Making Big Data Open in Collaborative Edges: A Blockchain-based Framework with Reduced Resource Requirements

Chenhan Xu*, Kun Wang*†, Guoliang Xu‡, Peng Li§, Song Guo†, and Jiangtao Luo‡
*Nanjing University of Posts and Telecommunications, Nanjing, China
Emails: xchank@outlook.com, kwang@njupt.edu.cn
†The Hong Kong Polytechnic University, Hong Kong, China
Email: song.guo@polyu.edu.hk
‡Chongqing University of Posts and Telecommunications, Chongqing, China
Emails: xugl@cqupt.edu.cn, luojt@cqupt.edu.cn
§The University of Aizu, Japan
Email: pengli@u-aizu.ac.jp

***Abstract*—With the emergence of edge computing in various applications domains, end users are now surrounded by a fast growing volume of data from edge devices belonging to different stakeholders. However, these edge devices cannot cooperate to share big data because of the distrust among them. In this paper, the blockchain is deployed in collaborative edges by exploiting the non-repudiation and non-tampering properties to enable trust. First, we develop a blockchain based big data sharing framework in collaborative edges for adapting to the limited computational and storage resources in edge devices. Then, a consensus mechanism called Proof-of-Collaboration (PoC) is proposed for computational resources reduction in our proposed framework, where edge devices offer their credits of PoC to compete for the block generation. Moreover, we put forward a futile transaction filter algorithm for transaction offloading, greatly reducing the storage resources occupied by the blockchain in edges. Extensive experiments are performed to demonstrate the superior performance of our proposal.**

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

I. 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]. Accordingly, edge computing emerges, since it may effectively decrease time latency via deploying services at the edge of network, such as mobile phones, surveillance cameras, and Internet of things (IoT) sensors [3], [4]. More and more users are surrounded by large scale edge devices and data belonging to different stakeholders [5], [6].

Unfortunately, these edge devices may not always cooperate with each other because they are in a distrusted environment [4]. In some cases, when one participant sharing its business data for others to read, some participants will deny that they have read them even though they are benefit from it. On the contrary, when one participant opens the access authority for others to write, some participants can maliciously tamper the 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.* [7] 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.* [8] investigated to calculate a set of credit scores to select a more reliable participant. However, the credit scores are only suggestions for edges and the malicious participant cannot be avoided. Our goal is to implement edge collaboration in distrusted environment.

To this end, we propose to deploy blockchain for big data sharing in collaborative edges. The blockchain is a public append-only ledger carrying all transactions that have been executed [9]. Every block carrying some transactions is committed to the global blockchain that every participant maintains, and thus, no one can reject to admit the transactions of data flow from edge applications that have been committed. Moreover, the blockchain can achieve global consensus on the whole sequence of transactions so that a conflicting transaction will be dropped once it is committed. Although non-repudiation and non-tampering properties of the blockchain is promising, there still exist some challenges as follows:

- Edge devices are heterogeneous in the aspects of computational resources. Therefore, some edge devices with limited computational resources cannot support both the operations of blockchain and big data.
- Edge devices have limited storage resources, which is hardened to store the whole ledger.

Different from traditional blockchain for edge computing [10], where the blockchain technology is employed without taking the limitation of resource into consideration, we design a green blockchain framework with reduced computational and storage resource requirements for big data sharing in

978-1-5386-3180-5/18/$31.00 ©2018 IEEE

Authorized licensed use limited to: University at Buffalo Libraries. Downloaded on January 25,2021 at 08:27:54 UTC from IEEE Xplore. Restrictions apply.

