MAKING BIG DATA OPEN IN EDGES: A RESOURCE-EFFICIENT BLOCKCHAIN-BASED APPROACH

Seminar Report

Submitted in partial fulfillment of the requirements for
the award of degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

of

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

Submitted By

AKHIL MURALI

Department of Computer Science & Engineering
Mar Athanasius College Of Engineering
Kothamangalam

MAKING BIG DATA OPEN IN EDGES: A RESOURCE-EFFICIENT BLOCKCHAIN-BASED APPROACH

Seminar Report

Submitted in partial fulfillment of the requirements for the award of degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

of

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

Submitted By

AKHIL MURALI

Department of Computer Science & Engineering
Mar Athanasius College Of Engineering
Kothamangalam

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
MAR ATHANASIUS COLLEGE OF ENGINEERING
KOTHAMANGALAM

CERTIFICATE

This is to certify that the seminar report entitled **Making Big Data Open in Edges : A Resource - Efficient Blockchain - based Approach** *submitted by* **Mr. AKHIL MURALI, Reg. No. MAC15CS007** *towards partial fulfillment of the requirement for the award of Degree of Bachelor of Technology in Computer science and Engineering from APJ Abdul Kalam Technological University for December 2018 is a bonafide record of the seminar carried out by him under our supervision and guidance.*

....	....	....
Prof. Joby George	**Prof. Neethu Subash**	**Dr. Surekha Mariam Varghese**
Faculty Guide	***Faculty Guide***	***Head Of Department***

Date:

Dept. Seal

ACKNOWLEDGEMENT

First and foremost, I sincerely thank the 'God Almighty' for his grace for the successful and timely completion of the seminar.

I express my sincere gratitude and thanks to Dr. Solly George, Principal and Dr. Surekha Mariam Varghese, Head Of the Department for providing the necessary facilities and their encouragement and support.

I owe special thanks to the staff-in-charge Prof. Joby george, Prof. Neethu Subash and Prof. Joby Anu Mathew for their corrections, suggestions and sincere efforts to co-ordinate the seminar under a tight schedule.

I express my sincere thanks to staff members in the Department of Computer Science and Engineering who have taken sincere efforts in helping me to conduct this seminar.

Finally, I would like to acknowledge the heartfelt efforts, comments, criticisms, co-operation and tremendous support given to me by my dear friends during the preparation of the seminar and also during the presentation without whose support this work would have been all the more difficult to accomplish.

ABSTRACT

Emergence of edge computing has witnessed a fast growing volume of data on edge devices belonging to different stakeholders. This data cannot be shared among them due to lack of trust. By exploiting blockchain's non-repudiation and non-tampering properties a blockchain-based big data sharing framework to support various applications across resource-limited edges were developed . Number of novel resource-efficient techniques for the framework is devised. Proof-of-Collaboration based consensus mechanism with low computation complexity is beneficial to the edge devices. Blockchain transaction filtering and offloading scheme can reduce storage overhead. This paper introduces new types of blockchain transaction and block to enhance the communication efficiency. Extensive experiments are conducted and the results demonstrate the superior performance of the proposed system.

Contents

List of Figures

List Of Abbrevations

CC	Collaboration Credit
E-TX	Express Transaction
FTF	Futile Transaction Filtering
HB	Hollow Block
IoT	Internet of Things
PBFT	Practical Byzantine Fault Tolerance
PoC	Proof of Collaboration
PoS	Proof of Stake
PoW	Proof of Work

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. Accordingly, edge computing emerges, effectively decreasing time latency via deploying services at the edge of network, such as mobile phones, surveillance cameras, and Internet of things (IoT) sensors. More and more users are surrounded by edge devices and data belonging to different stakeholders .

Unfortunately, these edge devices may not always co- operate with each other because they are in a distrusted environment. 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.

The distrust issues of big data sharing in collaborative edges are serious threat. A few previous works have investigated how to solve the distrust issues. One strategy is 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. Another strategy is 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. For solving the trust issues a Blockchain based strategy can be . The blockchain is a public append-only ledger carrying all transactions that have been executed. Every block carrying some trans- actions 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. The blockchain framework can prevent malicious double spending attack. This attack allows the malicious participant to deny that they have benefited from the collaboration.

Although non-repudiation and non-tampering proper- ties 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.

