Faster Service with Less Resource: A Resource Efficient Blockchain Framework for Edge Computing

Kaiyu Wang
Harbin Institute of Technology
Zhiying Tu
Harbin Institute of Technology(Weihai)
Zhenzhou Ji (jizhenzhou@hit.edu.cn)
Harbin Institute of Technology
Shufan He
Harbin Institute of Technology(Weihai)

Research Article

Keywords: Blockchain, Edge computing, PoL, Resource-efficient, Cloud-edge collaboration

Posted Date: June 10th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1719287/v1

Faster Service with Less Resource: A Resource Efficient Blockchain Framework for Edge Computing

Kaiyu Wang[1], Zhiying Tu[1,2], Zhenzhou Ji[1,2*] and Shufan He[2]

[1*]Department of Computer Science and Technology, Harbin Institute of Technology, Xidazhi Street No.92, Harbin, 150001, Heilongjiang, China.

[2*]Department of Computer Science and Technology, Harbin Institute of Technology(Weihai), Wenhua west Street No.2, Weihai, 264209, Shandong, China.

*Corresponding author(s). E-mail(s): jizhenzhou@hit.edu.cn;
Contributing authors: wang_kaiyu2103@126.com;
tzy_hit@hit.edu.cn; Sofan_He@163.com;

Abstract

By integrating edge computing with blockchain technology, traceable and immutable services can be provided to address the distrust issue between edge devices. However, the contradiction between the computing and storage requirements of blockchain deployment and the constrained resources of edge devices is the greatest obstacle standing in the way. This makes edge devices have to consume expensive costs to deploy a blockchain system for trusted collaboration. To address the above problem, this article proposes a resource-efficient blockchain framework for edge computing. First, we propose a cloud-edge collaboration consensus protocol called the Proof-of-Lottery (PoL) that completely offloads the computing requirement of consensus protocol execution from edge devices to the cloud. By introducing double-chain structure and dynamic difficulty target mechanism, PoL can significantly improve the throughput and latency of the blockchain transaction while remains decentralization and security. Then, the framework division mechanism based on edge-terminal consensus zone (ETCZ) is proposed to decrease the minimum storage requirement for resource constrained

edge devices. The multi-ETCZ PoL protocol is also designed for consensus in the multi-ETCZ blockchain framework. Finally, controlled experiments are tested to demonstrate that our framework can significantly reduce the resource requirements for edge devices on blockchain deployment while achieving high-throughput and low-latency services.

Keywords: Blockchain, Edge computing, PoL, Resource-efficient, Cloud-edge collaboration

1 Introduction

Recently, many service scenarios, such as Internet of Things (IoT) [1], Internet of Vehicle (IoV) [2], and intelligent manufacturing [6] et al., are requiring low-latency and high-throughput services [3]. To achieve this goal, edge computing has been widely applied via deploying services on distributed edge devices. However, unfortunately, the trusted cooperation between edge devices is not always satisfied. Some malicious edge devices may deny or even tamper service records and transaction data for self benefits [4]. These distrust issues cause negative impacts and will eventually lead to the non-collaboration in edge computing environment. To address this problem, the fusion of blockchain and edge computing is one of the most attractive research directions because of the traceability and immutability features of blockchain technology.

However, the computing and storage resources requirement limits the deployment of blockchain in the edge computing environment. How to reduce the deployment costs on edge devices is a valuable and challenging research problem. On one side, the large amount of computing power requirement in the consensus process brings heavy burdens to resource-constrained edge devices. On the other side, the current blockchain system requires all participating nodes to store the entire chain's data, which generates extreme storage consumption. Furthermore, limited by consensus and synchronization efficiency, blockchain generally has low throughput and high latency that is conflicted with the application demand of edge computing. Accordingly, we list two basic problems to be addressed:

P1: How to reduce the computing and storage resource requirements of edge devices for deploying and operating a blockchain system on the edge computing environment?

P2: How to achieve the low-latency and high-throughput blockchain services on edge computing environment?

In this paper, solutions are proposed from two aspects respectively, the reduction of computing power requirement and storage requirement. In the former aspect, in need of dynamic device management and scalability, PoW is still necessary and non-replaceable in blockchain consensus process [23]. The demand for computing power is an objective requirement. Therefore, current researches are mainly focus on reducing the mining difficulty to decrease the

computation pressure on edge devices [4, 25]. However, the computing power is valuable and constrained in edge devices. The reduced computing power still brings heavy burden and affects the performance for processing normal services. Instead of executing consensus protocol on edge devices, we notice that cloud computing can provide ample computing power with a relatively cheap cost compared to edge devices. By introducing the collaboration between the cloud and edge devices, we can design a novel blockchain consensus protocol to offload the computing power requirement from edge devices to the cloud.

In the storage reduction aspect, most service scenarios such as IoT [1], IoV [2], and spectrum sharing [23] are normally geography or logic enclosed. Sharding the blockchain into several smaller groups based on the distribution of enclosed sub-scenarios can significantly reduce the storage requirement for each participant node. However, the smaller group means less nodes and lower security. Compared to control the entire blockchain, attackers are more easier to control a single group to deny and tamper records. Therefore, the potential security hazard must be defended in the implementation of sharding mechanism.

To achieve our target, the listed three additional problems are also need to be addressed:

P3: How to integrate the cloud into the edge-based blockchain system without disrupting the decentralized feature?

P4: How to prevent the security reduction in each shard of the blockchain?

P5: How to achieve services between different shards?

To address the aforementioned four problems, this article proposes a resource-efficient blockchain service framework for edge computing that can reduce the computing and minimum storage resource requirements for edge devices while providing high-throughput and low-latency services. First, we design a new cloud-edge collaboration consensus protocol called the *Proof-of-Lottery* (PoL) to offload the huge amount of computation operations in the consensus process from edge devices to the cloud. In PoL, the traditional blockchain structure has been redesigned from a single chain to a double-chain structure. Based on that, PoL combines the PoW and PoS consensus approaches to allow edge devices to generate multiple blocks with a single round consensus. With the dynamic difficulty target function, the framework can achieve high-throughput and low-latency at the same time.

Second, we introduce the *Edge-Terminal Consensus Zone* (ETCZ) and the framework division mechanism based on ETCZ to reduce the minimum storage requirement of edge devices. Besides that, the multi-ETCZ PoL consensus protocol is proposed to achieve consensus in a multiple ETCZ framework. The cross-zone transaction process procedure has been proposed to achieve services between two different ETCZs.

Finally, with controlled experiments, the effectiveness of our framework is demonstrated. The computing requirement has been completely offloaded from edge devices to the cloud. The storage requirement is reduced to $\frac{1}{K}$ of the traditional structure blockchain storage consumption in a K-ETCZ framework.

Meanwhile, the TPS and transaction processing latency also have remarked improvements. The main contributions of this paper can be summarized as follows:

- We propose a resource-efficient blockchain service framework for edge computing to reduce the computing and minimum storage requirements for edge devices;
- We propose a new cloud-edge collaboration consensus protocol called the Proof-of-Lottery (PoL). In PoL, the computing power requirement has been completely offloaded from edge devices to the cloud. With the new designed blockchain structure, block generation rule, and the dynamic difficulty target function, the transaction throughput and latency has been significantly improved;
- We define the edge-terminal consensus zone (ETCZ), which is a set of devices belonging to an enclosed sub-scenario. By dividing the framework based on ETCZ, the minimum storage requirement for edge devices has been successfully reduced;
- The double-bet mechanism is provided in multi-ETCZ PoL protocol to prevent the security reduction because of the implementation of blockchain sharding;
- The effectiveness and performance of our proposed framework are demonstrated through comparison experiments.

2 RELATED WORK

2.1 Blockchain on the Edge

Blockchain technology has been first proposed by Satoshi Nakamoto in 2008 and used in cryptocurrencies widely ever since. The decentralization, immutability, and traceability features demonstrated by blockchain have received widespread attention from both academia and industry. In the meanwhile, with the limitation of network transmission bandwidth and extremely large data volumes, traditional cloud computing centers are struggling to provide low-latency and high-throughput services. The new distributed and closer to data source computing framework, edge computing [8], is proposed and gradually applied in many service scenarios. Because of the absence of trust in the cooperation between edge devices, blockchain technology is naturally being used to integrate with edge computing in an attempt to solve the above problem. In recently researches, blockchain has been used as a decentralized ledger to ensure trust and security in data transmission [9], trustworthy authentication [10], and storage of critical information [11] of edge computing.

Data transmission is the most common application scenario of blockchain in edge computing environment. Both IoT and IoV have high demands for data interaction. Xiaolong Xu et al. have proposed the BeCome which is a blockchain-enabled computation offloading method in edge computing [12]. Kang J et al. proposed a method of secure and efficient data sharing

between vehicular edge computing networks [13]. Yunlong Liu et al. designed a blockchain empowered secure data sharing architecture for distributed multiple parties to solve the privacy issue in federal learning data transmission [14]. In other application scenarios, such as software-defined networks [15], mobile-edge computing [16], and smart grid networks [17], blockchain has also been used to protect the security and privacy of data sharing.

Trust authentication is another function that edge computing hoping to achieve through the application of blockchain. Shaoyong Guo et al. proposed a distributed and trusted authentication system on edge computing to achieve secure authentication and collaborative sharing [18]. Matev et al. introduced an autonomous blockchain system to achieve the selection of the most convenient electric terminal charging station [19]. Tselios et al. applied blockchain in the software defined network (SDN) based cloud computing infrastructure to manage the security factor [20]. Zhonglin et al. proposed a security authentication scheme of 5G Ultra-Dense network based on blockchain [21].