collaborative edges. This framework, as shown in Fig. 1, is divided into four different layers, i.e., Application Programming Interface (API) layer, cache layer, blockchain layer, and storage layer. The details of the framework design are illustrated in Section III. API layer and blockchain layer can directly access data from cache layer, rather than from storage layer, which reduces the response time and makes our system adapted for big data sharing. This paper mainly focuses on the design of blockchain layer in the proposed framework, especially in green consensus mechanism and transaction offloading module. 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 can support trust in collaborative edges as well as reducing the computational and storage resources.
- We put forward a green consensus mechanism in the collaborative edges called Proof-of-Collaboration (PoC). Based on PoC, edge devices compete for new blocks via showing their collaboration credits instead of paying a significant amount of computation to solve a mathematic puzzle, which greatly saves the computational resources.
- We propose a futile transaction theory with the proof. This theory shows the former transaction, whose outputs are all referenced by the latter, is useless for the validation of new generated transaction. Furthermore, we design a novel transaction offloading module based on Futile Transactions Filter (FTF) algorithm, which contributes to reduce the storage resources occupied by the blockchain.
- We perform extensive experiments on 16 RaspberryPi micro computers to demonstrate the high performance of our proposal. These experiment results show that PoC mechanism can reduce at most 90% computational resources than PoW mechanism. Additionally, more than 95% storage resources can be reduced by transaction offloading module.

The structure of the paper is organized as follows. Section II reviews the related works on edge collaboration and blockchain technology. The green blockchain framework in collaborative edges is designed in Section III. Section IV 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 V. Section VI performs extensive experiments to show the performance of our proposal.

II. Related Works

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

A. Edge collaboration

With the emergence of edge computing technology, the edge collaboration issues are taken into great consideration [11]. Shi *et al.* [5] surveyed the edge computing and addressed 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.* [12] explored to implement mobile edge computing collaboration in 5G ecosystem. Wu *et al.* [13] proposed a two-step detection mechanism in mobile edge collaboration, where users' preferences are concerned for constructing virtual communities and collaborative clusters.

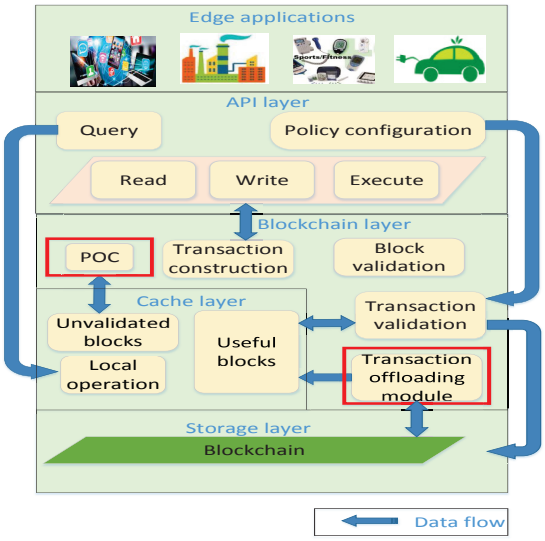


Fig. 1: Green blockchain framework in collaborative edges.

B. Blockchain technology

Blockchain technology has aroused great interests from both academic and industrial fields, including finance, e-health, distributed system, etc. Christidis *et al.* [14] presented a comprehensive survey on blockchain and addressed 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. Lewenberg *et al.* [15] designed a directed acyclic graph (DAG) structure for blockchain in order to enhance the throughout. Miller *et al.* [16] proposed a practical asynchronous HoneyBadgerBFT protocol, which can provide a high throughput.

III. Framework Design

In our proposal, we deploy blockchain on these edges, where every block contains multiple transaction logs of big data flow from 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 interface services for edge applications, and abstracts the functions of cache and blockchain layer to provide various calls for implementing collaboration. Specifically, API layer consists of different operations as follows:

- 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 other edge devices' operation record on local data, where the latest operations are stored

Authorized licensed use limited to: University at Buffalo Libraries. Downloaded on January 25,2021 at 08:27:54 UTC from IEEE Xplore. Restrictions apply.

in the local operation module in cache layer.

Cache layer is used for caching data in the system, containing cache for local operation, unvalidated blocks, and useful blocks.

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

- First, transaction construction module transforms the requests from the upper layer into transactions, which will be broadcast to the entire edge network for validation.
- Second, transaction validation module contains validation regulations, where the operation permission to local data is often set for other edge devices via modifying validation regulations. Besides, transaction and block validation modules guarantee rules, which are foundations of the green PoC consensus, as we will illustrate in Section IV.
- Finally, transaction offloading module first locates the blocks with useful transactions, 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.