Since the blockchain Different from existing blockchain for edge computing , where the blockchain technology is employed without taking the limitation of resources into consideration, A green blockchain framework is designed with reduced computational, storage, and network resource requirements for big data sharing in collaborative edges. This framework, as shown in Fig. 1.1, is divided into four layers, i.e., Application Programming Interface (API) layer, cache layer, blockchain layer, and storage layer. The 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, transaction offloading, and transaction construction.

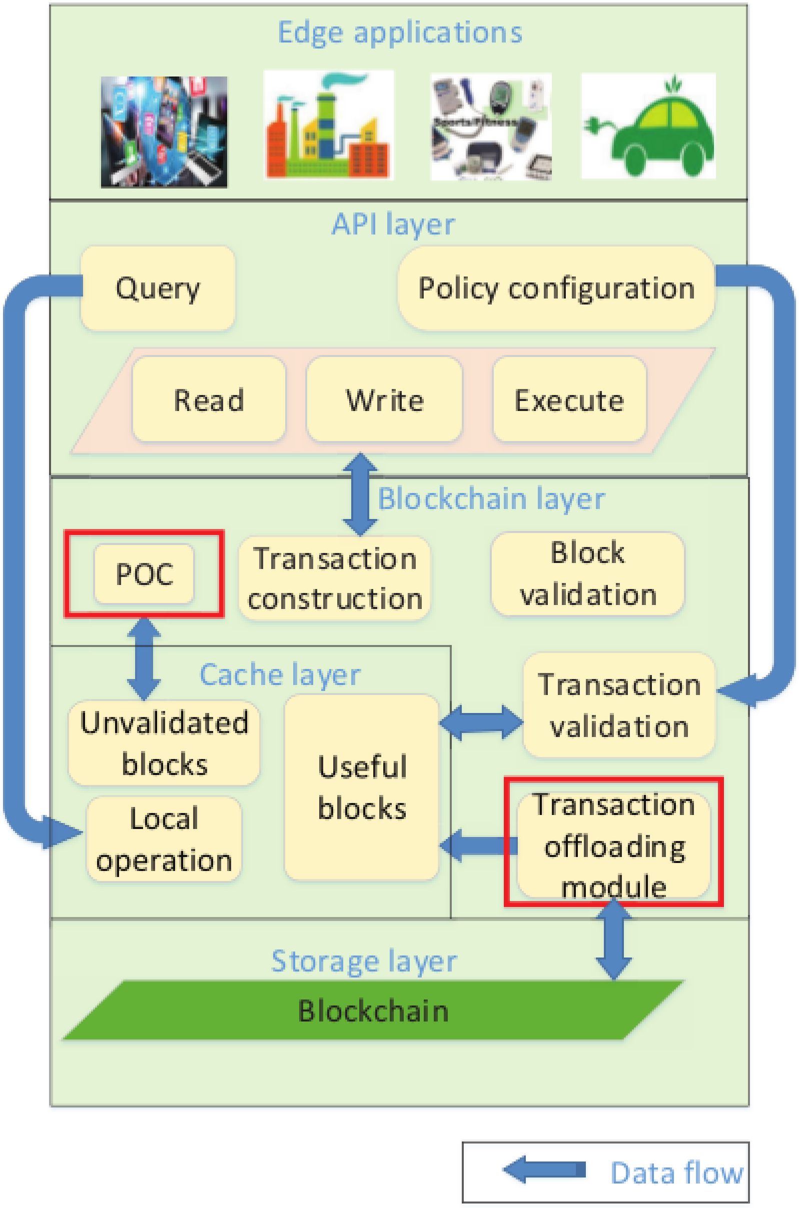


Figure 1.1: Green blockchain framework in collaborative edges

The strategies put froward are:

- Developed 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 con- sensus mechanism in the collaborative edges called

Proof-of-Collaboration (PoC). Based on this PoC con- sensus mechanism, edge devices compete for new blocks generation via showing their collaboration credits instead of paying a significant amount of computation to solve a mathematic puzzle, which greatly saves the computational resources in edge devices.

- Formulated 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 new generated trans- action. A novel transaction offloading module based on Futile Transactions Filter (FTF) algorithm is designed, which contributes to reduce the storage resources occupied by the blockchain.

- Proposes Express Transactions (E-TX) and Hollow Blocks to enhance the network efficiency of the pro- posed 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.

