This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2019.2897805, IEEE Transactions on Industrial Informatics

Performance Optimization for Blockchain-Enabled Industrial Internet of Things (IIoT) Systems: A Deep Reinforcement Learning Approach

Mengting Liu, F. Richard Yu, *Fellow, IEEE,* Yinglei Teng, *Member, IEEE,*
Victor C. M. Leung, *Fellow, IEEE,* and Mei Song

***Abstract*—Recent advances in industrial Internet of things (IIoT) provide plenty of opportunities for various industries. To address the security and efficiency issues of the massive IIoT data, blockchain is widely considered as a promising solution to enable data storing/processing/sharing in a secure and efficient way. To meet the high throughput requirement, this paper proposes a novel deep reinforcement learning (DRL) based performance optimization framework for blockchain-enabled IIoT systems, the goals of which are three-fold: 1) providing a methodology for evaluating the system from the aspects of scalability, decentralization, latency and security; 2) improving the scalability of the underlying blockchain without affecting the system's decentralization, latency and security; 3) designing a modulable blockchain for IIoT systems, where the block producers, consensus algorithm, block size and block interval can be selected/adjusted using the DRL technique. Simulations results show that our proposed framework can effectively improve the performance of blockchain-enabled IIoT systems and well adapt to the dynamics of IIoT.**

***Index Terms*—Blockchain, industrial internet of things (IIoT), performance optimization, deep reinforcement learning**

I. Introduction

As an emerging technology, the industrial Internet of things (IIoT), a.k.a. industrial Internet, has received great attention in various industrial sectors such as manufacturing, logistics, retailing, environmental monitoring, security surveillance, energy/utilities, aviation, healthcare, etc. [1]–[5]. With the advances in wireless communication and sensor network technologies, more and more smart objects are being involved in IIoT, where massive raw data is captured and processed to support decision making [6], [7]. Nowadays, most IIoT applications rely on centralized servers for data storing/processing and intermediaries for data transmission, which exposes the data to security risks and also introduces high operational cost and delay [8]–[11]. Therefore, data security and efficiency become critical concerns for IIoT [12].

To address the above issues, ***Blockchain*** is widely considered as a promising solution, which can build a secure and efficient environment for data storing/processing/sharing in IIoT [13], [14]. Blockchain, firstly used as a peer-to-peer (P2P) ledger for Bitcoin economic transactions [15], can guarantee data security and efficiency by enabling anonymous and trustful transactions and removing all kinds of intermediaries. Although blockchain technology brings significant benefits, it is challenging for traditional blockchain systems to provide the scalability necessary to meet the high transactional throughput requirement of IIoT.

In fact, scalability has become a key issue that prevents blockchain from being used as a generic platform for different services and applications [16]. Bitcoin, the first-known blockchain-based cryptocurrency, can only confirm an average of about 3-4 transactions per second (TPS) [17] while Ethereum improves the throughput to about 14 TPS [18], which is still not enough to deal with high frequency transactions scenarios such as IIoT. Recently, a number of teams are working on realizing general, scalable and deployable blockchain platforms, which can be divided into two categories. One is through *on-chain scaling* solutions, such as adjusting the block size and interval (e.g., BitcoinCash [19]), shifting the process of issuing blocks (e.g., Bitcoin-NG [20]), introducing new consensus mechanisms like Proof of Stake (PoS), Delegated Proof of Stake (DPoS) and Practical Byzantine Fault Tolerance (PBFT) (e.g., Cardano [21], EOS [22]), using Sharding technique (e.g., Zilliqa [23]), etc. The other is through *off-chain scaling* solutions that aim at reducing the redundancy on the main blockchain using Sidechains (e.g., Plasma [24]), Multi-chains (e.g., Cosmos [25], AION [26]), Lightning network [27], Payment channels (e.g., Raiden network [28], TeeChan [29]), etc.

Nevertheless, these emerging blockchain platforms are still facing significant challenges when applied in IIoT. For example, most emerging blockchain platforms only focus on increasing transactional throughput by sacrificing other important performance measures, such as decentralization, security or latency[1]. There is a well-known *Trilemma* in blockchain: a blockchain system can only at most have two

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.*(Corresponding author: Yinglei Teng)*

M. Liu, Y. Teng and M. Song are with Beijing Key Laboratory of Space-ground Interconnection and Convergence, Beijing University of Posts and Telecommunications, Beijing, 100876, China.

F. R. Yu is with the Department of Systems and Computer Engineering, Carleton University, Ottawa, ON K1S 5B6, Canada.

V. C. M. Leung is with the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC V6T 1Z4, Canada.

[1]In this paper, latency refers to "time to finality" that measures the time used for a transaction to be finalized/confirmed (specified in Section II-B).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2019.2897805, IEEE Transactions on Industrial Informatics

of the following three properties: decentralization, scalability and security. Therefore, these properties should be carefully considered in designing blockchain-enabled IIoT systems.

To address the above challenges, this paper proposes a performance optimization framework for blockchain-enabled IIoT systems to improve the performance of data security and efficiency, where the scalability/throughput can be optimized while taking into account of system decentralization, security and latency. In addition, to handle the dynamic and large-dimension characteristics of IIoT systems, a deep reinforcement learning (DRL) approach is adopted, which shows its superiority in dealing with dynamic and complicated problems [30], [31]. The main contributions of this paper are listed as follows.

- To our best knowledge, we are the first to present a performance optimization framework for blockchain-enabled IIoT systems to improve the performance of data security and efficiency, where the four-way trade-off, i.e., *scalability, decentralization, security and latency*, is considered.
- In order to facilitate performance optimization, we quantify the performance of blockchain systems from the aspects of scalability, decentralization, latency and security.
- To handle the dynamic and large-dimension characteristics of IIoT systems, we design a DRL-based algorithm to dynamically select/adjust the block producers, consensus algorithm, block size, and block interval to improve the performance.
- Simulation results show that the proposed performance optimization framework can well address the scalability issue for blockchain-enabled IIoT systems while guaranteeing other performance, and can facilitate a wide range of applications in blockchain-enabled IIoT systems.

The rest of this paper is organized as follows. We present the related works in Section II. Section III describes the system model. The performance analysis for blockchain systems is carried out in Section IV. In Section V, the proposed DRL-based algorithm is presented. Section VI discusses the simulation results. Finally, Section VII concludes the paper.

II. Related Works

In this section, we present some related works concerning blockchain-enabled IIoT systems and the four-way trade-off of blockchain systems to provide some necessary backgrounds.

A. Blockchain-enabled Industrial Internet of Things (IIoT)

Recent years have witnessed the rapid development of IIoT, which is reshaping various industrials such as agriculture, environmental monitoring, and security surveillance [1]. Meanwhile, blockchain technology can enable the storing, processing and sharing of the data captured from the IIoT devices using a distributed, decentralized and shared ledger [16], thus can overcome some obstacles of IIoT like security risks, high operational cost and frequency delay. In this sense, the convergence of IIoT and blockchain can facilitate the realization of IIoT, which has also attracted great attention from industry to academic. [8] describes how the integration of IIoT and blockchain will improve the security and efficiency of various industrial sectors such as supply chain, autonomous vehicle and manufacturing plant equipment. Based on blockchain technology, the authors of [9] present a trusted and resilient communication architecture for IIoT applications, which can achieve data assurance, resilience and accountability. To ensure the security and privacy of trading data and consumption in the smart grid energy trading scenario, blockchain is used together with several other technologies including multi-signatures, and anonymous encrypted messaging streams in [10]. Besides, a blockchain-based platform architecture is proposed for IIoT in [11], to provide trust between the participants and control over the distribution of resources. All of the above works have pointed out that the scalability is a major concern of blockchain-enabled IIoT systems. However, the performance of blockchain systems (scalability, decentralization, security or latency) hasn't been well investigated in these works.

