Computer Networks 274 (2026) 111827

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Game theory based incentive mechanism for a bi-tiered public blockchain system

Jing Li, Tao Xie *, Yiwei Liu

College of Computer science and technology, National University of Defense Technology, Changsha, 410073, Hunan, China

ARTICLE INFO

Keywords:
Blockchain
Incentive
Game theory
Reputation

ABSTRACT

Incentive mechanisms serve as the driving force behind blockchain operations, bearing significant implications for the practical applications of blockchain technology. However, existing incentive mechanisms predominantly focus on incentivization from a monetary perspective, with minimal incorporation of punitive measures, thus exhibiting limited efficacy in regulating system entities. Furthermore, the assurance of providing adequate incentives for the full nodes of the system cannot be guaranteed, potentially resulting in node attrition and posing threats to blockchain security. In this paper, we propose EVONIncentive, a bi-tiered blockchain hybrid incentive scheme based on game theory. It adopts a hybrid incentive form of monetary and reputation, introducing fee selection mechanisms, compensation mechanisms, and supervisory group penalty mechanisms, aiming to ensure the sustainability, fairness, and security of the Blockchain system. Two Stackelberg dynamic game models are constructed based on different issues, and through algorithm design and mathematical derivation, the optimal incentive allocation is obtained. Finally, through numerical simulation experiments, the specific values of relevant parameters such as mining rewards, transaction fee, and compensation are determined. A comparative experiment with the Ethereum 1.0 incentive mechanism is designed. The results from both game-theoretic analysis and numerical simulations demonstrate that EVONIncentive can enhance the sustainability and security of the system, showing significant advantages over Ethereum 1.0 in terms of system social welfare, user income gap, and fairness.

1. Introduction

Since the publication of the Bitcoin whitepaper titled “Bitcoin: A Peer-to-Peer Electronic Cash System” in 2008 [1], blockchain technology has emerged as a widely adopted technology in modern society. Its characteristics such as decentralization, security, immutability, and anonymity have propelled innovations in real-world applications across various domains including financial systems [2–4], supply chain [5,6], healthcare [7,8], and government governance [9,10].

Public chains represent the earliest researched and most widely applied domain within blockchain technology. Decentralization constitutes the intrinsic attribute of public chain technology, wherein consensus mechanisms such as PoW [1], PoS [11], and DPoS [12] facilitate the attainment of consensus among a broad spectrum of system entities through predefined rules, thus achieving the fundamental objective of decentralized trust in the absence of third-party authority. Consequently, consensus mechanism has remained central to research endeavors concerning public chains [13–21]. In the architecture of public chain technology, incentive mechanisms, operating through economic equilibrium, encourage node participation in the system, deter malicious node behavior, and provide impetus for the sustained operation of public chain systems.

However, compared to the increasingly refined design of consensus mechanisms today, incentive mechanism design still exhibits several shortcomings. Firstly, the predominant adoption of a singular incentive form, namely the use of monetary incentives to encourage node participation in system operation, as seen in Bitcoin’s utilization of block rewards and transaction fee. In reality, miners and validators engage in a repeated game, wherein malicious nodes may profit in the long run by deviating from protocol rules, forgoing short-term optimal strategies through attacks such as selfish mining [22] and block withholding attacks [23]. Furthermore, permissionless blockchain environments generally lack explicit incentives for forming cooperative alliances for the common good; instead, nodes may even seek to maximize their own gains at the expense of the system. Consequently, purely monetary incentive schemes can hardly guarantee participant honesty and cooperative behavior through incentive compatibility and cooperative game properties alone. Secondly, the lack of punitive measures against mali-

* Corresponding author.
E-mail addresses: lijing18@nudt.edu.cn (J. Li), hamishxie@vip.sina.com (T. Xie), 610002858@qq.com (Y. Liu).

https://doi.org/10.1016/j.comnet.2025.111827
Received 19 May 2025; Received in revised form 16 October 2025; Accepted 1 November 2025
Available online 4 November 2025

cious behavior, which similarly compromises system security. Thirdly, the neglect of cost burden on system full nodes, which comprises both bandwidth costs and storage costs. Bandwidth costs, incurred by continuously uploading and downloading data to propagate new transactions and blocks and to synchronize with the network, represent a recurring and fluid expense. Storage costs, on the other hand, arise from persistently maintaining a complete copy of the blockchain ledger and grow linearly over time with usage. Neglecting this burden leads to the attrition of full nodes, thereby posing a direct threat to the decentralization, security, and sustainability of public blockchains. Early incentive studies largely overlooked this issue, often due to their primary focus on consensus participants and simplified game-theoretic models. Over time, the severity of ignoring the cost burden on full nodes has become increasingly evident. Taking Ethereum 1.0 launched in 2014 [13] as an example, the massive transaction broadcasts and extensive data storage impose significant and escalating operating costs on full nodes responsible for complete block broadcasting, validation, and consistent storage. By 2020, the total cost of operating full nodes amounted to $20 million per month (approximately 10,000 full nodes[1]), with each full node incurring a monthly operating cost of $2,000,[2] while Ethereum users in the first half of 2020 paid an average monthly transaction fee of $7.3 million.[3] Failure to offset operating costs with transaction fee can lead to full node attrition, thereby diminishing the system's decentralization, posing security issues.[4] Fourthly, the oversight of user compensation. Using Ethereum 1.0 as an example again, transaction confirmation relies on a purely competitive bidding model. In essence, users with higher transaction fee can expedite their transaction confirmation, thus garnering more profits, while other regular users endure delayed or even unconfirmed transactions. Without adequate compensation, users may leave the system due to low returns or poor experiences, thereby compromising system vitality.

In addressing the aforementioned issues, this paper builds upon a previously proposed bi-tiered public chain system, EVONChain [24], introduces a hybrid incentive scheme termed EVONIncentive. EVONIncentive adopts a dual incentive approach combining monetary and reputation rewards. It provides an ample transaction fee as monetary incentives to the inner-layer cloud nodes (i.e., full nodes) responsible for transaction packaging and block creation, as well as for recording complete ledger data. Simultaneously, reputation-based incentives are employed, where the behavior of these nodes determines corresponding reputation rewards or penalties. Cloud nodes with reputations falling below a predefined threshold will be unable to receive monetary rewards until their reputation is restored to the threshold level. Outer-layer miner nodes participating in and successfully completing PoI (Proof of Intelligence) tasks receive reasonable system monetary rewards. For outer-layer user nodes solely engaging in transactions, mechanisms for transaction fee selection and compensation are designed. Multiple transaction fee options are provided to users to ensure that the minimum fee sufficiently covers the bandwidth and storage costs of system full nodes. Upon proposing a transaction, users are required to compensate other users who experience longer wait times due to their transaction submissions, thereby ensuring that all system users derive some benefit from transactions. This incentivizes user participation in the system and enhances its decentralization level.

The main contributions of this paper are as follows:

1) We have designed a hybrid incentive scheme based on monetary and reputation. Different incentive forms are devised according to the corresponding functions of nodes within the public chain system. Furthermore, in anticipation of potential malicious behaviors by nodes, we have designed corresponding supervisory and punitive mechanisms to ensure system security and stability.
2) We have proposed fee selection and compensation mechanisms. These mechanisms offer system user nodes transparent options for transaction fee, ensuring that the minimum transaction fee covers at least the operating costs of system full nodes. Additionally, compensation is provided for potential transaction delays experienced by users, incentivizing broad user participation in the system and enhancing decentralization level.
3) Through the integration of relevant game theory principles, we have constructed two Stackelberg game models for the proposed mechanisms. By deriving Nash equilibrium conditions in these game models, we have determined the optimal values or ranges of parameters within the mechanisms.
4) We have conducted comparative experiments between EVONIncentive and the incentive mechanism of Ethereum 1.0 using identical parameter settings. Experimental results indicate that EVONIncentive exhibits significant advantages over Ethereum 1.0 in terms of increasing system social welfare, reducing income disparities among users, and enhancing fairness.

Paper organization The rest of this article is organized as follows: Section 2 introduces the related work. Section 3 gives a brief introduction to the system model of EVONIncentive. Section 4 describes some key mechanisms of EVONIncentive in detail. Section 5 presents the security analysis. Section 6 introduces the Stackelberg game models and Section 7 shows the experimental results. Future work and conclusions of the study are in Section 8. The various notations used in this paper are mentioned in Table 1.

2. Related work

2.1. EVONChain

EVONChain [24] is a novel, scalable, and decentralized public chain architecture characterized by high throughput and fast confirmation. EVONChain hierarchically organizes the public chain network into an inner-layer high-performance cloud computing node network and an outer-layer low-performance user node network. By decoupling the block consensus process, inner-layer cloud nodes are responsible for operations such as transaction collection and validation, pending block packaging and block header broadcasting, verified block verification, and consistent storage. Meanwhile, outer-layer user nodes, which are low-performance nodes, handle the mining process, utilizing block headers instead of complete blocks for broadcasting in the outer-layer network, thereby significantly reducing the bandwidth burden on the low-performance outer-layer network. Moreover, the inner-layer network employs the graph chain accounting protocol ORIC [19], which increases the bandwidth utilization rate of the inner-layer network to the theoretical limit of 50 %. Outer-layer network nodes utilize the Proof of Intelligence (PoI) smart computing proof algorithm, a enhanced variants of PoW, instead of the traditional PoW mining algorithm, effectively resisting ASIC computation and ensuring the decentralization and security of the system. Simulation results demonstrate that under a 1.5 Gbps internal network bandwidth, EVONChain can achieve a throughput of nearly 375,000 transactions per second, with a block interval of 10 s and a orphan block rate of less than 7 %.

2.2. Incentive in blockchain system

The incentive mechanism of blockchain integrates factors such as the issuance mechanism and distribution mechanism of economic incentives into the blockchain technology system to drive the benign development cycle of the system. Han et al. [25] suggests that blockchain

[1] https://www.ethernodes.org/history
[2] https://medium.com/quiknode/welcoming-the-newest-member-of-the-quiknode-family-the-eth-archive-node-ac66201e0793
[3] https://studio.glassnode.com
[4] The issue of insufficient transaction fee in Ethernum was ultimately addressed in the London upgrade on August 4, 2021, through the proposal of EIP-1559, which altered the fee mechanism.

Table 1
Key notations.

Symbol	Explanation
$\mathcal{C}$	Set of cloud nodes
n_c	Number of cloud nodes
π_{c_i}	Profit function of cloud node i
R_i	Reputation value of cloud node i
R_{thre}	Reputation threshold
$R_{\max}$	Maximum reputation value
$\mathcal{B}$	System transaction pool
η	Unit byte cost for cloud nodes
$\eta_{bandwidth}$	Unit byte bandwidth cost for cloud nodes
$\eta_{storage}$	Unit byte storage cost for cloud nodes
$\boldsymbol{\zeta}$	Vector of unit byte transaction fee selection for user nodes
I	Number of options for unit byte transaction fee
ζ_i	The ith unit byte fee
$\mathcal{U}$	Set of user nodes
n_u	Number of user nodes
π_{u_i}	Profit function of user node i
f	Transaction fee
σ	Transaction size
$\boldsymbol{C}$	Vector of unit byte compensation
U_H/U_L	Utility of the users' transaction
U_H^{net}/U_L^{net}	Net utility of users
Ψ_i	Set of transactions packaged by cloud node i
λ_i	Transaction generation rate of user node i
ξ_i	Time cost coefficient of user node i
τ_i	Transaction waiting time of user node i
$\mathcal{M}$	Set of miner nodes
n_m	Number of miner nodes
π_{m_i}	Profit function of miner node i
M_0	Mining monetary reward for miner nodes
γ_i	Non-mining income factor of miner node i
p_i	Total computing power of miner i
p_i^m	Computing power devoted in the mining process by miner i
$\boldsymbol{P}$	Set of computing power devoted in mining by miner nodes
T	Main block time
D_p	Main block difficulty requirement
Φ	System throughput
k	Throughput adjustment parameter

incentive forms can generally be classified into two categories: monetary incentives and non-monetary incentives. Non-monetary incentives can further be categorized into credit incentives, reputation incentives, and gamified incentives. Huang et al. [26] discusses the issuance mechanism and distribution mechanism of blockchain tokens, providing examples of token issuance from the perspectives of computing, storage, and transmission, which are the most concerning aspects of the network. Yu et al. [27] proposes that blockchain incentive mechanisms include not only reward systems but also punishment systems. Liu et al. [28] outlines the compatibility of blockchain incentives, using game theory to analyze the motivations for interaction among different components in the blockchain. Studies in [29] investigate the incentive fairness of several blockchain protocols, including PoW, ML PoS, SL PoS, and C-PoS.