2.2 Resource Efficient Blockchain Approach

In the past researches, many previous works have investigated how to solve the distrust issue between edge devices by applying blockchain. However, the maintenance and operation of a blockchain system is very costly. New blockchain deployment schemes on edge computing are needed to reduce the resource requirements. Jiaping Wang et al. proposed a scale out blockchain based on asynchronous consensus zones that supports blockchain scale out and parallel processing. Meanwhile, Wang's work has contribution on storage reduction. The minimum storage requirement for miners are decreased with the increase of scale of consensus zones [7]. Chenhan Xu et al. introduced a resource-efficient blockchain approach in edges [22] in which new types of consensus protocol, the Proof-of-Collaboration, and new block structure Hollow Block are proposed respectively to decrease the computation complexity and storage requirement of edge devices. The scheme in [22] filtering futile transactions to release the storage space occupied by the transactions which outputs are all referenced. Xin Jie et al. introduced a resource-efficient DAG blockchain with the sharding for 6G networks [23]. Fan Zhang et al. introduced a resource-efficient mining framework for blockchain that uses trusted hardware called REM [24]. Yinqiu Liu et al. introduced a lightweight blockchain system for the industrial internet of things called the LightChain [25] which is suitable for power-constrained IIoT devices. In LightChain, a green consensus mechanism, Synergistic Multiple Proof is proposed for simulating the cooperation of IIoT devices. Zhixin Liu et al. propose an efficient QoS support for robust resource allocation method for blockchain-based femtocell networks [26].

3 FRAMEWORK DESIGN

In our proposal, the blockchain framework has been deployed and maintained in a cloud-edge collaboration method. The whole framework has been divided

into three layers according to the type of devices and network topology location, as shown in Fig.1. The cloud layer contains a traditional cloud center with the enormous amount of computing power and storage space. The edge layer is consisted of several heterogeneous edge devices which include edge infrastructures, bases stations, edge servers, IoT edge devices, etc. Terminal layer consists of many terminal devices that request services from edge infrastructures. The cloud layer and the edge layer are connected by an ad hoc network and a P2P network is composed of edge and terminal devices.

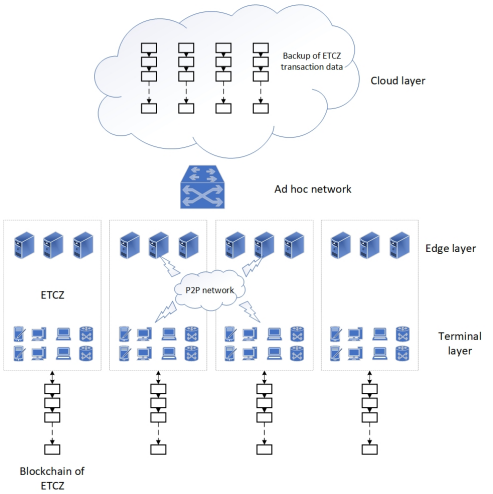


Fig. 1 The cloud-edge blockchain service framework

Between the edge and terminal layers, edge and terminal devices are divided into several zones according to the closure of service scenarios. Most service scenarios in edge computing are constituted by several sub-scenarios which are geographically or logically enclosed. For example, in the IoT smart home scenario and intelligent industrial production scenario, data is geographically and logically enclosed. There is no data interaction between different communities or industrial lines based on the consideration of security and privacy. These scenarios can be seen as complete-enclosed scenarios. And IoV is the classical example of near-enclosed service scenarios. Most of vehicles only share and transmit data within a small scope (a district or a city) in the most of time because of the limited locomotivity. Only in rare cases, the information will be send from one vehicle to another one who are thousands of kilometers away. Therefore, we vertically segment the framework according to the data interactive feature of service scenario to form several consensus zones called the *edge-terminal consensus zone* (ETCZ). The definitions are as follows:

Definition 1. Complete Enclosed sub-scenario. *In a complete enclose sub-scenario, all data interactions are within this sub-scenario.*

Definition 2. Near Enclosed sub-scenario. *In a near enclose sub-scenario, the majority data interactions are within this sub-scenario.*

Definition 3. Edge-Terminal Consensus Zone. *In edge computing environment, a group of edge and terminal devices which belong to a complete-enclose or near-enclose service scenario consist an edge-terminal consensus zone.*

In the proposed framework, each ETCZ maintains and operates a single blockchain that processes transactions generated within the ETCZ. The cloud is used to provide the computing power required for the consensus process and to store the global data of the entire framework. Since the cloud does not participate in any specific consensus process and only provides computing and storage resources, the overall framework is working in a decentralized manner.

4 CLOUD-EDGE COLLABORATION CONSENSUS

4.1 UTXO and Account based transaction model

There are two types of blockchain transaction models, the *Unspent Transaction Output* (UTXO) model and the account-based model. In the UTXO based blockchain, a transaction spends outputs from previous transactions and generates new outputs that can be spent in future transactions. In specific, a user generate a transaction to spend a mount of money/cryptocurrency, he/she needs to use one or more UTXOs to cover the cost and get the changes back as new UTXOs. This model is first used in Bitcoin [5] and applied many other blockchain systems [36, 37]. The account-based model is similar to the bank account which is used by Ethereum [27]. Blockchain nodes need to check if the account balance can cover the cost before approving the transaction. Compared to the UTXO, account based model is thought to be better for supporting smart contracts.

For the above two transaction models, data integrity has different importance levels in security. For the UTXO blockchain, the integrity of transaction records is necessary for verifying the newly generated transactions whether have enough inputs to cover the spend. Therefore, blockchain nodes without the entire blockchain data cannot process any transactions. However, for account-based blockchain, all nodes maintain a certain kind of data structure that supports searching to record the global state of all nodes. Therefore, the data integrity is not as important as the UTXO blockchain in transaction verifying, which is more used as a ledger for transaction traceability and working as some sort of credential for arbitration. In this paper, the proposed blockchain service framework is based on the account-based transaction model.

4.2 Protocol Design Premise

The blockchain has two types: the public chain and the consortium/private chain. If the blockchain system allows anyone to participate in without the authorization of privilege nodes or the third party, it is a public chain. Otherwise, the blockchain may be a consortium or private chain. The different

chain types result in different consensus protocols. Proof-of-Work (PoW) as the first public blockchain consensus protocol making a great success in Bitcoin [5]. Later, the concept of Proof-of-Stake (PoS) [33] has been proposed, and its main idea is that stakeholders show their assets as the substitutes of computing power to compete the blockchain's counting authority. In addition, Practical Byzantine Fault Tolerance (PBFT) [28] and its variants are widely used in consortium and private chains. The PBFT can tolerate up to a third of participants that occurs any form of failure by giving a fixed number of participants in advance.

In edge computing environment, the number of edge devices should adopt to the demand of users, which is not fixed. Therefore, most studies choose to implement a public blockchain system on edge devices network [22–26]. However, current consensus protocols are all have shortcomings that bring drawbacks in security and deployment aspects. For PoW, the enormous computing power consumption and low consensus efficiency limit its application. The PoW consensus requires blockchain nodes searching for a proper *nonce* that can satisfy the difficulty target. Because of the irreversibility of the hash operation, the searching process is similar to gold mining. Therefore, the PoW is also called as mining and nodes are called as miners. The mining process can be simplified as the following formula:

$$Find\ nonce \quad s.t.\ \mathbf{SHA256}(h \odot nonce) < target \tag{1}$$

in which $\odot$ is a string concatenate operator; *target* is the difficulty factor. The smaller the target is, the more difficulty the mining is. h represents the content of the newest block.

For avoiding fork, the PoW is designed to generate only one block in a fixed period. Meanwhile, the difficulty target is usually been set as a relatively low value, which means miners require to execute plenty of hash operations to search a proper nonce to generate only one block. Therefore, the PoW is energy consumptive and low performance. This increases the deployment difficulty of PoW blockchain on resource constrained edge devices. However, PoW is reliable and easy implementation. Besides, PoW protocol requires no prior statement of miners. Blockchain nodes can join or leave the system at will, which provides a flexible device management that is suitable for edge computing environment.

As the alternative to PoW, the PoS protocol solves PoW's energy consumption to achieve consensus in a more environmental friendly manner. By taking tokens as stakes, PoS selects the candidate in a roulette wheel way. However, the fairness and security of consensus rely on the chaos level of the generated random number. Unfortunately, the creation of completely random numbers is always the conundrum faced by academia. No matter the random seed for random numbers is selected within the blockchain or from outside, the risk of the random number prediction is still existed and threats the security of the PoS blockchain. Besides, PoS consensus is more likely to be centralized than

PoW. The centralization of tokens has less cost than the centralization of computing power. Therefore, the entire blockchain has the trend to be controlled by a small group of members with strong financial resources. Meanwhile, the existence of the Nothing-at-stake attack [34] and the Long-range attack [35] decreases the barrier of malicious behavior that further raises the potential safety hazard.

BFT-like consensus is another selection for blockchain systems. BFT-like consensus is based on the voting mechanism in which if a message receiving more than $n - f$ votes in a total $n = 3f + 1$ nodes distributed system, this message can be seen as achieving consensus if the number of byzantine nodes is less than f. BFT-like consensus can provide deterministic consensus no matter byzantine nodes are sending fault messages or just denying to respond. Therefore, it is very suitable for synchronizing data or state in a distributed or decentralized system. However, to prevent the Sybil attack, BFT-like consensus requires to pre-review nodes' authority. Besides, newly added nodes need to be carefully verified to prevent the number of byzantine nodes exceeds the system upper bound f. Therefore, BFT-like consensus can hardly be used in edge computing environment.