Related works

2.1 Edge collaboration

With the emergence of edge computing technology, the edge collaboration issues are taken into great consideration. [1] 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. 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. [2] 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. [3] 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 en- ergy consumption at the edge of mobile networks. [4] proposed to offload the mining task to the edge computing service providers, who make profit by providing computation resource. In the proposed scenario, Stackelberg game is used to optimize the price of the resource.

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. [5] 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. [6] 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. [7] 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.

Different from these previous works, The paper propose a a green blockchain framework to enable trust for big data sharing in collaborative edges. Then, put forward green PoC consensus mechanism in our framework to reduce com- putational 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, proposed a futile transaction theory and establish transaction offloading module based on FTF algorithm for reduction of storage resources occupied by blockchain. Finally, designd Express Transaction and Hollow Block to reduce the usage of resource in blockchain network.

Proposed work

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 this proposal, blockchain is deployed on these edges, where every block contains multiple transaction logs of big data flows among edge applications. For a more clear description of the proposal, it demonstrate a green blockchain framework in collaborative edges in this section. Proposed framework is divided into four layers, as shown in Fig. 1.1 .

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 transac- tion 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 opera- tions 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, unvalidated blocks, and useful blocks.

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

- 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.
- Transaction validation module contains val- idation 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.

- 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.

3.1 Green consensus mechanism

Blockchain is a distributed data structure and every participant keeps a copy of the entire blockchain. The first class component in blockchain is named transaction, which is a record of some asset transferring. These trans- actions 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.

3.1.1 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. If every participant can take part in blockchain operations, for example, competing to mine blocks or proposing transactions, the blockchain is public. Since the public chain is open and competitive, the participants in public chain network do not trust each other. On the contrary, the participants of a consortium chain network are privileged and white-listed.

The differences between two chain types result in differ- ent 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. 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:
Find n,

$$s.tSHA256(SHA256(h.n)) < target \tag{Equ. 3.1}$$

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

addition, Practical Byzantine Fault Tolerant (PBFT) and its variants are widely used in consortium chains, which tolerates up to a third of participants that occur any form of failure (Byzantine fault), given the number of participants 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 operations. Moreover, the number of edge devices, which should adapt to the demand of users, is not fixed. 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,.

3.1.2 Proof-of-Collaboration mechanism

Edge devices give their proof of contributing 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.

Collaboration credit

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

$$F = \frac{\varphi'}{\varphi * n} CC/kB \qquad \text{(Equ. 3.2)}$$

where φ is a predefined throughput threshold, φ' 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 φ' 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, φ equals to the maximum value of devices' network capacity. According to the definition of F , the framework will decrease F to encourage collaboration when the recent throughput is lower than predefined threshold, or increase F to reduce network overload when the throughput is higher than defined. Moreover, the larger the amount of edge devices is, the lower F will be in the framework.

In the framework, CC can be gained by two approaches. First, the block proposer can be rewarded a certain number of CC by adding a new block to the blockchain successfully. Second, the block proposer earns CC from the transactions carried by the block. The collaboration fee 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.

Proof-of-Collaboration

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

Rule 1 (Dynamic difficulty): The mining in the proposed PoC is different from Eq. (3.1). Mining in PoC is influenced by dynamic difficulty, which is different from various participants. It has the form as follows:

Find n,

$$SHA256(SHA256(h.n) < CC * P * target \qquad \text{(Equ. 3.3)}$$

where the target is the same as that in Eq. (3.1).

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 CC of the proposer and gives the same amount of CC as return, i.e., the payment changes CC of the proposer, but proposers do not lose CC . In addition, the new block pay the proposer extra CC*P *0.001 $\%$ as reward. According to the definition of P , the 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 P $\in$ [L, R] , where L is calculated by

$$L = \frac{n}{\theta} \qquad \text{(Equ. 3.4)}$$

and R = 3L . The value of θ can vary according to the 3.4) makes more in-tense 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 as explained in [9], of the needed computational resources is quite lower than that in PoW. gives the expectation of the needed computational resources E in PoW, which is

  $$E_{PoW} = \frac{Target_{max}}{target} 2^{32} \qquad \text{(Equ. 3.5)}$$

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

  $$E_{PoC} = \frac{Target_{max}}{CC * P * target} 2^{32} \quad = \frac{1}{CC * P} E_{PoW} \qquad \text{(Equ. 3.6)}$$

  $Target_{max}$ is the maximum target hash value. Constrained by Rule 2, only the winner of competition should clear its P . If an edge device fails in competing to propose a block, its 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 .

