Making Big Data Open in Edges: A Resource-Efficient Blockchain-Based Approach

Chenhan Xu, Kun Wang, *Senior Member, IEEE*, Peng Li, *Member, IEEE*, Song Guo, *Senior Member, IEEE*, Jiangtao Luo, *Senior Member, IEEE*, Baoliu Ye, *Member, IEEE*, and Minyi Guo, *Fellow, IEEE*

Abstract—The emergence of edge computing has witnessed a fast-growing volume of data on edge devices belonging to different stakeholders which, however, cannot be shared among them due to the lack of the trust. By exploiting blockchain's non-repudiation and non-tampering properties that enable trust, we develop a blockchain-based big data sharing framework to support various applications across resource-limited edges. In particular, we devise a number of novel resource-efficient techniques for the framework: (1) the PoC (Proof-of-Collaboration) based consensus mechanism with low computation complexity which is especially beneficial to the edge devices with low computation capacity, (2) the blockchain transaction filtering and offloading scheme that can significantly reduce the storage overhead, and (3) new types of blockchain transaction (i.e., Express Transaction) and block (i.e., Hollow Block) to enhance the communication efficiency. Extensive experiments are conducted and the results demonstrate the superior performance of our proposal.

Index Terms—Big data, blockchain, collaborative edges, transaction offloading, consensus mechanism

1 Introduction

With the significant improvements in cloud computing technologies, an increasing amount of services are deployed in the cloud, which might inevitably cause long time latency for users [1], [2], [3]. Accordingly, edge computing emerges, effectively decreasing time latency via deploying services at the edge of the network, such as mobile phones, surveillance cameras, and Internet of things (IoT) sensors [4], [5]. More and more users are surrounded by edge devices and data belonging to different stakeholders [6], [7].

Unfortunately, these edge devices may not always cooperate with each other because they are in a distrusted environment [5]. In some cases, malicious edge devices would deny that they have read shared business data from others even though they are benefited from it. Moreover, malicious devices can do tampering when accessing these business data. These distrust issues eventually cause non-collaboration in edges.

In this paper, we study the distrust issues of big data sharing in collaborative edges. A few previous works have investigated how to solve the distrust issues. Hussain et al. [8] proposed to first verify reputation via a centralized trusted third party before performing data operations. However, this approach may lead to high latency and the third party becomes more vulnerable. Kantert et al. [9] have studied to calculate credit scores to select a more reliable participant. However, the credit scores are only suggestions for edges and the malicious participant cannot be eliminated. Our goal is to implement edge collaboration in the distrusted environment.

To this end, we propose to deploy blockchains for big data sharing in collaborative edges. The blockchain is a public append-only ledger carrying all transactions that have been executed [10]. Every block carrying some transactions is committed to the global blockchain. Since every participant has a copy of the blockchain, no one can reject to admit the transactions of data flow from edge applications that have been committed. Moreover, the blockchain can reach global consensus on the whole sequence of transactions so that a conflicting transaction will be dropped once it is committed. For example, this blockchain framework can prevent malicious double-spending attack. This attack allows the malicious participant to deny that they have benefited from the collaboration.

- *C. Xu is with the Key Laboratory of Broadband Wireless Communication and Sensor Network Technology, Nanjing University of Posts and Telecommunications, Nanjing 210003, China. E-mail: xchank@outlook.com.*
- *K. Wang is with Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing University of Posts and Telecommunications, Nanjing 210003, China, and also with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China. E-mail: kwang@njupt.edu.cn.*
- *P. Li is with the School of Computer Science and Engineering, The University of Aizu, Aizu-Wakamatsu City, Fukushima 965-8580, Japan. E-mail: pengli@u-aizu.ac.jp.*
- *S. Guo is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China. E-mail: song.guo@polyu.edu.hk.*
- *J. Luo is with Electronic Information and Networking Research Institute, Chongqing University of Posts and Telecommunications, Chongqing 400065, China. E-mail: luojt@cqupt.edu.cn.*
- *B. Ye is with the School of Computer and Infomation, Hohai University, Nanjing 210046, China. E-mail: yebl@hhu.edu.cn.*
- *M. Guo is with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200000, China. E-mail: guomy@cs.sjtu.edu.cn.*

Manuscript received 23 June 2018; revised 11 Sept. 2018; accepted 11 Sept. 2018. Date of publication 20 Sept. 2018; date of current version 13 Mar. 2019. (Corresponding authors: Kun Wang and Minyi Guo.)
Recommended for acceptance by Y. Yang.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPDS.2018.2871449

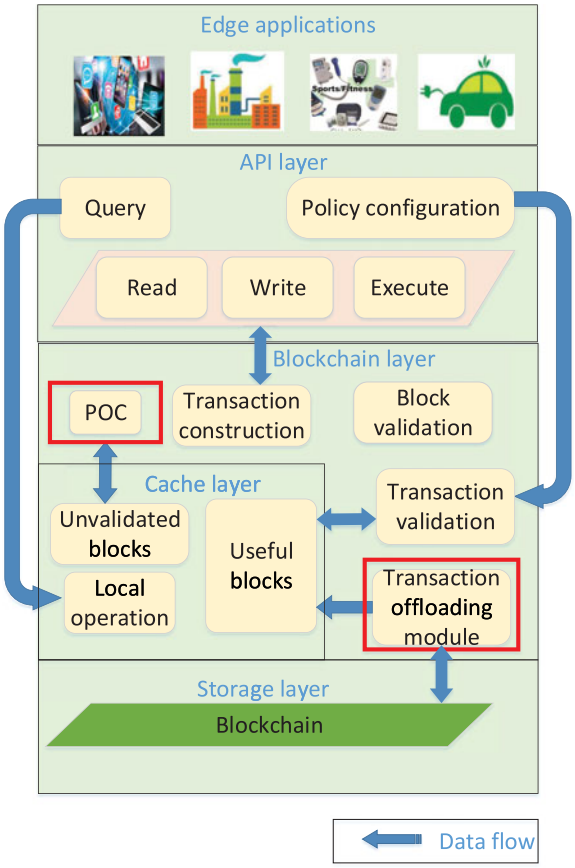


Fig. 1. Green blockchain framework in collaborative edges.

Although non-repudiation and non-tampering properties of the blockchain are promising, there are still some challenges as follows:

- Edge devices are heterogeneous of computational and network resources. Therefore, some edge devices with limited resources cannot support the operations related to blockchain and big data.
- Edge devices have limited storage resources, which are hardly to store the whole ledger.

Different from existing blockchain for edge computing [11], where the blockchain technology is employed without taking the limitation of resources into consideration, we design a green blockchain framework with reduced computational, storage, and network resource requirements for big data sharing in collaborative edges. This framework, as shown in Fig. 1, is divided into four layers, i.e., the Application Programming Interface (API) layer, the cache layer, the blockchain layer, and the storage layer. The details of the framework design will be presented in Section 3. The API layer and blockchain layer can directly access data from the cache layer, rather than from the storage layer, which reduces the response time and makes our system adapted for big data sharing. This paper mainly focuses on the design of the blockchain layer in the proposed framework, especially in green consensus mechanism, transaction offloading, and efficient blockchain network. Our contributions are summarized as follows:

- We develop a green blockchain framework for big data sharing in collaborative edges, considering the challenging issues arose from the properties of edge computing. This framework deploys a green consensus mechanism in the collaborative edges called Proof-of-Collaboration (PoC). Based on this PoC consensus mechanism, edge devices compete for generating new blocks via showing their collaboration credits instead of paying a significant amount of computation to solve a mathematics puzzle, which greatly saves the computational resources in edge devices.
- We propose a futile transaction theory with the proof. This theory shows the former transaction, whose outputs are all referenced by the latter transactions, is useless for the validation of newly generated transaction. Furthermore, we design a novel transaction offloading module based on Futile Transactions Filter (FTF) algorithm, which contributes to reducing the storage resources occupied by the blockchain.
- We propose Express Transactions (E-TX) and Hollow Blocks to enhance the network efficiency of the proposed framework. The smart contract based E-TX is designed for supporting the asynchronous validation of transaction. Moreover, Hollow Block which can significantly reduce redundancy in block propagation is proposed to further enhance the network resource efficiency.
- We perform extensive experiments on 16 RaspberryPi microcomputers to demonstrate the high performance of our proposal. These experimental results show that PoC mechanism can reduce at most 90 percent of computational resources than PoW mechanism. Additionally, more than 95 percent of storage resources and 27 percent of network resources can be reduced by the transaction offloading module and the optimizing methods, respectively. These practical mechanisms can make better use of the edge network capacity to support the trustful big data sharing.