4.3 Consensus Incentive

As we know, the incentive of consensus protocol is very vital for driving a public blockchain system. In our design, edge devices are stimulated by a new kind of asset called the *Service Contribution* which is denoted as $\mathbb{C}$. $\mathbb{C}$ is similar to the BTC in the Bitcoin and ETH (Gas) in the Ethereum. In the proposed framework, the edge device that wins in the PoL will be rewarded by $\mathbb{C}$. We normalize the reward of each datablock contained in the winning lottery as a single unit $\mathbb{C}$. Otherwise, transactions also need to be paid by $\mathbb{C}$ as the transaction fee for the process.

In the proposed framework, $\mathbb{C}$ can be obtained through two approaches, participating in PoL and trading from others. And the only way to create $\mathbb{C}$ is to participant in consensus. Therefore, terminals who want to get their transactions processed must purchase $\mathbb{C}$ from edge devices. Edge devices can benefit from participating in blockchain consensus and processing transactions, which can further encourage more devices to join the framework.

4.4 Proof-of-Lottery Protocol

PoW protocol is suitable for public blockchain systems because of its security and easy-to-implement features. Actually, by ignoring the differences in each miner's mining equipment (such as CPU, GPU and ASIC), PoW consensus can be abstracted as a lottery drawing process with computing power as lottery, the *nonce* is the winning number and the winner is the miner selected by PoW. In PoW consensus, all miners are doing the mining process until someone finds a *nonce* that satisfies the Equation. 1. Therefore, the computing power consumed by miners can be offloaded by moving all mining processes from

miners to a certain machine to reduce the computing power consumption. Because mining is an energy consuming, repeating processing and same for all miners, the offloading will not change the consensus effectiveness. Based on this idea, we propose the Proof-of-Lottery (PoL) consensus protocol. PoL moves the mining process to the cloud to free edge devices from the requirement of a huge amount of computing power. The demonstration of the PoL consensus is shown in Fig.2.

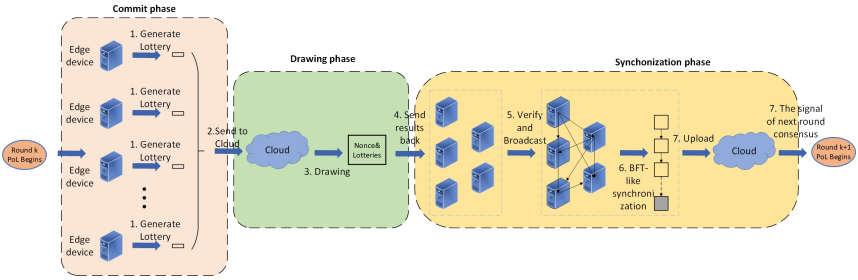


Fig. 2 The procedure illustration of PoL protocol

The PoL protocol is constituted by three phases, the *commit phase*, the *drawing phase* and the *synchronous phase*. After receiving the signal from the cloud, the commit phase begins. The edge device first generates a lottery which is the approximation of a PoW blockhead. The lottery contains the edge device's public key E_i, i.e. device ID, the number of datablocks Θ, the Merkel root of datablocks of this lottery M, the previous round lottery $\mathbb{L}_{j-1}$ and the count of failure round P_{E_i}, i.e. the count of rounds since last time E_i wins in the PoL. The lottery submitted by edge device E_i in round j can be denote as a $\mathbb{L}_j < E_i, \Theta, M, \mathbb{L}_{j-1}, P_{E_i} >$. One device ID is only allowed to submit one lottery.

Then, all edge devices send their lotteries to the cloud. The drawing phase begins. Similar to the PoW, in drawing phase, the cloud searches a proper nonce with all submitted lotteries in parallel method until a certain edge device's lottery satisfies the Equation 1. The detailed process can be expressed as the following equation:

$$Find\ nonce\ s.t. \begin{cases} \mathbf{SHA256}(\mathbb{L}_j^{E_1} \odot nonce) < target \\ \mathbf{SHA256}(\mathbb{L}_j^{E_2} \odot nonce) < target \\ \quad . \\ \quad . \\ \quad . \\ \mathbf{SHA256}(\mathbb{L}_j^{E_i} \odot nonce) < target \end{cases} \tag{2}$$

To improve the throughput of blockchain, PoL allows the winner to generate a block sequence with one round consensus. Considering the blockchain adopts the longest-chain mechanism, edge devices who generate more blocks will have a greater advantage. Therefore, the double-chain structure is adopted

in PoL that is similar to the Fruitchain [34]. The *lottery_Chain* is consisted of the winning lotteries, and the *data_Chain* is the chain of datablocks that are contained in winning lotteries of corresponding rounds. When lottery_chain forks, only datablocks that belong to the longest lottery_chain will be considered as valid. The double-chain structure diagram is shown in Fig.3.

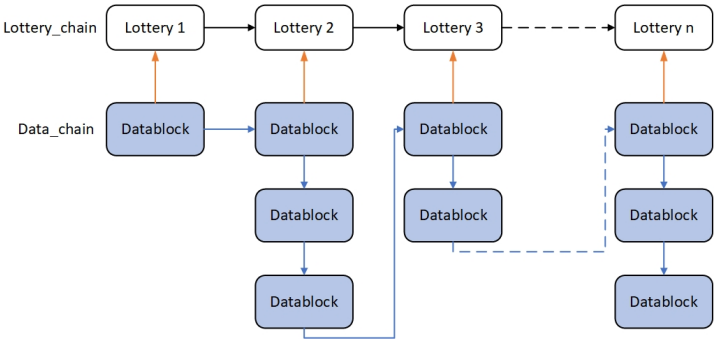


Fig. 3 The double-chain structure

Since the security of blockchain is not related to the number of data blocks. Edge devices are supported to generate a block sequence with at most Θ datablocks with one winning lottery. The value of Θ can be calculated based on the amount of $\mathbb{C}$ held by each device in the PoS way. Because the mining process has been offloaded from edge devices to the cloud, the lack of computing power requirement makes malicious edge device has the ability to launch the Sybil attack to masquerade as multiple devices by forging different IDs. To prevent this malicious behavior, in the calculation of Θ, we introduce the nonlinear stake share which is shown as follows:

$$\Theta_{E_i} = \lfloor \frac{(\mathbb{C}_{E_i})^{1+\varepsilon}}{\mathbb{C}_{total}} \times \Phi \rfloor \tag{3}$$

in which $\mathbb{C}_{E_i}$ is the service contribution held by the edge device E_i and $\mathbb{C}_{total}$ is the sum of all service contribution in the framework. The ε is the advantage factor, $\varepsilon \in (0, 1)$. Φ is initially been set as the number of edge devices. With the operation of framework, the Φ is updated to the average number of lotteries of last 10 rounds.

To accelerate the drawing process, we introduce the difficulty adjustment function $\mathbb{F}_{E_i}$ to dynamic adjusts each device's drawing difficulty target. The $\mathbb{F}_{E_i}$ is the function of the persistence P_{E_i}. The difficulty function is shown as follows:

$$\mathbb{F}(E_i) = P_{E_i} \tag{4}$$

The drawing process for device E_i with dynamic difficulty can be expressed as:

$$Find\ nonce \quad s.t. \quad \mathbf{SHA256}(\mathbb{L}_j^{E_i} \odot nonce) < target \times \mathbb{F}(E_i) \tag{5}$$

After selecting a winning lottery, the cloud broadcasts all lotteries to edge devices. The synchronous phase starts. Due to the BFT-like consensus approach can provide deterministic consensus and tolerant certain levels of byzantine fault, we adopt a BFT-like approach to complete the synchronization of datablocks. The synchronization has four phases, pre-prepare, prepare, log replication, and reply. All lottery submitters are divided into two kinds of roles in which the winner of the PoL is the leader and the rest lottery submitters are followers. The data synchronization is occurring between the current round consensus lottery submitters.

In the pre-prepare phase, after finding a winning lottery, the cloud will broadcast the winning information to all lottery submitters, which includes the nonce and all $|i| = 3f + 1$ lotteries with the signature of cloud. Then, the synchronization turns to prepare phase.

In the prepare phase, the follower who receives the winning lottery will do two operations, verification and broadcast. First, the follower verifies the validity of the information of winning lottery including the reference relationship (the previous lottery) in the lottery_chain and the correctness of the nonce under the corresponding difficulty target. If the winning lottery is valid, the owner of winning lottery will be identify as the leader. Then, the follower broadcasts the prepare message to other followers. If the follower receives more than $2f + 1$ prepares from other followers, it goes to the log replication phase.

In the log replication phase, followers request the datablock sequence from the leader. After receiving all of datablocks, the follower verify datablocks with the winning lottery. If datablock sequence matches the merkel root M in winning lottery, the follower broadcasts a reply message and turns to the reply phase.

In reply phase, after receives over $2f+1$ different followers' reply messages, the follower sends a finish message to the leader, which is the signature of the winning lottery. If the leader receives more than $2f + 1$ finish messages, it means the majority of lottery submitters are achieving consensus and finishing synchronous. The entire synchronous phase is over.

After the synchronous phase, the leader submits the lottery with more than $2f+1$ signatures to the cloud and uploads all Θ datablocks. Edge devices who do not submit lottery in this round or newly joined can request data backup from lottery submitters. Finally, all edge devices achieve data consensus. The current round PoL is completed. The cloud sends the beginning signal to all edge devices to start next round consensus. If the uploaded lottery has less than $2f + 1$ signatures or the current round PoL exceeds the limited time τ, the cloud sends error signal to restart the current round consensus.

In PoL, edge devices are allowed to generated a datablock sequence within one round of PoL consensus. Hence, the entire framework achieves high-throughput in transaction process. Meanwhile, the introduction of persistence P in the drawing process avoids edge devices unable to win for a long time to maintain the relative fairness to prevent the centralization phenomenon like PoS blockchain caused by great stake holders. The drawing process can be

completed with a larger target for a great probability. The transaction latency of the entire framework is also significantly improved. Therefore, the **P2** is addressed.

