Deep Reinforcement Learning based Performance Optimization in Blockchain-Enabled Internet of Vehicle

Mengting Liu*, Yinglei Teng*, F. Richard Yu†, Victor C. M. Leung‡, and Mei Song*
*Beijing Key Laboratory of Space-ground Interconnection and Convergence,
Beijing University of Posts and Telecommunications, Beijing, 100876, China
†Department of Systems and Computer Engineering, Carleton University, Ottawa, ON K1S 5B6, Canada
‡Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC V6T 1Z4, Canada

***Abstract*—The rapid development of Internet of Vehicles (IoV) necessitates a secure and reliable infrastructure to store and share the massive data. *Blockchain*, a distributed and immutable ledger, is widely considered as a promising solution to ensure data security and privacy for IoV. To deal with the massive IoV data, the scalability of blockchain becomes a critical issue, which should maximize transactional throughput as well as handling the dynamics of IoV scenarios. Therefore, this paper proposes a novel deep reinforcement learning (DRL) based performance optimization framework for blockchain-enabled IoV, where transactional throughput is maximized while guaranteeing the decentralization, latency and security of the underlying blockchain system. In this framework, we first carry out the performance analysis for blockchain systems from the aspects of scalability, decentralization, latency and security. Further, DRL technique is adopted to select block producers and adjust block size and block interval to adapt to the dynamics of IoV scenarios. Simulation results show that our proposed framework can effectively improve the throughput of blockchain-enabled IoV systems without affecting other properties.**

I. Introduction

Recent advances of Internet of Things (IoT) are driving the evolution of traditional Vehicle Ad-hoc Networks, Vehicle Telematics, and other Connected Vehicle Networks into the Internet of Vehicle (IoV) [1]. With the rapid growth of IoV, more and more IoV nodes are being evolved, where massive data is captured to improve traffic safety and efficiency, e.g., road traffic data, sensory data, driving habit data, car-to-car experience sharing, e-commerce transaction data [2]. This has brought huge challenges for IoV data storing and sharing. Currently, most vehicle networks rely on centralized servers to store and share IoV data, which is faced with several pitfalls [3]: Low data security and privacy, poor interoperability and compatibility among different nodes, and high storage cost and transaction cost.

To solve the above problems, ***Blockchain*** is deemed as a promising solution. Blockchain, firstly used as a peer-to-peer (P2P) ledger for Bitcoin transactions [4], can guarantee data security and privacy as well as reduce the cost: **1)** Blockchain enables the IoV nodes to store the generated data by submitting transactions that would be collected in form of blocks and verified by the validators in the blockchain system. **2)** Each owner manages their own keys, and blockchain allows the owners to decide which data to share and with whom to share. **3)** Storage cost and transaction cost will be greatly reduced using blockchain and smart contract [5].

Recently, there have been a number of works focusing on blockchain-enabled IoV application scenarios. In [6], a distributed architecture is proposed for the outward transmission of vehicle data in IoV scenarios, where blockchain is used to facilitate distributed data storage and security management. With the help of blockchain technology, a decentralized trust management system is proposed for vehicular networks in [3]. To select the most appropriate electric vehicle charging station, the authors of [7] design a blockchain-based autonomous selection system for vehicle networks utilizing smart contract. Based on the lightning network and smart contract, [8] puts forward a blockchain-based security model to enhance the security of trading between electric vehicles and charging piles. Additionally, for vehicular edge networks, consortium blockchain and smart contract are used to achieve secure data storage and sharing in [9].

Nevertheless, few works address the scalability issue for blockchain-enabled IoV scenarios, which is a main bottleneck for the application of blockchain to high-throughput scenarios like IoV. Meanwhile, there is a well-known four-way trade-off in blockchain: a blockchain system can only at most have several of the following three properties: scalability, decentralization, security and latency [5]. Therefore, this paper aims at improving transactional throughput for blockchain-enabled IoV while guaranteeing the system's decentralization, security and latency. Additionally, to handle the dynamic and large dimension characteristics of IoV scenarios, a deep reinforcement learning (DRL) approach is adopted, which shows its superiority in dealing with dynamic and complicated problems [10]. The main contributions are as follows.