B. The Four-way Trade-off of Blockchain Systems

Similar to the CAP (Consistency, Availability, tolerance to network Partitions) theorem of robust distributed systems [32]: "A robust distributed system can only simultaneously provide two out of the three properties", the challenge of scaling the blockchain systems can be considered as a four-way trade-off, which involves the following four properties:

Scalability: The scalability issue refers to the ability for the blockchain systems to process transactions. To make the blockchain technology more pervasive, the system should be able to handle the increasing volume of transactions in a wide range of applications [33].

Decentralization: Decentralization reflects the fragmentation of control over the whole system, which allows the system to achieve other goals: censorship resistance, open participation, immunity from certain attacks, and elimination of single points of failure [34].

Security: The blockchain systems should guarantee the immutability of the ledger, which is reflected by its general robustness to attacks such as 51% attack, Sybil attack, distributed deny of service (DDoS), etc. [35].

Latency: The latency of the blockchain is measured by time to finality (TTF) that is defined as the time for transactions to be finalized/confirmed and become irreversible [36].

Most existing blockchain systems can only achieve part of these four properties but sacrificing others. We give some examples that are in line with this observation as follows. Permissionless PoW systems (e.g., Bitcoin, Ethereum 1.0) can achieve good decentralization and security while suffer poor scalability and finality. Centralized block production systems (e.g., Cardano, EOS) attempt to achieve scalability by sacrificing the decentralization of block producers. Meanwhile, multi-chains systems (e.g., Cosmos, AION) gain scalability, decentralization and fast TTF at the expense of undertaking additional attack risks. Therefore, this motivates us to design a performance optimization framework for blockchain-enabled IIoT systems to address the four-way trade-off issue.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2019.2897805, IEEE Transactions on Industrial Informatics

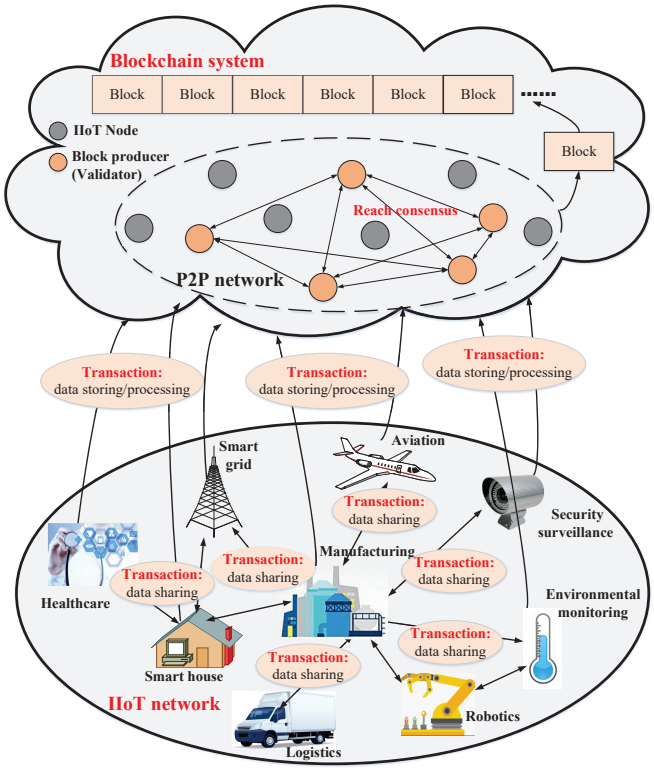


Fig. 1: Illustration of blockchain-enabled IIoT systems.

C. Performance Analysis of Blockchain Systems

Some attempts have been made to measure the performance of blockchain systems. In [37], performance analysis of two private blockchains, i.e., Ethereum and Hyperledger Fabric, is carried out in terms of average execution time, latency and throughput. However, the performance analysis is obtained through simulation while explicit quantitative measurement is absent. [38] presents a comparative assessment of decentralization for Bitcoin and Ethereum by several metrics including provisioned bandwidth, network structure, distribution of mining power, mining power utilization and fairness, where explicit quantification of these factors is still unaddressed and this assessment method only works for PoW systems. Besides, [35] and [39] provide a quantitative measurement for PoW blockchains, where the decentralization, security and latency are evaluated by the number of block producers, orphaning/fork probability and propagation time, respectively. Although providing some insights for performance analysis of blockchain systems, these works lack of comprehensiveness and generality. Therefore, this motivates us to design a comprehensive and general evaluation methodology to facilitate the performance optimization of blockchain systems.

III. System Model

As is shown in Fig. 1, we consider a blockchain-enabled IIoT system, which consists of two parts, i.e., the IIoT network that generates transactions of data storing/processing and data sharing, 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. IIoT Network

In IIoT networks, smart devices (e.g., industrial equipment, vehicles, drones, surveillance, etc.) are capable of using sensors to collect ambient data or using embedded cameras to capture the images or videos, which should be captured and stored/processed in a secure manner. Meanwhile, it's likely that the collected data needs to be shared between different industrial sectors for efficient decision making. Therefore, we consider two kinds of transactions[2] (data storing/processing and data sharing) continuously created by the smart devices. 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 the IIoT network, the block producers need to complete the following steps: i) *Generate a block*: collect, validate and package the transactions into a block. ii) *Append the block to the blockchain*: broadcast the generated block to other block producers, and add the block to their local blockchain after a consensus is reached on the new block [22]. Therefore, there are two key factors, the block producers (a.k.a. validators) , the models of which are given as follows.

1) Block Producers:

In this paper, we assume that there are N IIoT nodes 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 these 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 (specified in Section V-C). Assume the block producers are gathered at independent random positions in $\mathbb{R}^2$ according to an *inhomogeneous Poisson point process (PPP)* with density $\lambda(\boldsymbol{x})$ [40], where the location of node z_n is represented by the two-dimension coordinate $x_n \in \mathbb{R}^2$ and $\boldsymbol{x} = \{x_n\}$ is the location set. The density $\lambda(\boldsymbol{x})$ is defined such that $\mathbb{E}\{Num(A)\} = \iint_A \lambda(\boldsymbol{x})dx$ for any $A \subseteq \mathbb{R}^2$, where $Num(A)$ is the number of nodes in area A. In the blockchain-enabled IIoT system, we assume that these K block producers take turns to produce blocks with block size S^B (in *MB*) and block interval T^I (in *seconds*).

2) Consensus Models:

It's noted that there's no one-size-fits-all protocol considering varying system conditions [41]. This motivates us to design an *adaptive consensus algorithm* where a set of consensus protocols (i.e., PBFT [42], Zyzzyva [43], Quorum [44]) are considered as candidates. The most suitable consensus mechanism is dynamically selected according to the requirements

[2]A transaction is defined as an instance of changing ownership of tokens through digital signing portion of data and broadcasting it to the network [10].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2019.2897805, IEEE Transactions on Industrial Informatics

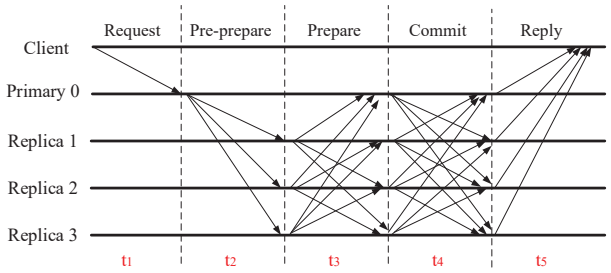


Fig. 2: Protocol communication pattern of PBFT.

of applications and the system conditions. In the following, we first give a brief introduction of these three consensus protocols, and then describe the assumptions of the consensus model adopted in this paper.

- A Brief Introduction of Consensus Protocols

(i) PBFT

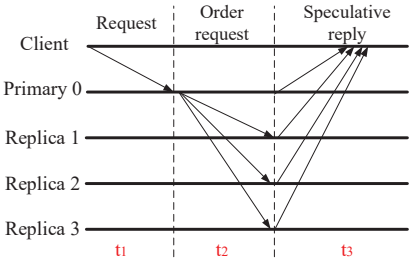
PBFT is a well-known and widely-used consensus protocol, which is adopted by EOS, Hyperledger, etc. It's considered as a very robust protocol since an consensus can be achieved as long as more than a fraction (2/3) of replicas are honest [42]. Although enjoying strong robustness, PBFT suffers significant performance drawbacks. This is mainly because the whole consensus process includes five phases: *Request, Pre-pare, Prepare, Commit and Reply*, as shown in Fig 2. The message exchanging among the replicas is time-consuming, which causes a long consensus latency. Besides, each replica has to validate the messages from most of the other replicas, which sets high demand for computing resources, such that one replica can't handle to many requests at one time.