- For the whole edge network, Rule 3 stipulates that the block proposer should wait 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 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. The forking wastes enormous computational resources. Since not all the edge devices in PoC compete at the same time, the forking rarely happens.

3.2 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 . Moreover, a new participant is expected to download these blocks before joining the blockchain network, if he intends to validate the new generated transactions. 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.

3.2.1 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. 3.1 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 . 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 out- put that has already been spent, the transaction containing this input is invalid, i.e., double-spending .

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.

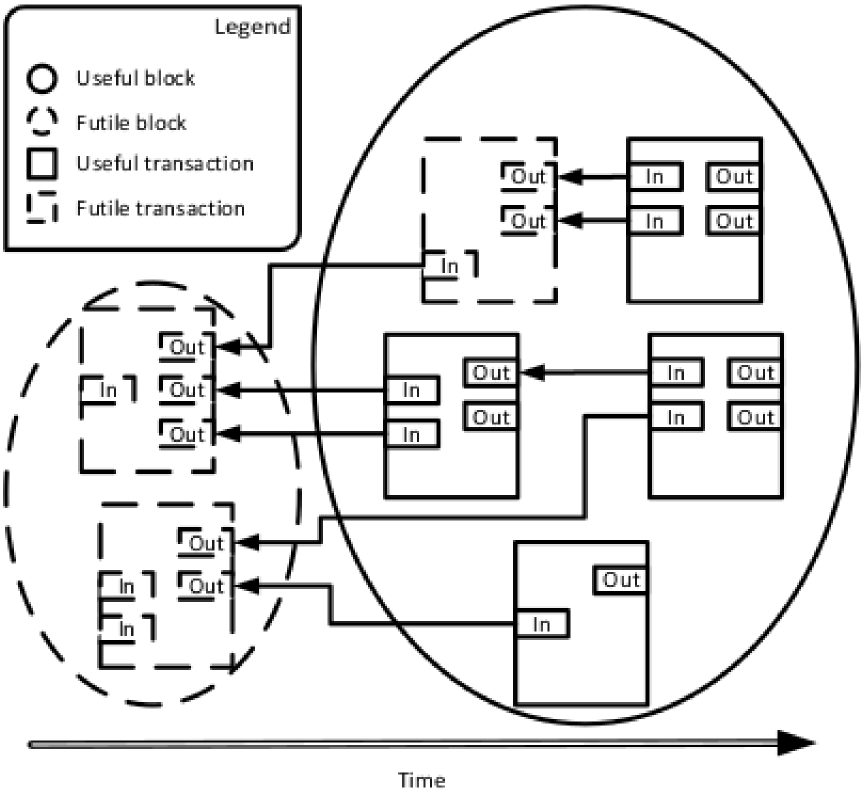


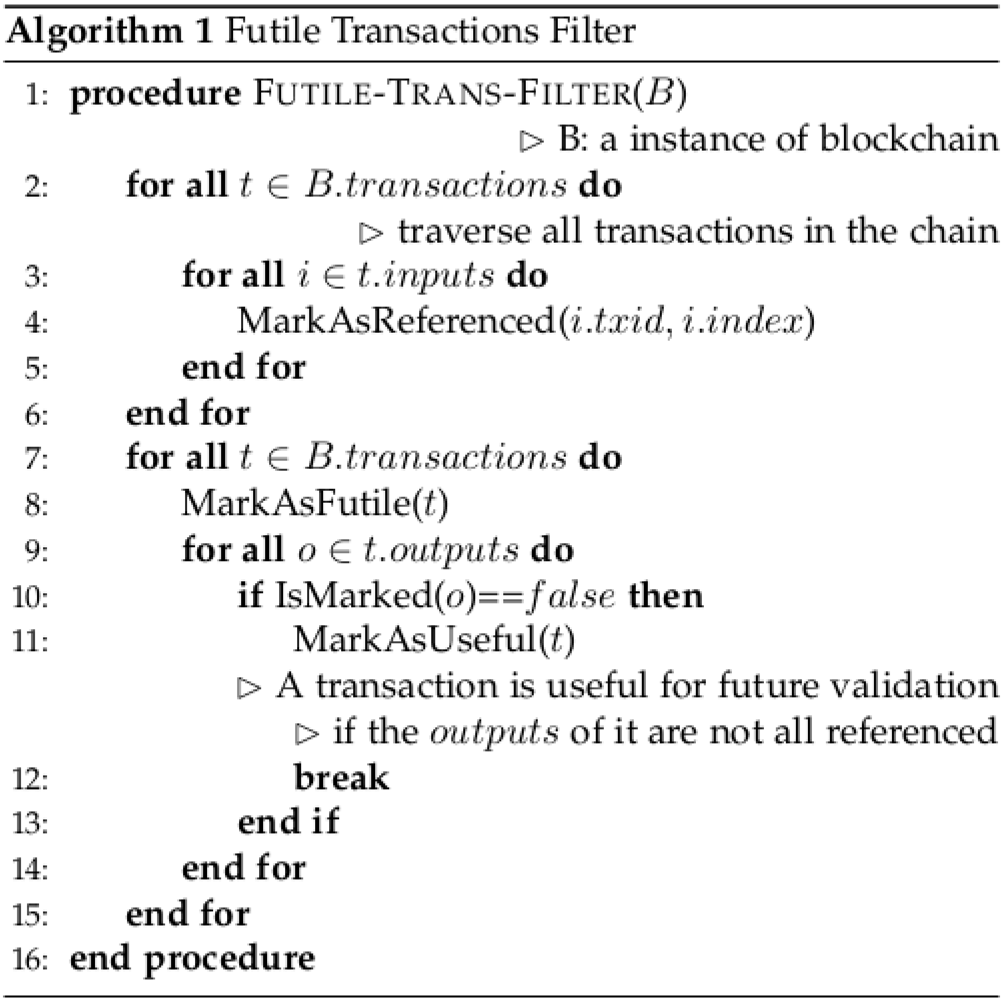
Figure 3.1: References among transactions.

3.2.2 Transaction filtering theory

Theorem 1 (Futile transaction): The transaction whose out- puts are all referenced by the latter transactions is useless for the validation of the new generated transaction.

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.

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 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 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, so that devices can more edge applications, making them efficient and green.



Algorithm 1 Futile Transactions Filter

```
procedure FUTILE-TRANS-FILTER($B$)
                                        ▷ B: a instance of blockchain
    for all $t \in B.transactions$ do
                                ▷ traverse all transactions in the chain
        for all $i \in t.inputs$ do
            MarkAsReferenced($i.txid$, $i.index$)
        end for
    end for
    for all $t \in B.transactions$ do
        MarkAsFutile($t$)