- To our best knowledge, we are the first to present a novel performance optimization framework for blockchain-enabled IoV, where the transactional throughput is improved without affecting the other three key properties:

978-1-5386-8088-9/19/$31.00 ©2019 IEEE

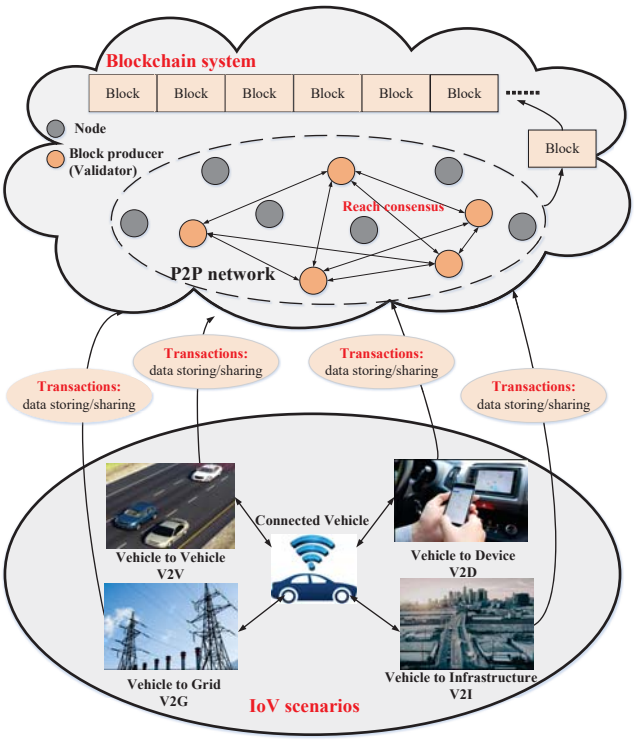

Fig. 1: Illustration of Blockchain-enabled IoV Scenarios.

decentralization, latency and security.

- In order to evaluate blockchain systems comprehensively, we provide a methodology to quantify the performance of blockchain systems from the aspects of scalability, decentralization, latency and security.
- To handle the dynamic features of IoV scenarios, we use DRL technique to select block producers, and adjust block size and block interval, in order to maximize the transactional throughput.
- Simulation results with different system parameters are presented to show the effectiveness of the proposed performance optimization framework for blockchain-enabled IoV, which provides some useful insights to guide the application of blockchain to future IoV.

The rest of this paper is organized as follows. Section II describes the system model. Performance analysis for blockchain systems is carried out in Section III. In Section IV, the proposed DRL-based performance optimization framework is presented. Section V shows and discusses the simulation results. Finally, Section VI concludes the paper.

II. System Model

As is shown in Fig. 1, we consider blockchain-enabled IoV scenarios, which include two parts i.e., IoV networks that generate transactions, and blockchain system that deals with the transactions in a trustless and secure manner. The models of these two parts are presented as follows.

A. IoV Networks

In IoV scenarios, a huge amount of data is generated from IoV nodes (e.g., vehicles, passengers, roadside facilities, infrastructure), which should be stored and shared in a secure way. Meanwhile, it's likely that the collected data needs to be shared among different IoV nodes to improve traffic safety and efficiency. Therefore, we consider two kinds of transactions (data storing and data sharing) continuously created by the IoV networks. Afterwards, these transactions are relayed to blockchain systems for storing/fetching the data into/from the distributed ledger, i.e., the underlying blockchain.

B. Blockchain System

To handle the transactions generated from IoV scenarios, block producers need to complete the following steps: i) Collect and package the transactions into a block. ii) Append the block to the blockchain by reaching a consensus on the newly generated block [11]. Therefore, there are two key factors, block producers (a.k.a. validators)[1] and consensus process, the models of which are given as follows.

1) Block Producers:

Assume that there are N nodes (i.e., block producer candidates) and K block producers in the blockchain system. The set of nodes is denoted as $\Phi_S = \{z_1, z_2, \ldots, z_N\}$, and the stake and computational resource of node $z_n, n = 1, \ldots, N$ are represented by Υ_n (in *token*) and c_n (in *GHz*), respectively. For clarity, we use $\boldsymbol{\Upsilon} = \{\Upsilon_1, \Upsilon_2, \ldots, \Upsilon_n\}$ and $\boldsymbol{c} = \{c_1, c_2, \ldots, c_n\}$ to denote the set of stakes and computational resources. Note that K block producers, represented by $\Phi_B = \{z_{b_1}, \ldots, z_{b_k}, \ldots, z_{b_K}\}, \Phi_B \subseteq \Phi_S$, are selected out of Φ_S according to certain rules (seen in Section IV-C). Besides, we assume that K block producers take turns to produce blocks with block size S^B (in *MB*) and block interval T^I (in *seconds*), which is in line with EOS [11].