2.3. Analysis on existing blockchain incentive mechanism

Analysis of existing blockchain incentive mechanisms primarily focuses on the interaction between users and miners, the interaction between transaction confirmation delay and transaction fee, and system security. Huberman et al. [30] suggests that Bitcoin users decide to join or leave the blockchain system based on utility functions, while miners weigh the cost of computing power against mining rewards. Tsabary and Eyal [31] argues that issuing transaction fee alone is insufficient to incentivize miners, leading to a mining gap. Jiang and Wu [32] posits that rational miners in the Bitcoin system are more willing to include high transaction fee transactions in blocks and generate large blocks, while low fee transactions are ignored. Ma et al. [33] models the mining game of miners as a dynamic game and analyzes the equilibrium state of the PoW consensus mechanism. Altman et al. [34] focuses on miner decision-making, establishing a non-cooperative game model to determine which blockchain miners will mine. Aldweesh [35] studies the rationality of Gas fee settings for CPU usage in Ethereum users. Yao [36] analyzes some user fee mechanisms in Bitcoin. Xue et al. [37] investigates the public cost model and private cost model of miners in Bitcoin mining pools. Rizun [38], Zhang and Preneel [39] propose that both block generation time and broadcast time affect transaction wait time and thus transaction fee. Jonathan and Thorsten [40] classifies miners into honest nodes and malicious nodes, analyzing the motivations of malicious nodes for double-spending attacks based on total computing power. Pagnotta [41] focuses on security threats caused by token economics in Bitcoin. Goffard [42] analyzes the probability of a malicious chain catching up with the public chain. Wang and Wu [43] suggests that new nodes should reward full nodes for paying for historical block records to enhance system security. Sompolinsky and Zohar [44] discusses the important role of incentive measures in Bitcoin protocol security.

2.4. Incentive mechanisms in collaborative and distributed systems

The use of incentive mechanisms to guide participant behavior is a well-established challenge in various decentralized and collaborative systems beyond blockchain. In crowdsourcing and human computation [45–47], incentive schemes are designed to encourage high-quality data contributions. Similarly, in peer-to-peer (P2P) networks [48,49], incentives help identify malicious nodes and encourage resource sharing. More recently, the rise of collaborative machine learning has introduced new dimensions to this problem. In federated learning (FL), incentive mechanisms aim to motivate data owners to contribute resources without compromising data privacy. For example, Zeng et al. [50] proposed a multi-dimensional game-theoretic model for cross-silo FL, analyzing how organizations strategically provision resources like data and computation to improve global model accuracy. Parallel to FL, the paradigm of client-assisted foundation model training has emerged, which leverages distributed computing resources from numerous clients for large-scale model pre-training. Here, incentive mechanisms like the online procurement auction proposed in [51] address unique challenges such as asynchronous client arrivals and multi-round training processes, with a focus on minimizing social cost while ensuring truthfulness. Although the studies above share the high-level goal of using incentives to foster cooperation, their design objectives and operational environments differ fundamentally from those of public blockchain systems. Works like [50,51] primarily aim at optimizing learning performance or resource utilization in a relatively structured environment. In contrast, public blockchains operate in a more adversarial setting, where the paramount goals include maintaining consensus security, ensuring ledger consistency against Byzantine behaviors, and preserving long-term decentralization. Therefore, our contribution in EVONIncentive is a novel hybrid incentive mechanism specifically tailored to the blockchain model, integrating monetary and reputation elements with fee selection and compensation to meet its unique security and sustainability requirements.

3. System model

The bi-tiered public chain architecture on which EVONIncentive relies is illustrated in Fig. 1. The overall architecture is primarily divided into two layers: the high-performance cloud nodes constitute the inner-layer peer-to-peer network, responsible for transaction collection and package, block generation, broadcast, verification and storage. Edge nodes form the outer-layer network. These outer-layer edge nodes can serve both as just users generating transactions and as miners participating in the Proof of Intelligence process. Interactions between inner and outer layer nodes occur through transactions and block headers. Transactions proposed by user nodes are collected and sent to the inner-layer

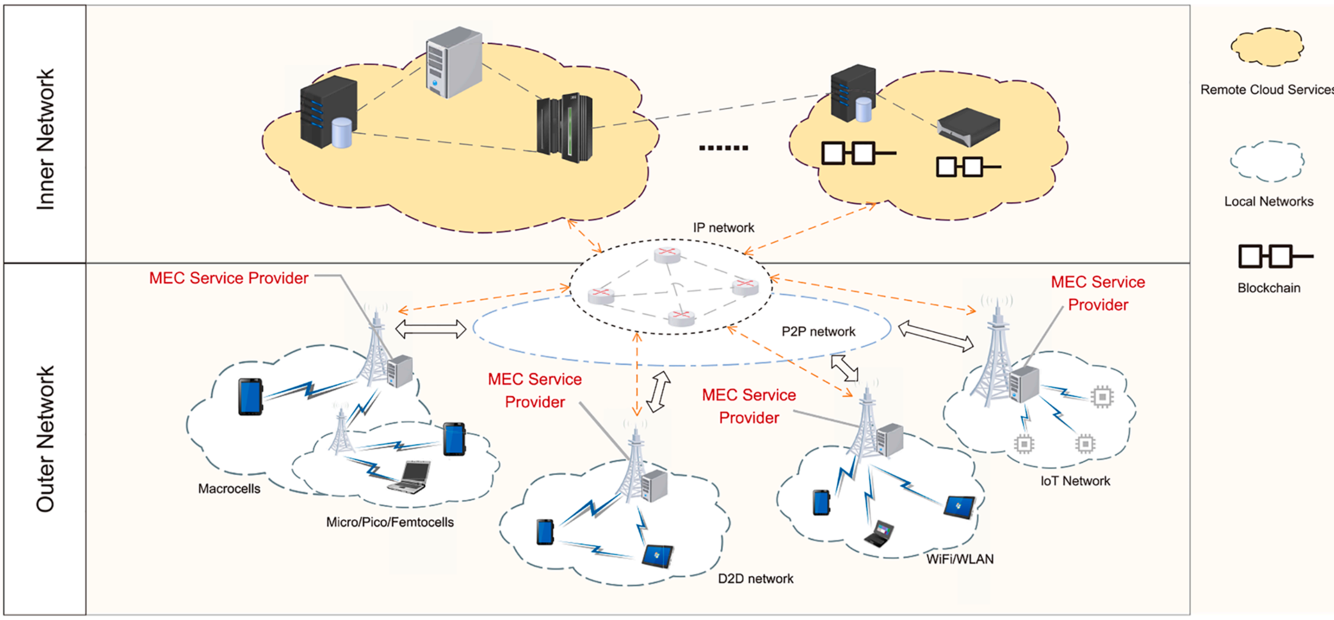


Fig. 1. Bi-tiered blockchain design with Cloud Computing and Mobile Edge Computing (MEC) [24].

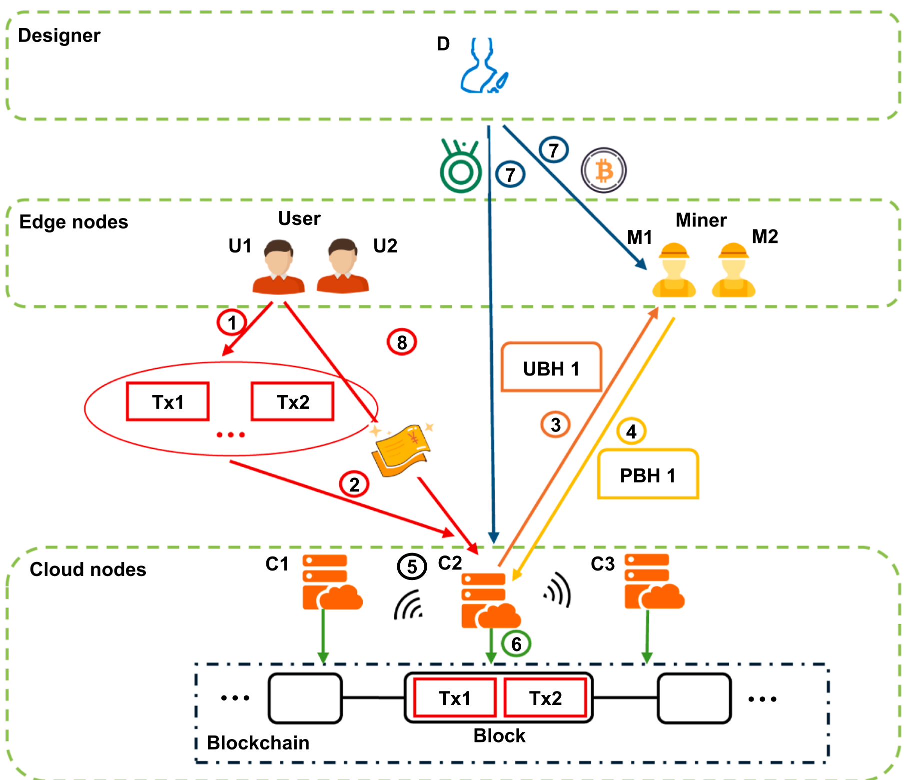


Fig. 2. Operating process of EVONIncentive.

network by Mobile Edge Computing (MEC) service providers.[5] Inner-layer nodes then package transactions into pending blocks, and the block headers of these pending blocks (Unproved Block Header, UBH) are sent to the outer-layer network via MEC. After competing in the Proof of Intelligence process, outer-layer nodes transmit the headers of proven blocks (Proved Block Header, PBH) back to the inner-layer. After verification by inner-layer nodes, complete proven blocks are formed and ultimately added to the blockchain, achieving consensus.

Fig. 2 illustrates the operational workflow of EVONIncentive, starting from the roles of designer, outer-layer user nodes, outer-layer miner nodes, and inner-layer cloud nodes. It mainly consists of five stages: mechanism determination, transaction generation, block packaging, mining, and incentive allocation. The detailed explanation is as follows:

① Mechanism determination by the designer: The system designer (D) first determines the specific mechanisms for edge nodes and cloud nodes. Taking Ethereum as an example, in 2021, the Ethereum

[5] MEC receives monetary rewards from the system based on the duration of its service, which is a fixed value and is not discussed in this paper

online community proposed Ethereum Improvement Proposal 1559 (EIP-1559), which changed the fee model to base fee + tip.

