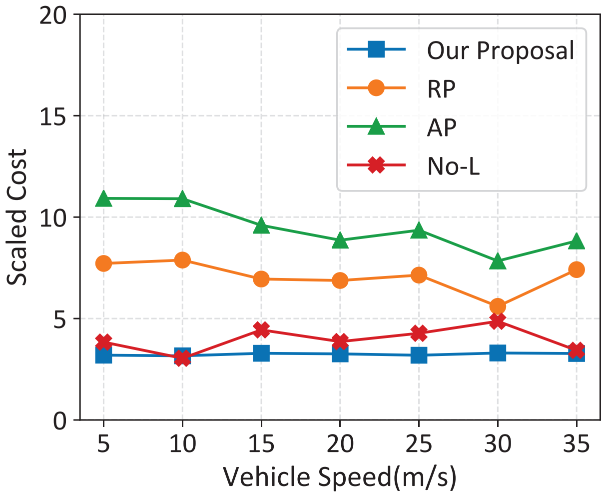

Fig. 6. Average cost comparison for different speed of vehicles.

did not use Lyapunov optimization in No-L, resulting in an unstable performance of the vehicle. It is obvious that our proposal is more efficient and stable than these baselines.

Fig. 7 illustrates the average cost comparison between our algorithm and different data block size baselines. It is observed that the cost of all approaches increases with the increase of the file block size, and our proposal has a lower cost than other baselines. The cost of No-L is similar to that of our method. It is noted that AP has the worst performance. The reason is that vehicles' wireless conditions and data queue are not considered when the DT caches data blocks in RSUs.

V. Conclusion

In this paper, we investigate the data synchronization problem in the DTV under the time-varying networks. Due to the variability of network channels and data transmission rates, we leverage the Lyponuv optimization to trade off network stability and the data transmission cost. Furthermore, we resort to A3C to find the optimal solution to the Lyapunov optimization problem. At the same time, we adopt the online joint decision-making method that the DT makes block cache decisions, and vehicles make transmission decisions to minimize the cost of data transmission while ensuring network stability. Simulation results show that our proposal achieves the lowest transmission cost under different conditions and has a more stable performance than other baselines. The joint game competition problem among RSUs will be considered in our future work.

References

[1] Tom H. Luan, Ruhan Liu, Longxiang Gao, Rui Li, Haibo Zhou. "The paradigm of digital twin communications[J]." in arXiv preprint arXiv:2105.07182, 2021.

[2] Uhlemann T H J, Lehmann C, Steinhilper R. "The digital twin: Realizing the cyber-physical production system for industry 4.0." Procedia Cirp, 2017.

[3] Zhaolong Ning, Kaiyuan Zhang, Xiaojie Wang, Lei Guo;Xiping Hu, Jun Huang, Bin Hu, Ricky Y. K. Kwok, "Intelligent Edge Computing in Internet of Vehicles: A Joint Computation Offloading and Caching Solution," in IEEE Transactions on Intelligent Transportation Systems, April 2021.

[4] Y. Huang, Y. Gao, K. Nahrstedt and W. He, "Optimizing File Retrieval in Delay-Tolerant Content Distribution Community," in 2009 29th IEEE International Conference on Distributed Computing Systems, 2009.

[5] M. H. Cheung, F. Hou, V. W. S. Wong and J. Huang, "DORA: Dynamic Optimal Random Access for Vehicle-to-Roadside Communications," in IEEE Journal on Selected Areas in Communications, May 2012.

[6] J. Kim, G. Caire and A. F. Molisch, "Quality-Aware Streaming and Scheduling for Device-to-Device Video Delivery," in IEEE/ACM Transactions on Networking, Aug. 2016.

[7] Z. Su, M. Dai, Q. Xu, R. Li and S. Fu, "Q-Learning-Based Spectrum Access for Content Delivery in Mobile Networks," in IEEE Transactions on Cognitive Communications and Networking, March 2020.

[8] Jinkai Zheng, Tom H. Luan, Longxiang Gao, Yao Zhang, Yuan Wu, "Learning Based Task Offloading in Digital Twin Empowered Internet of Vehicles." in arXiv preprint arXiv:2201.09076, 2021.