2) Consensus Models:

In this paper, we consider practical Byzantine fault tolerant (PBFT) as the consensus algorithm, which is considered as a very robust protocol since an consensus can be achieved as long as more than a fraction (2/3) of replicas are honest [12]. The whole consensus process includes five phases: *Request, Pre-pare, Prepare, Commit and Reply*, as shown in Fig 2.

In the blockchain system, we consider the block producer that generates a new block as the client $z_{b_c}, c = 1, 2, \ldots, K$, and the other block producers/validators are regarded as replicas $z_{b_i}, i = 1, 2, \ldots, K, i \neq c$. In other words, the client issues a block with a number of transactions (i.e., data storing or data sharing request) and broadcasts it to other validators to reach a consensus. During the consensus process, there is one replica that is designated as the primary. In this paper, we consider a Byzantine failure model as in [13].

As is shown in Fig. 2, the consensus process mainly involves exchanging and verifying messages. For the message exchanging, we model the time-varying transmission links as finite-state Markov channels (FSMC). Let $R_{b_i,b_j}(t)$

[1]Block producers and validators are used interchangeably in this paper.

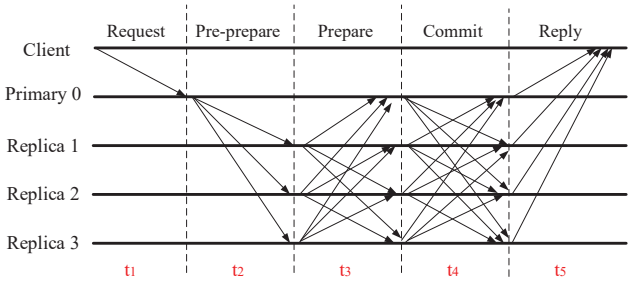

Fig. 2: Protocol communication pattern of PBFT [12].

denote the data transmission rate of the link connecting validator z_{b_i} and validator $z_{b_j}, i, j = 1, 2, \ldots, K, i, j \neq c$, which is partitioned and quantized into L levels, i.e., $\boldsymbol{r} = \{r_1, r_2, \ldots, r_L\}$. Then the $L \times L$ transition probability matrix w.r.t. $R_{b_i,b_j}(t)$ is defined as $\boldsymbol{p}(t) = [p_m(t)]_{L\times L}$, where $p_m(t) = \Pr\left[R_{b_i,b_j}(t+1) = y_2 \left| R_{b_i,b_j}(t) = y_1\right.\right]$ and $y_1, y_2 \in \boldsymbol{r}$. For message verification, we only consider the computing cost of the cryptographic operations as in [13], which include verifying signatures, generating message authentication codes (MACs), and verifying MACs, requiring α, β, and β CPU cycles, respectively.

III. Performance Analysis

This section carries out the performance analysis for blockchain systems from the aspects of scalability, decentralization, latency and security. For each property, we first clarify the concept and then provide a quantitative measure.

A. Scalability

Literally, blockchain refers to a *chain of blocks*, where each block contains a number of transactions. For blockchain systems, scalability is evaluated by transactional throughput, which is measured by the number of transactions per unit time that the system can process. The transactional throughput Ω (in transaction per second, i.e., TPS) directly depends on block size and block interval, which can be calculated by

$$\Omega\left(S^B, T^I\right) = \frac{\left\lfloor S^B/\chi \right\rfloor}{T^I}, \tag{1}$$

where S^B represents the block size, T^I is the block interval, and χ denotes the average size of transactions.

Observing (1), we can find an intuitive way to improve the throughput is to increase block size or to cut down the time interval between blocks. However, considering other properties, i.e., decentralization, latency and security, the adjustment of block size and block interval can't be conducted arbitrarily.