② Transaction generation by user nodes: User node U1 in the outer-layer edge nodes generates two transactions (Tx1 and Tx2) and, according to the systemś fee selection mechanism, assigns a unit byte fee to each transaction. The transaction fee equals the transaction size multiplied by the unit byte fee. After generation, transactions enter the transaction pool.

③ Block packaging by inner-layer cloud nodes: Cloud node C2 select Tx1 and Tx2 from the transaction pool, validate and package them into pending blocks. Through the connection channel between inner and outer layers constructed by MEC, the header of pending block 1 (UBH1) is sent to the outer-layer network.

④ Mining by miner nodes: Miner nodes M1 in the outer-layer edge nodes receive the header of pending block 1 and compete in the Proof of Intelligence process. Once a miner node finds a magic square that meets the difficulty requirements, it combines it with other data to form the header of proven block 1 (PBH1), which is then sent back to the inner-layer network.

⑤ Incentive allocation: After verifying the received header of the proven block, C2 packages the complete block and broadcast it within the inner-layer network. All cloud nodes incorporate this complete block into the global blockchain. After transaction execution, the final incentive allocation process takes place. Outer-layer miner nodes who complete the Proof of Intelligence receive corresponding system monetary rewards based on the block difficulty. The cloud node that generated the block receives all transaction fee included in the block and also receives system reputation rewards.

4. Mechanism design

4.1. Monetary + reputation hybrid incentive mechanism

Monetary incentives are typically used to provide tangible rewards as direct incentives for system entities, incentivizing node participation in the system from an economic perspective. Reputation incentives, on the other hand, focus more on encouraging mutual cooperation among nodes by regulating node behavior through reputation values. Each of these incentive forms has its limitations: monetary incentives provide incentive for node participation driven by self-interest but cannot control dishonest behavior after node participation, while reputation incentives may not meet the economic needs of nodes. Therefore, EVONIncentive adopts a hybrid incentive form of monetary + reputation to overcome these issues.

Incentives for Inner-Layer Cloud Nodes Inner-layer cloud nodes, as a small number of archival nodes in the system, are responsible for transaction processing and blockchain storage, which largely determine the direction of the global blockchain ledger and are also critical areas where the system is vulnerable to malicious attacks. On the one hand, if the revenue is insufficient to cover storage and bandwidth costs, it may lead to the exit of cloud nodes from the system, thereby greatly damaging the decentralization level of the system. On the other hand, malicious behavior may also occur within cloud nodes (such as selecting illegal transactions to form illegal blocks). Therefore, EVONIncentive adopts monetary incentives to ensure the continuous operation of cloud nodes and reputation incentives to supervise malicious behavior within cloud nodes.

The incentive allocation process for inner-layer cloud nodes is illustrated in Algorithm 1. The allocation scheme is as follows:

- Transactions legal: If the cloud node's reputation is not lower than the threshold, it will receive the transaction fee and be rewarded with reputation value, with its staked monetary returned. Otherwise, the system will confiscate the transaction fee and only reward the cloud node with reputation value, returning its staked monetary.
- Transactions illegal: If the cloud node's reputation is not lower than the threshold, its reputation value will reset to zero. The staked monetary will be used to reward the cloud node that reported the illegal behavior and the outer-layer miner nodes that finished mining. Otherwise, the cloud node will be directly removed from the system.

Algorithm 1 Incentive allocation for cloud nodes.

Input: f, R_{thre}, $R_{\max}$, Current R', Pledged currency f_c
Output: π_c, Updated R

CollectTXs();
loop
  upon event Received PBH **do**
    if IsNounceCorrect(PBH) **then**
      if IsFromUBH(PBH) **then**
        $Block \leftarrow$ GenFullBlock(PBH);
        InnerGossip($Block$)
      else
        InnerGossip(PBH);
      end if
    end if
end loop
if IsBlockTxsValid($Block$) **then**
  if IsTxsConfirmd($Block$) **then**
    if $R' \geq R_{thre}$ **then**
      $R \leftarrow \min(R' + 1, R_{\max})$;
      $\mu \leftarrow f$;
    else
      $R \leftarrow R' + 1, \pi_c \leftarrow 0$;
    end if
  end if
else
  if $R' \geq R_{thre}$ **then**
    $R \leftarrow 0, \pi_c \leftarrow -f_c$;
  else
    Get Removed;
  end if
end if

Here, the reputation threshold is set as a dynamic value, which is periodically updated (e.g., every N epochs) based on the reputation distribution of all cloud nodes across the network. The calculation formula is as follows:

$$R_{thre} = \mu(t) - m\sigma(t) \tag{1}$$

where $\mu(t)$ represents the mean reputation value of all active cloud nodes at time t, and $\sigma(t)$ denotes the standard deviation. The parameter m is a configurable security factor.

If the majority of nodes perform well (high μ , low σ), the threshold will be relatively high, making the system stricter toward underperforming nodes. Conversely, if many nodes perform poorly due to issues such as network problems (low μ, high σ), the threshold will be automatically lowered to avoid falsely excluding a large number of nodes, thereby ensuring system availability.

Incentives for Outer-Layer Miner Nodes Ensuring the participation of a large number of outer-layer miner nodes in the Proof of Intelligence (PoI) task is crucial for the decentralization of the entire system. To encourage widespread participation of outer-layer user nodes in the PoI process, EVONIncentive rewards miner nodes that propose magic squares satisfying the difficulty level with system tokens after transaction confirmation. The incentive allocation process for outer-layer miner nodes is described in Algorithm 2.

Algorithm 2 Incentive allocation for miner nodes.

```
Output: π_m
loop
    upon event Received UBH do
        StartMiningProcess(UBH)
end loop
loop
    upon event successfully mines PBH do
        SendInner(PBH)
end loop
if IsTxsConfirmed() then
    π_m ← SystemTokenReward()
end if
```

4.2. Fee selection mechanism

The fee selection mechanism aims to provide outer-layer user nodes with transparent options for transaction fee and to ensure that cloud nodes receive sufficient monetary rewards to offset their operational burdens. The design principle is that the minimum transaction fee should be at least equal to the bandwidth and storage costs incurred by all cloud nodes in the system due to that transaction[6]. Since both bandwidth and storage costs are influenced by the size of transaction data, it is not possible to directly determine the profitability of cloud nodes solely based on transaction fee (transactions with higher fee but larger data sizes may result in lower profits). Therefore, we introduce the concept of unit byte fee to uniformly measure transaction fee.

Definition 1 (Unit byte fee selection)**.**
Vector of unit byte fee selection $\boldsymbol{\zeta} = (\zeta_1, \zeta_2, \ldots, \zeta_I)$, where $I \geq 2$.

- $\zeta_1 > \zeta_2 > \ldots > \zeta_I \geq n_c\eta$, where n_c is the number of clode nodes in system, $\eta = \eta_{bandwidth} + \eta_{storage}$, representing the sum of unit byte storage cost and bandwidth cost for a single cloud node.
- The fee of one transaction generated by the user is: $f = \zeta_i\sigma$, where σ is the transaction size.

Outer-layer user nodes select different unit byte fee based on their actual situations to calculate transaction fee, which are attached to transactions. After transactions enter the transaction pool, inner-layer cloud nodes prioritize packaging transactions with higher unit byte fee into the pending block body. Following Proof of Intelligence, complete block validation, and block chaining confirmation, the cloud node that packaged the transaction will receive the corresponding transaction fee.

4.3. Compensation mechanism

The generation and execution of a transaction not only incur costs for the inner-layer cloud nodes but also introduce additional waiting time for other users' transactions. Since inner-layer cloud nodes use a bidding model when selecting transactions for packaging, where transactions with higher byte fee are prioritized, this can lead to delays in processing other transactions. System users derive utility from the proposed transactions. For analytical purposes, based on the disparity in utility among user nodes, EVONIncentive models system users as high-utility users and low-utility users, denoted as U_H and U_L, respectively.

In extreme cases, if high-utility user nodes within the system endlessly generate high-fee transactions to earn more profit, transactions from other user nodes will be indefinitely delayed. In such cases, where the experience is affected without any compensation, nodes are likely to opt out of the system or cease participation, thus harming the system's vitality and decentralization.

The compensation mechanism involves charging users who submit transactions with monetary to encourage users to generate transactions more conservatively and compensate other users for the potential impact on their experience and income due to possible extended waiting times. Similarly, compensation is measured in terms of unit byte compensation.

Definition 2 (Compensation)**.**
Vector of unit byte compensation $\boldsymbol{C} = (C_{HH}, C_{HL}, C_{LH}, C_{LL})$, where C_{HH} represents the unit byte compensation paid by high-utility users to low-utility users, and the rest follow similarly.

Therefore, whenever user node i submits a transaction, the compensation he need to pay is:

$$C_i = \begin{cases} \sigma[(n_H - 1)C_{HH} + n_L C_{HL}] \\ \sigma[n_H C_{LH} + (n_L - 1)C_{LL}] \end{cases} \tag{2}$$

where σ is the transaction size, and n_H and n_L represent the numbers of high-utility users and low-utility users, respectively.

It is worth noting that when summing up the income of all users, the compensation paid/received between users essentially offsets each other. Therefore, from an overall perspective, the compensation mechanism does not affect the system's financial balance.

Table 2
Block Illegal Proof (Duplicate Transactions).

Field	Size	Description
Proof Header		
Hash1	32 bytes	Proof hash
Type	4 bytes	Illegal type (duplicate transactions)
Hash2	32 bytes	Illegal block hash
Proof Content		
Tx	Variable	Original transaction data
BH0	32 bytes	Hash of the first block
Path0	Variable	Merkle proof path
BH1	32 bytes	Hash of the second block
Path1	Variable	Merkle proof path

4.4. Supervisory group punishment mechanism

Before cloud nodes package transactions and generate unproven blocks, they need to stake a certain amount of monetary. When the proven blocks generated through the smart workload proof process are propagated within the inner-layer network, they undergo verification by other cloud nodes, forming the supervisory group within the network.

If a cloud node within the supervisory group detects that the received complete block is illegal, it immediately generates a block illegal proof based on the specific illegal content and forwards it to the remaining inner-layer cloud nodes. Taking the example of discovering illegal transactions within a block–duplicate transactions, the block illegal proof generated by the cloud node is as shown in Table 2.

5. Security analysis

The hybrid incentive mechanism of EVONIncentive, integrating monetary and reputation with a supervisory group, is designed to significantly enhance the security of the blockchain system against rational Byzantine behaviors. The core security argument lies in altering the incentive structure to make malicious actions economically irrational.

Deterrence through Punishment: As defined in Algorithm 1, a cloud node that packages illegal transactions faces severe penalties: its reputation is reset to zero, and its staked monetary is confiscated and redistributed to the reporter and miners. This punitive measure ensures that the expected payoff of any malicious attempt becomes negative, considering the loss of staked assets and all future rewards (which are

[6] Bandwidth and storage costs for non-transaction data within blocks can be negligible, for example, in Bitcoin, non-transaction data size accounts for only 0.1 % of all data size

gated by a reputation threshold R_{thre}). For a rational node seeking to maximize profit, honesty becomes the dominant strategy.

Cost of Attacks: The reputation system raises the cost of sustained attacks. An attacker cannot simply create a new identity (Sybil attack) to avoid punishment, as a new identity would start with zero reputation and be unable to receive block rewards until it earns a positive reputation through honest work. This forces attackers to acquire and maintain a high-reputation node, which is a costly and slow process, making attacks economically prohibitive.