4.5 Protocol Analysis

The drawing process is similar to the PoW protocol, which security and effectiveness has already been proved. Besides, by offloading the drawing process from miners to the cloud, malicious devices cannot launch attacks through withholding mined blocks and forking the blockchain artificially, such as the selfish-mining attack[38], the whale attack[39], the BWH attack [40], and the FAW attack [41]. However, without the requirement of computing power, edge devices have the ability to submit several lotteries by masquerading as several devices through the Sybil attack. We will prove this malicious behavior will not bring devices any advantage in benefit in PoL.

Theorem 1. *The PoL protocol can prevent rational edge devices from submitting several lotteries.*

Proof. PoL cannot prevent edge devices from irrational behaviors, i.e. behaviors without the consideration of benefits. Then, we believe that the PoL can prevent the rational edge device to submit several lotteries if the device cannot obtain higher revenue from it.

Because of the extremely large solution space of the SHA256 algorithm, the winning probability in PoL exhibits memoryless feature regarding the time of nonce searching. Therefore, we use the exponentially distribution to represent the winning probability of a device in drawing process.

$$Pro(t) = 1 - e^{-\kappa t} \tag{6}$$

In PoL, each lottery may has a different difficulty target. Then, κ of device E_i can be expressed by difficulty function as $\kappa_{E_i} = \frac{\sum_{\mathbb{L}_j} \mathbb{F}(E_m)}{\mathbb{F}(E_i)}$. For device E_i, the expected revenue in one round PoL is:

$$R_{E_i} = \Theta \times \int_{t=0}^{\infty} t \times (1 - e^{-\kappa t}) = \frac{\Theta}{\kappa_{E_i}} \tag{7}$$

If the device chooses to submit n lotteries by masquerading a device as n devices with different IDs E_i^{χ}, $\chi \in (1, n)$, each masqueraded device will hold $\mathbb{C}_{E_i^{\chi}}$ tokens in which $\mathbb{C}_{E_i} = \sum_{1\sim n} \mathbb{C}_{E_i^{\chi}}$. The total expected revenue for the submitting n lotteries is:

$$\sum_{1\sim n} R^*_{E_i^{\chi}} = \sum_{1\sim n} \Theta^*_{E_i^{\chi}} \times \int_{t=0}^{\infty} t \times (1 - e^{\kappa^* t}) = \sum_{1\sim n}^{i} \frac{\Theta^*_{E_i^{\chi}}}{\kappa^*} \tag{8}$$

in which $\kappa^* = \frac{\sum_{\mathbb{L}_j - E_i^{\chi}} f(E_i^{\chi}) + \sum_{1\sim n} f(E_i^{\chi})}{\sum_{1\sim n} f(E_i^{\chi})}$, $\Theta^*_{E_i^{\chi}} = \lfloor \frac{(\mathbb{C}_{E_i^{\chi}})^{1+\varepsilon}}{\mathbb{C}_{total}} \times \Phi \rfloor$.

Obviously, we have $R_{E_i} > \sum_{1\sim n}^{i} R^*_{E_i^\chi}$ (detailed derivation is shown in appendix A). Therefore, for an edge device with a certain amount of $\mathbb{C}$, submitting multiple lotteries within one round PoL has no advantage in expected revenue. The prove is complete.

For irrational devices, the irrational behaviors will cause revenue losses.

Theorem 2. *For irrational edge devices, the more lotteries they submit the less expected revenue will be.*

Proof. We assume an irrational edge device E_{ir} launches the Sybil attack to masquerade as n devices with different IDs. The Θ is a function about the $\mathbb{C}_{E_{ir}}$. Then, we have the Θ's derivation about $\mathbb{C}_{E_i}$ as:

$$\frac{\partial \Theta}{\partial \mathbb{C}_{E_{ir}}} = \lfloor (1+\varepsilon) \frac{(\mathbb{C}_{E_{ir}})^{\varepsilon}}{\mathbb{C}_{total}} \times \Phi \rfloor \tag{9}$$

Obviously, Θ's derivation is always positive and it's value is positively related with the value of $\mathbb{C}_{E_{ir}}$. The smaller $\mathbb{C}_{E_{ir}}$ is the smaller Θ will be. Therefore, the following in-equation is established when $n > 1$.

$$\Theta(\mathbb{C}_{E_{ir}}) \geq \sum_{1\sim n}^{i} \Theta(\mathbb{C}^i_{E_{ir}}) \tag{10}$$

In the extreme case when n is big enough, we have $\Theta(\mathbb{C}^i_{E_{ir}}) = 0$. The prove is complete.

So far, PoL successfully offloads the computation operations of consensus protocol execution from edge devices to the cloud. The part of **P1** is addressed. Meanwhile, the cloud center is only responsible for providing the computing power and storing space, rather than being involved in the specific consensus process, such as packing transactions and verifying blocks. Therefore, the consensus of PoL is achieved in a decentralized way. Blockchain's decentralized feature is not disrupted because of the integration of the cloud. The **P3** is addressed.

5 EDGE-TERMINAL CONSENSUS ZONE

In this section, we introduce the edge-terminal consensus zone in the blockchain service framework for reducing the minimum storage requirement of edge devices. In a traditional blockchain system, participants are required to store the entire chain's data, which is necessary for verifying transactions and tracing service records. Meanwhile, the abundant data backup held in the decentralized way provides immutability. However, the huge storage consumption excludes some storage limited edge devices from blockchain. Taking Bitcoin as an example, the total size of the local ledger is over 200GB and grows at a rate of approximately 150 MB per day [31]. Some lightweight class edge devices cannot afford such large storage requirements.

Besides, the edge computing environment is constituted by different types of edge devices with different computing and storage resources. How to reduce

the participant barrier while taking full advantage of devices with ample storage space is vital to blockchain security. To address the above problem, the edge-terminal consensus zone is proposed to reduce the minimum storage requirement for edge devices. Furthermore, devices with abundant storage sources are encouraged to participate in as much as ETCZ's consensus with the cooperation of the multi-ETCZ PoL protocol.

We define an ETCZ as $< K, S, W, \mathbb{E}_S >$ in which K is the number of ETCZ in the whole framework; S is the zone index; W is the number of edge devices in this ETCZ and $\mathbb{E}_S$ is the set of in-zone edge devices' public key. Each ETCZ maintains an individual blockchain as shown in Fig.4. By shading the entire blockchain service framework based on the service scenarios, multiple parallel blockchains are maintained and operated within respective ETCZs. The minimum storage requirement of edge devices is reduced to a single ETCZ. In the multi-ETCZ framework, the edge device's in-zone ID in ETCZ S is represented as E_i^S. One in-zone ID is only allowed to submit one lottery in this ETCZ. Edge devices are sllowed to participate in multiple ETCZs' consensus to underutilize their storage space. In our framework, an edge device can choose to participate in any one or several ETCZs consensus at will. Based on this, we give the following definitions:

Definition 4. Isolated Edge Device (IED). *For an edge device, if it only participates in one ETCZ consensus, we define it as an isolated edge device.*

Definition 5. Coupled Edge Device (CED). *For an edge device, if it participates in several ETCZs' consensus, we define it as a coupled edge device. The number of ETCZs it participated in is the dimension of CDE.*

5.1 Multi-ETCZ PoL consensus protocol

The multi-ETCZ PoL consensus protocol of the multi-ETCZ blockchain framework is similar to the PoL as we described in section 4 with following improvements. In the commit phase, all edge devices from all ETCZs generate lotteries to submit for drawing. One in-zone ID is allowed to generate only one lottery. Each in-zone ID is independent on service contribution counts. For a CDE E_i who participates in total $\mathbb{S}$ ETCZs, $\mathbb{C}_{E_i} = \sum_{S \in \mathbb{S}} \mathbb{C}_{E_i}^S$. Specially, for IDE, $|\mathbb{S}| = 1$.

In the drawing process, each ETCZ will generate one winning lottery. Total $|K|$ winning lotteries will be generated in a K-ETCZ framework. To encourage edge devices to participate in more ETCZs, we propose the *double-bet* mechanism in which, for coupled edge devices, if it's lottery is winning in one ETCZ, he is also winning in all ETCZs that he submits lotteries. The winning rules of multi-ETCZ PoL are as follows:

Rule 1. Priority to win. *The lottery with a proper nonce that satisfies the difficulty target has the priority to win in its ETCZ.*

Rule 2. Harder to win. *If more than one winning lotteries exist, the one with a smaller difficulty target is judged as the winning one.*

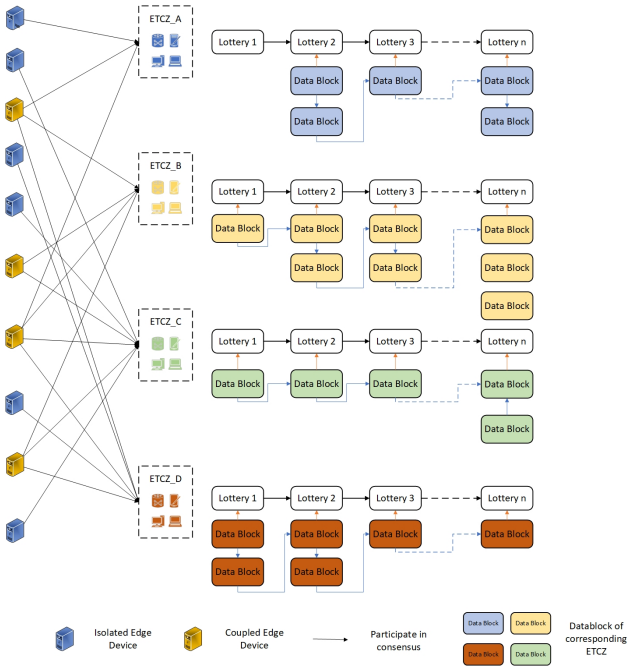


Fig. 4 The illustration of multi-ETCZ consensus

Rule 3. Higher dimension to win. *When there is no winning lottery generated in ETCZ A, but two CDEs who participate in A win in other ETCZs B and C at the same time, the CDE with a higher dimension will win in A.*

Rule 4. The winner reset difficulty. *If a CDE wins in one ETCZ, all ETCZs he participates reset his difficulty function as 0.*

For the winner in ETCZ S, the Θ is changed to Θ_S:

$$\Theta_S = \lfloor \frac{(\mathbb{C}_{E_i}^S)^{1+\varepsilon}}{\mathbb{C}_{total}^S} \times \Phi \rfloor \tag{11}$$

in which $\mathbb{C}_{total}^S$ is the total service contribution held by all devices in ETCZ S. The total number of datablock generated in the entire framework is $\Theta = \sum_{\mathbb{S}}^{S \in \mathbb{S}} \Theta_s$.