B. Decentralization

To measure the decentralization of blockchain systems, we utilize *Gini coefficient*, which was well studied as a measurement for the inequality of wealth or income [14]. In this paper, we focus on the decentralization of block producers w.r.t. stakes distribution[2], which is calculated by

$$G(\boldsymbol{\Upsilon}) = \frac{\sum\limits_{z_{b_i}\in\Phi_B}\sum\limits_{z_{b_j}\in\Phi_B}\left|\Upsilon_{b_i}-\Upsilon_{b_j}\right|}{2\sum\limits_{z_{b_i}\in\Phi_B}\sum\limits_{z_{b_j}\in\Phi_B}\Upsilon_{b_i}} = \frac{\sum\limits_{z_{b_i}\in\Phi_B}\sum\limits_{z_{b_j}\in\Phi_B}\left|\Upsilon_{b_i}-\Upsilon_{b_j}\right|}{2K\sum\limits_{z_{b_i}\in\Phi_B}\Upsilon_{b_i}}, \tag{2}$$

Note that the value of Gini coefficient is within $[0, 1]$, where 0 and 1 denote the highest decentralized and the highest centralized, respectively. In other words, the more uniform or decentralized the stakes distribution, the closer the coefficients are to 0. Conversely, the more centralized, the closer the coefficients are to 1. In order to guarantee the decentralization of block producers from the aspect of stakes distribution, the following constraint should be satisfied.

$$G(\boldsymbol{\Upsilon}) \leq \eta_s, \tag{3}$$

where $\eta_s \in [0, 1]$ is the threshold of decentralization.

C. Latency/Time to Finality (TTF)

We evaluate the latency of the blockchain system by TTF, i.e., time to finality, which measures how long it takes to receive a reasonable guarantee that the transaction written in blockchain is irreversible. Recall that the transaction processing includes two phases, i.e., generate a block and reach a consensus on the generated block among the validators. In this sense, the TTF for the transactions includes the block generation time (i.e., block interval) and the time for the block to be validated, which is denoted by

$$T^F = T^I + T^C, \tag{4}$$

where T^C is the consensus latency, i.e., the time cost for the validators to authenticate the generated block. For simplicity, we divide the whole validation process into two parts, i.e., messages delivering and messages verification (verifying signatures, generating and verifying MACs). Therefore, we can calculate the consensus latency T^C by

$$T^C = T^D + T^V, \tag{5}$$

where the time cost for message delivery T^D and validation T^V are calculated by [15]

$$\begin{aligned} T^D &= \tfrac{1}{M}\left(t_1 + t_2 + t_3 + t_4 + t_5\right) \\ &= \tfrac{1}{M}\left(\begin{array}{l} \min\left\{\frac{MS^B}{R_{b_c,b_p}}, \mathcal{T}\right\} + \min\left\{\max\limits_{i\neq c,p}\frac{MS^B}{R_{b_p,b_i}}, \mathcal{T}\right\} + \\ \min\left\{\max\limits_{i\neq j;i,j\neq c}\frac{MS^B}{R_{b_i,b_j}}, \mathcal{T}\right\} + \\ \min\left\{\max\limits_{i\neq j}\frac{MS^B}{R_{b_i,b_j}}, \mathcal{T}\right\} + \min\left\{\max\limits_{i\neq c}\frac{MS^B}{R_{b_i,b_c}}, \mathcal{T}\right\} \end{array}\right). \end{aligned} \tag{6}$$

and

$$T^V = \tfrac{1}{M}\max_{k=1,\ldots,K;k\neq c}\left\{\frac{\mathcal{O}_{b_k}}{c_{b_k}}\right\}. \tag{7}$$

where M denotes the batch size, and the computational load per batch for the primary z_{b_p} and the replica z_{b_i} are

[2]For simplicity, we only measure the decentralization of blockchain systems in terms of stakes distribution, and this method can be easily extended to measure other aspects.

$\mathcal{O}_{b_p} = M\alpha + [2M + 4(K + f - 1)]\beta$ and $\mathcal{O}_{b_i} = M\alpha + [M + 4(K + f - 1)]\beta$, respectively.

In order to meet the delay requirement of IoV scenarios, we assume that one block should be issued and validated within a number of consecutive block intervals, i.e., $\omega\,(\omega > 1)$ block intervals[3]. Specifically, the time cost for transaction finality should satisfy the following constraint.

$$T^F \leq \omega \times T^I. \tag{8}$$

D. Security

For the PBFT consensus algorithm, the security can be guaranteed under all network conditions as long as a fraction of participants are honest [12]. Therefore, the loyalty of the validators are very critical for PBFT. To guarantee the security of blockchain systems, the number of malicious validators f should be restricted by the following constraint.

$$f \leq F, \tag{9}$$

where $F = \lfloor \frac{K-1}{3} \rfloor$ represents for the maximum tolerable number of malicious validators.