The structure of the paper is organized as follows. Section 2 reviews the related works on edge collaboration and blockchain technology. The green blockchain framework in collaborative edges is designed in Section 3. Section 4 demonstrates the technical details of our proposed green PoC consensus mechanism. How transaction offloading module helps to reduce the storage resources is illustrated in Section 5. Section 6 investigates the network patterns of the proposed framework. We illustrate the proposed express transaction and hollow block in this section. Section 7 performs extensive experiments to show the performance of our proposal.

2 Related Works

In this section, we give an overview of edge collaboration and blockchain technology.

2.1 Edge Collaboration

With the emergence of edge computing technology, the edge collaboration issues are taken into great consideration [12]. Shi et al. [6] surveyed the edge computing and investigated the challenges and opportunities. They explained the definition of edge computing and demonstrated many case studies, such as cloud offloading, video analytics, smart city, and edge collaboration. Tran et al. [13] explored to implement

mobile edge computing collaboration in 5G ecosystem. The authors illustrated a context-aware and dynamic collaboration infrastructure in the edge of Radio Access Network (RAN) consisting of mobile edge devices, edge services, and base stations, where the heterogeneous resources are merged at edges. Zhang et al. [14] developed a novel computing paradigm for big data sharing called Firework in collaborative edges, where virtual shared data views are built and data is transmitted to users through predefined interfaces. This framework guarantees users' privacy as well as solving the response latency issue by pushing data to the network edges. Wu et al. [15] proposed a two-step detection mechanism in mobile edge collaboration, where users' preferences are concerned for constructing virtual communities and collaborative clusters. Moreover, a video coding sharing mechanism based on users identities is developed for flexible video distribution and decreasing energy consumption at the edge of mobile networks. To adapt blockchain to the edge computing, Xiong et al. [16], [17] proposed to offload the mining task to the edge computing service providers, who make profit by providing computational resource. In the proposed scenario, Stackelberg game is used to optimize the price of the resource. Stanciu [11] proposed to use blockchain as a distributed control system confirming to the IEC standard. Samaniego et al. [18], [19] leveraged blockchain as the carrier of the virtual resource, which is a kind of micro-services, to reduce the computation moving cost on edge hosts.

2.2 Blockchain Technology

Blockchain technology has aroused great interests from both academic and industrial fields, including finance, e-health, distributed system, etc. Christidis et al. [20] presented a comprehensive survey on blockchain and claimed that the blockchain can be employed to construct a resilient distributed system in which participants could interact with each other without a trusted third party. They demonstrated that the combination of blockchain and IoT can make significant improvements. Azaria et al. [21] designed a blockchain based system called MedRec for electronic medical record management. In the MedRec, medical stakeholders such as medical scientist and public health authorities are involved as miners. Weber et al. [22] adapted blockchain for business. The trust of blockchain underpins the international business process. The authors performed three case studies to illustrate the feasibility of their proposed solution.

There were several works focusing on the inner mechanism of blockchain technologies. Saito et al. [23] proposed that blockchain can be regarded as a probabilistic state machine, where the amount of participants is uncertain and participants cannot make commitment on the decisions. Eyal et al. [24] developed a novel Bitcoin-NG (Next Generation) protocol to improve the scalability, which belongs to a kind of Byzantine fault tolerant protocol. In addition, a set of metrics standard is introduced to quantify the security and efficiency of the blockchain protocols. Lewenberg et al. [25] designed a Directed Acyclic Graph (DAG) structure for blockchain to enhance the throughout, where a block could reference many predecessors. Miller et al. [26] proposed a practical asynchronous HoneyBadgerBFT protocol. This protocol can ensure normal operation without any time assumptions on a wide area network more than 100 nodes, and the throughput can achieve up to tens of thousands of transactions per second. Milutinovic et al. [27] designed a time- and energy- efficient blockchain consensus algorithm based on extra trusted execution environments. The core function of this algorithm is constructed on the random number generation of the underpinned trusted execution environment. Turesson et al. [28] proposed to use deep learning model training rather than calculate hash in Proof-of-Work consensus. Luu et al. [29] proposed a secure sharding protocol for blockchain mining, which is also used in Ethereum for improving the throughput and scalability. Zheng et al. [30] gave an overview of classic blockchain technologies from the aspects of the architecture and the consensus. Based on this work, Bach et al. [31] performed a further comparative analysis of several classic and modern blockchain consensus algorithms from complexity, scalability, security, and rewarding method.

Different from these previous works, we first develop a green blockchain framework to enable trust for big data sharing in collaborative edges. Then, we put forward green PoC consensus mechanism in our framework to reduce computational resources in edges, where edge devices give their proof of contributing collaboration to compete for block generation, rather than wasting computational resources to solve mathematic puzzle. Furthermore, we propose a futile transaction theory and establish transaction offloading module based on FTF algorithm for reduction of storage resources occupied by blockchain. Finally, we design Express Transaction and Hollow Block to reduce the usage of the network resource in blockchain.

3 Framework Design

Edges consist of edge infrastructures, base stations, edge servers, and IoT edge devices, etc. Every edge links to the network served by different Internet Service Providers (ISPs). In our proposal, we deploy blockchain on these edges, where every block contains multiple transaction logs of big data flows among edge applications. For a more clear description of our proposal, we demonstrate a green blockchain framework in collaborative edges in this section. Our proposed framework is divided into four layers, as shown in Fig. 1.

API layer offers interfaces for edge applications, which abstract the functions of cache and blockchain layer to provide various calls for implementing edge collaboration. Specifically, API layer contains following operations:

- Read, write, and execute operations abstract transaction construction in the blockchain layer.
- Policy configuration is designed to set the operation permission to local data for other edge devices.
- Query operation can query operation record of other edge devices on local data, where the latest operations of local data are stored in the local operation module in cache layer.

The cache layer is designed to accelerate the responses to the calls, and it contains local operations, invalidated blocks, and useful blocks.

The blockchain layer implements the content of blockchain in edges, including several modules as follows:

- First, transaction and block construction module transforms the requests from the upper layer into

transactions or blocks, which will be broadcast to the entire edge network for validation.
- Second, transaction validation module contains validation rules, where the operation permission to local data is often set for other edge devices via modifying validation rules. Besides, transaction and block validation modules guarantee rules, which are foundations of the green PoC consensus, as we will illustrate in Section 4.
- Finally, transaction offloading module locates the blocks with useful transactions, and then the useful blocks are updated to the cache layer. This module is designed to reduce storage resources occupied by blockchain.

Storage layer in the bottom provides persistent storage service for the upper layers.

4 Green Poof-of-Collaboration Consensus Mechanism

Blockchain is a distributed data structure and every participant keeps a copy of the entire blockchain [32], [33], [34]. The first class component in blockchain is named transaction, which is a record of some asset transferring. These transactions generated by different devices are validated via a whole blockchain network, and are packaged into a block by a miner. Then, miners keep consistency of blocks validation via performing consensus mechanism. Finally, a valid block is added to the blockchain.

4.1 Different Chain Types and Consensus

The blockchain has two types: a public chain and a consortium (private) chain. If anyone can participate in a blockchain network, the blockchain is naturally public or consortium [35]. If every participant can take part in blockchain operations, for example, competing to mine blocks or proposing transactions, the blockchain is public [20]. Since the public chain is open and competitive, the participants in public chain network do not trust each other [36]. On the contrary, the participants of a consortium chain network are privileged and white-listed.

The differences between the two chain types result in different kinds of potential applied consensus protocols. Giving any participant an opportunity to mine blocks, Proof-of-Work (PoW) makes a great success in Bitcoin, which is the biggest public chain in the world [33], [34]. PoW requires participants that compete for mining blocks to give the proof of their work. This proof is a kind of mathematical puzzle that is easy to be validated but extremely hard to be solved, i.e., solving these kinds of puzzles consumes fabulous amount of computational resources. In most cases, the puzzle has the following form:

$$\begin{aligned} &\text{Find } n \\ &\quad \text{s.t.} \quad \text{SHA256}(\text{SHA256}(h.n)) < \text{target}, \end{aligned} \tag{1}$$

where "." is a string concatenate operator, and h represents the content of the newest block. The smaller the target is, the more difficult the mining is. Later, the concept of Proof-of-Stake (PoS) [37] has been proposed, and its main idea is that stakeholders should show their stake of assets to compete mining. It is a promising replacer of PoW, since it requires quite less computational resources than that of PoW.

In addition, Practical Byzantine Fault Tolerant (PBFT) and its variants are widely used in consortium chains, which tolerate up to a third of participants that occur any form of failure (Byzantine fault), given the number of participants in advance and fixed [38].

Within the context of collaborative edges, as mentioned above, every edge device is a participant of the network, and may require to perform blockchain operations. Moreover, the number of edge devices, which should adapt to the demand of users, is not fixed. As we mentioned in Section 1, the blockchain based edge collaboration urges to pursue a green solution because of the limited computational and storage resources. Hence, inspired by PoS and PoW, we will illustrate PoC in details in next subsection.