(ii) Zyzzyva

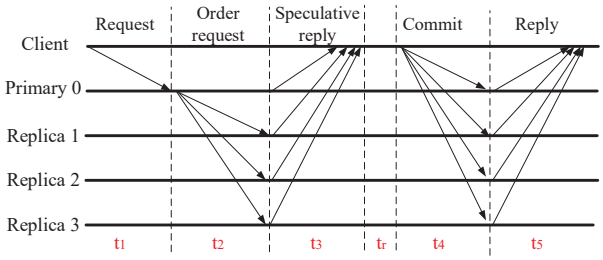
Zyzzyva is a speculative BFT protocol where the client sends a request to a replica (a primary) and then this primary replica forwards it to other replicas [43]. These replicas execute the request speculatively and send their replies back to the client. Same with BPFT, the consensus can be reached if there are no more than 1/3 faulty replicas, but the difference is that there is no message exchanging between replicas for Zyzzyva. The performance of this protocol varies for different cases, as is shown in Fig. 3. In fast case, i.e., all the replicas are honest, the replies that client received from all the replicas match with each other. In this case, the consensus can be reached in a very short time. In another case, if there's any faulty replica, complex recovery phases will be launched and a recovery time t_r is required, thus the performance may be significantly deteriorated.

(iii) Quorum

Quorum is widely considered as the "fastest" BFT protocol, where only one-phase message pattern is involved, as is shown in Fig. 4: a client broadcasts its request to all the replicas, and the replicas send their replies back [44]. The simple consensus process brings a short consensus latency as well as a high throughput. However, it's extremely vulnerable to attacks since the consensus can't be reached once any replica is malicious. Besides, both PBFT and Zyzzyva can process a number of



(a) *One-phase (Fast case)*



(b) *Two-phase*

Fig. 3: Protocol communication pattern of Zyzzyva.

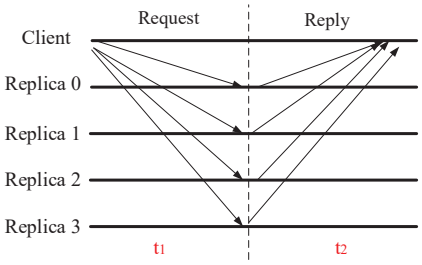


Fig. 4: Protocol communication pattern of Quorum.

requests at one time use batching while Quorum is not able to support batching.

- Consensus Model Assumptions

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 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 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 where several assumptions are made: 1) The client and the primary is correct and there are f faulty replicas. 2) The faulty replicas can behave arbitrarily, such as sending incorrect messages, sending the correct messages too slowly, or sending messages too quickly to introduce overhead. 3) The consensus system is an asynchronous network where synchronous intervals occur infinitely. During a synchronous interval, any message sent is delivered within a timeout $\mathcal{T}$. 4) The correct replicas schedule messages from other replicas in a round-robin fashion [45].

Note that the consensus process mainly involves exchanging

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2019.2897805, IEEE Transactions on Industrial Informatics

and verifying messages [41]. For the message exchanging, we model the time-varying transmission links as finite-state Markov channels (FSMC). Let $R_{b_i,b_j}(t)$ 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 [45], which includes verifying signatures, generating message authentication codes (MACs), and verifying MACs, requiring α, β, and β CPU cycles, respectively.

IV. Performance Analysis

In this section, we first use *transactional throughput* to measure system's scalability. Then the other properties, i.e., decentralization, latency and security, are served as the scalability constraints to address the four-way trade-off issue.

A. Scalability

Literally, blockchain refers to a *chain of blocks*, where each block contains a number of transactions. In blockchain systems, scalability can be evaluated by transactional throughput, which directly depends on two performance-related parameters: block size and block interval. one is *block size*, i.e., the number of bytes can be contained in each block, which determines how many transactions can be included in one block. The other is *block interval*, i.e., the average time required for the block producer to produce a new block, which captures the block release rate. Considering these two factors, the transactional throughput 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 (the number of bytes can be contained in each block), T^I is the block interval (the average time required for the block producer to produce a new block), and χ denotes the average size of transactions.

Observing (1), we can find an intuitive way to improve the on-chain transactional throughput is to increase the block size or to cut down the time interval between blocks. However, since the generated blocks have to be verified among the validators based on a consensus mechanism, the selection of validators and consensus algorithm is also of great importance. As revealed by the four-way trade-off, the scalability of blockchain systems is affected by the other three factors, i.e., decentralization, latency and security, which also means that the adjustment should not be conducted arbitrarily for the block size, block interval, selection of block producers, and consensus algorithm. In the following, we quantify these factors in blockchain systems.

B. Scalability Constraints

1) 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 [46]. Due to its advantages in evaluating "inequality", Gini coefficient has been widely used in many fields, such as capturing "system fairness" [47], "contrast intensity" [48], "resource difference degree" [49], etc.

In this paper, we focus on the decentralization of block producers and consider two typical factors (i.e., stakes distribution and geographical locations)[3]. To describe the decentralization w.r.t. stakes distribution, the Gini coefficient of the block producers' stakes can be calculated by (2). The details of Gini coefficient are given in **Appendix A**.

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

For the decentralization w.r.t. geographical locations, we recall that the assumption that the block producers are distributed according to an inhomogeneous PPP with density $\lambda(\boldsymbol{x})$ at $\boldsymbol{x}\in\mathbb{R}^2$. Since the density distribution $\lambda(\boldsymbol{x})$ is a continuous function of $\boldsymbol{x}$, the Gini coefficient can be calculated in terms of integration as follows [48].

$$G(\boldsymbol{\lambda})=\frac{\int_\Xi\int_\Xi|\lambda(\boldsymbol{x})-\lambda(\boldsymbol{y})|d\boldsymbol{y}d\boldsymbol{x}}{2\int_\Xi\int_\Xi\lambda(\boldsymbol{x})d\boldsymbol{y}d\boldsymbol{x}}=\frac{\int_\Xi\int_\Xi|\lambda(\boldsymbol{x})-\lambda(\boldsymbol{y})|d\boldsymbol{y}d\boldsymbol{x}}{2K},\tag{3}$$

where the density set is $\boldsymbol{\lambda}=\{\lambda(\boldsymbol{x})\},\boldsymbol{x}\in\Xi$, and the block producers are scattered in region $\Xi\subseteq\mathbb{R}^2$.

Note that the values of Gini coefficient are 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 distribution of stakes/locations, the closer the coefficients are to 0. To guarantee the decentralization of block producers from the aspects of stakes distribution and geographical locations, the following constraints should be satisfied.

$$G(\boldsymbol{\Upsilon})\leq\eta_s,\tag{4}$$

and

$$G(\boldsymbol{\lambda})\leq\eta_l.\tag{5}$$

where $\eta_s,\eta_l\in[0,1]$ are the thresholds of decentralization w.r.t. stakes distribution and geographical locations, respectively.

2) 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, or in other words, is finalized. 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

$$T^{F,\delta}=T^I+T^{C,\delta},\tag{6}$$

where $T^{C,\delta}$ is the consensus latency, i.e., the time cost for the

[3]This method can be easily extended to measure the system's decentralization from other aspects.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2019.2897805, IEEE Transactions on Industrial Informatics

validators to authenticate the generated block, which depends on the adopted consensus algorithm. We use $\delta = 0, 1, 2$ to denote three different consensus protocols, i.e., PBFT, Zyzzyva, Quorum, respectively. 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 by

$$T^{C,\delta} = T^{D,\delta} + T^{V,\delta}, \tag{7}$$

where the derivation of message delivery time $T^{D,\delta}$ and the validation time $T^{V,\delta}$ for these three consensus protocols $\delta = 0, 1, 2$ are given in **Appendix A**.