IV. DRL-based Performance Optimization Framework for Blockchain-Enabled IoV

To handle the dynamic and large-dimension characteristics of IoV scenarios such as the transaction size and the features of nodes in the blockchain system (e.g., stake, available computing resources), we resort to DRL technique. To implement the DRL-based algorithm, we identify the state space, action space and reward function as follows.

A. State Space

We define the state space at decision epoch $t\,(t = 1, 2, \ldots)$ as a union of the average transaction size χ, stakes distribution $\boldsymbol{\Upsilon}$, computing capability of nodes $\boldsymbol{c} = \{c_k\}$, and the date transmission rate of the links between each pair of nodes $\boldsymbol{R} = \{R_{i,j}\}$, which is denoted as

$$\mathcal{S}^{(t)} = [\chi, \boldsymbol{\Upsilon}, \boldsymbol{c}, \boldsymbol{R}]^{(t)}. \tag{10}$$

B. Action Space

In order to maximize the transactional throughput, several parts of the blockchain system should be adjusted to adapt to the dynamic environment, which includes block producers $\boldsymbol{a}$, block size S^B and block interval T^I. Formally, the action space at decision epoch t is expressed by

$$\mathcal{A}^{(t)} = \left[\boldsymbol{a}, S^B, T^I\right]^{(t)}, \tag{11}$$

where the block producers indicator is $\boldsymbol{a} = \{a_n\}, a_n \in \{0, 1\}, z_n \in \Phi_S$ with $a_n = 1$ representing node z_n is chosen as a block producer while $a_n = 0$ otherwise. Besides, block size $S^B \in \left\{0.2, 0.4, \ldots, \dot{S}\right\}$ and block interval $T^I \in \left\{0.5, 1, \ldots, \dot{T}\right\}$ are restricted by block size limit $\dot{S}$ and maximum block interval $\dot{T}$.

[3]This paper assumes that the transactions should be finalized within a number of consecutive block time, which is in line with the concept of EOS [11]. More general case will be considered in future works.

C. Reward Function

The reward function is defined to maximize the transactional throughput while guaranteeing the decentralization, finality and security of the blockchain system, i.e., a decision should be made in each epoch to solve the following problem.

$$\begin{aligned} \mathcal{P}1: \quad & \max_{\mathcal{A}} \quad Q(\mathcal{S}, \mathcal{A}) \\ C1: \quad & G(\boldsymbol{\Upsilon}) \leq \eta_s, \\ C2: \quad & T^F \leq \omega \times T^I, \\ C3: \quad & f \leq F. \end{aligned} \tag{12}$$

where the decentralization of block producers w.r.t. stakes distribution is guaranteed by $C1$ (i.e., Eq. (3)), the finality and security of the underlying blockchain system are ensured by $C2$ (i.e., Eq. (6)) and $C3$ (i.e., Eq. (7)), respectively, and $Q(\mathcal{S}, \mathcal{A})$ is the action-value function calculated by $Q(\mathcal{S}, \mathcal{A}) = \mathbb{E}\left[\sum_{t=0}^{\infty} \mu^t \mathcal{R}^{(t)}\left(\mathcal{S}^{(t)}, \mathcal{A}^{(t)}\right) \middle| \mathcal{S}^{(0)} = \mathcal{S}, \mathcal{A}^{(0)} = \mathcal{A}\right]$ with the discount factor $\mu \in (0, 1]$ that reflects the tradeoff between the immediate and future rewards, and we define the immediate reward as

$$\mathcal{R}^{(t)}\left(\mathcal{S}^{(t)}, \mathcal{A}^{(t)}\right) = \begin{cases} \frac{\lfloor S^B/\chi \rfloor}{T^I}, & if\ C1 - C3\ are\ satisfied, \\ 0, & otherwise. \end{cases} \tag{13}$$

Note that if any constraint is not satisfied, it means that the blockchain system has a poor performance in decentralization, latency or security. Therefore, we set the reward to be 0 for this case to avoid this invalid situation. Finally, we present the proposed DRL-based framework in **Algorithm 1**.