        for all $o \in t.outputs$ do
            if IsMarked($o$)==$false$ then
                MarkAsUseful($t$)
            ▷ A transaction is useful for future validation
                ▷ if the $outputs$ of it are not all referenced
                break
            end if
        end for
    end for
end procedure
```

3.3 Blockchain network optimization

Investigation is carried out about the network utilization pattern of the proposed PoC. Based on the pattern, the existing design of transactions and blocks are changed to enhance the network efficiency of the proposed green blockchain framework.

3.3.1 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 trans- action need to be propagated over the entire blockchain net- work 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. From blockchain participant's perspective, transactions and blocks come at different time. Moreover, the propagation of transactions and blocks are later than the validations of them, which means when the participant is perform- ing validation, its network is idle. For transactions or blocks, as illustrated above, the propagation 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

$$T_{max} = \sum_{i \in \Gamma_{max} \wedge V} T_i^v + \sum_{j \in \Gamma_{max} \wedge E} T_j^e \qquad \text{(Equ. 3.7)}$$

where Γ 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. (3.7) indicates that the propagation of transactions or blocks are synchronous.

Considering 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 transaction queue of participant i , according to the propagation of transactions,

$$| Q_1 \wedge Q_2 \wedge ... \wedge Q_i | \approx | Q_T | \qquad \text{(Equ. 3.8)}$$

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.

3.3.2 Express transaction

Based on the CC Express Transaction (E-TX) is proposed, which supports asynchronous transaction validation. Once a participant receives an E-TX, the participant first propagates it, and then performs the validation. It is stipulated that 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.