In IIoT networks, the smart devices usually expect to receive the finality of transactions within a short time. In order to meet the delay requirement of IIoT networks, we assume that one block should be issued and validated within a number of consecutive block intervals, i.e., $\omega\,(\omega > 1)$ block intervals[4]. Specifically, the TTF satisfy the following constraint.

$$T^{F,\delta} \leq \omega \times T^I, \delta = 0, 1, 2. \tag{8}$$

3) Security:

For security, the first generation blockchain consensus algorithm (PoW, PoS, DPoS) can only offer high probability of security. In theory, someone could use enough ($> 51\%$) mining power/stake to mine/mint an alternative "longer" blockchain that goes all the way back to genesis [35], [39]. However, for the BFT-type protocols, unambiguous finality can be reached under all network conditions as long as a fraction of participants are honest [42]. Therefore, the loyalty of the validators are very critical for BFT-type consensus algorithms. To guarantee the security of blockchain systems with consensus algorithm δ, the number of malicious validators f should be restricted by the following constraint.

$$f \leq F^\delta, \delta = 0, 1, 2, \tag{9}$$

where $F^0 = F^1 = \left\lfloor \frac{K-1}{3} \right\rfloor$ and $F^2 = 0$ represent for the maximum tolerable number of malicious validators.

It can be observed from (1) and (8) that the effects of block size and block interval act in two ways. For one thing, increasing block size or reducing block interval can bring an improvement of transactional throughput, as is illustrated in (1). For another, the time for the transactions to be finalized (i.e., TTF) increases with larger block size since there are more transactions to be validated at a time. Meanwhile, as shown in (8), the decrease of block interval imposes a stricter constraint on consensus delay. Additionally, the consensus delay is closely related to the chosen validators and the adopted consensus algorithm, which is restricted by (4), (5) and (9). Therefore, the adjustment of block size and block interval or the selection of block producers and consensus algorithm should be conducted elaborately, in order to reach the four-way trade-off.

[4]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 [22]. More general case will be considered in future works.

V. DRL-based Performance Optimization Framework

To handle the dynamic and large-dimension characteristics of IIoT systems, we resort to the DRL approach. The DRL framework [50] contains an offline deep neural network (DNN) construction phase that can approximate the action-value function with corresponding states and actions, and an online dynamic deep Q-learning phase for action selection, system control, and dynamic network updating. 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 transaction size χ, stakes distribution $\boldsymbol{\Upsilon}$, geographical location of IIoT nodes $\boldsymbol{x}$, computing capability of IIoT nodes $\boldsymbol{c} = \{c_k\}$, and the date transmission rate of the links between each pair of IIoT nodes $\boldsymbol{R} = \{R_{i,j}\}$, which is denoted as

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

B. Action Space

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

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

where the block producers indicator is $\boldsymbol{a} = \{a_n\}, a_n \in \{0, 1\}, \sum_{n=1}^{N} a_n = K, z_n \in \Phi_S$ with $a_n = 1$ representing node z_n is chosen as a block producer while $a_n = 0$ otherwise. Besides, the action w.r.t. consensus algorithm selection is denoted by $\delta = \{0, 1, 2\}$, which means PBFT, Zyzzyva, and Quorum is selected as the consensus protocol, respectively. Additionally, using limited fractional methods, block size $S^B \in \left\{0.2, 0.4, \ldots, \dot{S}\right\}$ and block interval $T^I \in \left\{0.5, 1, \ldots, \dot{T}\right\}$ with block size limit $\dot{S}$ and maximum block interval $\dot{T}$.

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.

$$\begin{aligned} \mathcal{P}1: \quad & \max_{\mathcal{A}} \quad Q\left(\mathcal{S}, \mathcal{A}\right) \\ C1: \quad & G\left(\boldsymbol{\Upsilon}\right) \leq \eta_s, G\left(\boldsymbol{\lambda}\right) \leq \eta_l, \\ C2: \quad & T^{F,\delta} \leq \omega \times T^I, \delta = 0, 1, 2, \\ C3: \quad & f \leq F^\delta, \delta = 0, 1, 2. \end{aligned} \tag{12}$$

where $Q\left(\mathcal{S}, \mathcal{A}\right)$ is the action-value function calculated by $Q\left(\mathcal{S}, \mathcal{A}\right) = \mathbb{E}\left[\sum_{t=0}^{\infty} \mu^t \mathcal{R}^{(t)}\left(\mathcal{S}^{(t)}, \mathcal{A}^{(t)}\right) \middle| \mathcal{S}^{(0)} = \mathcal{S}, \mathcal{A}^{(0)} = \mathcal{A}\right]$ with the discount factor $\mu \in (0, 1]$ that reflects the tradeoff

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2019.2897805, IEEE Transactions on Industrial Informatics

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 constraints $C1-C3$ can't be satisfied, it means that the modulated blockchain system has a poor performance in decentralization, TTF or security. Therefore, we set the reward to be 0 for this case to avoid this invalid situation. Finally, we formally present the proposed DRL-based performance optimization framework in **Algorithm 1**.

Algorithm 1: DRL-based Performance Optimization Framework for Blockchain-enabled IIoT systems

Offline DNN construction:
1) Load the historical state transition profiles and $Q\left(\mathcal{S},\mathcal{A}\right)$ value estimates in experience memory $\mathcal{D}$;
2) Pre-train the DNN (main Q network) with input pairs $\left(\mathcal{S},\mathcal{A}\right)$ and the corresponding estimated $Q\left(\mathcal{S},\mathcal{A}\right)$.
Online learning:
for *each decision epoch* t **do**
  /** Modulating the blockchain system ***/
  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 $Q\left(\bullet\right)$ is estimated by the main Q network;
  2) Execute $\mathcal{A}^{(t)}$ to select the block producers and consensus algorithm, and adjust the 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 $L\left(\theta\right)=\left[y^{(i)}-Q\left(\mathcal{S}^{(i)},\mathcal{A}';\theta\right)\right]^2$ every G steps.
end

VI. Simulation Results and Discussions

In the simulation, we consider a blockchain-enabled I-IoT system with 100 IIoT nodes and 21 block producers scattering over a 1km-by-1km area. The DNN included in the proposed DRL-based framework was implemented using PyTorch, which is a fast and flexible deep learning framework [51]. For the software environment, we utilized PyTorch 0.4.0 with Python 3.6 in Window 10 system. For different simulation scenarios, we trained the DRL-based framework with different initial parameters. The parameters settings used in the simulations are summarized in Table I.

For comparison, four baseline schemes are considered in the simulation part: 1) *Proposed scheme without consensus algorithm selection (CAS)*: the validators use the PBFT protocol to reach consensus. 2) *Proposed scheme with fixed block size*: the blocks generated by the 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 maximizing the immediate reward.

TABLE I: Simulation Parameters

Parameter	Value
The geographical coverage area of nodes	1km×1km
The number of nodes, N	100
The number of block producers, K	21
Average transaction size, χ	200B
The stake of node z_n, Υ_n	1-50 *token*
The computing resource of node z_n, c_n	10-30GHz
The data transmission rate of node z_n, R_n	10-100Mbps
Block size limit $\dot{S}$	8MB
Maximum block interval $\dot{T}$	10s
The threshold of decentralization w.r.t. stakes distribution and geographical locations, η_s, η_l	0.2, 0.3
The number of intervals that a new block should be validated, ω	6
The computing cost for verifying signatures and generating/verifying MACs, α, β	2MHz/1MHz
The batch size, M	3

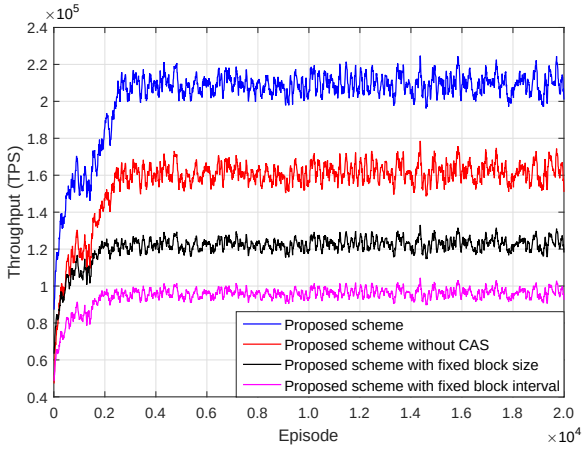


Fig. 5: Convergence performance of different schemes.