Security against Specific Attacks:

- Double-Forging: A cloud node attempting to broadcast conflicting blocks would be easily detected by the supervisory group. The first valid block received by honest nodes will be accepted. The subsequent conflicting block will be identified as illegal, triggering the punishment mechanism described in Table 2 and Algorithm 1.
- Transaction Censorship: While cloud nodes prefer transactions with higher fees, our fee selection and compensation mechanisms ensure that even low-fee transactions from U_L users are eventually processed if they meet the minimum fee requirement ($\zeta_{\min} \geq n_c * \eta$). This is because systematically ignoring them would lead to a loss of compensation income for other users and could harm the cloud node's reputation for fair packaging in the long run.
- Spam Attacks: The requirement to attach a fee $\geq n_c * \eta$ per byte and to pay compensation for the waiting time imposed on others makes launching spam attacks prohibitively expensive for an adversary. The attacker would bear the full cost of the system's resource consumption.

In conclusion, by strategically employing slashing conditions, reputation gates, and a verifiable punishment protocol, EVONIncentive aligns individual rationality with system-wide security goals, thereby creating a more robust and resilient economic environment against rational Byzantine participants compared to purely monetary incentive schemes.

6. Game analysis

The main roles within the system are divided into four types: system designers, outer edge user nodes, outer edge miner nodes, and inner cloud nodes. In this section, we mainly address the following two questions, constructing Stackelberg dynamic game models for each, analyzing equilibrium conditions to determine the optimal values or ranges of parameters.

- Problem 1: How to determine the system monetary reward for miner nodes after completing proof of intelligent workload?
- Problem 2: How to determine the transaction fee for successful block generation by cloud nodes and compensation for other users?

6.1. Game model I: Miner node incentive determination

In this subsection, we propose a two-stage Stackelberg game model between the system designer and outer edge miner nodes.

Miner Nodes Miner nodes constitute the subset of outer edge nodes interested in participating in the system's mining process. Among them, there are indeed full-time miner nodes (dedicating all computational power and resources to the PoI process); however, more commonly, a portion of computational resources is allocated to mining while the rest serves other life purposes. Therefore, for miner i, its profit function is defined as follows:

$$\pi_{m_i}(M_0, p_i^m) = M_0\mathbf{P}(p_i^m) + \gamma_i(p_i - p_i^m) \tag{3}$$

Where M_0 is the system monetary reward value for successfully completing one PoI process meeting the main block difficulty requirements. If a node completes one PoI process meeting the sub block difficulty requirements, the system monetary reward is $\frac{M_0}{k}$ (k is the throughput parameter in the graph chain protocol ORIC [19], used to measure the difficulty multiplier of main and sub blocks).The probability $\mathbf{P}(p_i^m) = \frac{p_i^m}{\sum_{i=1}^{n_m} p_i^m}$ represents the likelihood of miner i successfully completing one corresponding PoI process, calculated as the proportion of the computing power devoted by that miner node relative to the total network computing power. Here, p_i denotes the total computing power of miner i, p_i^m represents the computing power devoted by miner i in the mining process, and γ_i is the income factor indicating miner i's allocation of computing power for other purposes. The planning problem for miner i is as follows:

$$\begin{aligned} max \quad & \pi_{m_i}(M_0, p_i^m) \\ s.t. \quad & 0 \leq p_i^m \leq p_i \end{aligned} \tag{4}$$

System Designer The primary goal of the system designer is to incentivize more nodes to participate in the system's proof-of-intelligence (PoI) process, ensuring a high level of decentralization in the system. However, setting excessively high mining monetary rewards is clearly disadvantageous for maintaining stable system inflation and the value of the system monetary. Therefore, the system designer tends to provide a mining incentive that is as small as possible but still profitable for miner nodes. The problem statement is as follows:

$$\begin{aligned} & min \quad M_0 \\ s.t. \quad & \sum_{i=1}^{n} p_i^m \geq \frac{D_p}{T} \end{aligned} \tag{5}$$

Where D_p is the difficulty requirement of the system's main blocks, and T is the generation period of the main blocks. Constraint 5 ensures that the total computational power of the system can meet the system settings for generating a main block within a given period.

Due to the dynamic interaction process between the system designer and the miners, the specific value of the mining monetary reward formulated by the system designer will affect each miner's decision on the amount of computational power to devote tn the system. The total computational power of the system will also influence the system designer's formulation of the mining monetary reward value. To determine the optimal value of the system monetary reward, we divide the process into two stages and conduct a specific analysis using backward induction. In the first stage, the system designer informs the system miners of the specific value of the mining monetary reward. In the second stage, the miners adjust the amount of computational power devoted based on the value of the mining monetary reward and return the results to the system designer. Through continuous adjustments, a balance is achieved where the value of the system monetary reward reaches equilibrium.

6.1.1. Stage II: Miner nodes

Miner Node Game Model Miner nodes' game is defined as the tuple $\phi_m(M, P, \Xi)$

- Players: Set of miner nodes M
- Strategies: Any miner node $i \in M$ decides to devote computational power p_i^m to the proof-of-intelligence (PoI) process, where the set of all miners' devoted computational power is $P = (p_1^m, p_2^m, \dots, p_{n_m}^m)$
- Payoff: Set $\Xi = (\pi_{m_i}, \forall i \in M)$ containing the payoffs of all miner nodes, where the payoff function is defined as in Eq. (3)

In the miner node game model, each miner's objective is to maximize its profit. Upon receiving the system designer's determined monetary reward value, each miner decides the amount of computational power to devote in the mining process as their response. Below is the definition of equilibrium in this process, along with proofs of its existence and uniqueness.

Definition 3 (Miner Game Equilibrium)**.** Given the system mining reward M_0, the set of miner nodes' devoted computational power $\Pi^* = (p_1^{m*}, p_2^{m*}, \dots, p_{n_m}^{m*})$ constitutes a Nash equilibrium of the miner game if and only if for all $i \in M$, it holds that:

$$\begin{aligned} &\pi_{m_i}(M_0, p_i^m, d_i^*) \le \pi_{m_i}(M_0, p_i^{m*}, d_i^*) \\ &d_i^* = \sum_{l \neq i} p_l^{m*}, \quad \forall l \in M \end{aligned} \tag{6}$$

Theorem 1. *In the miner node game model, the Nash equilibrium exists and is unique.*

Proof. According to constraint 4, the range of miner node i's devoted computational power p_i^m in the system is $[0, p_i]$, and the payoff function $\pi_{m_i}(p_i^m)$ is continuous in the range $[0, p_i]$.

By computing the first and second derivatives of the payoff function, we have:

$$\begin{aligned} \frac{\partial \pi_{m_i}(p_i^m)}{\partial p_i^m} &= \frac{M_0 d_i}{(p_i^m + d_i)^2} - \gamma_i \\ \frac{\partial^2 \pi_{m_i}(p_i^m)}{\partial p_i^{m2}} &= -\frac{2M_0 d_i}{(p_i^m + d_i)^3} < 0 \end{aligned} \tag{7}$$

The second partial derivative being less than 0 indicates that the payoff function $\pi_{m_i}(p_i^m)$ is strictly concave with respect to p_i^m, hence a Nash equilibrium exists.

Since the payoff function is continuous and strictly concave over the range $[0, p_i]$, there is only one optimal solution, making the equilibrium unique.

□

Miner i's optimal strategy is to adjust the devoted computational power p_i^m to maximize their payoff. Assuming M_0 is a given value by the system designer, to obtain the optimal solution, it is evident that:

$$\frac{\partial \pi_{m_i}(p_i^m)}{\partial p_i^m} = 0 \tag{8}$$

Since p_i^m is greater than 0, we have:

$$p_i^{m*} = \sqrt{\frac{M_0 d_i}{\gamma_i}} - d_i \tag{9}$$

After obtaining the miner's devoted computational power, it is substituted into the system designer's planning problem to minimize the quantity of M_0.

6.1.2. Stage I: System designer

In this stage, the system designer contemplates each miner's anticipated strategy, aiming to minimize the determined value of the system monetary incentive. For each miner i, by substituting their optimal strategy's computational power devotement into the constraint provided by Eq. (5), we obtain:

At this point, the planning problem for the system designer can be reformulated as:

$$\begin{aligned} &min \quad S(M_{0_i}) = \sum_{i \in M} M_{0_i} \\ &s.t. \quad \sqrt{\frac{M_{0_i} d_i}{\gamma_i}} \ge \frac{D_p}{T} \end{aligned} \tag{10}$$

To address the planning problem described in Eq. (10), the key focus lies in determining the total amount of computational power d_i devoted by other miners in the Proof of Intelligence (PoI) process. Based on the result of d_i and Eq. (10), M_0 can be calculated. Subsequently, each miner i computes the optimal value of computational power devotement p_i^{m*} based on the resulting value of the mining incentive M_0.

Drawing on the principles of Gauss-Seidel iteration, we propose an iterative algorithm to ultimately determine the value of the mining incentive for outer-layer mining nodes. Iterative cycles are denoted by r, where in each iteration, every mining node updates their computational power devotement p_i^m based on their calculated M_0. Eventually, when the variance of M_0 calculated by each mining node falls below a predetermined threshold, we consider the value of M_0 at that point as the stable value of the mining incentive under equilibrium conditions, which can be employed within the system.

Within a single iteration, the update of the computational power devotement p_i^m for nodes is a sequential process. Assuming there are n_m miners in the system at this moment, each marked from 1 to n_m, we divide each iteration into n_m slots. Before the start of each iteration, each miner node initializes their mining computational power devotement based on their computational resource status. In the ith slot of the rth iteration ($i \le n_m$), d_i is calculated as follows:

Miner i determines their own computational power devotement p_i^m based on Eq. (9) and the result of d_i. The updated set of computational power devotements by miners is then provided to miner node $i+1$ for further calculation. The specific iterative process is outlined in Algorithm 3.

Algorithm 3 Iterative algorithm for monetary incentive.

Input: p_i^m, γ_i, variance threshold α
Output: M_0, Updated p_i^m

$r \leftarrow 1$
while variance $> \alpha$ or r < 100 **do**
  for $i \in M$ **do**
    $d_i \leftarrow \sum_{l<i} p_l^m + \sum_{l>i} p_l^m$
    $M_{0_i} \leftarrow$ Solve Formula 10
    $p_i^m \leftarrow \sqrt{\frac{M_{0_i} d_i}{\gamma_i}} - d_i$
    Substitute p_i^m, update p^m set
  end for
  Calculate variance of M_0
  $r++$
end while
$M_0 \leftarrow \overline{M_{0_i}}$

We now proceed to demonstrate the convergence of the algorithm.

Theorem 2. *Algorithm 3 converges to the global optimum.*

Proof. Algorithm 3 is based on the Gauss-Seidel iteration. According to Proposition 2.3 in Chapter 3 of the literature [52], if the following conditions are satisfied, the algorithm can converge to the global optimum:

1. $S(M_{0_i})$ is continuous and differentiable, the constraints are convex and compact.
2. Given a fixed M_{0_i}, miner i can uniquely and optimally determine p_i^m.
3. When a feasible solution for p_i^m is given, $S(M_{0_i})$ has a unique minimum value.

Since $S(M_{0_i})$ is a linear addition over (M_{0_i}), condition 1 is obviously satisfied.

According to Theorem 1, the payoff function $\pi_{m_i}(p_i^m)$ is a concave function with a unique solution in the given compact space, satisfying condition 2.