The PoL can avoid devices to submit multiple lotteries within one ETCZ, which has been proven in Theorem 1. Based on Theorem 1, we will prove that multi-ETCZ PoL can encourage devices to become CDEs with the help of double-bet mechanism.

Theorem 3. *Edge devices with adequate storage space are encouraged to becoming CDEs.*

Proof. We believe that if the edge device can obtain higher expected revenue by becoming a CDE, the theorem can be seen as established. As we analyzed in Theorem 1 and 2, for a device E_i, the winning probability of a lottery in ETCZ S is decided by this lottery's difficulty function $\mathbb{F}(E_i)$ and the difficulty

criterion of this ETCZ, $\sum_{\mathbb{L}^s} \mathbb{F}$. The ETCZ with a lower difficulty criterion will attract other devices to participate in. Therefore, after the entire framework achieves stability, each ETCZ tends to have an approximately equal criterion. Therefore, we consider each ETCZ has the same difficulty criterion $\sum_{\mathbb{L}^*} \mathbb{F}$, so do the total service contribution $\mathbb{C}^*_{total}$. Then, the winning probability for a lottery in each ETCZ can be expressed as $\frac{1}{\kappa} = \frac{\mathbb{F}(E_i)}{\sum_{\mathbb{L}^s} \mathbb{F}}$.

For an edge device E_i with adequate storage space and total service contribution $\mathbb{C}_{E_i}$. The expected revenue for an IDE E_i is:

$$R^{IDE}_{E_i} = \lfloor \frac{(\mathbb{C}_{E_i})^{1+\varepsilon}}{\mathbb{C}^*_{total}} \times \Phi \rfloor \times \frac{1}{\kappa} \tag{12}$$

If E_i wants to become a m dimension CDE, he has to splits $\mathbb{C}_{E_i}$ into m shares that each has $\mathbb{C}^m_{E_i}$, $\mathbb{C}_{E_i} = \sum_{1\sim m} \mathbb{C}^m_{E_i}$. With the double-bet mechanism, the expected revenue for E_i as a CDE is:

$$R^{CDE}_{E_i} = \sum_{1\sim m} \lfloor \frac{(\mathbb{C}^m_{E_i})^{1+\varepsilon}}{\mathbb{C}^*_{total}} \times \Phi \rfloor \times \frac{m}{\kappa} \tag{13}$$

By simplifying the same variable, we only need to compare $(\mathbb{C}_{E_i})^{1+\varepsilon}$ and $m \sum_{1\sim m} (\mathbb{C}^m_{E_i})^{1+\varepsilon}$. When $\varepsilon \leq 1$, we have $(\mathbb{C}_{E_i})^{1+\varepsilon} \leq m \sum_{1\sim m} (\mathbb{C}^m_{E_i})^{1+\varepsilon}$ (the detailed derivation is shown in appendix B). Besides, the higher dimension a CDE has the greater gap will be between the revenue of $R^{CDE}_{E_i}$ and $R^{IDE}_{E_i}$. The prove is complete.

We adopt the same difficulty function as PoL in multi-ETCZ PoL for all edge devices to independently verify winning lotteries. Because of Rule 4, a CDE's all lotteries have the same difficulty function value. Therefore, edge devices can verify lotteries under the double-bet mechanism.

After the drawing process, the synchronous phase for each ETCZ is the same as the PoL. After receiving $|K|$ winning lotteries, the cloud sends the beginning signal to all edge devices to start the next round consensus. Otherwise, the cloud sends signals to all edge devices to restart the current round PoL. In multi-ETCZ PoL, one round successful consensus means that all ETCZs are successfully executing the PoL protocol and accomplishing the synchronous on new generated block sequence. For ETCZs that are failure in the current round consensus, multi-ETCZ PoL will start to re-drawing in these ETCZs. Then, the multi-ETCZ PoL consensus is completed.

For each ETCZ in $\mathbb{S}$, edge devices only store datablocks that are generated in their ETCZ. Then, edge devices that are tight on storage space can choose to become an IED which only requires about $\frac{1}{K}$ of total storage space compared with the traditional blockchain architecture. The part of **P1** is addressed. Meanwhile, edge devices are encouraged to become CDEs by having a higher winning probability and expected revenue with the double-bet mechanism. Considering the heterogeneity of edge devices, the deployment barrier of blockchain has been reduced while without wasting the storage space of high-performance devices. More low-performance devices that are originally

excluded from blockchain consensus can also participate. The security level and tamper difficulty of blockchain are improved. Then, the **P4** is addressed.

5.2 Cross-zone transaction process

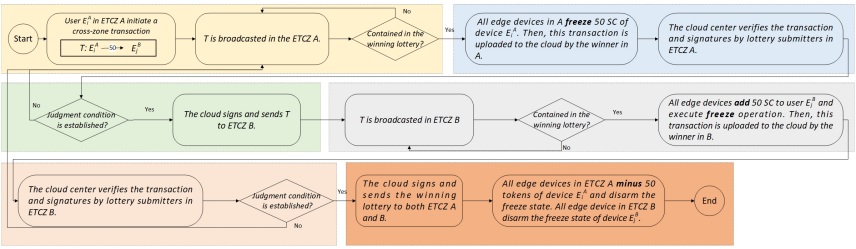


Fig. 5 The cross-zone transaction processing procedure

Although the ETCZ attempts to contain a complete enclosed service scenario, the demand for cross-zone transactions still exists. For example, an edge device E_i^A who wants to participate in a new ETCZ B. Then, this device needs to transmit some service contribution to this new in-zone ID E_i^B. Therefore, in this section, a cross-zone transaction processing procedure is proposed to solve it. For simplicity, we take a transfer transaction as an example in which E_i^A wants to transfer 50 $\mathbb{C}$ to E_i^B. We define this transaction as $T^{50}_{E_i^A \to E_i^B}$, abbreviating for T.

Obviously, a cross-zone transaction requires both ETCZs that are involved to achieve consensus. Because of the data segregation between different ETCZs, a cross-zone transaction cannot be executed directly. Therefore, we divide a cross-zone transaction into the combination of two in-zone transactions and execute them in corresponding ETCZs respectively. With the collaboration of the cloud, the cross-zone transaction between two data-isolated ETCZs can be completed. The illustration of the cross-zone processing procedure is shown in Fig.5.

The execution of a cross-zone transaction can be divided into 6 phases, *Preparation*, *Initiator consensus*, *Initiator judgement*, *Target consensus*, *Target judgement* and *View update*. In *Preparation*, device E_i^A generates and broadcasts a cross-zone transaction T in A. Then, if T has been successfully packaged in a datablock belonging to the winning lottery, the preparation is over. The procedure turns to the initiator consensus.

In initiator consensus, all edge devices in ETCZ A will do a **freeze** operation to device E_i^A that any action causes E_i^A's service contribution to be less than 50 will be considered invalid until T is completed. When ETCZ A achieves consensus on T, the data block that contains this transaction will be uploaded to the cloud center and stored temporarily which is consistent with the synchronization of PoL. If the consensus is not achieved, the T is interrupted and re-executed from preparation. Then, the initiator consensus phase is over.

In initiator judgement, the cloud center verifies the signatures and sends T to the target zone B with a digital signature. Then, the process enters the target consensus.

In target consensus, T will also be processed in B. After achieving the consensus about T, all edge devices add 50 $\mathbb{C}$ to E_i^B and freeze this amount of service contribution. The procedure enters the target judgement.

In target judgement, the cloud verifies datablocks uploaded by ETCZ B. If the judgement condition is established, the procedure enters the view update phase.

The datablock backup from B will be uploaded to the cloud to update the view of B. The view of A is also updated by the block backup uploaded in the initiator consensus. The cloud signs on T to show the process of T is complete and broadcasts it to all devices. Then, all edge devices in A minus 50 tokens from E_i^A and disarm the freeze state. Edge devices in B disarm the freeze state of E_i^B. If the judgement condition fails, the procedure returns to the target consensus phase. The target consensus phase will repeat until T is complete. The whole procedure is over.