4.2 Proof-of-Collaboration Mechanism

Edge devices give the proof of their contribution to collaboration rather than solve meaningless mathematical puzzle to obtain the privileges of collaboration. The key concept of Proof-of-Collaboration is that participants contribute to the big data sharing so that they can also benefit from other participants' collaboration. More specifically, the green PoC consensus mechanism is designed as follows.

4.2.1 Collaboration Credit

In our design, the edge collaboration is underpinned by a new asset called Collaboration Credit ($\mathcal{CC}$), which is slightly similar to BTC in Bitcoin [33] and ETH (GAS) in Ethereum [39]. This means that the data flow from edge applications recorded by transactions, i.e., collaborations, must be paid using $\mathcal{CC}$ in the proposed framework. The $\mathcal{CC}$ used for this payment is dynamically determined by collaboration fee $\mathcal{F}$ as

$$\mathcal{F} = \frac{\psi'}{\psi \times n} \quad \mathcal{CC}/\text{kB}, \tag{2}$$

where ψ is a pre-defined throughput threshold, ψ' represents the average throughput of the entire network during recent 100 blocks, and n denotes the number of edge devices in the network. The average network throughput ψ' can be calculated by dividing the total size of transactions in recent 100 blocks by the time consumption of generating these 100 blocks. In practice, ψ equals to the maximum value of devices' network capacity. According to the definition of $\mathcal{F}$, the framework will decrease $\mathcal{F}$ to encourage collaboration when the recent throughput is lower than the pre-defined threshold, or increase $\mathcal{F}$ to reduce network overload when the throughput is higher than defined. Moreover, the larger the amount of edge devices is, the lower $\mathcal{F}$ will be in the framework.

In the framework, $\mathcal{CC}$ can be gained by two approaches. First, the block proposer can be rewarded a certain number of $\mathcal{CC}$ by adding a new block to the blockchain successfully. Second, the block proposer earns $\mathcal{CC}$ from the transactions carried by the block. The collaboration fee $\mathcal{F}$ is used to evaluate the contribution, i.e., to prevent selfish applications requesting shared data without sharing their own data. If an edge application leverages data flow from other applications, it must contribute to the edge collaboration.

4.2.2 Proof-of-Collaboration

In the framework, the way to propose a block is related to the Persistence $\mathcal{P}$, which is defined as the time since the last $\mathcal{CC}$ changes. Our proposal has the following three core rules, underpinned by $\mathcal{CC}$ and $\mathcal{P}$, to guarantee itself a green blockchain:

Rule 1 (Dynamic Difficulty). *The mining in the proposed PoC is different from Eq. (1). Mining in PoC is influenced by dynamic difficulty, which is different from various participants. It has the form as follows:*

$$\begin{aligned} &Find\ n \\ &s.t. \quad SHA256(SHA256(h.n)) \ < \ \mathcal{CC} \times \mathcal{P} \times target, \end{aligned} \tag{3}$$

where the target is the same as that in Eq. (1).

Rule 2 (Winner Initialization). *The block proposer must pay for himself when constructing the new block. The operation of constructing the new block costs the $\mathcal{CC}$ of the proposer and gives the same amount of $\mathcal{CC}$ as return, i.e., the payment changes $\mathcal{CC}$ of the proposer, but proposers do not lose $\mathcal{CC}$. In addition, the new block pay the proposer extra $\mathcal{CC} \times \mathcal{P} \times 0.001$ percent as reward. According to the definition of $\mathcal{P}$, the $\mathcal{P}$ of a block proposer will be set to 0 when he successfully adds a block to the blockchain.*

Rule 3 (Partial competition). *A block proposer must have $\mathcal{P} \in [\mathcal{L}, \mathcal{R}]$, where $\mathcal{L}$ is calculated by*

$$\mathcal{L} = \frac{n}{\Theta}, \tag{4}$$

and $\mathcal{R} = 3\mathcal{L}$. The value of Θ can vary according to the collaboration demand. A higher Θ in Eq. (4) makes more intense competition, and a lower Θ decreases the security of the blockchain. The typical value of Θ is 0.75.

The guarantees provided by these three rules are manifold:

- *For a single edge device*, the expectation of the needed computational resources is quite lower than that in PoW. [40] gives the expectation of the needed computational resources in PoW, which is

$$\mathbb{E}_{\mathrm{PoW}} = \frac{target_{\max}}{target} \times 2^{32}. \tag{5}$$

  However, according to *Rule 1*, the expectation in PoC is

$$\mathbb{E}_{\mathrm{PoC}} = \frac{target_{\max}}{\mathcal{CC} \times \mathcal{P} \times target} \times 2^{32} = \frac{1}{\mathcal{CC} \times \mathcal{P}} \mathbb{E}_{\mathrm{PoW}}. \tag{6}$$

  Constrained by *Rule 2*, only the winner of competition should clear its $\mathcal{P}$. If an edge device fails in competing to propose a block, its $\mathcal{P}$ is preserved. This provides the its superiority in the next round of competition for proposing block. However, the failed nodes in PoW waste their all computation [41].
- *For the whole edge network*, *Rule 3* stipulates that the block proposer should wait $\mathcal{L}$ to rejoin the competition for proposing the next block. This makes only a part of edge devices in the network try to mine at the same time, and reduces $\mathcal{L}/n$ computational resources for the whole network. However, all the nodes in PoW compete to mine all the time. Besides, the all-nodes-competition in PoW makes a high possibility that more than one node propose valid blocks, i.e., fork [34]. The fork wastes enormous computational resources. Since not all the edge devices in PoC compete at the same time, the fork rarely happens.

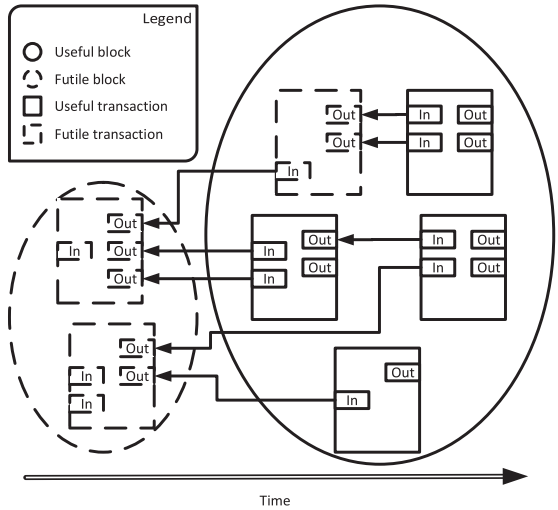


Fig. 2. References among transactions.

5 Transaction Offloading

In traditional blockchain, the historical blocks are stored in every node. As we mentioned in previous section, as continuous running of the blockchain, the size of these blocks becomes larger and larger. Edge devices will not be able to afford the storage size sooner or later [42], [43]. Moreover, a new participant is expected to download these blocks before joining the blockchain network, if he intends to validate the new generated transactions [34]. Within the edge context, this download operation costs enormous network resources, which makes edge collaboration inefficient. In this section, we first present how the transactions are organized. Then, the proposed transaction filtering theory is illustrated in details.

5.1 Transaction Organization

In the blockchain, every transaction references one or more previous transactions to support its validity. The structure of mutual-reference transactions is depicted in Fig. 2. In the *inputs* field, the transaction references a list of *outputs* which belong to one or more previous transactions, and indicates the indexes of *outputs* in transactions where they belong to. In the blockchain, the node that performs transaction validation is called full node. The full node takes more than ten procedures to verify whether a transaction is valid [44]. The most essential idea is to check the assets which are used to pay for the new generated transaction. Hence, for each *input* in the transaction being validated, the full node will check whether the referenced *output* exists. If not, the transaction will be rejected. Additionally, the full node also protects blockchain against double-spending issue, which is denoted in *Remark 1*. This is because the same asset cannot be spent more than once.

Remark 1 (Double-spending). If one *input* references an *output* that has already been spent, the transaction containing this input is invalid, i.e., double-spending [45].

These validation procedures enlighten us that the blockchain network can only preserve blocks whose transactions might be referenced, which benefits us to resolve storage and network crisis of blockchain in the edge. Motivated by this, we propose a novel transaction offloading module, which reduces the storage resource occupation of the blockchain, based on a Futile Transactions Filter algorithm. We illustrate the technical details in the following subsection.