A. Performance of Convergence

Fig. 5 shows the convergence performance of our proposed DRL-based performance optimization scheme. From Fig. 5, we can observe that the transactional throughput is very low at the beginning of the learning process. However, as the number of episodes increases, the throughput increases and reaches a stable state after around 4000 episodes, which verifies the convergence performance of our proposed scheme. Besides, it can also be found that the proposed scheme can receive higher throughput than that of the other three DRL-based baselines, which shows the advantage of our proposed framework.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2019.2897805, IEEE Transactions on Industrial Informatics

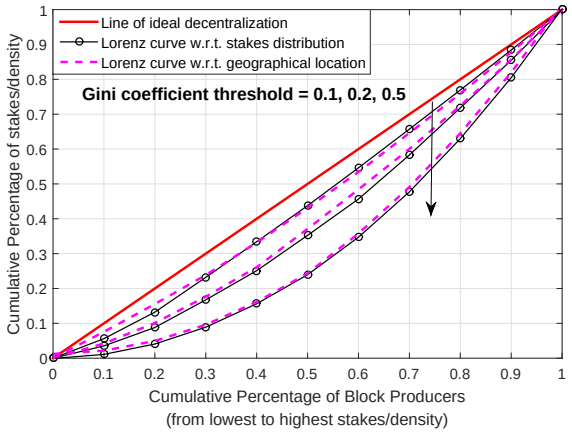


Fig. 6: Decentralization performance of block producers.

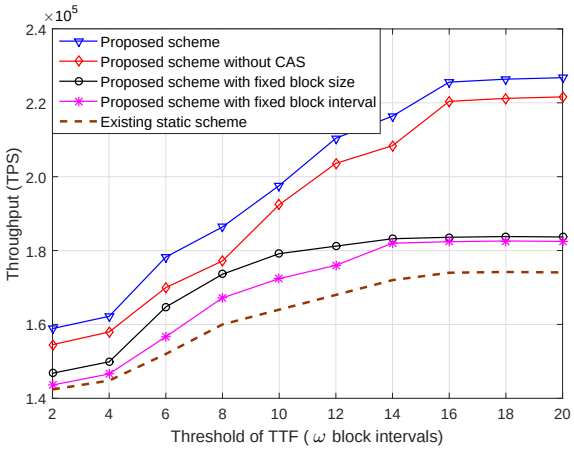


Fig. 7: Throughput vs. threshold of TTF.

B. Decentralization Performance

Fig. 6 describes the decentralization performance of the blockchain system. Different from [35] and [39], 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 and geographical location. We can see that as the threshold of Gini coefficient decreases, the Lorenz curve gradually approaches to the ideal decentralized one (the red line), i.e., the blockchain system becomes more decentralized. It reveals that Gini coefficient is an effective metric that is able to measure the decentralization of blockchain system from different aspects in a quantitative way.

C. Performance Comparison with the Baselines

We explore the effects of different parameters on the performance of the blockchain-enabled IoT network in Fig. 7 - Fig. 10. Specifically, the performance of our proposed DRL-based framework is compared with that of the baselines with different thresholds of TTF, average transaction sizes, average computational resources of validators, and block size limits. We can make the following observations.

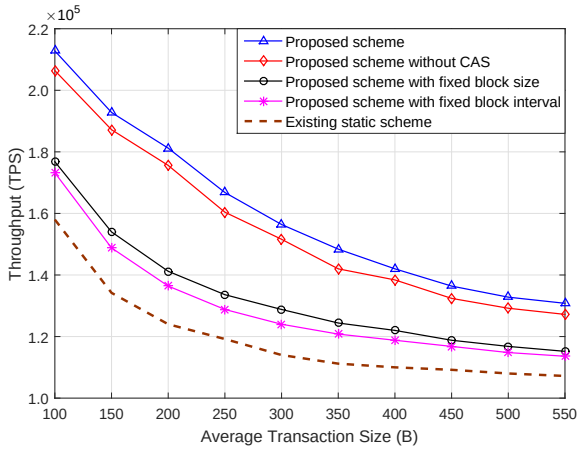


Fig. 8: Throughput vs. average transaction size.

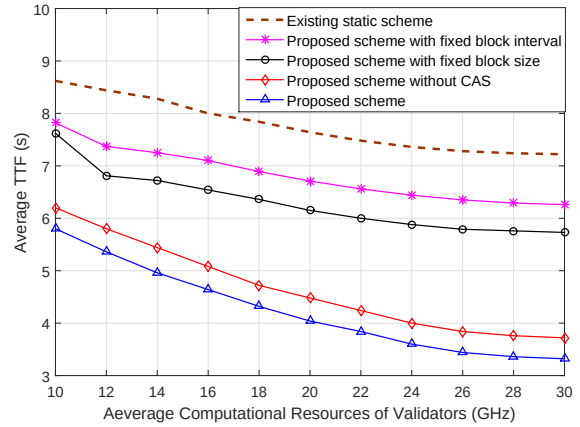


Fig. 9: Finality latency vs. average computational resources of validators.

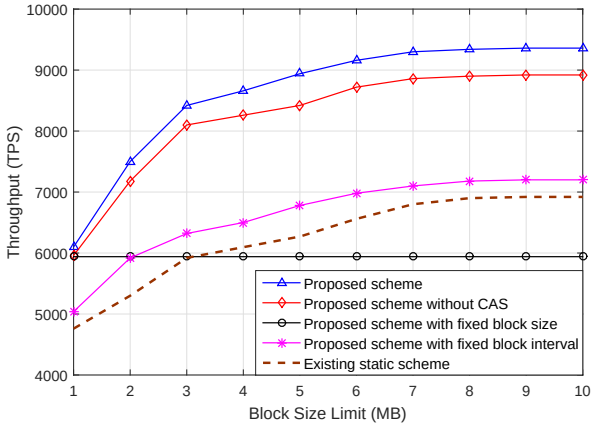


Fig. 10: Throughput vs. block size limit.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2019.2897805, IEEE Transactions on Industrial Informatics

1) The effect of the threshold of TTF on the system's tractional throughput is depicted in Fig. 7. We can see that all the schemes gain a higher transactional throughput with the 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. 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, we can find the existing static scheme shows the poorest performance, which reveals the superiority of DRL-based solutions.

2) Fig. 8 examines the transactional throughput with different transaction sizes, which makes sense when considering different types of transactions, such as the transactions of storing the data of a shipment and a vehicle may be different. One observation is that the throughput of the blockchain system for all the schemes decreases with the increasing transaction size. The reason behind is obvious that one block can hold less number of transactions for larger-size transactions. Another observation is that our proposed scheme can obtain the highest throughput with the variation of average transaction size, then follows the proposed scheme without CAS, the fixed block size scheme and the fixed block interval scheme, and the lowest is the existing static scheme. Similar observations can be made from Fig. 9: it takes the least time (lowest average TTF) for the proposed scheme to confirm transactions when compared with the baselines.

3) In Fig. 10, we discuss the impact of block size limit $\dot{S}$ on the throughput, which is also a controversial topic for scaling blockchain. Fig. 10 shows that the blockchain-enabled IoT network can handle more transactions with the increase of block size limit, which works for all the schemes except the fixed block size scheme. However, the throughput does not increase infinitely since the TTF constraint restricts the maximum number of transactions in one block. This also provides some insights into the design of blockchain systems in real scenarios that increasing the block size limit is not always helpful considering the other properties such as latency.