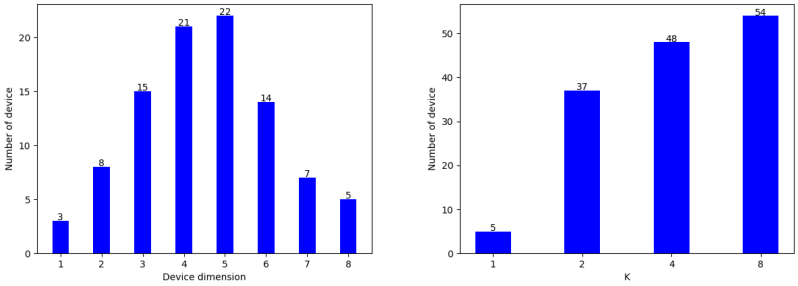
The execution of a cross-zone transaction has been successfully divided into two in-zone transactions to be executed respectively without designing a special procedure. Meanwhile, the security of the cross-zone transaction is ensured through independent verification of ETCZs on both sides of the transaction and the cloud center. Then, the **P5** is addressed.

6 EXPERIMENTAL ANALYSIS

6.1 Experiment settings

In this section, we test the proposed cloud-edge collaboration blockchain service framework that is composed by a cloud server and four client computers in a distributed structure. The cloud server equips a Xeon Core CPU with 3.50 GHz, 256GB RAM, and a 100MB bandwidth internet. The client computer equips an i5 Core CPU with 2.90 GHz, 16GB RAM, and 50MB bandwidth internet. Within an ETCZ, edge devices are simulated on the client computer through virtual machines created by VMware. The propagation delay between cloud and edge devices adopts the average delay time between the cloud server and client computer. The delay time of edge devices between different ETCZs adopts the average delay time between client computers. The propagation time of edge devices in the same ETCZ is simulated by the instant virtual network simulator, Mininet. The hash operation is performed by *hashlib.sha256* library in Python 3.8. The standard drawing difficulty target is set to require approximately 2^{32} times hash operations to be satisfied. The distribution of transaction number per block is $U(1500, 2000)$ with each transaction 1KB.

In experiments, we simulate 100 edge devices' behaviors under three kinds of service scenarios, the complete enclose scenario (A), the nearly enclose scenario (B), and the random chaos scenario (C). The dimension of CDE obeys *Normal distribution*. The detailed distribution of devices is shown in Fig6.



(a) The distribution of devices with different dimension (b) The average number of devices within an individual ETCZ

Fig. 6 The distribution of devices in experiments

For devices whose dimension is smaller than K, the participated ETCZs are randomly selected. For each scenario, we divide the scenario into several ETCZs that are averagely deployed on four client computers. For the complete enclose scenario, there is no data interactive between different ETCZs. For the nearly enclose scenario, we assume that each transaction has a 10% individual probability of becoming a cross-zone transaction. The random chaos scenario is the kind of scenario that data interactive is randomly distributed. Therefore, we set that the probability of a transaction becoming a cross-zone transaction is individual and positive linearly related to the number of the ETCZs in the framework. The Φ has been set as 100 which is the number of edge devices, $\varepsilon = 0.3$ and $\tau = 45s$ in all experiments.

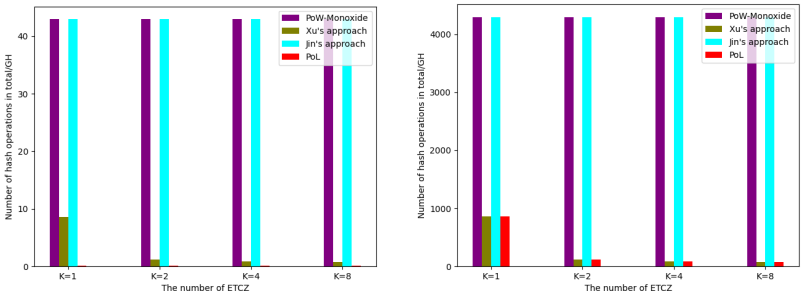
To objective evaluation our proposed framework's performance, we compare our framework with another two state-of-art resource-efficient blockchain approaches, Xu's approach [22] and Jin's approach [23] with a scalable PoW blockchain as the baseline that is implemented based on the idea from Monoxide [7]. Controlled experiments are tested under three kinds of service scenarios. We set 100 edge devices uniformly dividing into 1, 2, 4, and 8 ETCZs respectively. Both complete enclose scenario and nearly enclose scenario are supported to be divided into at most 8 different ETCZs without breaking the cross-zone transaction distribution.

6.2 Experiment results

6.2.1 Computing power comparison

The Fig.7 shows the comparison of computing power requirement in one round consensus between the PoL and the other three protocols. The 7(a) illustrates the computing power requirement on edge devices. The PoL offloads the entire drawing process from edge devices to the cloud. Therefore, the number of hash operations executed on edge devices is approaching zero. Specifically, hash operations are only required in the generation of lotteries. Meanwhile, the PoW-Monoxide baseline and Jin's approach have no target adjustment mechanism. Therefore, these two approaches have the same computing power

requirement on edge devices. Xu's approach and PoL adopt the same target adjustment mechanism that results in the same expected number of hash operation for one round consensus. However, in Xu's approach, the hash operations are executed on edge devices. Therefore, PoL outperforms three other approaches to reducing the computing power on edge and also has the optimal performance in one round consensus.



(a) The average number of hash operations requirement on the edge device

(b) The average number of total hash operations requirement

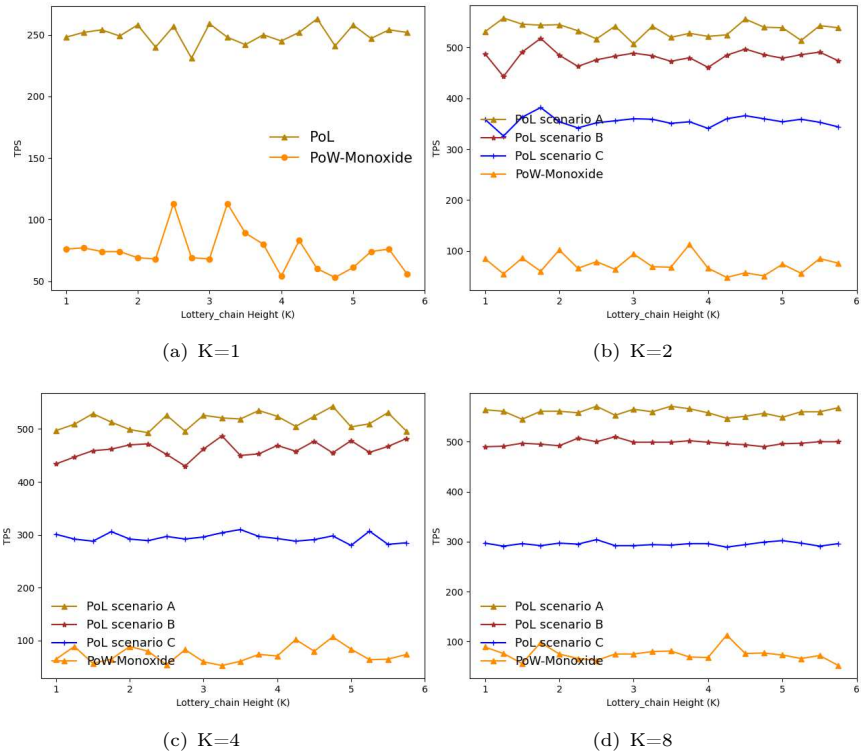
Fig. 7 The required hash operations in one round consensus

6.2.2 TPS comparison

The Fig.8 illustrates the TPS of our proposed framework for each ETCZ under three kinds of service scenarios. In each experiment, we test the average TPS (transaction per second) of 250 successful rounds of PoL from the lottery_chain height from 1000 to 5750 for a stable performance. All experiments are based on the independent simulated mining time by *hashlib.sha256* library in Python 3.8.

In Fig.8(a), we compare the TPS between PoL and PoW-Monoxide when $K = 1$. In this experiment, three service scenarios have no difference because no ETCZ has been divided. The result shows that PoL has a significant advantage compared with PoW-Monoxide in TPS due to the double-chain structure allowing devices to generate multiple datablocks in one round consensus. The time consumption of the drawing process has been evenly distributed to each datablock. Meanwhile, the difficulty function further reduces the drawing difficulty. Therefore, we can have an average of 3.78 times TPS in PoL compared to PoW.

In Fig.8(b-d), we test the TPS of multi-ETCZ blockchain framework under three kinds of service scenarios with 2, 4, and 8 ETCZs respectively. With the increase of K, for complete enclose scenario and near enclose scenario, the TPS has been slightly improved. This is because the higher K is each ETCZ will contain more devices according to Fig.6(b) resulting in a higher average value of P_{E_i} in the drawing process. However, for the random chaos scenario, considering a cross-zone transaction is processed as two independent in-zone transactions in both ETCZs, the increasing proportion of cross-zone



(a) K=1 (b) K=2

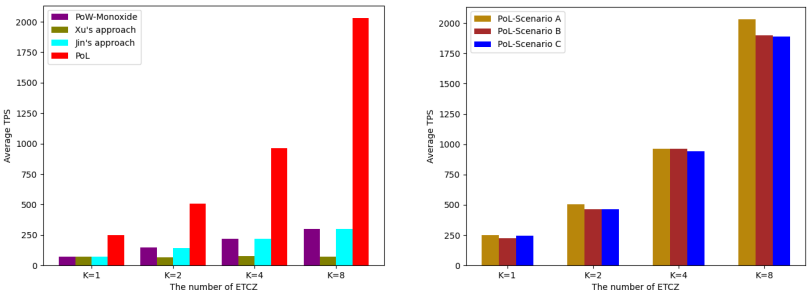
(c) K=4 (d) K=8

Fig. 8 Transaction per second comparison

transactions will significantly pull down the TPS which is reflected in 8(b) and (c). In the comparison of 8(c) and (d), the positive feedback from the reduced drawing time consumption balances the negative feedback from the increasing proportion of cross-zone transactions. The TPS has no significant variation. From experiments, we can see that the high-throughput feature of edge computing is well satisfied. For a clear illustration, the Fig.9 (a) illustrates the comparison for TPS between PoL and three approaches. The Fig.9 (b) shows the trends of the entire framework's TPS with the changing of K for both three scenarios with PoW-Monoxide as the baseline. The experiments prove our framework has excellent scalable and high throughput in the comparison with the other state-of-art approaches.