5.2 Transaction Filtering Theory

As illustrated in Section 5.1, the *outputs* of valid transactions are referenced by later ones. Based on the transaction organization, we have the following theorem:

Theorem 1 (Futile transaction). *The transaction whose* outputs *are all referenced by the latter transactions is useless for the validation of the new generated transaction.*

Proof. According to *Remark 1*, a transaction whose *outputs* are all referenced by the latter transactions cannot be referenced further, or the double-spending issue will occur. If a new generated transaction references several previous valid transactions, we know this transaction is valid. The previous transactions are not involved in the process of validation. Hence, the *Theorem 1* is true. □

Theorem 1 underpins our proposed FTF algorithm, as shown in *Algorithm 1*. The FTF excepts the entire blockchain stored in the edge device where it runs as an input. In lines 2-6, the FTF goes through all the transactions in the given blockchain, searches every *outputs* referenced by other transactions' *inputs*, and marks these *outputs* as *referenced*. After that, the FTF goes through all the transactions again, marks the useful (non-futile) and futile transactions, as shown in lines 7-15, respectively. Hence, the time complexity of *Algorithm 1* is $O(n)$, where n represents the number of transactions in the blockchain.

Algorithm 1. Futile Transactions Filter

```
1: procedure FUTILE-TRANS-FILTER(B)
                                   ▷ B: a instance of blockchain
2:   for all t ∈ B.transactions do
                                   ▷ traverse all transactions in the chain
3:     for all i ∈ t.inputs do
4:       MarkAsReferenced(i.txid, i.index)
5:     end for
6:   end for
7:   for all t ∈ B.transactions do
8:     MarkAsFutile(t)
9:     for all o ∈ t.outputs do
10:      if IsMarked(o)==false then
11:        MarkAsUseful(t)
                   ▷ A transaction is useful for future validation
                   ▷ if the outputs of it are not all referenced
12:        break
13:      end if
14:    end for
15:  end for
16: end procedure
```

After FTF finishes the filtering of futile transactions, the transaction offloading module locates the blocks that carry useful transactions, and updates them to the cache layer. The futile blocks, i.e., the blocks only carry futile transactions, will be sent to stakeholder's clouds for backup. Then, these blocks will be dropped from edge devices. Because the *Algorithm 1* does not change the distribution and the amount of computational resource of the whole network, it does not increase the risk of being attacked by "51 percent attack". The offloading module runs periodically, and maintains the amount of blocks at a low level all the time. For edge devices, the offloading module can reduce fabulous storage resources occupied by blockchain, so that devices can operate more edge applications, making them efficient and green.

6 Blockchain Network Optimization

In this section, we first investigate the network utilization pattern of the proposed PoC. Based on the pattern, we improve the existing design of transactions and blocks to enhance the network efficiency of the proposed green blockchain framework.

6.1 Network Utilization Pattern

In blockchain, the consensus mechanisms such as PoW, PBFT, and the proposed PoC have common basic procedures in communication. On the one hand, all new transactions need to be propagated over the entire blockchain network for validation. On the other hand, if a participant finds the solution of the current proof, the participant propagates its block to all the participants seeking for acceptance [46].

From blockchain participant's perspective, transactions and blocks come at different time. Moreover, the propagations of transactions and blocks are later than the validations of them [47], which means when the participant is performing validation, its network is idle.

For transactions or blocks, as illustrated above, the propagations and validations are mutually exclusive. If we model the blockchain network as a connected graph, the total propagation time $T_{\max}$ of a transaction or block can be formulated by

$$T_{\max} = \sum_{i \in \Gamma_{\max} \wedge V} T_i^v + \sum_{j \in \Gamma_{\max} \wedge E} T_j^c, \tag{7}$$

where $\Gamma_{\max}$ is the longest path in the network, V is the set of all participants, set E represents all connections between participants, T_i^v represents the time of validation in participants i, and T_j^c represents the time of propagation in connection j. Eq. (7) indicates that the propagations of transactions or blocks are synchronous.

We further consider the relationship of the new transactions and the new block. In blockchain, each participant maintains a transaction queue. Once a new block is proposed by a participant, it carries all the transactions in the queue. As mentioned above, new transactions need to be propagated over the entire blockchain network. Thus, most of the transactions that a new block carries overlap with the transactions in participants' queues, i.e., participants waste network resources to receive existing transactions in the block propagation process. We use Q_i to denote the

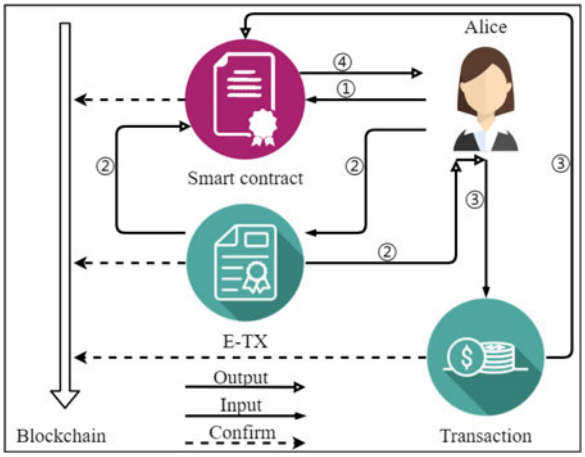


Fig. 3. The procedures for issuing an E-TX.

transaction queue of participant i, according to the propagations of transactions, we have

$$|Q_1 \wedge Q_2 \wedge \cdots \wedge Q_i| \approx |Q_T|, \tag{8}$$

where Q_T represents the transactions carried by the new block.

The idle and waste of the network resources are two main problems that limit big data application in collaborative edges. To address these two problems, we design new types of transaction and block. In the following two subsections, we illustrate the technical details of them.

6.2 Express Transaction

Based on the $\mathcal{CC}$ that is defined in Section 4.2, we propose the express transaction, which supports asynchronous transaction validation. Once a participant receives an E-TX, the participant first propagates it, and then performs the validation. We stipulate the issuer of an E-TX should pay the extra "express deposit", whose value equals to the value of the outputs of the E-TX. The purpose of the express deposit is to guarantee the validity of the E-TX. If the E-TX passes the validation, the express deposit will return to the issuer.

As shown in Fig. 3, technically, we use smart contract [48] to support the proposed E-TX. If Alice wants to issue an E-TX, she should first write a smart contract for managing the express deposit (procedure ①). This smart contract only responds to the transactions from Alice. Then, Alice constructs her E-TX, where the first and second outputs must be set as the express deposit to the above smart contract and any $\mathcal{CC}$ to Alice, by indicating the E-TX type in the header of the E-TX. The rest inputs and outputs of the E-TX can be customized, which is similar to the common transactions. After that, Alice starts propagating. Once the participants in the network receive the header of an E-TX, they propagate the header together with the body of the E-TX at first rather than validate it (procedure ②). This method reduces the total propagation time $T_{\max}$ to

$$T'_{\max} = \max_{i \in \Gamma_{\max} \wedge V} T_i^v + \sum_{j \in \Gamma_{\max} \wedge E} T_j^c. \tag{9}$$

If the E-TX that Alice issues is valid, it will be packaged to a block and added to the blockchain. Finally, Alice issues a common transaction, whose only input references the second output of the previous E-TX, and sets the smart contract as the only output (procedure ③). This procedure asks the smart contract to return the express deposit (procedure ④). If the E-TX is invalid, the final procedure will fail because the transaction cannot pass validation. The pseudo codes of the entire smart contract are shown in *Algorithm* 2. Lines 7-13 show when the smart contract is initialized, it has state 0. After the issuing of the E-TX, the smart contract receives express deposit and transfers to state 1. Lines 14-20 correspond to the return of the deposit. Then the smart contract turns to state 2, which indicates its finish. Note that an extra validation of the smart contract should be added to the procedures of transaction validation. The validation of the smart contract is to avoid the risk of that an issuer can retrieve express deposit after issuing an invalid E-TX.

Algorithm 2. Smart Contract for Express Deposit Management

```
1: procedure MANAGE-EXPRESS-DEPOSIT(t)
                                              ▷ t: transaction
2:   static P_KEY ← key_i
                                  ▷ key_i: public key of the issuer
3:   static state ← 0
4:   static deposit ← None
5:   if t.key = P_KEY then
6:     switch(state)
7:       Case 0:
8:       if t.output[0].value = t.value/2 and
9:       t.output[1].target = P_KEY then
10:        state ← 1
11:        deposit ← t
12:      end if
13:      return
                              ▷ state 0: receive express deposit
14:      Case 1:
15:      if t.input.length = t.output.length = 1 and
16:      t.input[0].from = deposit.output[1] then
17:        SEND(P_KEY, t.value + deposit.value/2)
18:        state ← 2
19:      end if
20:      return
                                  ▷ state 1: return deposit
21:      Case 2:
22:      return
                              ▷ state 2: smart contract finishes
23:  end if
24: end procedure
```

In the next subsection, we illustrate the proposed lightweight block structure called hollow block in detail.

6.3 Hollow Block

In this part, the data structure in the block called Merkle Tree [33] is introduced first. Then, we detail the mechanism of building the proposed hollow block based on the Merkle Tree.