VII. Conclusion

In this paper, we proposed a novel DRL-based performance optimization framework for blockchain-enabled IIoT systems, where the scalability of the blockchain was improved while guaranteeing the system's decentralization, latency and security. In our proposed framework, we first provided a quantitative measurement for the performance of blockchain systems. Then the on-chain transactional throughput of the blockchain system was maximized by selecting the block producers and consensus algorithm, and adjusting the block size and block interval using the DRL technique. Simulation results demonstrated that the proposed framework can achieve higher throughput than the baselines with various system parameters. Future work is progress to consider other consensus algorithms and the decentralization of blockchain systems from other aspects.

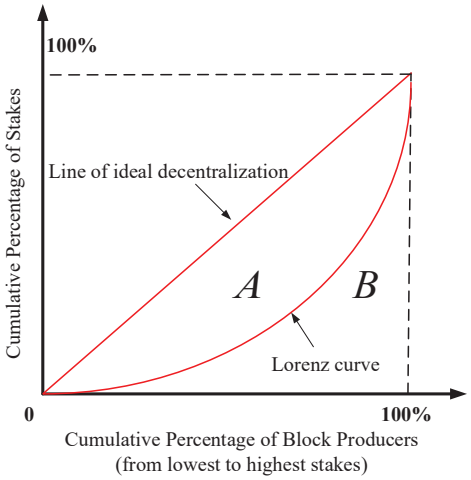


Fig. 11: An illustration of decentralization (take stakes distribution for example).

Appendix A
Gini Coefficient

Gini coefficient was first introduced by Corrado Gini in 1912 [46], which was primarily used as a measurement for the inequality of wealth or income. There are two commonly accepted versions for the definition of Gini coefficient.

The *first* version is based on the Lorenz curve, where Gini coefficient is defined as a ratio of the areas $\frac{area(A)}{area(A+B)}$ [47], [49] (an example is shown in Fig. 11). In this case, Gini coefficient can be interpreted as the degree of derivation from the line of ideal decentralization.

In the *second* version, Gini coefficient is defined as half of the relative mean absolute difference, which is mathematically equivalent to the Lorenz curve definition [52]. Specifically, the mean absolute difference is calculated by the average absolute difference of all pairs of items of the population, and the relative mean absolute difference is the mean absolute difference divided by the average. Hence, the expression of Gini coefficient is given by [48], [52]

$$G = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}|x_i - x_j|}{2\sum_{i=1}^{n}\sum_{j=1}^{n}x_i} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}|x_i - x_j|}{2n\sum_{i=1}^{n}x_i}, \tag{14}$$

where x_i is the wealth or income of person i, and the total number of persons is n. In this paper, we use the second definition to calculate the Gini coefficient in order to measure the decentralization of block producers.

Appendix B
Performance Analysis of Consensus Protocols

To decide *when* to implement *which* protocol, we evaluate the performance of these three protocols (i.e., PBFT, Zyzzyva, Quorum) from the aspects of fault-tolerance and consensus delay. Note that both PBFT and Zyzzyva can process a number of requests at one time use batching while Quorum is not able to support batching [41].

(i) PBFT

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2019.2897805, IEEE Transactions on Industrial Informatics

PBFT can tolerance at most $\left\lfloor \frac{K-1}{3} \right\rfloor$ faulty replicas [42]. As is shown in Fig. 2, there are five phases in the PBFT communication pattern. For example, in the *Request* phase, the client z_{b_c} sends a request for block validation to the primary $(z_{b_p}, p = 1, \ldots, K, p \neq c)$, then the primary verifies one MAC for each request. Note that each request contains one signature that requires verification for each replica (validator) during the consensus process. **1)** In the *Request* phase, the client z_{b_c} sends a request for block validation to the primary (validator $z_{b_p}, p = 1, \ldots, K, p \neq c$), then the primary verifies one MAC for each request. Note that each request contains one signature that requires verification for each replica (validator) during the consensus process. **2)** In the *Pre-prepare* phase, the primary processes a batch of requests (M validation requests for M generated blocks) in a single pre-prepare message and forwards this message to all the replicas. In this phase, the primary generates $K-1$ MACs to send the pre-prepare message and each replica needs to verify one MAC. **3)** For the *Prepare* phase, each replica authenticates the pre-prepare message and generates $K-1$ MACs to all the other replicas, and verifies $K-2$ MACs when they receive them. Meanwhile, the primary needs to verify $K-1$ MACs received from all the replicas. **4)** In the *Commit* phase, all the replicas including the primary exchange the message between each other, so the primary and any replica first send and then receive $K-1$ commit messages, which need to generate $K-1$ MACs and verify $K-1$ MACs, respectively. **5)** In the *Reply* phase, the primary and each replica generates a final MAC for each request in the batch to reply to the client.

Therefore, we can conclude that for each batch with bath size M, the primary z_{b_p} needs to verify M signatures and complete $2M + 4(K-1)$ MAC operations. Meanwhile, the replica $z_{b_i}, (i \neq c, p)$ needs to very M signature and complete $M + 4(K-1)$ MAC operations. Concerning the case with $f\left(f \leq F^0\right)$ faulty replicas, a correct replica will process at most two messages from a faulty server per message. This is because a correct replica processes messages from other replicas in round-robin order, and the faulty replicas may send messages too quickly in order to introduce overhead and further retard the consensus process [45]. Considering the worst case, the computational load per batch for the primary z_{b_p} and the replica z_{b_i} are $\mathcal{O}^0_{b_p} = M\alpha + [2M + 4(K+f-1)]\beta$ and $\mathcal{O}^0_{b_i} = M\alpha + [M + 4(K+f-1)]\beta$, respectively. So the validation time $T^{V,0}$ of each request can be calculated by $T^{V,0} = \frac{1}{M} \max\limits_{k=1,\ldots,K;k\neq c} \left\{ \frac{\mathcal{O}^0_{b_k}}{c_{b_k}} \right\}$.

Recall that any message sent is delivered within a timeout $\mathcal{T}$. Hence, the message delivery time $T^{D,0}$ of each request can be derived as follows.

$$\begin{aligned} T^{D,0} &= \tfrac{1}{M}\left(t_1 + t_2 + t_3 + t_4 + t_5\right) \\ &= \tfrac{1}{M}\left(\begin{array}{l} \min\left\{\frac{MS^B}{R_{b_c,b_p}}, \mathcal{T}\right\} + \min\left\{\max\limits_{i\neq c,p} \frac{MS^B}{R_{b_p,b_i}}, \mathcal{T}\right\} + \\ \min\left\{\max\limits_{i\neq j;i,j\neq c} \frac{MS^B}{R_{b_i,b_j}}, \mathcal{T}\right\} + \\ \min\left\{\max\limits_{i\neq j} \frac{MS^B}{R_{b_i,b_j}}, \mathcal{T}\right\} + \min\left\{\max\limits_{i\neq c} \frac{MS^B}{R_{b_i,b_c}}, \mathcal{T}\right\} \end{array} \right). \end{aligned} \tag{15}$$

(ii) Zyzzyva

Zyzzyva can also tolerate $\left\lfloor \frac{K-1}{3} \right\rfloor$ faulty replicas to guarantee the system security, i.e., $F^1 = \left\lfloor \frac{K-1}{3} \right\rfloor$. Similarly, we can also evaluate Zyzzyva protocol as follows.