IV. Green Poof-of-collaboration consensus mechanism

Blockchain is a distributed data structure where every participant keeps a entire copy [9]. The first class citizen in blockchain is 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 will be added to the blockchain.

A. Different consensus

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 [9]. 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 is:

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

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. At this time, the concept of Proof-of-Stake (PoS) [14] is proposed, and the main idea of PoS is that stakeholders should show their stake of assets to compete for 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 tolerates up to a third of participants to occur any form of failure (Byzantine fault) when the number of participants is known in advance and fixed.

Within the context of collaborative edges, as mentioned above, every edge device is a participant of the network, and may require to perform blockchain operation. Moreover, the number of edge devices, which should adapt to the demand of users, is not fixed. As we mentioned in Section I, 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 detailedly illustrate PoC in next subsection.

B. Proof-of-Collaboration mechanism

Edge devices give their proof of contributing collaboration rather than solving meaningless mathematical puzzle to obtain the privileges of collaboration. More specifically, the green PoC consensus mechanism is designed as follows.

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 [9] and ETH (GAS) in Ethereum [17]. 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 collaboration fee. This collaboration fee $\mathcal{F}$ is dynamically determined by $\mathcal{F} = \frac{\psi'}{\psi \times n}$ $\mathcal{CC}$/kB, 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. According to the definition of $\mathcal{F}$, the framework will decrease $\mathcal{F}$ to encourage collaboration when the recent throughput is lower than pre-defined, 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.

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:*

$$\begin{aligned} &\textit{Find } n \\ &\quad \textit{s.t.} \quad \textit{SHA256(SHA256}(h.n)) < \mathcal{CC} \times \mathcal{P} \times \textit{target} \end{aligned} \tag{2}$$

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*

Authorized licensed use limited to: University at Buffalo Libraries. Downloaded on January 25,2021 at 08:27:54 UTC from IEEE Xplore. Restrictions apply.

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 $\mathcal{CC} \times \mathcal{P} \times 0.001\%$ 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 $\mathcal{L} = \frac{n}{\Theta}$ and $\mathcal{R} = 3\mathcal{L}$. A higher Θ makes more intense competition.*

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. [9] gives the expectation of the needed computational resources in PoW, which is $\mathbb{E}_{\text{PoW}} = \frac{target_{\max}}{target} \times 2^{32}$. However, according to **Rule 1**, the expectation in PoC is

$$\mathbb{E}_{\text{PoC}} = \frac{target_{\max}}{\mathcal{CC} \times \mathcal{P} \times target} \times 2^{32} = \frac{1}{\mathcal{CC} \times \mathcal{P}} \mathbb{E}_{\text{PoW}}. \quad (3)$$

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 [9]. 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 [9]. The forking wastes enormous computational resources. Since not all the edge devices in PoC compete at the same time, the forking rarely happens.

V. Transaction Offloading

In traditional blockchain, the historical blocks are stored in every node. As we mentioned in previous section, with 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 [18], [19]. Moreover, a new participant is expected to download these blocks before joining the blockchain network, if he intends to validate the new generated transactions [9]. Within the edge context, this download operation costs enormous network resources, which makes edge collaboration inefficient. In this section, we first glance at how the transactions are organized. Then, the proposed transaction filtering theory is illustrated in details.

A. Transaction organization

In the blockchain, every transaction references one or more previous transactions to support its validity. 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 [9]. 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 [9].*

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.

B. Transaction filtering theory

As illustrated in Section V-A, the *outputs* of valid transactions are referenced by later ones. Based on the approach of 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: For one thing, 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. For another, 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** proves right. ■

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, and finds 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.

After FTF finishes the filtering of futile transactions, the transaction offloading module locates the blocks that carry useful transactions, updates them to the cache layer. The futile blocks, i.e., the blocks only carry futile transactions, will be sent to stakeholders' 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

Authorized licensed use limited to: University at Buffalo Libraries. Downloaded on January 25,2021 at 08:27:54 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1 Futile Transactions Filter

```
procedure FUTILE-TRANS-FILTER(B)
    for all t ∈ B.transactions do
        for all i ∈ t.inputs do
            MarkAsReferenced(i.txid, i.index)
        end for
    end for
    for all t ∈ B.transactions do
        MarkAsFutile(t)
        for all o ∈ t.outputs do
            if IsMarked(o)==false then
                MarkAsUseful(t)
                break
            end if
        end for
    end for
end procedure
```

does not increase the risk of being attacked by "51% 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, which makes devices carry more edge applications, i.e., efficient and green.