Algorithm 1: DRL-based Performance Optimization Framework for Blockchain-enabled IoV Scenarios

for *each decision epoch* t **do**
- / * ** Performance Optimization * ** /
- 1) A random action is selected with probability ε, otherwise $\mathcal{A}^{(t)} = \arg\max_{\mathcal{A}} Q\left(\mathcal{S}^{(t)}, \mathcal{A}^{(t)}\right)$ where $Q(\bullet)$ is estimated by the main Q network;
- 2) Execute $\mathcal{A}^{(t)}$ to select block producers, and adjust block size and block interval;
- / * * * * * Updating * * * * * /
- 1) Observe the reward $\mathcal{R}^{(t)}$ and the next state $\mathcal{S}^{(t+1)}$;
- 2) Store the experience $\left(\mathcal{S}^{(t)}, \mathcal{A}^{(t)}, \mathcal{R}^{(t)}, \mathcal{S}^{(t+1)}\right)$ into the experience memory $\mathcal{D}$;
- 3) Randomly sample a mini-batch of state transitions $\left(\mathcal{S}^{(i)}, \mathcal{A}^{(i)}, \mathcal{R}^{(i)}, \mathcal{S}^{(i+1)}\right)$ from experience memory $\mathcal{D}$;
- 4) Calculate the target Q-value from the target Q network by $y^{(i)} = \mathcal{R}^{(i)} + \gamma \max_{\mathcal{A}'} Q\left(\mathcal{S}^{(i+1)}, \mathcal{A}'\right)$.
- 5) Update the target Q network with loss function $L(\theta) = \left[y^{(i)} - Q\left(\mathcal{S}^{(i)}, \mathcal{A}'; \theta\right)\right]^2$ every G steps.

end

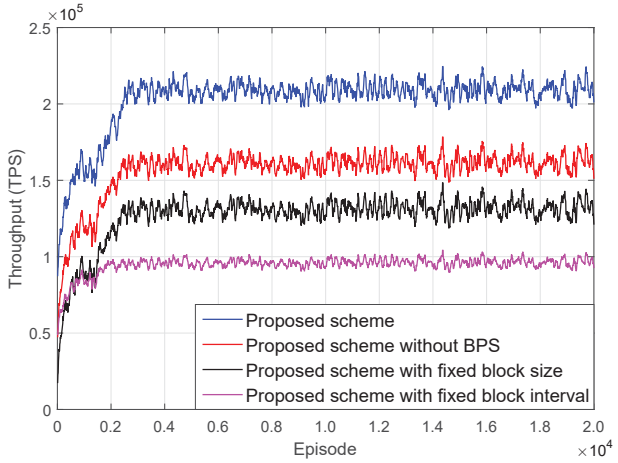

Fig. 3: Convergence performance of different schemes.

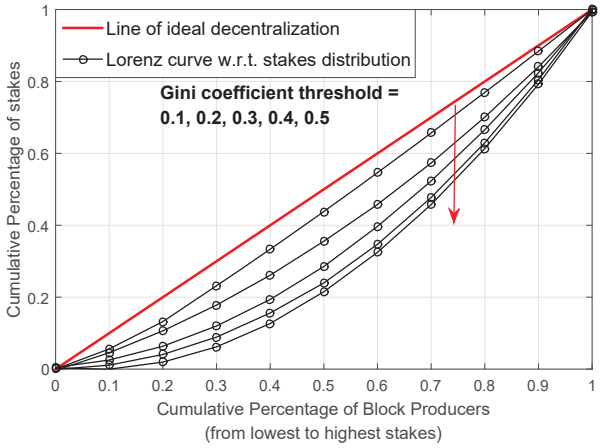

Fig. 4: Decentralization performance of block producers w.r.t. stakes distribution.

V. Simulation Results and Discussions

In the simulation, we consider a blockchain-enabled IoV scenario with 100 IoV nodes and 21 block producers. The DNN included in the proposed DRL-based framework was implemented using PyTorch, which is a fast and flexible deep learning framework. For the software environment, we utilized PyTorch 0.4.0 with Python 3.6 in Window 10 system.

For performance comparison, four baseline schemes are considered: 1) *Proposed scheme without block producers selection (BPS)*: block producers are randomly determined. 2) *Proposed scheme with fixed block size*: the blocks generated by block producers in different intervals are with the same size ($4MB$). 3) *Proposed scheme with fixed block interval*: the frequency of issuing blocks is fixed (every 1 second). 4) *Existing static scheme*: the decisions are determined by optimizing the immediate reward in a greedy way without using DRL tecnique.