$S(M_{0_i})$ is a linear function, and there exists a unique minimum value within the given range of values, thus satisfying condition 3. □

Algorithm 3 is designed to compute the exact Nash Equilibrium of the miner game defined in Section 6.1.1. It is an iterative best-response algorithm based on the Gauss-Seidel method, where each miner sequentially updates their strategy (computing power devotion p_i^m) in response to the current system incentive M_0 and the strategies of other miners. Unlike heuristic or approximation algorithms for NP-hard problems, Algorithm 3 is designed to converge to the exact equilibrium point of the defined convex game, as guaranteed by Theorem 2. Therefore, in the limit of infinite iterations, the output M_0 and the corresponding strategy profile P^m constitute a precise solution to the Stackelberg game.

We further discuss the approximation ratio of Algorithm 3 under a finite number of iterations.

The Gauss-Seidel iteration, upon which our algorithm is based, is known to converge linearly for strongly convex problems. The miner's profit function π_{m_i} is strictly concave in p_i^m (as shown in Theorem 1), and the constraint set is compact and convex. The overall problem of minimizing M_0 subject to the total power constraint is a convex optimization problem given the miners' best-response strategies. The update rule for each miner in each slot is given by Eq. (9). The convergence rate of the Gauss-Seidel method can be characterized by the contraction property of the update mapping.

Let $P^{m(r)} = (p_1^{m(r)}, p_2^{m(r)}, \ldots, p_n^{m(r)})$ be the strategy profile at the end of iteration r, and $P^{m(*)}$ be the exact equilibrium profile. The error after r iterations can be bounded by:

$$||P^{m(r)} - P^{m(*)}|| \le A \cdot \delta^r \tag{11}$$

where A is a constant and $0 < \delta < 1$ is the contraction factor, which depends on the specific structure of the best-response functions and the order of updates. Consequently, the error in the resulting system incentive M_0^r is also bounded:

$$|P^{m(r)} - P^{m(*)}| \le B \cdot \delta^r \tag{12}$$

where B is another constant related to A and the sensitivity of M_0 to changes in P^m. This linear convergence rate implies that the algorithm can achieve an ϵ-approximate solution with a number of iterations $r = \mathcal{O}(log(\frac{1}{\epsilon}))$. The threshold α directly controls this ϵ. Therefore, for any desired precision $\epsilon > 0$, Algorithm 3 can compute an incentive value $M_0^{(r)}$ such that $|M_0^{(r)} - M_0^*| < \epsilon$ within a logarithmic number of steps. The ratio between the algorithm's output $M_0^{(r)}$ and the true optimum M_0^* satisfies $1 - \frac{\epsilon}{M_0^*} \le \frac{M_0^{(r)}}{M_0^*} \le 1 + \frac{\epsilon}{M_0^*}$, effectively providing an approximation ratio that can be made arbitrarily close to 1 by choosing a sufficiently small threshold α.

6.2. Game model II: Incentives for cloud nodes and compensation determination for user nodes

In this subsection, we propose a three-stage Stackelberg game model involving the system designer, outer edge user nodes, and inner cloud nodes.

Cloud Nodes Inner cloud nodes select transactions from the transaction pool and include them in pending blocks. Subsequently, the corresponding block headers of the pending blocks are transmitted to the outer peer-to-peer network via MEC nodes. After miner nodes complete the PoI process, the proven block headers are sent back to the inner network. Once the complete block is broadcasted and validated, legitimate blocks are successfully added to the chain, and the transactions within are confirmed. Cloud nodes that generate the blocks receive transaction fee attached to all transactions included in the block. All inner cloud nodes store the block and bear their respective bandwidth and storage costs. The transaction pool is updated by removing transactions already included in blocks.

Rationally, cloud nodes aim to select transactions with higher transaction fee per byte when packing blocks to maximize their own profits. The probability of a cloud node successfully mining a block (where the pending block generated by a cloud node becomes a proven block through the work proof of miner nodes) is related to the reputation value of the cloud node, similar to PoS where the probability of node block mining is related to the node's wealth. The higher the reputation value of a cloud node $i \in C$, the easier it is for the corresponding pending block header to be prioritized by the outer miner nodes. Suppose the probability of successful block production by cloud node i is α_i, we have $\sum_{i \in C} \alpha_i = 1$.

Let's assume that the jth transaction proposed by user node i' is denoted as $tx_{i'j}$, with its corresponding transaction size represented by $\sigma_{i'j}$, and the unit byte fee as $\zeta_{i'j}$. In a certain round k[7], the set of transactions packed by cloud node i into a block is denoted as $\psi_{i,k}$. Then, the transaction fee obtained by cloud node i in this round k is represented as $f_{i,k}$ and can be expressed as:

$$f_{i,k} = \alpha_i \sum_{tx_{i'j} \in \psi_{i,k}} \zeta_{i'j} \sigma_{i'j} \tag{13}$$

Simultaneously, all cloud nodes within the system are required to unconditionally forward complete blocks and maintain a consistent ledger. Each successful block generation by a cloud node implies that all cloud nodes must bear the cost of forwarding and storing that generated block. Assuming that all cloud nodes use the same storage technology and have the same bandwidth conditions, the cost of storing one byte is denoted as η_1, and the cost of bandwidth per byte is denoted as η_2. The total cost per byte is defined as $\eta = \eta_1 + \eta_2$. In round k, the expenses borne by cloud node i can be expressed as:

$$S_{i,k} = \sum_{l \in C} \alpha_l \sum_{tx_{l'j} \in \psi_{l,k}} \sigma_{l'j} \eta \tag{14}$$

Therefore, in round k, the profit function of cloud node i is defined as:

$$\pi_{c_{i,k}} = \alpha_i \sum_{tx_{i'j} \in \psi_{i,k}} \zeta_{i'j} \sigma_{i'j} - \sum_{l \in C} \alpha_l \sum_{tx_{l'j} \in \psi_{l,k}} \sigma_{l'j} \eta \tag{15}$$

User Nodes Outer edge user nodes earn profits by proposing transactions. Based on the unit byte fee vector determined by the system designer $\zeta = (\zeta_1, \zeta_2, \ldots, \zeta_I)$, user nodes consider factors such as their financial resources, transaction utility, and urgency to reasonably select unit byte fee for generating transactions. Assuming that the transaction generation of each user $i \in U$ follows a Poisson distribution[8], there exist different transaction generation rates λ_i based on different unit byte fee, given by:

$$\begin{aligned} &\lambda_{i_1} + \lambda_{i_2} + \ldots + \lambda_{i_I} \le \frac{\Phi}{n_u} \\ &\lambda_{i_1}, \lambda_{i_2}, \ldots, \lambda_{i_I} \ge 0 \end{aligned} \tag{16}$$

Where Φ represents the system's transactions processed per second, n_u denotes the number of outer layer user nodes, and constraint 16 ensures that theoretically, the system can include all user-generated transactions in the blockchain.

We differentiate user nodes into high utility users and low utility users based on the utility of their transactions respectively. In simple terms, high utility users are those who can gain greater off-chain profits after engaging in transactions, such as enterprises reaching offline cooperation agreements through on-chain transactions. Low utility users, on the other hand, are more ordinary users who simply engage in on-chain

[7] To simplify the analysis, we assume that cloud nodes generate blocks in rounds, meaning that in one round, a cloud node can generate at most one pending block

[8] Modeling transaction generation as a Poisson process is a mainstream practice in blockchain systems, several seminal works [23,30,53] in blockchain analysis have successfully employed the Poisson assumption to model transaction arrival times, with predictions exhibiting a close agreement with observed market behavior and system dynamics. While practical applications may indeed exhibit more complex patterns, this is beyond the scope of this paper.

transactions, such as purchasing original virtual assets through on-chain payments. The utility of users is denoted by U_H and U_L.

Clearly, a user's profit is also related to the transaction waiting time τ, as the time cost of delayed confirmation can result in profit loss for the user. The user's transaction waiting time is defined as the time interval from transaction generation to transaction confirmation, i.e., $\tau_{ij} = t_{ij}^{con} - t_{ij}^{gen}$, where $t_{ij}^{com} \leftarrow \infty$ if the transaction is not eventually confirmed. The user's time cost is defined as $\xi\tau$, where ξ is the user's time cost coefficient.

In summary, the profit of user nodes consists of four parts: transaction fee, transaction utility, time cost, and compensation. The profit for user node i due to transaction tx_{ij} is given by:

$$\pi_{u_{ij}} = \begin{cases} U_H - [\zeta_{ij} + (n_H - 1)C_{HH} + n_L C_{HL}]\sigma_{ij} - \xi\tau_{ij} \\ U_L - [\zeta_{ij} + [n_H C_{LH} + (n_L - 1)C_{LL}]\sigma_{ij} - \xi\tau_{ij} \\ -\xi\tau_{ij} \end{cases} \tag{17}$$

It should be noted that in addition to earning revenue from transactions initiated by themselves, user nodes also receive compensation when transactions proposed by other user nodes are confirmed. Therefore, the total profit function of user node i is defined as:

$$\pi_{u_i} = \sum_{j=1}^{tx(i)} \pi_{u_{ij}} + \sum_{l \in \mathcal{N}, l \neq i} \sum_{j=1}^{tx(l)} \sigma_{lj} C_{xx} \tag{18}$$

Where $tx(i)$ represents the number of transactions proposed by user node i.

System Designer The main objectives of the system designer are: to encourage user nodes to pay transaction fee that are sufficient to cover the total operating costs of the system's full nodes, ensuring the sustainability of the system and compensate users whose own transactions are delayed due to transactions initiated by others, while suppressing situations where a minority of user nodes propose transactions without limits, ensuring fairness in the system, and to shorten the average transaction waiting time as much as possible, thereby increasing social welfare and improving the scalability of the system. Therefore, the system designer needs to reasonably design the unit byte fee $\boldsymbol{\zeta}$ and the unit byte compensation $\boldsymbol{C}$, aiming to maximize the social welfare ϑ of the system. The problem is formulated as follows:

$$\begin{aligned} & max \quad \vartheta = \sum_{i \in C} \pi_{c_i} + \sum_{i \in U} \pi_{u_i} \\ & s.t. \quad \zeta_{min} \geq n_c \eta \\ var. \quad & \boldsymbol{\zeta} = (\zeta_1, \zeta_2, \ldots, \zeta_I) \quad \boldsymbol{C} = (C_{HH}, C_{HL}, C_{LH}, C_{LL}) \end{aligned} \tag{19}$$

Where the system's social welfare ϑ is represented as the sum of the earnings of all system-wide cloud nodes and user nodes,[9] n_c is the number of internal cloud nodes, and η is the total unit byte cost burden. Constraint 19 ensures that the total cost of system full nodes is covered.

There exists dynamic interaction among the system designer, external user nodes, and internal cloud nodes. The unit byte fee and unit byte compensation set by the system designer will affect the generation of transactions by user nodes, while the additional transaction fee of user nodes will also affect the transaction selection process of cloud nodes. Similarly, we divide this interaction process into three stages and analyze them step by step in reverse order. In the first stage, the system designer determines the unit byte fee and unit byte compensation parameters and invests them in the system operation. In the second stage, user nodes selectively generate transactions based on the unit byte fee and unit byte compensation, combined with their own situations. In the third stage, cloud nodes selectively pick transactions from the transaction pool and package them to generate the pending proof blocks.

[9] Mining node revenue is allocated by the system and can be considered as overall balanced income and expenditure, therefore not included in the consideration of social welfare.