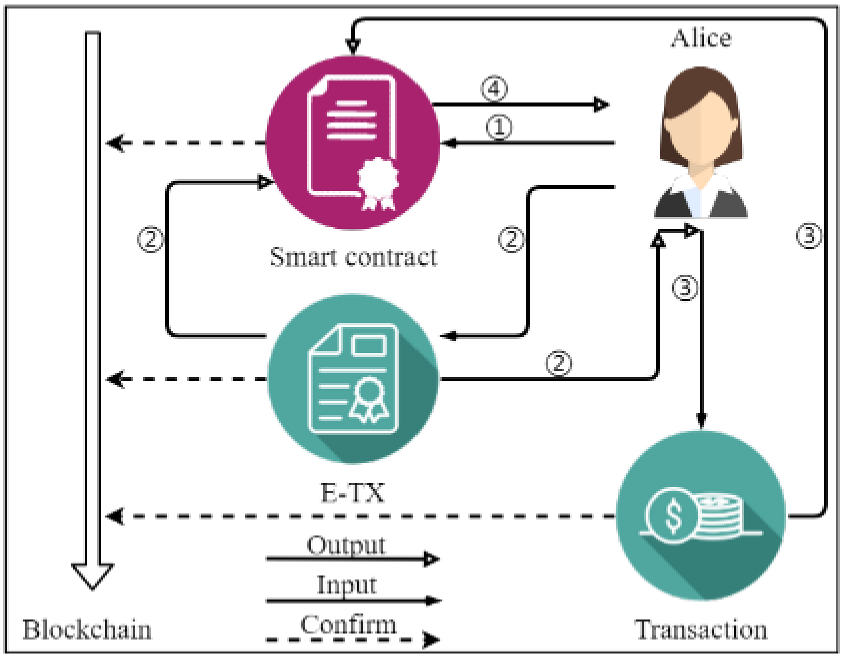
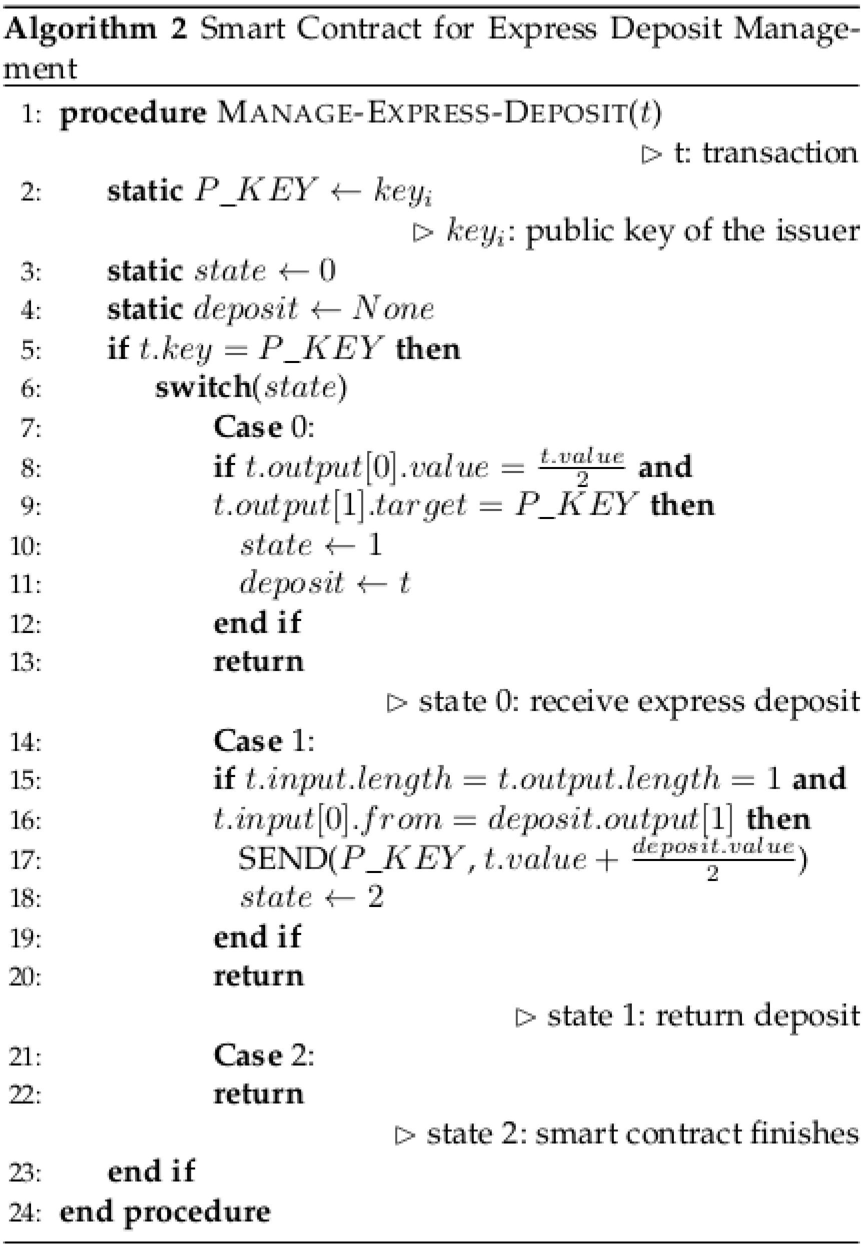