A. Performance of Convergence

The convergence performance of our proposed DRL-based performance optimization scheme is presented in Fig. 3. It demonstrates a good convergence performance of our proposed scheme: the transactional throughput is very low at the beginning of the learning process, whereas it increases and reaches a stable state after around 4000 episodes. Besides, we can find that the proposed scheme can receive higher throughput than that of the other three DRL-based baselines. The reasons behind are as follows. 1) In our proposed DRL-based framework, block producers can be selected, and block size and block interval can be adjusted to maximize the throughput in an adaptive way. 2) The proposed scheme without BPS may fail to reach a consensus in the case when more than 1/3 of the replicas (validators) are malicious. 3) For the proposed scheme without block size or block interval, it fails to reach a higher throughput since the block size or block interval can't be adjusted to adapt to the dynamic environment.

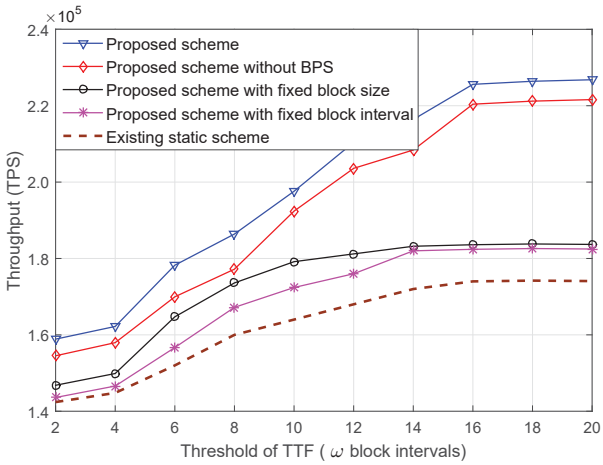

Fig. 5: Throughput vs. threshold of TTF.

B. Decentralization Performance

Fig. 4 describes the decentralization performance of the underlying blockchain system. Different from [16], where the decentralization performance is measured by the number of block producers, we use a more general metric, *Gini coefficient*, to capture the decentralization of the blockchain system w.r.t. stakes distribution. It can be found that the Lorenz curve gradually approaches to the ideal decentralized one (the red line) with the decrease of threshold, i.e., the blockchain system becomes more decentralized. It reveals that Gini coefficient can be considered as an effective metric to measure the decentralization of blockchain system in a quantitative way.

C. Performance Comparison with the Baselines

The effect of the threshold of TTF on the system's tractional throughput is depicted in Fig. 5. We can see that all the schemes gain a higher transactional throughput with the

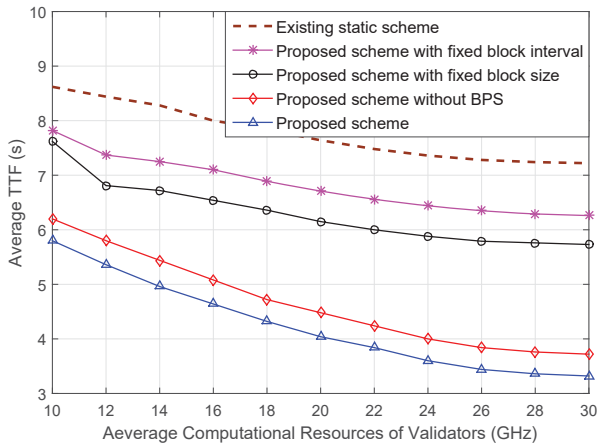

Fig. 6: Latency vs. average computational resources of validators.

increase of the threshold of TTF. This is because the validators can handle more transactions in one block with a more relaxed latency constraint. Besides, it can be observed that our proposed scheme can achieve higher throughput consistently than the baselines since there are some drawbacks of the baselines as illustrated before. Meanwhile, it's noted that the fixed block size scheme acts better than the fixed block interval scheme in the case of strict TTF constraints. A reasonable explanation is that the fixed block size scheme can adjust the block interval to deal with the low TTF threshold situation. Moreover, it's not surprising to find that the existing static scheme shows the poorest performance, which reveals the superiority of DRL-based solutions.

Fig. 6 examines the latency performance of different schemes with different average computational resources of validators. One observation is that the average TTF of the blockchain system for all the schemes decreases with the increase of average computational resources. The reason behind is obvious that a consensus can be reached more quickly among more computing capable validators. Another observation is that it takes the least time (lowest average TTF) for the proposed scheme to confirm transactions when compared with the baselines, then follows the proposed scheme without BPS, the fixed block size scheme and the fixed block interval scheme, and the longest latency is introduced by the existing static scheme.