6.2.1. Stage III: Cloud nodes

Cloud Node Game Model The game of cloud nodes in round k is defined as the tuple $\phi_{c,k}(C, \mathcal{T}_k, \mathcal{R}_k)$

- Players: The set of cloud nodes C
- Strategies: Each cloud node $i \in C$ selects a set of transactions $\psi_{i,k}$ to enter the pending block generated in round k, and all transactions selected by cloud nodes in the system in the current round k is $\mathcal{T}_k = \prod_{i \in C} \psi_{i,k}$
- Payoffs: The set $\mathcal{R}_k = (\pi_{c_{i,k}}, \forall i \in C)$ contains the earnings of all cloud nodes, with the payoff function defined in Eq. (15)

In the cloud node game model, cloud nodes will weigh between the unit byte fee of transactions and the total cost per byte, taking into account the strategies of other cloud nodes to maximize their own earnings. Below is the definition of equilibrium in this process, the proof is detailed in Appendix A.[10]

Definition 4 (Cloud Node Game Equilibrium). Given the unit byte fee vector $\boldsymbol{\zeta}$, in round k, the transaction set $\psi_{i,k}$ chosen by cloud node $i \in C$ constitutes a Nash equilibrium of the cloud node game if the following conditions hold:

$$\pi_{c_{i,k}}(\boldsymbol{\zeta}, \psi_{i,k}, \prod_{l \in C, l \neq k} \psi^*_{l,k}) \leq \pi_{c_{i,k}}(\boldsymbol{\zeta}, \psi^*_{i,k}, \prod_{l \in C, l \neq k} \psi^*_{l,k}) \tag{20}$$

Theorem 3. *In the cloud node game model, let $\mathcal{B}_k$ denote the total transaction pool in round k, and $(\psi^*_{i,k}, i \in C)$ constitute the equilibrium strategy for cloud node i, where:*

$$\psi^*_{i,k} = \begin{cases} max(\mathcal{B}_k) & If \zeta_{max} \geq \eta \\ \varnothing & If \zeta_{max} < \eta \end{cases} \tag{21}$$

According to Theorem 3, if the unit byte fee of a transaction is not lower than the unit byte cost of cloud node i, i.e., $\zeta \geq \eta$, cloud node i may choose to include this transaction in the block to seek profit. However, the transaction fee only covers the cost of cloud node i. But alll cloud nodes need to transmit and finally store the block, resulting in a larger overall unit byte cost for the system, namely $n_c\eta$. In this case, the transaction fee is insufficient to cover the total cost of the system's cloud nodes. Nevertheless, cloud nodes within the system will continue to persistently select such transactions for block inclusion, exacerbating the overall loss for cloud nodes. This further confirms one of the problems we identified at the beginning of this paper: transaction fee cannot cover the operational costs of the entire system's nodes, leading to a reduction in decentralization and sustainability of the system.

6.2.2. Stage II: User nodes

User Node Game Model The game of user nodes in round k is defined by the tuple $\phi_u(U, \Gamma_u, \Upsilon)$:

- Player: Set of user nodes U
- Strategy: Each user node $i \in U$ generates transactions at a rate vector λ_i, corresponding to different unit byte fee ζ_i. User nodes have different transaction generation rates, i.e., $\lambda_i = (\lambda_{i_1}, \lambda_{i_2}, \ldots, \lambda_{i_I})$. The overall transaction generation rate of all user nodes in the system is $\Gamma_u = \Pi_{i \in U} \lambda_i$.
- Payoff: The set $\Upsilon = (\pi_{u_i}, \forall i \in U)$ contains the payoff of all user nodes, with the payoff function defined in Eq. (18).

In the user node game model, considering that cloud nodes prioritize transactions with higher fee, user nodes weigh between higher unit byte fee and longer transaction waiting time. They consider factors such

[10] Due to the space limit, we leave the rest of proofs in our Online Appendix: https://github.com/LDoinA/EVONInventiveProof

as the urgency of their transactions, transaction utility, and personal financial capacity to generate transactions. Due to the existence of compensation mechanisms, traditionally high utility users may become low utility users (paying higher compensation fee resulting in smaller net utilities), and vice versa. Therefore, we further introduce the concept of net utility:

Definition 5 (Net Utility)**.** Given different utilities U_H and U_L for user nodes, and unit byte compensation $\boldsymbol{C} = (C_{HH}, C_{HL}, C_{LH}, C_{LL})$, the net utility U_{net} is defined as:

$$U_{net} = \begin{cases} U_H - \sigma[(n_H - 1)C_{HH} + n_L C_{HL}] \\ U_L - \sigma[n_H C_{LH} + (n_L - 1)C_{LL}] \end{cases} \tag{22}$$

For user nodes with higher net utility, we collectively refer to them as high-net users, with their corresponding net utility denoted as U_H^{net}, for user nodes with lower net utility, we collectively refer to them as low-net users, with their corresponding net utility denoted as U_L^{net}.

$$\tau_i = \begin{cases} \dfrac{\lambda_{i_1}}{\Phi - \sum_{l\in U} \lambda_{l_1}} + \sum_{q=2}^{I} \dfrac{\Phi\lambda_{i_q}}{(\Phi - \sum_{x=1}^{q-1}\sum_{l\in U}\lambda_{l_x})(\Phi - \sum_{x=1}^{q}\sum_{l\in U}\lambda_{l_x})} \\ \quad \text{If} \zeta_1 > \zeta_2 > \ldots > \zeta_I \geq \eta \\ \dfrac{\lambda_{i_1}}{\Phi - \sum_{l\in U} \lambda_{l_1}} + \sum_{q=2}^{Y} \dfrac{\Phi\lambda_{i_q}}{(\Phi - \sum_{x=1}^{q-1}\sum_{l\in U}\lambda_{l_x})(\Phi - \sum_{x=1}^{q}\sum_{l\in U}\lambda_{l_x})} \\ \quad \text{If} \zeta_1 > \zeta_2 > \ldots \zeta_Y \geq \eta > \ldots > \zeta_I \\ \dfrac{\lambda_{i_1}}{\Phi - \sum_{l\in U} \lambda_{l_1}} \\ \quad \text{If} \zeta_1 \geq \eta > \zeta_2 > \ldots > \zeta_I \\ \infty \quad \text{If} \eta > \zeta_1 > \zeta_2 > \ldots > \zeta_I \end{cases}$$

Since the profit function of user nodes includes losses due to time costs, before discussing the equilibrium of the game regarding user node profits, we first need to discuss the average transaction waiting time for users. As mentioned earlier, the transaction generation of user nodes follows a Poisson distribution, while the block generation time follows an exponential distribution. Therefore, we consider the entire process from transaction generation to block generation as an $M/M/1$ queuing system, where transactions with higher unit byte fee are always prioritized by cloud nodes to enter the block, following preemptive priority scheduling. According to the basic principles of queueing theory, we can derive the formula for calculating the average transaction waiting time τ_i for user node i as follows:

Lemma 1. *For user node $i \in U$, its average transaction waiting time τ_i is shown in the top of next page. Due to the space limit, the proof is detailed in Online Appendix B*

To simplify the analysis, we follow the assumption made in [54], which assumes that users of the same type in the system will adopt the same strategy, and each user node will choose only one unit byte fee ζ_i to generate transactions when reaching equilibrium. Next, we provide the equilibrium definition for the user node game model and prove it.

Definition 6 (Equilibrium in User Node Game)**.** Given the unit byte fee vector $\boldsymbol{\zeta}$ and the unit byte compensation vector $\boldsymbol{C}$, the transaction generation rate vector λ_i decided by user node $i \in U$ constitutes a Nash equilibrium in the user node game if and only if the following conditions hold:

$$\pi_{u_i}(\boldsymbol{\zeta}, \boldsymbol{C}, \lambda_i, \Pi_{l\in U, l\neq i}\lambda_l^*) \leq \pi_{u_i}(\boldsymbol{\zeta}, \boldsymbol{C}, \lambda_i^*, \Pi_{l\in U, l\neq i}\lambda_l^*) \tag{23}$$

where $\Pi_{l\in U, l\neq i}\lambda_l^*$ represents the transaction generation rate vector of other user nodes in the system under equilibrium conditions, excluding user node i.

Theorem 4. *In the user node game model, suppose λ_H^* represents the equilibrium strategy of high-net user nodes, and λ_L^* represents the equilibrium strategy of low-net user nodes. Together, they constitute the equilibrium strategy of all users in the system, shown at the top of next page. The proof is detailed in Appendix C.*

$$\lambda_{H/L}^* = \begin{cases} \lambda_{H/L}(\zeta_1, U_H^{net}, U_L^{net}) & \text{If} \varpi(\zeta_2, U_H^{net}, U_L^{net}) > \zeta_1\sigma \\ \lambda_{H/L}(\zeta_i, U_H^{net}, U_L^{net}) & \text{If} \varpi(\zeta_{i+1}, U_H^{net}, U_L^{net}) > \zeta_i\sigma, \varpi(\zeta_i, U_H^{net}, U_L^{net}) \leq \zeta_{i-1}\sigma \\ \lambda_{H/L}(\zeta_I, U_H^{net}, U_L^{net}) & \text{If} \varpi(\zeta_I, U_H^{net}, U_L^{net}) \leq \zeta_{I-1}\sigma \end{cases}$$

$$\varpi(\zeta_I, U_H^{net}, U_L^{net}) = max(\zeta_i\sigma + \frac{\xi[(n_H^{net} - 1)\Lambda_H + n_L^{net}\Lambda_L]}{\Phi(\Phi - n_H^{net}\Lambda_H - n_L^{net}\Lambda_L)}, \\ U_L^{net} - \frac{\xi}{\Phi} - \frac{\xi\Lambda_L(2\Phi - n_H^{net}\Lambda_H - n_L^{net}\Lambda_L])}{\Phi(\Phi - n_H^{net}\Lambda_H - n_L^{net}\Lambda_L)^2})$$

$$\lambda_H = \begin{cases} 0 \text{If} U_H^{net} \leq \zeta_i\sigma + \dfrac{\xi}{\Phi} \\ h_1 \text{If} U_H^{net} > \zeta_i\sigma + \dfrac{\xi}{\Phi}, U_L^{net} \leq \zeta_i\sigma + \dfrac{\xi}{\Phi - h_1 n_H^{net}} \\ min(\dfrac{\Phi}{n_H^{net} + n_L^{net}}, \dfrac{(\Phi - h_2)[h_2(U_H^{net} - \zeta_i\sigma) - \xi]}{n_H^{net}[h_2(U_H^{net} - \zeta_i\sigma) - \xi) + n_L^{net}[h_2(U_L^{net} - \zeta_i\sigma) - \xi)]}) \\ \text{If} U_L^{net} > \zeta_i\sigma + \dfrac{\xi}{\Phi - h_1 n_H^{net}} \end{cases}$$

$$\lambda_L = \begin{cases} 0 \text{If} U_L^{net} \leq \zeta_i\sigma + \dfrac{\xi}{\Phi - h_1 n_H^{net}} \\ \dfrac{\Phi - n_H^{net}\Lambda_H}{n_L^{net}} - \dfrac{\xi(n_L^{net} - 1) + \sqrt{\xi^2(n_L^{net} - 1)^2 + 4n_L^{net}(U_L^{net} - \zeta_i\sigma)\xi(\Phi - n_H^{net}\Lambda_H)}}{2N_{net}^{l2}(U_L^{net} - \zeta_i\sigma)} \\ \text{If} U_L^{net} > \zeta_i\sigma + \dfrac{\xi}{\Phi - h_1 n_H^{net}} \end{cases}$$

$$h_1 = min(\frac{\Phi}{n_H^{net} + n_L^{net}}, \frac{\Phi}{n_H^{net}} - \frac{\xi(n_H^{net} - 1) + \sqrt{\xi^2(n_H^{net} - 1)^2 + 4\xi\Phi n_H^{net}(U_H^{net} - \zeta_i\sigma)}}{2n_H^{net^2}(U_H^{net} - \zeta_i\sigma)})$$

$$h_2 = \frac{\sqrt{\xi^2(n_H^{net} + n_L^{net} - 1)^2 + 4\xi\Phi[n_H^{net}(U_H^{net} - \zeta_i\sigma) + n_L^{net}(U_L^{net} - \zeta_i\sigma)]}}{2[n_H^{net}(U_H^{net} - \zeta_i\sigma) + n_L^{net}(U_L^{net} - \zeta_i\sigma)]} \\ + \frac{\xi(n_H^{net} + n_L^{net} - 1)}{2[n_H^{net}(U_H^{net} - \zeta_i\sigma) + n_L^{net}(U_L^{net} - \zeta_i\sigma)]}$$