VI. Experiment

In this section, we first present the environment of our experiment, and then analyze the experiment results.

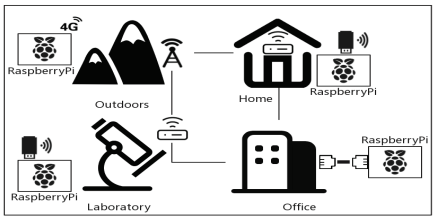


Fig. 2: Experiment platform.

A. Experiment setup

Our experiment is conducted on the RaspberryPi 2 model B which equips with a 900MHz quad-core ARM Cortex-A7 CPU, in Raspbian operation system [20]. Fig. 2 depicts our experiment platform. In the experiment, there are 16 RaspberryPi micro computers, which are deployed in our homes, laboratories, outdoors, and offices. These RaspberryPi micro computers access networks in wired and wireless ways.

We develop our proposed green blockchain using Python language 3.6. The implementation is multi-processed and we wrote customized codes for serialization and unserialization.

TABLE I: 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.

The settings of experiments are shown in TABLE I. To involve the factor of heterogeneous in computational resources, the experiments are divided into group A, B, and C. Different groups have distinct computational resources limitation. For example, TABLE I 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 5.25V. We also compare the performance of PoC with that of PoW, which is used in Bitcoin and Ethereum.

The data used for experiment is randomly generated. If it is not specified, every transaction has five inputs and outputs, while the number of transaction per block $\tau \sim U(50, 1000)$.

B. Experiment results

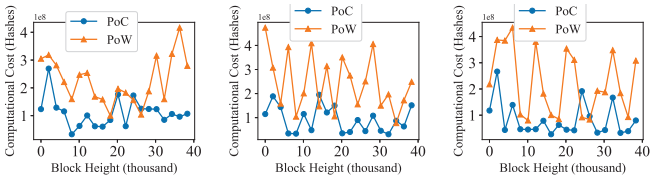
The results of computational resources cost comparison are shown in Fig. 3. The blockchain height refers to the amount of blocks in the blockchain. The computational cost is the number of hash operation that a block proposer tried, and is evaluated by hashes. Constrained by **Rule 1**, our green blockchain with PoC keeps mining easily in different groups, which reduces at most 90% computational resources for a single miner.

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 Fig. 4 and Fig. 5, respectively. This metric is defined as Block Proposer Hashes (BPH), which is calculated by 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. Fig. 4 and Fig. 5 show that our proposed PoC does reduce quite a lot of computational resources and energy cost on the edge network scale. We further demonstrate the accumulative energy cost and computational resource usage of group A in Fig. 6 and Fig. 7 (the curves of group B and C are similar to that of group A). With the proposed framework running, the superiorities on energy and computational resources reduction of PoC become more and more obvious.

In Fig. 8, we demonstrate the storage costs of PoW and PoC. We set that the transaction offloading module runs one time when every 200 blocks are generated. The results show that 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.

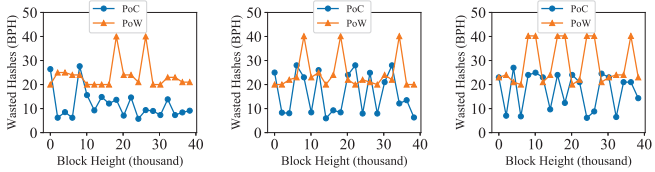
In summary, the results of experiments show that our proposed green blockchain is able to reduce enormous computational and storage resources, which helps to solve the critical challenges we describe in Section I.

Authorized licensed use limited to: University at Buffalo Libraries. Downloaded on January 25,2021 at 08:27:54 UTC from IEEE Xplore. Restrictions apply.