6.3.1 Merkle Tree

In the Merkle tree, every leaf node contains the hash of a transaction and every non-leaf node carries with the hash of the concatenation of its child nodes' hashes. The Merkle tree

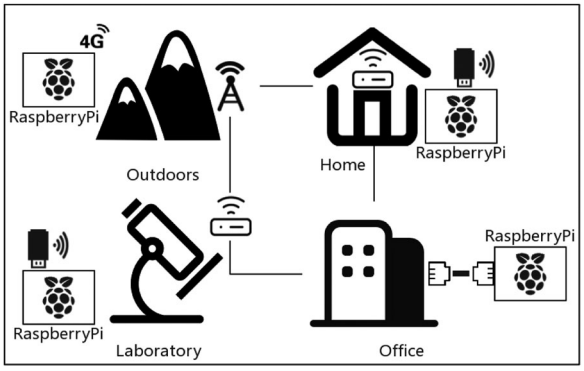


Fig. 4. Experiment platform.

supports efficient validation of the transaction integration. The Bitcoin stores the root of Merkle tree in every block for helping new participants download blockchains. If some transactions are tampered or broken during the downloading process, participants can utilize merkle tree to locate the transactions and download them rather than download the whole block again [49].

6.3.2 Redundancy Reduction

To reduce the redundancy among the new block and participants' queues of transactions, we propose the hollow block. In our design, different from the traditional blockchain, the hollow block consists of a block header together with the hash list of the transactions it should contains. We replace the Merkle root in the header of the hollow block with the cryptographic hash of the entire hash list, compressing the binary Merkle tree in traditional blockchain to a two layer Merkle tree. Note that the hollow block does not carry any transaction, which significantly reduces the network usage in block propagation. When receiving a new hollow block, the participant sorts the transactions in queue by chronological. Then, the participant calculates the cryptographic hash of these transactions and compares it with the Merkle root of the received hollow block. If they are equal, the participant packages these transactions into the hollow block and adds the hollow block to the blockchain. If they are not equal, the participant compares the hash list in hollow block with the transactions in queue to find all missing transactions, and then downloads them from the blockchain network. Note that the timestamp is determined by the transaction so that the sort results in different participants can keep consistence. Moreover, this timestamp will be determined before its submission. There is an appropriate tolerance in transaction propagation for its timestamp not being consistent with the current time.

The above mechanisms in consensus mechanism, blockchain storage, and blockchain network make blockchain a practical technology in collaborative edges by reducing the resource requirements of the blockchain. In the next section, the performance analyses of the proposed mechanisms are given.

7 Experiment

In this section, we first present the experiment settings. Then the experiment results are analyzed from computation, network, and storage aspects, respectively. Finally, we discuss the security limitation and generality of the proposed mechanisms.

TABLE 1
The Settings of the Experiments

Cores	Power (W)	Hash Rate (MH/s)	A	B	C
0 (idle)	1.2075	0	0	0	0
1	1.5225	0.133	2	4	6
2	1.785	0.266	6	4	2
3	1.995	0.414	6	4	2
4	2.2575	0.554	2	4	6

Note: Power is measured by PF9800 dynamometer.

7.1 Experiment Setup

Our experiment is conducted on the RaspberryPi 2 model B which equips with a 900 MHz quad-core ARM Cortex-A7 CPU, in Raspbian operation system [50]. Fig. 4 depicts our experiment platform, where there are 16 RaspberryPi micro computers that are deployed at our homes, laboratories, outdoors, and offices. These RaspberryPi micro computers access both wired and wireless networks.

We develop our proposed green blockchain using Python 3.6. The implementation is multi-processed. Since the Pickle module in Python uses extra spaces to serialize transactions and blocks [51], we write customized codes for serialization and unserialization.

The settings of experiments are illustrated in Table 1. We measure the average hash rate and power of the used RaspberryPis, where the hash operation is performed by Python's hashlib.sha256() library, and the power is measured with power supply in 5.25 V. To involve the factor of heterogeneous in computational resources, the experiments are divided into groups A, B, and C. Different groups have distinct computational resources limitation. For example, Table 1 indicates group A consists of two RaspberryPis running one core, six RaspberryPis running two cores, six RaspberryPis running three cores, and the rest running four cores. We measure the average hash rate and power of them, where the hash operation is performed by Python's hashlib.sha256() library, and the power is measured with power supply at 5.25 V. The data used for experiment is randomly generated. If it is not specified, every transaction has five inputs and outputs, while the number of transactions per block $\tau \sim U(50, 1000)$.

In next subsection, we give a comparison of the performance of PoC with that of PoW, which is used in Bitcoin and Ethereum. We also show how much the performance of throughput, storage, and network can be enhanced by the proposed mechanisms.

7.2 Experiment Results

The results of computational resources cost comparison are shown in Fig. 5. The blockchain height refers to the amount of blocks in the blockchain. The computational cost is the number of hash operation that a block proposer tried, and is evaluated by hashes. Constrained by *Rule 1*, our green blockchain with PoC keeps mining easily in different groups, which reduces at most 90 percent of computational resources for a single miner.

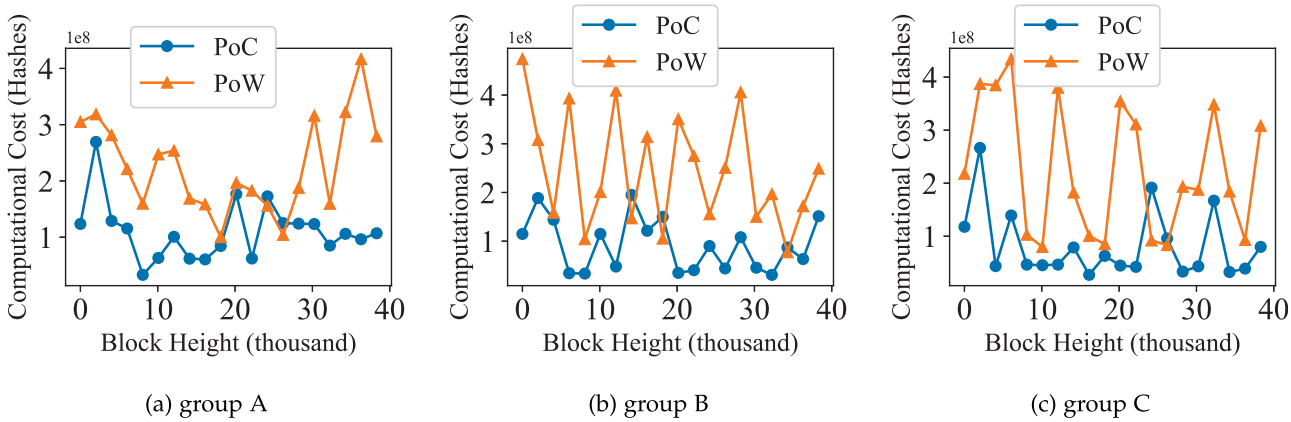


Fig. 5. Comparison of computational cost.

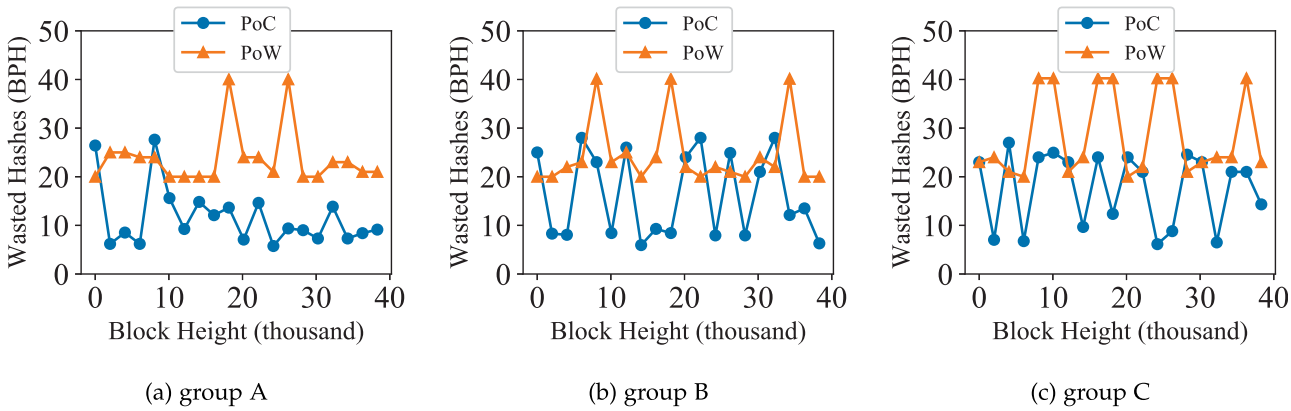


Fig. 6. Comparison of wasted hashes.