From Theorem 4, we can observe that when $\varpi(\zeta_2, U_H^{net}, U_L^{net})$ exceeds $\zeta_1\sigma$, all users in the system will choose ζ_1, opting for the highest unit byte fee to generate transactions and enjoying the shortest waiting time. This indicates that users are more tolerant of cost expenses than waiting time. As ζ_1 increases, users gradually find it difficult to afford such high cost expenses and tend to choose a lower unit byte fee ζ_2, at the expense of slightly longer waiting times compared to ζ_1. This trend continues, and when $\varpi(\zeta_{i+1}, U_H^{net}, U_L^{net})$ exceeds $\zeta_i\sigma$ while $\varpi(\zeta_i, U_H^{net}, U_L^{net})$ is less than $\zeta_{i-1}\sigma$, users will all generate transactions at the unit byte fee ζ_i. Finally, when $\varpi(\zeta_I, U_H^{net}, U_L^{net})$ is less than $\zeta_{I-1}\sigma$, users will all choose the lowest unit byte fee ζ_I.

6.2.3. Stage one: System designer

In this stage, the system designer determines the specific values of the unit byte fee vector $\boldsymbol{\zeta}$ and the unit byte compensation $\boldsymbol{C}$, encouraging user nodes to pay sufficient transaction fee and transaction compensation to alleviate the bandwidth and storage burdens of all system nodes as well as the waiting costs of other user nodes, to maximize social welfare. The specific planning problem is described in Eq. (19).

Theorem 5. *The optimal solution for the unit byte fee vector $\boldsymbol{\zeta}$ and unit byte compensation $\boldsymbol{C}$ determined by the system designer is shown at the top of next page after another, which can cover the total expenses of the system's cloud nodes while achieving optimal social welfare. The proof is detailed in Appendix D.*

When $U_H \le n_c\eta\sigma + \frac{\xi}{\Phi}$, both high-utility and low-utility users in the system have relatively low utility levels, and the revenue is insufficient to cover the transaction fee required to compensate for the total expenses of the system's cloud nodes as well as the waiting costs of other users. Therefore, users tend not to generate transactions. When $U_H > n_c\eta\sigma + \frac{\xi}{\Phi}$ and $U_L < n_c\eta\sigma + \frac{\xi(n_H+n_L)^2}{\Phi n_L^2}$, high-utility users in the system have sufficiently high utility levels, so they will generate transactions and pay sufficient transaction fee and compensation. However, low-utility users still tend not to generate transactions. When $U_L \ge n_c\eta\sigma + \frac{\xi(n_H+n_L)^2}{\Phi n_L^2}$, both types of users in the system will generate transactions.

Vector of unit byte transaction fee $\boldsymbol{\zeta} = (\zeta_1, \zeta_2, \dots, \zeta_I)$, $\forall q \in [1, I]$, we have

$$\zeta_q = n_c\eta + \frac{(I-q)\xi}{\Phi\sigma}$$

Vector of unit byte compensation $\boldsymbol{C} = (C_{HH}, C_{HL}, C_{LH}, C_{LL})$ satisfies:

- If $U_H \le n_c\eta\sigma + \dfrac{\xi}{\Phi}$

$$\begin{cases} (n_H-1)C_{HH} + n_L C_{HL} = 0 \\ n_H C_{LH} + (n_L-1)C_{LL} = 0 \end{cases}$$

- If $U_H > n_c\eta\sigma + \dfrac{\xi}{\Phi}$

$$\begin{cases} (n_H-1)C_{HH} + n_L C_{HL} = \dfrac{U_H}{\sigma} - \zeta_I - \dfrac{\xi[\Phi-(n_H-1)g_1 - n_L g_2]}{(\Phi - n_H g_1 - n_L g_2)^2\sigma} \\ n_H C_{LH} + (n_L-1)C_{LL} = \dfrac{U_L}{\sigma} - \zeta_I - \dfrac{\xi[\Phi - n_H g_1 - (n_L-1)g_2]}{(\Phi - n_H g_1 - n_L g_2)^2\sigma} \end{cases}$$

Where:

$$\begin{cases} g_1 = min(\dfrac{\Phi}{n_H+n_L}, \dfrac{1}{n_H}[\Phi - \sqrt{\dfrac{\Phi\xi}{U_H - n_c\eta\sigma}}]) \\ g_2 = \begin{cases} \text{If} U_L < n_c\eta\sigma + \dfrac{\xi(n_H+n_L)^2}{\Phi n_L^2} \\ \dfrac{\Phi}{n_H+n_L} - \dfrac{1}{n_L}\sqrt{\dfrac{\Phi\xi}{U_L - n_c\eta\sigma}} \qquad \text{Others} \end{cases} \end{cases}$$

7. Numerical experiment

In this section, we conducted numerical simulations of EVONIncentive to determine the parameters corresponding to the proposed mechanism and validate its effectiveness.

Based on Game Model I, we implemented Iterative Algorithm 3 and determined the optimal monetary incentive for miner nodes for different non-mining revenue factors. We also studied the changes in miner node computational power devotement under different scenarios.

Based on Game Model II, we focused on simulating the transaction generation strategies of user nodes. We determined the options for transaction unit byte fee and unit byte compensations for different user utilities and time cost coefficients. We analyzed the changes in the average unit byte fee of user nodes and the unit byte compensations for different types of users.

Since incentives are designed based on consensus, EVONChain adopts a PoI consensus mechanism, which is actually an enhanced variant of PoW. Therefore, we considered introducing a similar PoW-based incentive design from Ethereum 1.0 protocol for comparative testing. We incorporated its incentive scheme into the numerical simulation process and compared it with EVONIncentive under the same parameter conditions. We examined social welfare, user profit gaps, and Jain fairness index under changes in user utility and node count, demonstrating the advantages of EVONIncentive.

Table 3
Parameter settings for numerical simulation.

Outer User Node Count	$n_u = 5000$
Outer Miner Node Count	$n_m = 2000$
Inner Cloud Node Count	$n_c = 100$
Total Computation Power of Miner Nodes	$p = 1$
Main Block Time (seconds)	$T = 20$
System Throughput	$\Phi = 375{,}000$
Average Transaction Size (bytes)	$\sigma = 250$
User Transaction On-Chain Utility Ratio	$U_H : U_L = 2:1$
User Type Count Ratio	$n_H : n_L = 1:3$
Cloud Node Unit Byte Bandwidth Cost [55]	$\eta_{bandwidth} = 7 \times 10^{-11}$
Cloud Node Unit Byte Storage Cost [56]	$\eta_{storage} = 1 \times 10^{-10}$
Cloud Node Total Unit Byte Cost (USD/byte)	$\eta = 1.7 \times 10^{-10}$
Fee Options	$I = 10$

Table 4
Parameter Settings for Comparative Testing.

Number of User Nodes	$n_u = 600{,}000$
Number of Full Nodes	$n_c = 6500$
System Throughput	$\Phi = 16$
Average Transaction Size (bytes)	$\sigma = 150$
User On-chain Utility Ratio	$U_H : U_L = 2:1$
User Type Quantity Ratio	$n_H : n_L = 1:3$
Cloud Node Unit Byte Total Cost (USD/byte)	$\eta = 1.7 \times 10^{-10}$
Fee Options	$I = 10$

7.1. Experimental settings

Referring to settings and results of EVONChain, we adopted the parameter settings as shown in Table 3:

Consider a total of 5000 outer edge nodes, all serving as external user nodes in the system, freely generating transactions. The user nodes are divided into high-ulitity and low-ulitity users based on different on-chain utilities, with a utility ratio of 2:1 and a quantity ratio of 1:3.

Additionally, there are 2000 outer nodes willing to devote their computational power to the Proof-of-Intelligence (PoI) process of the system, acting as outer miner nodes, to receive the system's monetary incentives. It is assumed that each miner node has the same computational power, represented as 1 computational unit.

There are 100 high-performance cloud nodes in the inner layer, responsible for transaction processing, block and block header broadcasting, and storage of blockchain data.

The system's main block generation cycle is 20 s, with a throughput of approximately 375,000 TPS (transactions per second), and an average transaction size of 250 bytes.

In the comparative testing, to ensure the reliability of the experimental results, we set the blockchain-related parameters based on the Ethereum 1.0 historical data, as shown in Table 4:

As of the London upgrade, there were approximately 600,000 active user nodes and about 6500 full nodes responsible for accounting in the Ethereum network, with a system throughput of around 16 TPS and an average transaction size of 150 bytes.

7.2. Miner node incentives

We measure the incentives for miner nodes by the amount of system monetary they can obtain after completing the PoI process, denoted as M_0. Fig. 3 illustrates the system's optimal monetary reward obtained through the monetary incentive sequential iterative algorithm under different non-mining income factor conditions, as well as the variation in average computational power devoted by miner nodes.

When the non-mining income factor γ is large, miner nodes earn more income from non-mining tasks, resulting in weaker incentives to devote computational power to the PoI process. Therefore, a larger system mining monetary reward M_0 is needed to incentivize more miner computational power devotement. The trendlines in the graph confirm

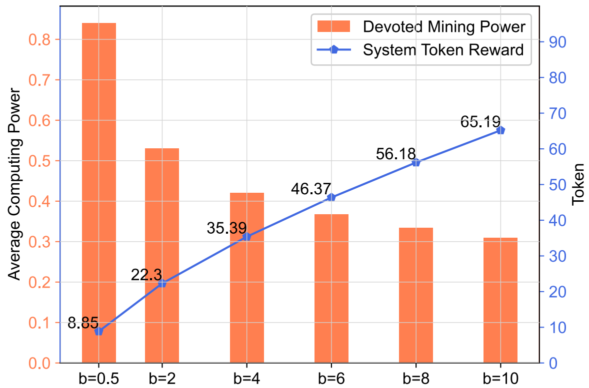


Fig. 3. System's Optimal Monetary Reward and Average Computational Power Devoted by Miner Nodes under Different Non-mining Income Factors.

this intuition. When the non-mining income factor is 0.5, the optimal monetary reward allocated by EVONIncentive is 8.85, which gradually increases with the increase in the non-mining income factor. When the non-mining income factor is 10, the optimal monetary reward is 65.19.

Additionally, we observe the variation in average computational power devoted into the PoI process with respect to γ. It can be seen that as the non-mining income factor increases, the average miner input power gradually decreases. When the non-mining income factor is 0.5, the average miner input power is greater than 0.8 units, accounting for 80 % of the total computational power. This indicates that participation in the PoI process dominates the miner node's computational power allocation at this point. As the non-mining income factor increases, the average miner input power gradually decreases. When the non-mining income factor is 10, the average miner input power decreases to approximately 0.3 units, indicating a decreased enthusiasm among miner nodes to participate in the PoI.