6.2.3 Storage reduction comparison

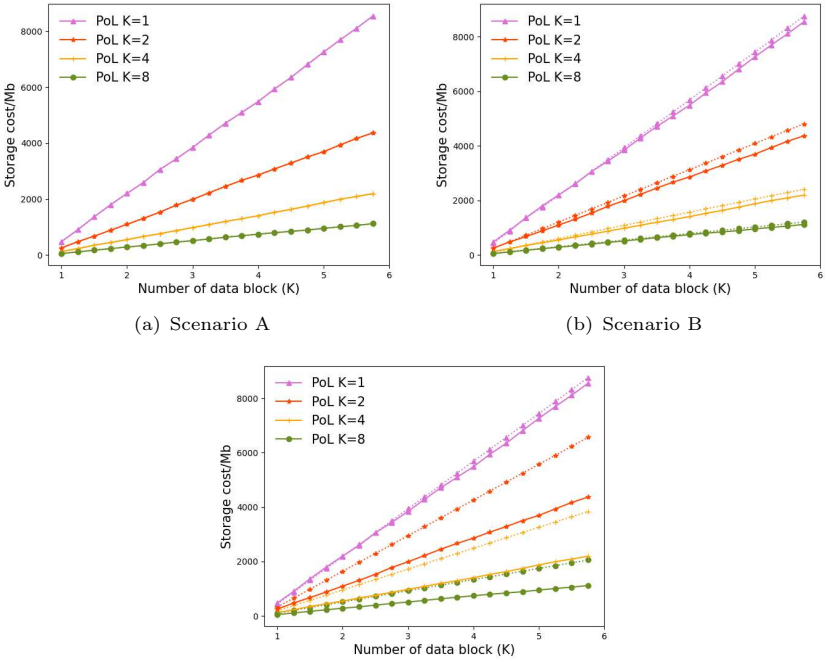
For storage requirement reduction, because of the existence of different blockchain models, there is no direct comparison between different storage reduction approaches. Therefore, we use the idea indicators as the evaluation criterion. For a K-ETCZ framework, the ideal minimum storage requirement for edge devices is $\frac{1}{K}$.



(a) TPS comparison between different approaches

(b) TPS of PoL with different K

Fig. 9 TPS comparison



(a) Scenario A

(b) Scenario B

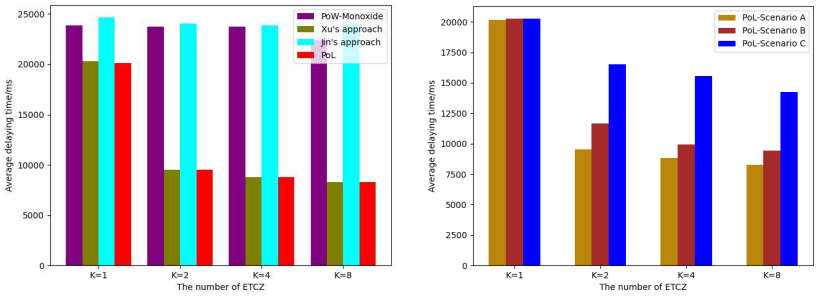
(c) Scenario C

Fig. 10 Storage cost comparison for three scenarios

In Fig.10, we test the variation of the minimum storage consumption for an edge device with the operation of the framework. In Fig.10(a-c), we compare the storage consumption for both three scenarios with the increasing of data blocks. For the proposed framework with K ETCZs, the minimum storage consumption for an edge device is $\frac{1}{K}$ of traditional single-chain blockchain which is represented as the standard value in Fig.10 with a dotted line. The actual storage consumption is represented with a solid line. In experiments, we can see that the deviation of actual value and ideal value comes from the generation

of cross-zone transactions. As we described in section 5, a cross-zone transaction should be independently executed and stored by both ETCZs involved. This makes the edge device may need to store transactions that are generated from other ETCZ resulting in the deviation between the actual value and the ideal value. In contrast, the more ETCZs have been divided the greater deviation will be due to the rising proportion of cross-zone transactions, especially in the random chaos scenario. Despite the existence of deviation, the proposed framework still successfully decreases the minimum storage requirement for edge devices.

6.2.4 Transaction latency comparison



(a) Delaying comparison between different approaches

(b) Delaying of PoL with different K

Fig. 11 Delaying comparison

In Fig.11(a), we compare the average transaction delay between PoL and other approaches. In the experiment, Jin's approach has the same delay as our baseline due to the similar consensus process. Xu's approach and PoL adopt a similar difficulty target adjustment mechanism which is based on persistence P_{E_i}. With the increase of K, the minimum storage requirement for edge devices is reduced that leads to more devices can participate in consensus. Therefore, the higher value K is, the easier for Jin's approach and PoL to select winning devices. The mining process (for Jin's approach) and the drawing process (for PoL) for all ETCZs have been significantly shortened which leads to a lower transaction latency.

In Fig.11(b), we test the transaction delaying time for three scenarios. Even for the random chaos scenario, the division of ETCZ improves the average transaction delaying time with the raising of K. For the complete and nearly enclose scenario, the delaying time improvement is more remarkable. Therefore, with the illustration of the above experiments, the performance of the framework can be markedly improved with a proper division of ETCZ. The effectiveness and performance of our research are proven.

7 CONCLUSIONS AND FUTURE WORK

The biggest obstacle to the fusion of blockchain technology and edge computing is the contradiction between the great computing and storage requirements and the constrained hardware of edge devices. To address this challenging issue, we propose a resource-efficient blockchain framework that can complete offload the computing requirement for edge devices and reduces the minimum storage requirement to $\frac{1}{K}$ of the traditional structure blockchain. Meanwhile, with the target difficulty adjustment function and the double-bet mechanism, the transaction throughput and latency of the blockchain framework have been significantly improved that can satisfy the high-throughput and low-latency service demand of the edge computing environment. Finally, experiments verify the performance, efficiency, and effectiveness of our framework with the comparison with other start-of-art resource-efficient blockchain approaches.

Our ongoing work is to utilize prior domain knowledge to further improve our framework. In the current version of the framework, the cross-zone transaction requires at least two rounds of consensus to complete the processing in ETCZs of both sides involved under the collaboration of the cloud. This greatly limits the efficiency and relies on the trusted level of the cloud. Besides, the cross-transaction is not support withdrawal which makes the cross-transaction may have to wait for a long time to be processed again if the first time consensus fails. In addition, only two different ETCZs are currently supported for cross-zone transactions. There is no good solution for a cross-zone transaction that involves multiple ETCZs yet to further improves the parallelism of the blockchain transaction. We will continue to explore new methods to solve the above problems.

8 Declarations

Ethics approval and consent to participate. This article is compliance with ethical standards and the consent for publication is not applicable for it.

Availability of data and materials. Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

Competing interests. All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this article.

Funding. This article is funded by the National Key Research and Development Program of China 2020YFB2009503.

Authors' contributions. Kaiyu Wang is contributed to this article's methodology, formal analysis, experiments, investigation, and writing-original. The writing-review and editing were performed by Zhiying Tu and Shufan. Zhenzhou Ji has contribution on writing-editing, resource and supervision. All authors read and approved the final manuscript.

Acknowledgements. Thanks to the support of the National Key Research and Development Program of China 2020YFB2009503.

References

[1] Ferrag M A, Derdour M, Mukherjee M, et al. Blockchain technologies for the internet of things: Research issues and challenges. IEEE Internet of Things Journal, 2018, 6(2): 2188-2204.

[2] Contreras-Castillo J, Zeadally S, Guerrero-Ibañez J A. Internet of vehicles: architecture, protocols, and security. IEEE internet of things Journal, 2017, 5(5): 3701-3709.

[3] Khan W Z, Ahmed E, Hakak S, et al. Edge computing: A survey. Future Generation Computer Systems, 2019, 97: 219-235.

[4] Xu, Chenhan, et al. "Making big data open in edges: A resource-efficient blockchain-based approach." IEEE Transactions on Parallel and Distributed Systems 30.4 (2018): 870-882.

[5] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system. Decentralized Business Review, 2008: 21260. URL http://www.bitcoin.org.

[6] He B, Bai K J. Digital twin-based sustainable intelligent manufacturing: A review. Advances in Manufacturing, 2021, 9(1): 1-21.

[7] Wang, Jiaping, and Hao Wang. "Monoxide: Scale out blockchains with asynchronous consensus zones." 16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19). 2019.

[8] Khan W Z, Ahmed E, Hakak S, et al. Edge computing: A survey. Future Generation Computer Systems, 2019, 97: 219-235.

[9] Kang J, Yu R, Huang X, et al. Blockchain for secure and efficient data sharing in vehicular edge computing and networks. IEEE Internet of Things Journal, 2018, 6(3): 4660-4670.

[10] Guo S, Hu X, Guo S, et al. Blockchain meets edge computing: A distributed and trusted authentication system. IEEE Transactions on Industrial Informatics, 2019, 16(3): 1972-1983.

[11] Bahutair M, Bouguettaya A. Blockchain-based Trust Information Storage in Crowdsourced IoT Services.2021 IEEE International Conference on Web Services (ICWS). IEEE, 2021: 608-617.

[12] Xu, Xiaolong, et al. "BeCome: Blockchain-enabled computation offloading for IoT in mobile edge computing." IEEE Transactions on Industrial

Informatics 16.6 (2019): 4187-4195.