As is shown in Fig. 3, in the *fast case*, the primary z_{b_p} needs to verify M signature and complete $2M + K - 1$ MAC operations while replica $z_{b_i}, (i \neq c, p)$ requires to verify M signature and complete $M+1$ MAC operations for each batch. In this sense, the computational load per batch for the primary z_{b_p} and the replica z_{b_i} are $\mathcal{O}^{1^{(1)}}_{b_p} = M\alpha + (2M + K - 1)\beta$ and $\mathcal{O}^{1^{(1)}}_{b_i} = M\alpha + (M+1)\beta$, respectively. Therefore, the validation time $T^{V,1^{(1)}}$ and message delivery time $T^{D,1^{(1)}}$ of each request can calculated by $T^{V,1^{(1)}} = \frac{1}{M} \max\limits_{k=1,\ldots,K;k\neq c} \left\{ \frac{\mathcal{O}^{1^{(1)}}_{b_k}}{c_{b_k}} \right\}$, and

$$\begin{aligned} T^{D,1^{(1)}} &= \tfrac{1}{M}\left(t_1 + t_2 + t_3\right) \\ &= \tfrac{1}{M}\left(\begin{array}{l} \min\left\{\frac{MS^B}{R_{b_c,b_p}}, \mathcal{T}\right\} + \min\left\{\max\limits_{i\neq c,p} \frac{MS^B}{R_{b_p,b_i}}, \mathcal{T}\right\} \\ + \min\left\{\max\limits_{i\neq c} \frac{MS^B}{R_{b_i,b_c}}, \mathcal{T}\right\} \end{array} \right). \end{aligned} \tag{16}$$

In *two-phase case* where there are $f\left(f \leq F^1\right)$ faulty replicas, the computational load per batch for the primary z_{b_p} and the replica z_{b_i} are $\mathcal{O}^{1^{(2)}}_{b_p} = M\alpha + (4M + K + f - 1)\beta$ and $\mathcal{O}^{1^{(2)}}_{b_i} = M\alpha + (3M+1)\beta$, respectively. Therefore, the the validation time $T^{V,1^{(2)}}$ and message delivery time $T^{D,1^{(2)}}$ of each request can calculated by $T^{V,1^{(2)}} = \frac{1}{M} \max\limits_{k=1,\ldots,K;k\neq c} \left\{ \frac{\mathcal{O}^{1^{(2)}}_{b_k}}{c_{b_k}} \right\}$, and

$$\begin{aligned} T^{D,1^{(2)}} &= \tfrac{1}{M}\left(t_1 + t_2 + t_3 + t_r + t_4 + t_5\right) \\ &= \tfrac{1}{M}\left(\begin{array}{l} \min\left\{\frac{MS^B}{R_{b_c,b_p}}, \mathcal{T}\right\} + \min\left\{\max\limits_{i\neq c,p} \frac{MS^B}{R_{b_p,b_i}}, \mathcal{T}\right\} + \\ \min\left\{\max\limits_{i\neq c} \frac{MS^B}{R_{b_i,b_c}}, \mathcal{T}\right\} + t_r + \\ \min\left\{\max\limits_{i\neq c} \frac{MS^B}{R_{b_c,b_i}}, \mathcal{T}\right\} + \min\left\{\max\limits_{i\neq c} \frac{MS^B}{R_{b_i,b_c}}, \mathcal{T}\right\} \end{array} \right). \end{aligned} \tag{17}$$

(iii) Quorum

The robustness of Quorum is poorest among these three protocols since it fails to reach consensus once there's any faulty replica, i.e., $F^2 = 0$. The consensus process only involves two phases – *Request* and *Reply*. Note that there's no primary replica and batching doesn't work for Quorum. We can derive that the computational load per request for the replica $z_{b_k}, k = 1, \ldots, K, k \neq c$ is $\mathcal{O}^2_{b_k} = \alpha + 2\beta$. Therefore, the the validation time $T^{V,2}$ and message delivery time of each request $T^{D,1^{(2)}}$ can calculated by $T^{V,2} = \max\limits_{k=1,\ldots,K;k\neq c} \left\{ \frac{\mathcal{O}^2_{b_k}}{c_{b_k}} \right\}$ and $T^{D,2} = t_1 + t_2 = \min\left\{\max\limits_{i\neq c} \frac{S^B}{R_{b_c,b_i}}, \mathcal{T}\right\} + \min\left\{\max\limits_{i\neq c} \frac{S^B}{R_{b_i,b_c}}, \mathcal{T}\right\}$.

REFERENCES

[1] L. D. Xu, W. He, and S. Li, "Internet of things in industries: A survey," *IEEE Trans. Ind. Informat.*, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.

[2] J. Kang, R. Yu, X. Huang, M. Wu, S. Maharjan, S. Xie, and Y. Zhang, "Blockchain for secure and efficient data sharing in vehicular edge computing and networks," *IEEE Internet of Things*, accepted, Nov. 2018.

[3] H. Liu, Y. Zhang, and T. Yang, "Blockchain enabled security in electric vehicles cloud and edge computing," *IEEE Network Magazine*, vol. 32, no. 3, pp. 78 – 83, May/June 2018.

[4] J. Kang, R. Yu, X. Huang, S. Maharjan, Y. Zhang, and E. Hossain, "Enabling localized peer-to-peer electricity trading among plug-in hybrid electric vehicles using consortium blockchains," *IEEE Trans. Ind. Informat.*, vol. 13, no. 6, pp. 3154 – 3164, Dec. 2017.

[5] Z. Zhou, B. Wang, Y. Guo, and Y. Zhang, "Blockchain and computational intelligence inspired incentive-compatible demand response in internet of electric vehicles," *IEEE Trans. on Emerging Topics in Computational Intelligence*, accepted, Nov. 2018.

[6] H. Wang, O. L. Osen, G. Li, W. Li, H.-N. Dai, and W. Zeng, "Big data and industrial internet of things for the maritime industry in northwestern norway," in *Proc. IEEE Region 10 Conf.* Macao, Nov. 2015, pp. 1–5.

[7] J. He, J. Wei, K. Chen, Z. Tang, Y. Zhou, and Y. Zhang, "Multitier fog computing with large-scale iot data analytics for smart cities," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 677–686, April 2018.

[8] D. Miller, "Blockchain and the internet of things in the industrial sector," *IT Professional*, vol. 20, no. 3, pp. 15–18, May/Jun. 2018.

[9] X. Liang, J. Zhao, S. Shetty, and D. Li, "Towards data assurance and resilience in iot using blockchain," in *Proc. IEEE Military Commun. Conf. (MILCOM)*. Baltimore, MD, Oct. 2017, pp. 261–266.

[10] N. Z. Aitzhan and D. Svetinovic, "Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams," *IEEE Trans. on Dependable and Secure Comput.*, vol. 15, no. 5, pp. 840–852, Sept./Oct. 2018.

[11] N. Teslya and I. Ryabchikov, "Blockchain-based platform architecture for industrial iot," in *Proc. 21st Conf. of Open Innovations Association (FRUCT)*. Helsinki, Nov. 2017, pp. 321–329.

[12] Z. Li, J. Kang, R. Yu, D. Ye, Q. Deng, and Y. Zhang, "Consortium blockchain for secure energy trading in industrial internet of things," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3690 – 3700, Aug. 2018.

[13] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *Proc. IEEE Int. Congress on Big Data (BigData Congress)*. Honolulu, HI, Jun. 2017, pp. 557–564.

[14] W. Chen, M. Ma, Y. Ye, Z. Zheng, and Y. Zhou, "Iot service based on jointcloud blockchain: The case study of smart traveling," in *Proc. IEEE Symposium on Service-Oriented Syst. Eng. (SOSE)*. Bamberg, Mar. 2018, pp. 216–221.

[15] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2009. http://www.bitcoin.org/bitcoin.pdf; accessed on 7 Sept. 2018.

[16] F. R. Yu, J. M. Liu, Y. He, P. B. Si, and Y. H. Zhang, "Virtualization for distributed ledger technology (vdlt)," *IEEE Access*, vol. 6, pp. 25 019–25 028, April 2018.

[17] "Bitcion charts and graphs - blockchain," https://www.blockchain.com/zh-cn/charts; accessed on 7 Sept. 2018.

[18] "Ethereum project," https://www.ethereum.org/; accessed on 7 Sept. 2018.

[19] "Bitcoincash: Peer-to-peer electronic cash," https://www.bitcoincash.org/; accessed on 7 Sept. 2018.

[20] I. Eyal, A. E. Gencer, E. G. Sirer, and R. V. Renesse, "Bitcoin-ng: A scalable blockchain protocol," in *Proc. 13th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*. Santa Clara, CA, March 2016, pp. 45–59.

[21] A. Kiayias, A. Russell, B. David, and R. Oliynykov, "Ouroboros: A provably secure proof-of-stake blockchain protocol," Aug. 2017. https://whitepaperdatabase.com/cardano-ada-whitepaper/; accessed on 7 Sept. 2018.