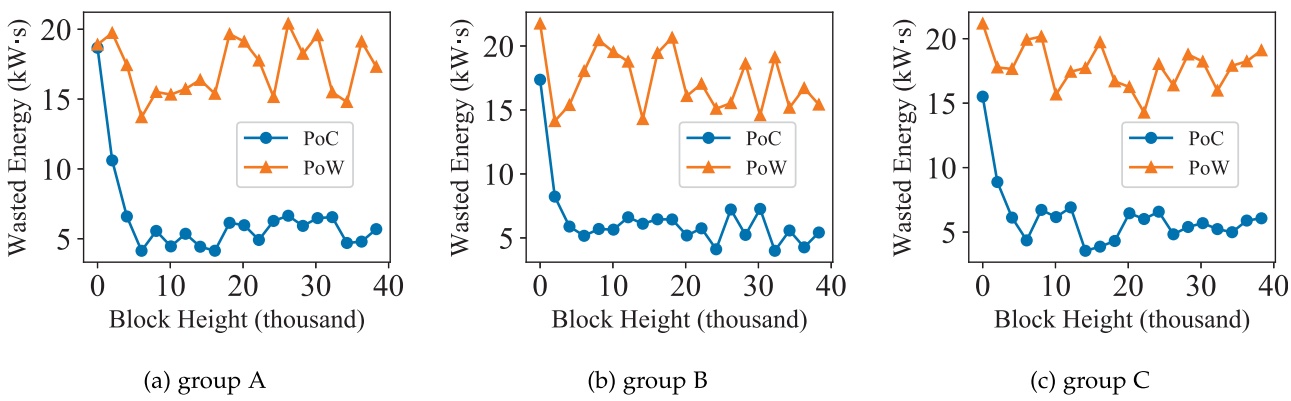


Fig. 7. Comparison of wasted energy.

To show that the *Rule 2* and *Rule 3* help to reduce computational resources, we illustrate the amount of wasted computational resources and energy in Figs. 6 and 7, respectively. This metric is defined as Block Proposer Hashes (BPH), which is calculated by $\text{BPH} = \frac{hashes_a}{hashes_p} - 1$, where $hashes_a$ represents the number of hash operations that all devices has performed to compete for proposing a new block, and $hashes_p$ is the number of hash operations that a block proposer has tried. Since *Rule 2* and *Rule 3* require a block proposer to stop mining for a while, only a part of the blockchain network devices perform mining at the same time. Figs. 6 and 7 show that our proposed PoC does reduce quite a lot of computational resources and energy cost on the edge network scale. In Fig. 7, the first thousands of blocks consume much more energy then the later blocks. This is because devices do not cumulate enough $\mathcal{CC}$ in the initial stage, which can reduce the mining consumption according to the *Rule 1*. After this stage, PoC is able to save tons of energy for the edge network. We further demonstrate the accumulative energy cost and computational resource usage of group A in Figs. 8 and 9 (the curves of group B and C are similar to that of group A). As the proposed framework running, the superiorities on energy and computational resources reduction of PoC become more and more obvious.

In Fig. 10, we demonstrate the storage costs of PoW and PoC. We let the transaction offloading module run one time when every 200 blocks are generated. The results show that

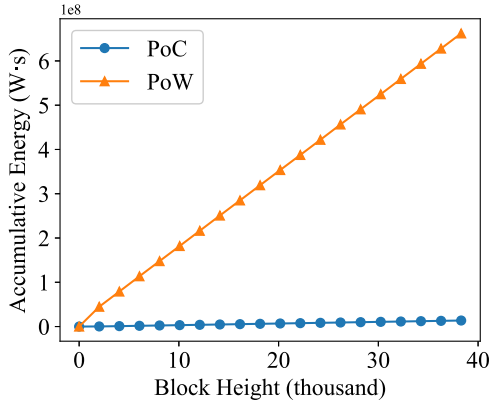


Fig. 8. Accumulative energy.

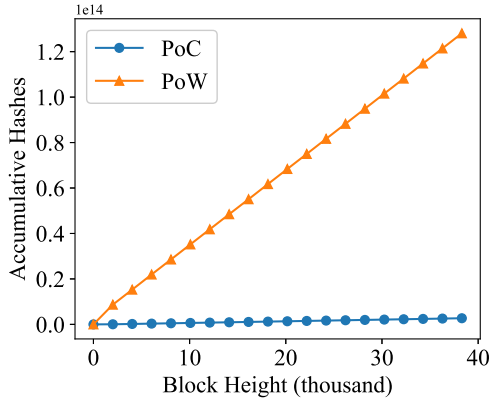


Fig. 9. Accumulative hashes.

the storage cost grows linearly without transaction offloading. Under the control of this offloading module, the storage cost of our green blockchain grows slowly and is stabilized at a low level. This is because the proposed module can recognize the transactions that cannot be further referenced and upload them to the cloud for reducing storage.

We compare the network throughput of traditional blockchain and blockchain with E-TX and hollow block (indicated by HB) in Fig. 11. The throughput of blockchain with E-TX is 17 percent higher than traditional blockchain. Hollow block further enhances the throughput to 23 percent higher than traditional blockchain. The enhancement is mainly contributed by the asynchronous validation of E-TX and the redundancy reduction provided by hollow block.

To show the capability of redundancy reduction, we demonstrate the network usage of recording transactions into blockchain with hollow block. Fig. 12 shows that the hollow blocks reduce about 27 percent network usage of a single participant by removing the redundancy between participant's queue and new blocks.

During the experiment, the average latency of submitting a transaction is 38 ms, which is lower than the requirement of latency (100 ms [52]) in 5G edge and IoT network [53], [54]. For each data sharing operation, the proposed method will bring 290 Bytes of communication overhead on average. In summary, the results of experiments show that our proposed green blockchain is able to reduce enormous computational, storage, and network resources, which helps to solve the critical challenges we describe in Section 1.

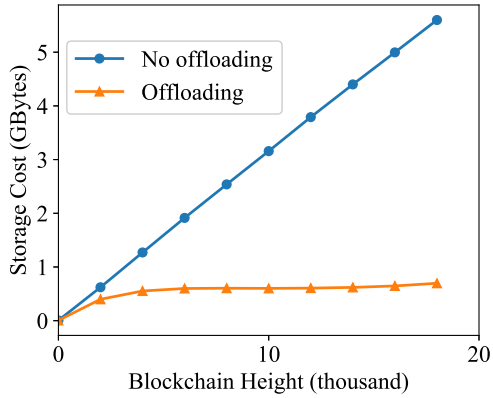


Fig. 10. Comparison of storage cost.

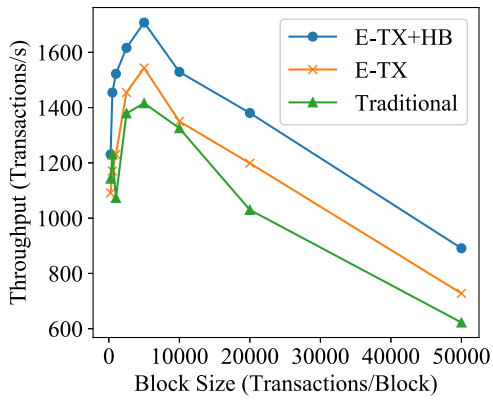


Fig. 11. Comparison of throughput.

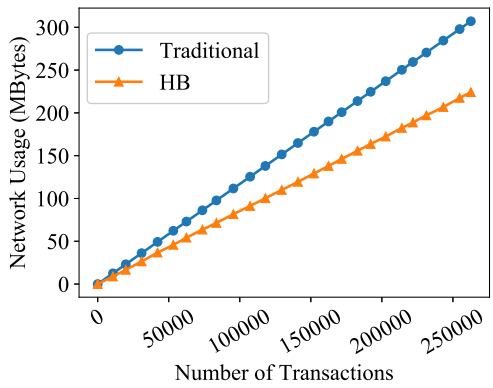


Fig. 12. Comparison of network usage.

7.3 Discussion

Security Limitation. The proposed PoC mechanism reduces the waste of computational resource, enabling blockchain based edge collaboration. Previous work indicates that the security of the PoW protocol is based on the "wasted" computation [55]. However, this assert is based on the threat model that the adversary aims to control the whole system. This model considers the blockchain system as an entity, which actually is an extreme case. In more general models, the adversary is able to start a double-spending attack if he owns more computational resource than any of the participants. In this situation, the security performance is determined by the participant owning the most computational resource. Therefore, the security level of PoW and PoC in general cases can be the same.

Generality of Futile Transaction. The transaction offloading module and its theory are based on Bitcoin-like organization of transaction. The basic idea of futile transaction also works for Ethereum and other blockchain implementation that are based on transaction reference. The Futile Transactions Filter in other implementation may be more complex. For Ethereum, the filter should analyze uncle block [39] in addition.