[13] Kang J, Yu R. et al. "Blockchain for secure and efficient data sharing in vehicular edge computing and networks." IEEE Internet of Things Journal 6.3 (2018): 4660-4670.

[14] Lu, Yunlong, et al. "Blockchain and federated learning for privacy-preserved data sharing in industrial IoT." IEEE Transactions on Industrial Informatics 16.6 (2019): 4177-4186.

[15] Gao, Ying, et al. "Blockchain based IIoT data sharing framework for SDN-enabled Pervasive Edge Computing." IEEE Transactions on Industrial Informatics 17.7 (2020): 5041-5049.

[16] Liu, Lei, et al. "Blockchain-enabled secure data sharing scheme in mobile-edge computing: An asynchronous advantage actor–critic learning approach." IEEE Internet of Things Journal 8.4 (2020): 2342-2353.

[17] Gai, Keke, et al. "Permissioned blockchain and edge computing empowered privacy-preserving smart grid networks." IEEE Internet of Things Journal 6.5 (2019): 7992-8004.

[18] Guo, Shaoyong, et al. "Blockchain meets edge computing: A distributed and trusted authentication system." IEEE Transactions on Industrial Informatics 16.3 (2019): 1972-1983.

[19] Pustišek, Matevž, Andrej Kos, and Urban Sedlar. "Blockchain based autonomous selection of electric vehicle charging station." 2016 international conference on identification, information and knowledge in the Internet of Things (IIKI). IEEE, 2016.

[20] Tselios, Christos, Ilias Politis, and Stavros Kotsopoulos. "Enhancing SDN security for IoT-related deployments through blockchain." 2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN). IEEE, 2017.

[21] Chen, Zhonglin, et al. "A security authentication scheme of 5G ultra-dense network based on block chain." IEEE Access 6 (2018): 55372-55379.

[22] Xu, Chenhan, et al. "Making big data open in edges: A resource-efficient blockchain-based approach." IEEE Transactions on Parallel and Distributed Systems 30.4 (2018): 870-882.

[23] Xie Jin, et al. "Resource-efficient DAG Blockchain with Sharding for 6G Networks." IEEE Network (2021).

[24] Zhang, Fan, et al. "REM: Resource-efficient mining for blockchains." 26th USENIX Security Symposium (USENIX Security 17). 2017.

[25] Liu, Yinqiu, et al. "LightChain: a lightweight blockchain system for industrial internet of things." IEEE Transactions on Industrial Informatics, 15.6 (2019): 3571-3581.

[26] Liu, Zhixin, et al. "Efficient QoS support for robust resource allocation in blockchain-based femtocell networks." IEEE transactions on industrial informatics 16.11 (2019): 7070-7080.

[27] Wood, Gavin. "Ethereum: A secure decentralised generalised transaction ledger." Ethereum project yellow paper 151.2014 (2014): 1-32.

[28] Castro, Miguel, and Barbara Liskov. "Practical byzantine fault tolerance." OSDI. Vol. 99. No. 1999.

[29] Douceur, John R. "The sybil attack." International workshop on peer-to-peer systems. Springer, Berlin, Heidelberg, 2002.

[30] Aponte-Novoa, Fredy Andres, et al. "The 51% Attack on Blockchains: A Mining Behavior Study." IEEE Access, 9 (2021): 140549-140564.

[31] "Bitcoin Chart" https://www.blockchain.com/zh-cn/charts.

[32] Androulaki, Elli, et al. "Hyperledger fabric: a distributed operating system for permissioned blockchains." Proceedings of the thirteenth EuroSys conference. 2018.

[33] King, Sunny, and Scott Nadal. "Ppcoin: Peer-to-peer crypto-currency with proof-of-stake." self-published paper, August 19.1 (2012).

[34] Nothing at stake attack Ethereum. URL https://github.com/ethereum/wiki/wiki/Proof-of-Stake-FAQs/what-is-the-nothing-at-stake-problem-and-how-can-it-be-fixed.

[35] Long range attack Ethereum. URL https://blog.ethereum.org/2014/05/15/long-range-attacks-the-serious-problem-with-adaptive-proof-of-work/.

[36] Kokoris-Kogias, Eleftherios, et al. "Omniledger: A secure, scale-out, decentralized ledger via sharding." 2018 IEEE Symposium on Security and Privacy (SP). IEEE, 2018.

[37] Luu, Loi, et al. "A secure sharding protocol for open blockchains." Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. 2016.

[38] Eyal, Ittay, and Emin Gün Sirer. "Majority is not enough: Bitcoin mining is vulnerable." International conference on financial cryptography and data security. Springer, Berlin, Heidelberg, 2014.

[39] Liao, Kevin, and Jonathan Katz. "Incentivizing blockchain forks via whale transactions." International conference on financial cryptography and data security. Springer, Cham, 2017.

[40] Bag, Samiran, Sushmita Ruj, and Kouichi Sakurai. "Bitcoin block withholding attack: Analysis and mitigation." IEEE Transactions on Information Forensics and Security 12.8 (2016): 1967-1978.

[41] Kwon, Yujin, et al. "Be selfish and avoid dilemmas: Fork after withholding (faw) attacks on bitcoin." Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. 2017.

9 Appendix

9.1 Appendix A

In the comparison between R_{E_i} and $\sum_{1\sim n}^{i} R^*_{E_i^\chi}$, first we have $\kappa_{E_i} = \frac{\sum_{\mathbb{L}_j} \mathbb{F}(E_m)}{\mathbb{F}(E_i)}$ strictly greater than $\kappa^* = \frac{\sum_{\mathbb{L}_j - E_i^\chi} f(E_i^\chi) + \sum_{1\sim n} f(E_i^\chi)}{\sum_{1\sim n} f(E_i^\chi)}$. This can be deduced to be established if $\mathbb{F}(E_i)$ is strictly smaller than $\sum_{1\sim n} f(E_i^\chi)$. Only one lottery will be selected in the classical single-ETCZ PoL. Therefore, $\mathbb{F}(E_i) < \sum_{1\sim n} f(E_i^\chi)$ is always satisfied. Then, the function of κ can be expressed as the following function:

$$\kappa = \frac{\lambda + \varphi}{\varphi} \tag{14}$$

in which $\lambda = \sum_{\mathbb{L}_j} \mathbb{F}(E_m) - \mathbb{F}(E_i) = \sum_{\mathbb{L}_j - E_i^\chi} f(E_i^\chi)$ is a constant value, and φ is the sum of difficulty target of device E_i's lotteries. The derivative of κ about φ is:

$$\frac{\partial \kappa}{\partial \varphi} = \frac{-\lambda}{\varphi^2} \tag{15}$$

which is always negative. Therefore, we have $\kappa_{E_i} > \kappa^*$.

With a constant value of $\mathbb{C}_{E_i}$ and $\varepsilon \in (0, 1)$, it is easy to derive that $\Theta_{E_i} > \sum_{1\sim n}^{i} \Theta^*_{E_i^\chi}$ (the detailed derivation is similar to appendix B). Then, with a greater molecule and smaller denominator in the revenue expression, R_{E_i} is proven to be strictly greater than $\sum_{1\sim n}^{i} R^*_{E_i^\chi}$.

9.2 Appendix B

We define two functions $f_1 = (\mathbb{C}_{E_i})^{1+\varepsilon}$ and $f_2 = m \sum_{1\sim m} (\mathbb{C}^m_{E_i})^{1+\varepsilon}$. For f_1, we have

$$\frac{\partial f_1}{\partial \varepsilon} = (\mathbb{C}_{E_i})^{1+\varepsilon} \times ln\mathbb{C}_{E_i} \tag{16}$$

f_1 is positive related with the value of ε.

For f_2, we have

$$\frac{f_2}{m} = \sum_{1 \sim m} (\mathbb{C}_{E_i}^m)^{1+\varepsilon} \tag{17}$$

f_2 is negative related with the value of m. Then, we set ε as the maximum value 1 and m as the minimum value 2. The value of f_1 and f_2 are

$$f_1(\varepsilon = 1) = (\mathbb{C}_{E_i})^2 \tag{18}$$

$$f_2(m = 2) = 2 \sum_{1 \sim 2} (\mathbb{C}_{E_i}^2)^2 = 2[(\mathbb{C}_{E_i}^1)^2 + (\mathbb{C}_{E_i}^2)^2] \tag{19}$$

We substitute into $\mathbb{C}_{E_i} = \mathbb{C}_{E_i}^1 + \mathbb{C}_{E_i}^2$. Then, we have

$$f_2(m = 2) - f_1(\varepsilon = 1) = (\mathbb{C}_{E_i}^1)^2 + (\mathbb{C}_{E_i}^2 - 2\mathbb{C}_{E_i}^1) \times \mathbb{C}_{E_i}^2 \tag{20}$$

in which we have $f_2(m = 2) - f_1(\varepsilon = 1) > 0$ when $\mathbb{C}_{E_i}^1 \neq \mathbb{C}_{E_i}^2$. Else, $f_2(m = 2) = f_1(\varepsilon = 1)$. Then, we can prove that no matter how to split $\mathbb{C}$, the CDE can always obtain higher expected revenue than IDE with the same amount of service contribution. Besides, we can prove that the CDE with higher dimension will have higher through repeated the above derivation. For a low-dimension CDE E_i, he can split his service contribution in ETCZ i to participate in ETCZ j, $\mathbb{C}_{E_i}^i = \mathbb{C}_{E_i*}^i + \mathbb{C}_{E_i}^j$. Then, it is easy to be proven that E_i can obtain higher revenue. Therefore, with a constant $\mathbb{C}$, the higher dimension the CDE is the higher expected revenue he will have.