[22] I. Grigg, "Eos - an introduction," July 2017. https://eos.io/documents/EOS-An-Introduction.pdf; accessed on 7 Sept. 2018.

[23] ZILLIQA, "The zilliqa technical whitepaper v0.1," Aug. 2017. https://docs.zilliqa.com/whitepaper.pdf; accessed on 7 Sept. 2018.

[24] J. Poon and V. Buterin, "Plasma: Scalable autonomous smart contracts," Aug. 2017. http://plasma.io/; accessed on 7 Sept. 2018.

[25] "Cosmos: Internet of blockchains," 2017. https://cosmos.network/; accessed on 7 Sept. 2018.

[26] "Aion: The internet, decentralized," 2018. https://aion.network/; accessed on 7 Sept. 2018.

[27] J. Poon and T. Dryja, "The bitcoin lightning network: Scalable off-chain instant payments," Jan. 2016. https://lightning.network/lightning-network-paper.pdf; accessed on 7 Sept. 2018.

[28] "The raiden network," 2015. https://raiden.network/; accessed on 7 Sept. 2018.

[29] J. Lind, I. Eyal, P. Pietzuch, and E. G. Sirer, "Teechan: Payment channels using trusted execution environments," *[online] arXiv:1612.07766 [cs.CR]*, Dec. 2016.

[30] 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.

[31] Z. Xu and *et. al*, "Experience-driven networking: A deep reinforcement learning based approach," *[online] arXiv:1801.05757 [cs.NI]*, Jan. 2018.

[32] E. A. Brewer, "Towards robust distributed systems," in *Proc. the nineteenth annual ACM symposium on Principles of distributed computing*. Portland, Oregon, USA, July 2000, pp. 1–7.

[33] K. Zhang and H. Jacobsen, "Towards dependable, scalable, and pervasive distributed ledgers with blockchains," in *Proc. IEEE 38th Int. Conf. on Distributed Comput. Syst. (ICDCS)*. Vienna, July 2018, pp. 1337–1346.

[34] M. Snider, K. Samani, and T. Jain, "Delegated proof of stake: Features & tradeoffs," Mar. 2018, https://multicoin.capital/wp-content/uploads/2018/03/DPoS-Features-and-Tradeoffs.pdf; accessed on 7 Sept. 2018.

[35] U. Klarman, S. Basu, A. Kuzmanovic, and E. G. Sirer, "bloxroute: A scalable trustless blockchain distribution network whitepaper v1.0," *BLOXROUTE LABS, WHITEPAPER*, Mar. 2018.

[36] K. Samani, "Models for scaling trustless computation," https://multicoin.capital/2018/02/23/models-scaling-trustless-computation/; accessed on 7 Sept. 2018.

[37] P. K. Sharma, M. Chen, and J. H. Park, "A software defined fog node based distributed blockchain cloud architecture for iot," *IEEE Access*, vol. 6, pp. 115–124, Sept. 2017.

[38] A. E. Gencer, S. Basu, I. Eyal, R. van Renesse, and E. G. Sirer, "Decentralization in bitcoin and ethereum networks," *[online] arXiv:1801.03998 [cs.CR]*, Mar. 2018.

[39] Z. H. 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.

[40] D. Stoyan, W. Kendall, and J. Mecke, "Stochastic geometry and its applications," *New York, NY, USA: Wiley*, 1996.

[41] A. Singh, T. Das, P. Maniatis, P. Druschel, and T. Roscoe, "Bft protocols under fire," in *Proc. 5th USENIX Symposium on Networked Systems Design and Implementation*, Jan. 2008, pp. 1–16.

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

[43] R. Kotla, L. Alvisi, M. Dahlin, A. Clement, and E. Wong, "Zyzzyva: speculative byzantine fault tolerance," *SIGOPS Oper. Syst. Rev.*, vol. 41, no. 6, pp. 45–48, Oct. 2007.

[44] R. Guerraoui, N. Knezevic, V. Quema, and M. Vukolic, "The next 700 bft protocols," in *Proc. 5th European conf. on Computer syst. (EuroSys'10)*. New York, NY, USA: ACM, April 2010, pp. 363–376.

[45] 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*, April 2009, pp. 153–168.

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

[47] L. Dai, Y. Jia, L. Liang, and Z. Chang, "Metric and control of system fairness in heterogeneous networks," in *Proc. 23rd Asia-Pacific Conf. on Commun. (APCC)*. Perth, WA, Dec. 2017, pp. 1–5.

[48] Z. Lin, F. Wen, Y. Ding, and Y. Xue, "Data-driven coherency identification for generators based on spectral clustering," *IEEE Trans. on Industrial Informatics*, vol. 14, no. 3, pp. 1275–1285, Mar. 2018.

[49] D. Wu, G. Zeng, L. Meng, W. Zhou, and L. Li, "Gini coefficient-based task allocation for multi-robot systems with limited energy resources," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 1, pp. 155–168, Jan. 2018.

[50] I. Goodfellow, Y. Bengio, and A. Courville, "Deep learning," *Book in preparation for MIT Press*, 2016, http://www.deeplearningbook.org/; accessed on 7 Sept. 2018.

[51] A. Paszke, "Pytorch tutorials," 2018, https://pytorch.org/tutorials/; accessed on 7 Sept. 2018.

[52] A. Sen, "On economic inequality," *Oxford: Oxford University Press*, 1977.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2019.2897805, IEEE Transactions on Industrial Informatics

Mengting Liu received her B.S. degree from Minzu University of China, Beijing, China, in 2013. She is currently pursuing the Ph.D. degree with Beijing University of Posts and Telecommunications (BUPT), Beijing, China. From Sept. 2017 to Sept. 2018, she was a visiting Ph.D. student with the University of British Columbia, Vancouver, Canada. Her current research interests include Blockchain, deep reinforcement learning, and mobile edge computing.

F. Richard Yu (S'00-M'04-SM'08-F'18) received the PhD degree in electrical engineering from the University of British Columbia (UBC) in 2003. From 2002 to 2006, he was with Ericsson (in Lund, Sweden) and a start-up in California, USA. He joined Carleton University in 2007, where he is currently a Professor. He received the IEEE Outstanding Service Award in 2016, IEEE Outstanding Leadership Award in 2013, Carleton Research Achievement Award in 2012, the Ontario Early Researcher Award (formerly Premiers Research Excellence Award) in 2011, the Excellent Contribution Award at IEEE/IFIP TrustCom 2010, the Leadership Opportunity Fund Award from Canada Foundation of Innovation in 2009 and the Best Paper Awards at IEEE ICNC 2018, VTC 2017 Spring, ICC 2014, Globecom 2012, IEEE/IFIP TrustCom 2009 and Int'l Conference on Networking 2005. His research interests include wireless cyber-physical systems, connected/autonomous vehicles, security, distributed ledger technology, and deep learning.

Yinglei Teng (M'12) received the B.S. degree from Shandong University, China, in 2005, and the Ph.D. degree in electrical engineering from the Beijing University of Posts and Telecommunications (BUPT) in 2011. She is currently an Associate Professor with the School of Electronic Engineering, BUPT. Her current research interests include UDNs and massive MIMO, IoTs and Blockchains.

Victor C. M. Leung (S'75-M'89-SM'97-F'03) is a Professor of Electrical and Computer Engineering and holder of the TELUS Mobility Research Chair at the University of British Columbia, Vancouver, Canada. He has co-authored more than 1200 technical papers in the areas of wireless networks and mobile systems. Dr. Leung is a Fellow of the Royal Society of Canada, the Canadian Academy of Engineering and the Engineering Institute of Canada.

Mei Song received the B.E. and M.E. degrees from Tianjin University, China. She is currently a Professor in Beijing University of Posts and Telecommunications. Her current research interests include integrated design technology, VLSI&CAD system, resource allocation and mobility management in heterogeneous and cognitive networks, cooperative communication, and other advanced technology in future communication networks.