8 Conclusion and Future Works

With the explosive growing deployment of mobile digital devices and sensors, we have witnessed the emergence of edge computing. Edge computing technology enables computation to be performed at the edge of network, where users are now surrounded by large-scale edge devices of different enterprises. However, enterprises do not trust others, directly leading to non-collaboration among edge devices of different enterprises. This paper studies the distrust issues and employs the blockchain to enable trustful big data sharing in edge collaboration. Although the non-repudiation and non-tampering properties of the blockchain is promising, the limited computational, network, and storage resources in edge devices bring challenges to the design of the blockchain. To this end, We construct a green blockchain framework. First, we propose a green PoC consensus mechanism in our framework, where edge devices give their proof of contributing collaboration rather than consuming enormous computational resources to solve the mathematic puzzle for the privilege of collaboration. Second, we propose the futile transaction theory and design a transaction offloading module based on the FTF algorithm in our framework to reduce storage resources occupied by the blockchain. Third, Express Transaction and Hollow Block are proposed to enhance the network of the green blockchain framework. Finally, extensive experiments show the advantages and superiority of our proposal.

This paper mainly focuses on designing the blockchain layer in our proposed framework. How to design these layers in our proposed framework in a green and efficient manner is still an open issue. We are motivated to complete the whole framework design and further improve the performance of our proposal in the future.

Acknowledgments

This work is supported by National Key Research and Development Program of China (2018YFB1004700); NSFC (61872195, 61872310, 61832005, 61572262, 61872240); The preliminary version of this paper, titled Making Big Data Open in Collaborative Edges: A Blockchain-based Framework with Reduced Resource Requirements, was published in IEEE ICC 2018 [1].

References

[1] C. Xu, K. Wang, G. Xu, P. Li, S. Guo, and J. Luo, "Making big data open in collaborative edges: A blockchain-based framework with reduced resource requirements," in *Proc. IEEE Int. Conf. Commun.*, May 2018, pp. 1–6.

[2] K. Wang, J. Mi, C. Xu, Q. Zhu, L. Shu, and D.-J. Deng, "Real-time load reduction in multimedia big data for mobile internet," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 12, no. 5s, pp. 76:1–76:20, Oct. 2016. [Online]. Available: http://doi.acm.org/10.1145/2990473

[3] X. Zhou, K. Wang, W. Jia, and M. Guo, "Reinforcement learning-based adaptive resource management of differentiated services in geo-distributed data centers," in *Proc. IEEE/ACM 25th Int. Symp. Quality Serv.*, Jun. 2017, pp. 1–6.

[4] S. K. Sharma and X. Wang, "Live data analytics with collaborative edge and cloud processing in wireless iot networks," *IEEE Access*, vol. 5, pp. 4621–4635, 2017.

[5] M. Du, K. Wang, X. Liu, S. Guo, and Y. Zhang, "A differential privacy-based query model for sustainable fog data centers," *IEEE Trans. Sustainable Comput.*, 2017. DOI: 10.1109/TSUSC.2017.2715038.

[6] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.

[7] H. Jiang, K. Wang, Y. Wang, M. Gao, and Y. Zhang, "Energy big data: A survey," *IEEE Access*, vol. 4, pp. 3844–3861, 2016.

[8] M. Hussain and B. M. Almourad, "Trust in mobile cloud computing with lte-based deployment," in *Proc. IEEE 14th Int. Conf. Scalable Comput. Commun. Associated Workshops*, Dec. 2014, pp. 643–648.

[9] J. Kantert, S. Edenhofer, S. Tomforde, and C. Mller-Schloer, "Representation of trust and reputation in self-managed computing systems," in *Proc. IEEE Int. Conf. CIT/IUCC/DASC/PICOM*, Oct. 2015, pp. 1827–1834.

[10] C. Xu, K. Wang, and M. Guo, "Intelligent resource management in blockchain-based cloud datacenters," *IEEE Cloud Comput.*, vol. 4, no. 6, pp. 50–59, Nov. 2017.

[11] A. Stanciu, "Blockchain based distributed control system for edge computing," in *Proc. 21st Int. Conf. Control Syst. Comput. Sci.*, May 2017, pp. 667–671.

[12] K. Wang, Y. Wang, X. Hu, Y. Sun, D. J. Deng, A. Vinel, and Y. Zhang, "Wireless big data computing in smart grid," *IEEE Wireless Commun.*, vol. 24, no. 2, pp. 58–64, Apr. 2017.

[13] T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili, "Collaborative mobile edge computing in 5g networks: New paradigms, scenarios, and challenges," *IEEE Commun. Mag.*, vol. 55, no. 4, pp. 54–61, Apr. 2017.

[14] Q. Zhang, X. Zhang, Q. Zhang, W. Shi, and H. Zhong, "Firework: Big data sharing and processing in collaborative edge environment," in *Proc. 4th IEEE Workshop Hot Top. Web Syst. Technol.*, Oct. 2016, pp. 20–25.

[15] D. Wu, Q. Liu, H. Wang, D. Wu, and R. Wang, "Socially aware energy-efficient mobile edge collaboration for video distribution," *IEEE Trans. Multimedia*, vol. 19, no. 10, pp. 2197–2209, Oct. 2017.

[16] Z. Xiong, S. Feng, D. Niyato, P. Wang, and Z. Han, "Optimal pricing-based edge computing resource management in mobile blockchain," in *Proc. IEEE Int. Conf. Commun.*, May 2018, pp. 1–6.

[17] Z. Xiong, Y. Zhang, D. Niyato, P. Wang, and Z. Han, "When mobile blockchain meets edge computing," *IEEE Commun. Mag.*, vol. 56, no. 8, pp. 33–39, Aug. 2018.

[18] M. Samaniego and R. Deters, "Hosting virtual iot resources on edge-hosts with blockchain," in *Proc. IEEE Int. Conf. Comput. Inf. Technol.*, Dec. 2016, pp. 116–119.

[19] M. Samaniego and Deters, "Using blockchain to push software-defined iot components onto edge hosts," in *Proc. Int. Conf. Big Data Adv. Wireless Technol.*, 2016, pp. 58:1–58:9. [Online]. Available: http://doi.acm.org/10.1145/3010089.3016027

[20] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[21] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "Medrec: Using blockchain for medical data access and permission management," in *Proc. 2nd Int. Conf. Open Big Data*, Aug. 2016, pp. 25–30.

[22] I. Weber, X. Xu, R. Riveret, G. Governatori, A. Ponomarev, and J. Mendling, "Untrusted business process monitoring and execution using blockchain," in *Business Process Management*, M. La Rosa, P. Loos, and O. Pastor, Eds. Cham, Switzerland: Springer, 2016, pp. 329–347.

[23] K. Saito and H. Yamada, "What's so different about blockchain? blockchain is a probabilistic state machine," in *Proc. IEEE 36th Int. Conf. Distrib. Comput. Syst. Workshops*, Jun. 2016, pp. 168–175.

[24] I. Eyal, A. E. Gencer, E. G. Sirer, and R. van Renesse, "Bitcoin-ng: A scalable blockchain protocol," *CoRR*, vol. abs/1510.02037, 2015. [Online]. Available: http://arxiv.org/abs/1510.02037

[25] Y. Lewenberg, Y. Sompolinsky, and A. Zohar, "Inclusive block chain protocols," in *Proc. FC*, 2015, pp. 528–547.

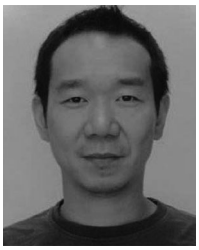
[26] A. Miller, Y. Xia, K. Croman, E. Shi, and D. Song, "The honey badger of bft protocols," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 31–42.