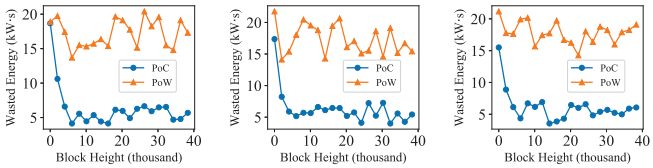
(a) group A (b) group B (c) group C

Fig. 3: Comparison of computational cost.



(a) group A (b) group B (c) group C

Fig. 4: Comparison of wasted hashes.



(a) group A (b) group B (c) group C

Fig. 5: Comparison of wasted energy.

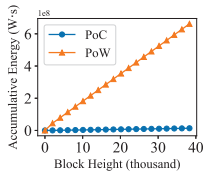


Fig. 6: Accumulative energy.

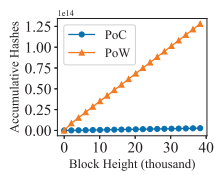


Fig. 7: Accumulative hashes.

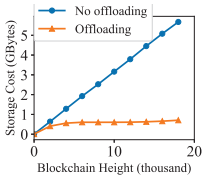


Fig. 8: Comparison of storage cost.

VII. Conclusion and Future Works

This paper studies the distrust issues and employs the blockchain to enable trust big data sharing in edge collaboration. We construct a green blockchain framework. For one thing, 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 mathematic puzzle for the privileges of collaboration. For another, we propose the futile transaction theory and design a transaction offloading module based on FTF algorithm in our framework to reduce storage resources occupied by the blockchain. 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 a open issue. We are motivated to complete the whole framework design and further improve the performance of our proposal in the future.

Acknowledgment

This work is supported by NSFC (61572262, 61772286), China Postdoctoral Science Foundation (2017M610252), China Postdoctoral Science Special Foundation (2017T100297), JSPS KAKENHI (16K16038), and Strategic Information and Communications R&D Promotion Programme (SCOPE No.162302008), MIC, Japan.

References

[1] 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

[2] X. Zhou, K. Wang, W. Jia, and M. Guo, "Reinforcement learning-based adaptive resource management of differentiated services in geo-distributed data centers," in *2017 IEEE/ACM 25th International Symposium on Quality of Service (IWQoS)*, June 2017, pp. 1–6.

[3] 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.

[4] M. Du, K. Wang, X. Liu, S. Guo, and Y. Zhang, "A differential privacy-based query model for sustainable fog data centers," *IEEE Transactions on Sustainable Computing*, vol. PP, no. 99, pp. 1–1, 2017.

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

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

[7] M. Hussain and B. M. Almourad, "Trust in mobile cloud computing with lte-based deployment," in *2014 IEEE 14th Intl Conf on Scalable Computing and Communications and Its Associated Workshops*, Dec 2014, pp. 643–648.

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

[9] (2017, Dec.) Bitcoin developer documentation. Bitcoin community. [Online]. Available: https://bitcoin.org/en/developer-documentation

[10] A. Stanciu, "Blockchain based distributed control system for edge computing," in *2017 21st International Conference on Control Systems and Computer Science (CSCS)*, May 2017, pp. 667–671.

[11] 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 Communications*, vol. 24, no. 2, pp. 58–64, April 2017.

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

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

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

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

[16] A. Miller, Y. Xia, K. Croman, E. Shi, and D. Song, "The honey badger of bft protocols," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016.

[17] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum Project Yellow Paper*, vol. 151, 2014.

[18] K. Wang, H. Li, Y. Feng, and G. Tian, "Big data analytics for system stability evaluation strategy in the energy internet," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 1969–1978, Aug 2017.

[19] K. Wang, Y. Shao, L. Shu, C. Zhu, and Y. Zhang, "Mobile big data fault-tolerant processing for ehealth networks," *IEEE Network*, vol. 30, no. 1, pp. 36–42, January 2016.

[20] RaspberryPi. (2015) Raspberry pi model b. [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-1-model-b/

Authorized licensed use limited to: University at Buffalo Libraries. Downloaded on January 25,2021 at 08:27:54 UTC from IEEE Xplore. Restrictions apply.