Figure 3.2: The procedures for issuing an E-TX.

As shown in Fig. 3.2, technically, we use smart contract to support the proposed E-TX. If Alice wants to issue an E-TX, she should first write a smart contract for managing (1). This smart contract only the express deposit (procedure 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 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 reduces the total propagation time T_{max} to

$$T'_{max} = \max_{i\in\Gamma_{max}\wedge V} T_i^v + \sum_{j\in\Gamma_{max}\wedge E} T_j^e \tag{Equ. 3.9}$$

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 (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 corresponds 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 a invalid E-TX.



Algorithm 2 Smart Contract for Express Deposit Management

```
procedure MANAGE-EXPRESS-DEPOSIT($t$)    ▷ t: transaction
    static $P\_KEY \leftarrow key_i$    ▷ $key_i$: public key of the issuer
    static $state \leftarrow 0$
    static $deposit \leftarrow None$
    if $t.key = P\_KEY$ then
        switch($state$)
            Case 0:
            if $t.output[0].value = \frac{t.value}{2}$ and
            $t.output[1].target = P\_KEY$ then
                $state \leftarrow 1$
                $deposit \leftarrow t$
            end if
            return    ▷ state 0: receive express deposit
            Case 1:
            if $t.input.length = t.output.length = 1$ and
            $t.input[0].from = deposit.output[1]$ then
                SEND($P\_KEY, t.value + \frac{deposit.value}{2}$)
                $state \leftarrow 2$
            end if
            return    ▷ state 1: return deposit
            Case 2:
            return    ▷ state 2: smart contract finishes
    end if
end procedure
```

3.3.3 Hollow block

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 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

Redundancy reduction

To reduce the redundancy among the new block and participants' queues of transactions, Hollow block is proposed. Which is different from the traditional blockchain blocks, 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 receives 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 time-stamp is determined by the transaction so that the sort results in different participants can keep consistence. Moreover, this time-stamp will be determined before its submission. There is an appropriate tolerance in transaction propagation for its times-tamp not being consistent with the current time.

Conclusion

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. Although 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 solve the above problems a green blockchain framework is designed . First, a green PoC consensus mechanism is proposed in the framework, where edge de- vices give their proof of contributing collaboration rather than consuming enormous computational resources to solve mathematic puzzle for the privilege of collaboration. Second, the futile transaction theory is proposed and a transaction offloading module is designed based on FTF algorithm in the 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 are conducted to prove the advantages and superiority of the proposed framework.

References

[1] 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.

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

[3] Z. Xiong, S. Feng, D. Niyato, P. Wang, and Z. Han, ““Optimal pricing-based edge computing resource management in mobile blockchain,” ” *2018 IEEE International Conference on Communications (ICC),* May 2018, pp. 1–6

[4] Z. Xiong, Y. Zhang, D. Niyato, P. Wang, and Z. Han, “”When mobile blockchain meets edge computing,” *IEEE Communications Magazine, vol. 56, no. 8, pp. 33–39, August 2018.*

[5] K. Christidis and M. Devetsikiotis “ “Blockchains and smart contracts for the internet of things,” ” *IEEE Access, vol. 4, pp. 2292–2303, 2016.,* 2017.

[6] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, ““Medrec: Using blockchain for medical data access and permission management,” ” *2016 2nd International Conference on Open and Big Data (OBD), Aug 2016, pp. 25–30.*

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

[8] K. J. O’Dwyer and D. Malone, ““Bitcoin mining and its energy footprint,” ”*n ISSC 2014/CIICT 2014, June 2014, pp. 280–285.*

[9] S. King and S. Nadal ““Ppcoin: Peer-to-peer crypto-currency with proof-of-stake,” ” *self-published paper,* August, vol. 19, 2012.