[27] M. Milutinovic, W. He, H. Wu, and M. Kanwal, "Proof of luck: An efficient blockchain consensus protocol," in *Proc. 1st Workshop Syst. Softw. Trusted Execution*, 2016, pp. 2:1–2:6. [Online]. Available: http://doi.acm.org/10.1145/3007788.3007790
[28] H. Turesson, A. Roatis, H. Kim, and M. Laskowski, "Deep learning models as proof-of-useful work: A smarter, utilitarian scheme for achieving consensus on a blockchain," *Soc. Sci. Res. Netw.*, Jul. 2018. Available at SSRN: https://ssrn.com/abstract=3206258.
[29] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, and P. Saxena, "A secure sharding protocol for open blockchains," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 17–30. [Online]. Available: http://doi.acm.org/10.1145/2976749.2978389
[30] 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 Big Data (BigData Congr.)*, Jun. 2017, pp. 557–564.
[31] L. M. Bach, B. Mihaljevic, and M. Zagar, "Comparative analysis of blockchain consensus algorithms," in *Proc. 41st Int. Conv. Inf. Commun. Technol. Electron. Microelectronics*, May 2018, pp. 1545–1550.
[32] M. Swan, *Blockchain: Blueprint for a New Economy*, 1st ed. Sebastopol, CA, USA: O'Reilly Media, Inc., 2015.
[33] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system, 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf
[34] Bitcoin developer documentation, Bitcoin community, Dec. 2017. [Online]. Available: https://bitcoin.org/en/developer-documentation
[35] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for iot security and privacy: The case study of a smart home," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops*, Mar. 2017, pp. 618–623.
[36] R. Beck, J. S. Czepluch, N. Lollike, and S. Malone, "Blockchain-the gateway to trust-free cryptographic transactions," in *Proc. Eur. Conf. Inf. Syst.*, 2016, Art. no. ResearchPaper153.
[37] S. King and S. Nadal, "Ppcoin: Peer-to-peer crypto-currency with proof-of-stake," *Self-Published Paper*, vol. 19, 2012. pp. 1–6.
[38] M. Castro, B. Liskov, et al., "Practical byzantine fault tolerance," in *Proc. 3rd Symp. Operating Syst. Des. Implementation*, 1999, vol. 99, pp. 173–186.
[39] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum Project Yellow Paper*, vol. 151, 2014. pp. 1–32.
[40] K. J. O'Dwyer and D. Malone, "Bitcoin mining and its energy footprint," in *Proc. 25th IET Irish Signals Syst. Conf/China-Ireland Int. Conf. Inf. Commun. Technol.*, Jun. 2014, pp. 280–285.
[41] M. Vukolić, "The quest for scalable blockchain fabric: Proof-of-work vs. bft replication," in *Proc. Int. Workshop Open Problems Netw. Secur.*, 2015, pp. 112–125.
[42] K. Wang, H. Li, Y. Feng, and G. Tian, "Big data analytics for system stability evaluation strategy in the energy internet," *IEEE Trans. Ind. Informat.*, vol. 13, no. 4, pp. 1969–1978, Aug. 2017.
[43] K. Wang, Y. Shao, L. Shu, C. Zhu, and Y. Zhang, "Mobile big data fault-tolerant processing for ehealth networks," *IEEE Netw.*, vol. 30, no. 1, pp. 36–42, Jan. 2016.
[44] Ivanzar, Bitcoin protocol, Bitcoin Wiki, Aug. 2017. [Online]. Available: https://en.bitcoin.it/wiki/Protocol_rules
[45] G. O. Karame, E. Androulaki, and S. Capkun, "Double-spending fast payments in bitcoin," in *Proc. ACM Conf. Comput. Commun. Secur.*, 2012, pp. 906–917. [Online]. Available: http://doi.acm.org/10.1145/2382196.2382292
[46] M. Ali, J. C. Nelson, R. Shea, and M. J. Freedman, "Blockstack: A global naming and storage system secured by blockchains." in *Proc. USENIX Annu. Tech. Conf.*, 2016, pp. 181–194.
[47] P. Koshy, D. Koshy, and P. McDaniel, "An analysis of anonymity in bitcoin using p2p network traffic," in *Financial Cryptography and Data Security*, N. Christin and R. Safavi-Naini, Eds. Berlin, Germany: Springer, 2014, pp. 469–485.
[48] K. Delmolino, M. Arnett, A. Kosba, A. Miller, and E. Shi, "Step by step towards creating a safe smart contract: Lessons and insights from a cryptocurrency lab," in *Financial Cryptography and Data Security*, J. Clark, S. Meiklejohn, P. Y. Ryan, D. Wallach, M. Brenner, and K. Rohloff, Eds. Berlin, Germany: Springer, 2016, pp. 79–94.
[49] Y. Yuan and F. Y. Wang, "Towards blockchain-based intelligent transportation systems," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst.*, Nov. 2016, pp. 2663–2668.
[50] RaspberryPi, Raspberry pi model b, 2015. [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-1-model-b/
[51] pickle: Python object serialization, Python.org, Jun. 2017. [Online]. Available: https://docs.python.org/3/library/pickle.html
[52] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computinga key technology towards 5g," *ETSI White Paper*, vol. 11, no. 11, pp. 1–16, 2015.
[53] X. He, K. Wang, H. Huang, T. Miyazaki, Y. Wang, and S. Guo, "Green resource allocation based on deep reinforcement learning in content-centric iot," *IEEE Trans. Emerging Top. Comput.*, 2018. doi: 10.1109/TETC.2018.2805718.
[54] X. He, K. Wang, H. Huang, and B. Liu, "Qoe-driven big data architecture for smart city," *IEEE Commun. Mag.*, vol. 56, no. 2, pp. 88–93, Feb. 2018.
[55] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the security and performance of proof of work blockchains," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 3–16. [Online]. Available: http://doi.acm.org/10.1145/2976749.2978341

Chenhan Xu is currently a research assistant with the Key Laboratory of Broadband Wireless Communication and Sensor Network Technology, Nanjing University of Posts and Telecommunications, China. His current research interests include big data, cloud computing, and machine learning.

Kun Wang (M'13-SM'17) received the BEng and PhD degrees in computer science from Nanjing University of Posts and Telecommunications, Nanjing, China, in 2004 and 2009, respectively. From 2013 to 2015, he was a postdoc fellow with Electrical Engineering Department, University of California, Los Angeles (UCLA), California. In 2016, he was a research fellow with the School of Computer Science and Engineering, the University of Aizu, Aizu-Wakamatsu City, Fukushima, Japan. He is currently a research fellow with the Department of Computing, the Hong Kong Polytechnic University, Hong Kong, China, and also a full professor with the School of Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing, China. His current research interests include big data, wireless communications and networking, smart grid, energy Internet, and information security technologies. He has published more than 100 papers in referred international conferences and journals. He has received Best Paper Award at IEEE GLOBECOM16. He serves as an associate editor of IEEE Access, editor of the *Journal of Network and Computer Applications*, the *Journal of Communications and Information Networks*, the *EAI Transactions on Industrial Networks and Intelligent Systems* and guest editors of IEEE Access, Future Generation Computer Systems, Peer-to-Peer Networking and Applications, and the *Journal of Internet Technology*. He was the symposium chair/co-chair of IEEE IECON16, IEEE EEEIC16, IEEE WCSP16, IEEE CNCC17, etc. He is a senior member of the IEEE and member of ACM.

Peng Li (M'12) received the BS degree from Huazhong University of Science and Technology, China, in 2007, the MS and PhD degrees from the University of Aizu, Japan, in 2009 and 2012, respectively. He is currently an associate professor with the University of Aizu, Japan. His research interests include wireless communication and networking, specifically wireless sensor networks, green and energy-efficient mobile networks, and cross-layer optimization for wireless networks. He also has interests on cloud computing, big data processing and smart grid. He is a member of the IEEE.



Song Guo (M'02-SM'11) received the PhD degree in computer science from the University of Ottawa and was a professor with the University of Aizu. He is a full professor with the Department of Computing, The Hong Kong Polytechnic University. His research interests include big data, cloud computing and networking, and distributed systems with more than 400 papers published in major conferences and journals. His work was recognized by the 2016 Annual Best of Computing: Notable Books and Articles in Computing in ACM Computing Reviews. He is the recipient of the 2017 IEEE Systems Journal Annual Best Paper Award and other five Best Paper Awards from IEEE/ACM conferences. He was an associate editor of the *IEEE Transactions on Parallel and Distributed Systems* and an IEEE ComSoc distinguished lecturer. He is now on the editorial board of the *IEEE Transactions on Emerging Topics in Computing*, the *IEEE Transactions on Sustainable Computing*, the *IEEE Transactions on Green Communications and Networking*, and the *IEEE Communications*. He also served as general, TPC and symposium chair for numerous IEEE conferences. He currently serves as an officer for several IEEE ComSoc Technical Committees and a director in the ComSoc Board of Governors. He is a senior member of the IEEE.

Jiangtao Luo received the PhD degree from Shanghai Institute of Optics and Fine Mechanics, Chinese Academe of Sciences. He is currently a full professor with Electronic Information and Networking Research Institute, Chongqing University of Posts and Telecommunications. He is a senior member of the IEEE.

Baoliu Ye received the PhD degree in computer science from Nanjing University, China, in 2004, and was a professor with Nanjing University from 2014. He is a full professor and the dean of the School of Computer and Infomation, Hohai University. His current research interests mainly include distributed systems, cloud computing, wireless networks with more than 70 papers published in major conferences and journals. He served as the TPC co-chair of HotPOST'12, HotPOST'11, P2PNet'10. He is the regent of CCF, the Secretary-General of CCF Technical Committee of Distributed Computing and Systems, and a member of the IEEE.

Minyi Guo (F'17) received the PhD degree in computer science from the University of Tsukuba, Tsukuba, Japan. He is currently a Zhiyuan chair professor with Shanghai Jiao Tong University, Shanghai, China. His research interests include pervasive computing, parallel and distributed processing, and parallelizing compilers. In 2007, he received the Recruitment Program of Global Experts and Distinguished Young Scholars Award from the National Natural Science Foundation of China. He is on the editorial board of the *IEEE Transactions on Parallel and Distributed Systems* and the *IEEE Transactions on Computers*. He is a fellow of the IEEE.