VI. Conclusion

This paper presented a novel DRL-based performance optimization framework for blockchain-enabled IoV scenarios, where the scalability of the underlying blockchain was improved while guaranteeing other properties including decentralization, latency and security. In our proposed framework, we first provided a quantitative measurement for the performance of blockchain systems. Then using DRL technique, the transactional throughput of the blockchain system was maximized by selecting block producers and adjusting block size and block interval. Simulation results demonstrated that the proposed framework can achieve higher throughput than the baselines with various system parameters. Future work is in progress to consider an adaptive consensus algorithm for blockchain-enabled IoV scenarios.

Acknowledgment

This work was supported in part by the National Key R&D Program of China (No. 2018YFB1201500), in part by the National Natural Science Foundation of China under Grant No. 61771072, in part by the Beijing Natural Science Foundation under Grant No. L171011, in part by the Beijing Major Science and Technology Special Projects under Grant No. Z181100003118012, and in part by the scholarship from China Scholarship Council under Grant No. 201706470059.

References

[1] F. Yang, S. Wang, J. Li, Z. Liu, and Q. Sun, "An overview of internet of vehicles," *China Communications*, vol. 11, no. 10, pp. 1–15, Oct. 2014.

[2] O. Kaiwartya and *et al.*, "Internet of vehicles: Motivation, layered architecture, network model, challenges, and future aspects," *IEEE Access*, vol. 4, pp. 5356–5373, Sept. 2016.

[3] Z. Yang, K. Yang, L. Lei, K. Zheng, and V. C. M. Leung, "Blockchain-based decentralized trust management in vehicular networks," *IEEE Internet of Things Journal*, pp. 1–1, May 2018.

[4] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2009. http://www.bitcoin.org/bitcoin.pdf; accessed on 13 Oct. 2018.

[5] F. R. Yu, "vdlt: A service-oriented blockchain system with virtualization and decoupled management/control and execution v0.1," *[online] arXiv:1809.00290 [cs.NI]*, Sept. 2018.

[6] T. Jiang, H. Fang, and H. Wang, "Blockchain-based internet of vehicles: Distributed network architecture and performance analysis," *IEEE Internet of Things Journal*, pp. 1–1, Oct. 2018.

[7] M. Pustiek, A. Kos, and U. Sedlar, "Blockchain based autonomous selection of electric vehicle charging station," in *Proc. Int. Conf. on Identification, Information and Knowledge in the Internet of Things (IIKI)*. Beijing, Oct. 2016, pp. 217–222.

[8] X. Huang, C. Xu, P. Wang, and H. Liu, "Lnsc: A security model for electric vehicle and charging pile management based on blockchain ecosystem," *IEEE Access*, vol. 6, pp. 13 565–13 574, Mar. 2018.

[9] J. Kang and *et al.*, "Blockchain for secure and efficient data sharing in vehicular edge computing and networks," *IEEE Internet of Things Journal*, pp. 1–1, Oct. 2018.

[10] Y. He, Z. Zhang, F. R. Yu, N. Zhao, H. Yin, V. C. M. Leung, and Y. Zhang, "Deep-reinforcement-learning-based optimization for cache-enabled opportunistic interference alignment wireless networks," *IEEE Trans. Veh. Tech.*, vol. 66, no. 11, pp. 10 433–10 445, Nov. 2017.

[11] I. Grigg, "Eos - an introduction," July 2017. https://eos.io/documents/EOS-An-Introduction.pdf; accessed on 13 Oct. 2018.

[12] M. Castro and B. Liskov, "Practical byzantine fault tolerance and proactive recovery," *ACM Trans. Comput. Syst.*, vol. 20, no. 4, pp. 398–461, 2002.

[13] A. Clement, E. Wong, L. Alvisi, and M. Dahlin, "Making byzantine fault tolerant systems tolerate byzantine faults," in *Proc. 6th USENIX Symposium on Networked Systems Design and Implementation*, 2009, pp. 153–168.

[14] C. Gini, "Variability and mutability," *Journal of The Royal Statistical Society*, vol. 76, pp. 619–622, May 1913.

[15] M. Liu, F. R. Yu, Y. Teng, V. C. M. Leung, and M. Song, "Performance optimization for blockchain-enabled industrial internet of things (iiot) systems: A deep reinforcement learning approach," *IEEE Trans. on Industrial Informatics*, published online, Feb. 2019.

[16] Z. Xiong, Y. Zhang, D. Niyato, P. Wang, and Z. Han, "Edge computing resource management and pricing for mobile blockchain," *[online] arXiv:1710.01567v1 [cs.CR]*, Oct. 2017.