7.3. Cloud and user node incentives

Upon confirmation of blocks, all transaction fee contained within are awarded to the cloud node responsible for block creation. Additionally, the compensation paid by outer-layer user nodes upon transaction completion serves as an incentive for other user nodes remaining within the system.

7.3.1. Unit byte fee

In this subsection, we investigate how the transaction utility U_H and time cost coefficient ξ affect the average selected unit byte fee by users. We set the transaction utility for high-utility users to $[7\times10^{-6}, 1\times10^{-5}]$ and derive the transaction utility for low-utility users based on the utility ratio from Table 3. The time cost coefficient is set to $[1\times10^{-5}, 1\times10^{-3}]$. Under this setting, the average TPS ranges from 93,750 to 372,645, aligning with the throughput experimental results of EVONChain. Fig. 4 illustrates the variation in the average selected unit byte fee by users under different transaction utilities and time cost coefficients.

Fig. 4 provides direct experimental validation for the sustainability claim of EVONIncentive. The lines representing total cost burden correspond to the bandwidth and storage expenses per byte for all cloud nodes, $n_c\eta$. Observing this as a benchmark, we note that regardless of changes in user parameters, EVONIncentive consistently provides sufficient transaction fee for all full nodes in the system to cover the total costs.This ensures that the economic model is sustainable, as the revenue generated from transactions is sufficient to cover the costs that incentivize full nodes to remain in the network, preventing their attrition.

Fig. 4(a) indicates that when user utility is low, user nodes tend to select the lowest unit byte fee to ensure their own profitability. As user utility increases, the selected unit byte fee by user nodes also increases gradually until the highest option is chosen. At this point, user nodes have greater incentive to drive transaction confirmations to obtain chain utility rather than saving costs.

Fig. 4(b) reflects a similar trend. When user transaction utility is adequate, user nodes are willing to pay higher unit byte fee due to the higher cost of waiting time, enabling faster transaction confirmations.

7.3.2. Unit byte compensation

To limit high-utility users from generating a large number of high-cost transactions, resulting in indefinite delays for transactions of other ordinary user nodes, EVONIncentive introduces the concept of compensation as a remedy for new transaction generation preempting system common resources. User parameters are set as in the previous subsection. Fig. 5 illustrates the variation in compensation required to pay by different types of users for transaction generation under different transaction utilities and time cost coefficients.

Fig. 5(a) demonstrates that the unit byte compensation increases with the increase in transaction utility, and high-utility users need to pay significantly higher compensation after generating transactions compared to low-utility users. When U_H is less than 8.5×10^{-6} and U_L is less than 4.25×10^{-6}, low-utility users do not generate transactions due to insufficient chain utility, resulting in an average compensation of 0. As user utility increases, the unit byte compensation to be paid by high-utility users gradually increases, and that for low-utility users also increases but to a lesser extent.

Fig. 5(b) indicates that when user chain utility is sufficient and remains constant, both types of users generate transactions. With changes in the time cost coefficient, the compensation values remain relatively constant, with high-utility users still requiring to pay significantly higher compensation than low-utility users.

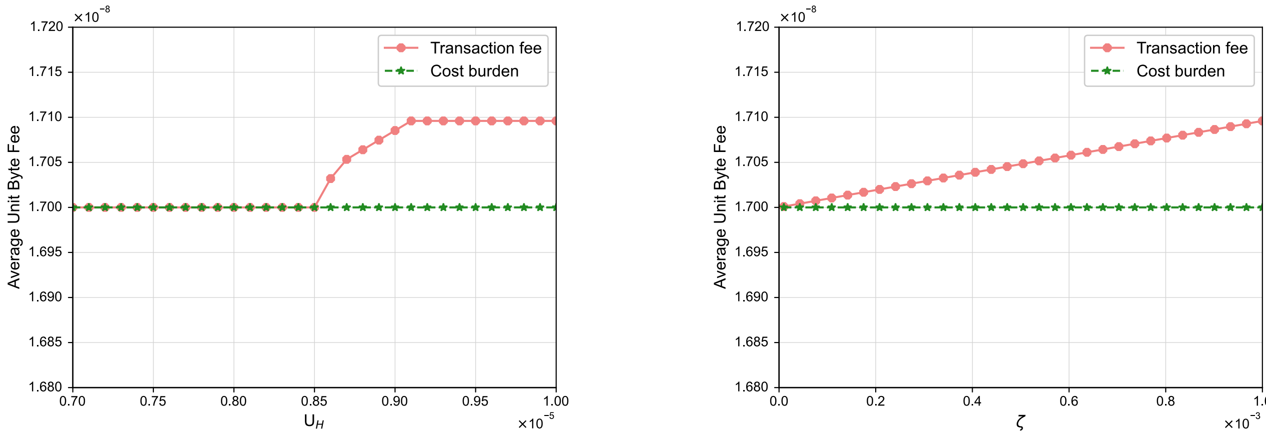
7.4. Comparative testing

Next, we will conduct numerical simulations under the same experimental settings for both Ethereum1.0′s incentive mechanism and EVONIncentive to demonstrate the advantages of EVONIncentive through comparison. Referring to the parameters set in Table 4, we investigate the impact of changes in user transaction utility U_H and the number of user nodes n_u on system social welfare, user income, and fairness. The range of transaction utility U_H is set to $[1.4\times10^{-4}, 6.7\times10^{-4}]$, resulting in a daily transaction volume of between 1.1 million and 1.38 million transactions, which aligns with the average daily transaction volume range of Ethereum 1.0 before the London upgrade, between 1.1 million and 1.35 million transactions. Similarly, the pre-London upgrade daily average active user range of Ethereum 1.0 is $[4.28\times10^5, 7.95\times10^5]$, which we adopt as the range for the number of user nodes n_u. The time cost coefficient ξ is uniformly set to 2×10^{-6}.

7.4.1. System social welfare

Fig. 6 illustrates the variation in system social welfare generated by EVONIncentive and Ethereum 1.0 incentive mechanism with changes in user chain utility U_H. It can be observed that the system social welfare of both incentive mechanisms increases with the increase in chain utility, but the social welfare level of EVONIncentive is consistently higher than that of the Ethereum 1.0 incentive mechanism, and the gap gradually widens.

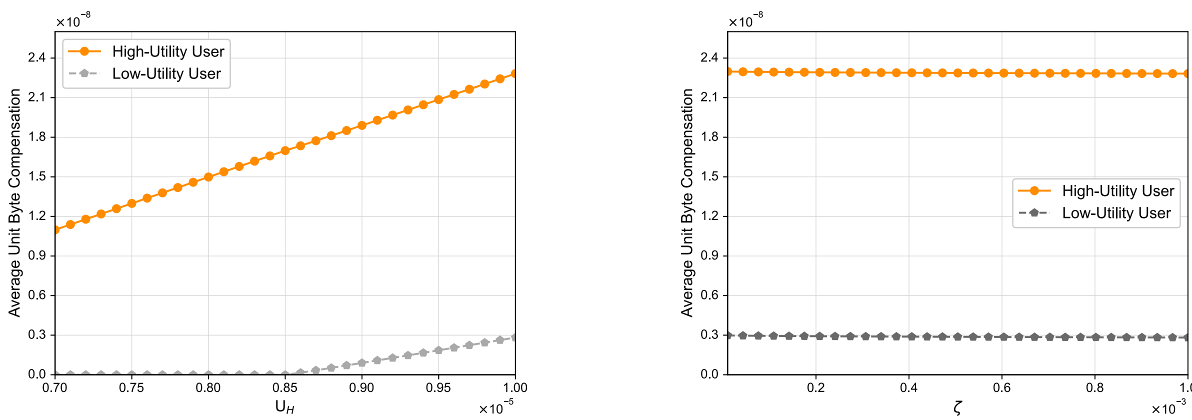
When the user chain utility is low, the social welfare of the Ethereum 1.0 incentive mechanism is negative, while the social welfare level of EVONIncentive remains non-negative. This phenomenon occurs because EVONIncentive ensures that users must generate transactions with fee sufficient to cover the total expenses of all system cloud nodes, while the Ethereum 1.0 incentive mechanism does not restrict user transaction fee, allowing users to generate transactions with fee lower than the total expenses of all system cloud nodes, resulting in negative overall system



(a) Average unit byte fee selected by users under different transaction utilities. $\xi = 1 \times 10^{-3}$

(b) Average unit byte fee selected by users under different time cost coefficients. $U_H = 1 \times 10^{-5}$

Fig. 4. Average unit byte fee selected by users under different user parameters.



(a) Average unit byte compensation required by users under different transaction utilities. $\xi = 1 \times 10^{-3}$

(b) Average unit byte compensation required by users under different time cost coefficients. $U_H = 1 \times 10^{-5}$

Fig. 5. Variation in average unit byte compensation for different user types with changes in user parameters.

social welfare. The possibility of system cloud nodes exiting the system increases when their expenses exceed their revenue, thereby weakening the system's sustainability.

The experimental results indicate that system social welfare is unaffected by the number of user nodes and is not included in the discussion.

7.4.2. User income

In Fig. 7, we plot the levels of income for high-utility and low-utility users under EVONIncentive and Ethereum 1.0 incentive mechanism with changes in user chain utility U_H and the total number of users n_u.

Fig. 7(a) illustrates that the time average income of all user types increases with the increase in transaction utility. However, under equivalent conditions, the income of high-utility users in the Ethereum 1.0 incentive mechanism is approximately 300 % higher than that of low-utility users, resulting in a significant wealth gap among system users. In comparison, the income gap between high-utility and low-utility users in EVONIncentive is lower (experimental data indicate that the income of high-utility users is still slightly higher than that of low-utility users), primarily due to the introduction of the compensation mechanism in EVONIncentive, which provides sufficient compensation to low-utility user nodes, increasing their incentive to remain in the system.

Fig. 7(b) demonstrates that the time average income of users decreases with the increase in the total number of users. This is because the increase in the total number of users intensifies transaction competition in the system, affecting income. However, the income gap among users in EVONIncentive remains smaller than that in the Ethereum 1.0 incentive mechanism.

7.4.3. Fairness

We utilize the Jain fairness index to measure the fairness of EVONIncentive and the Ethereum 1.0 incentive mechanism. The calculation

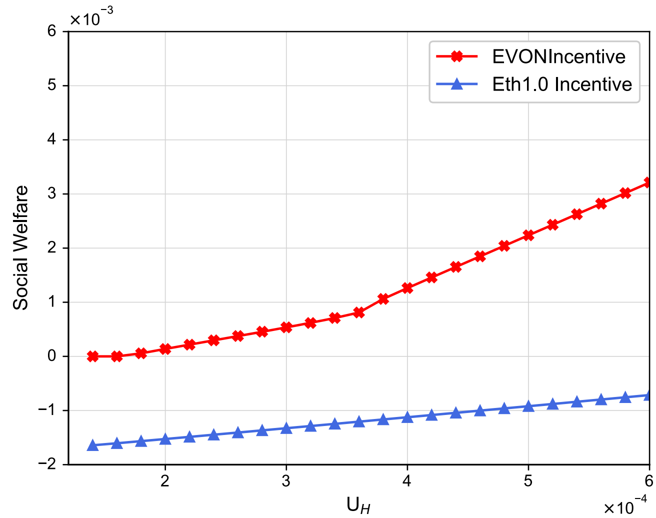
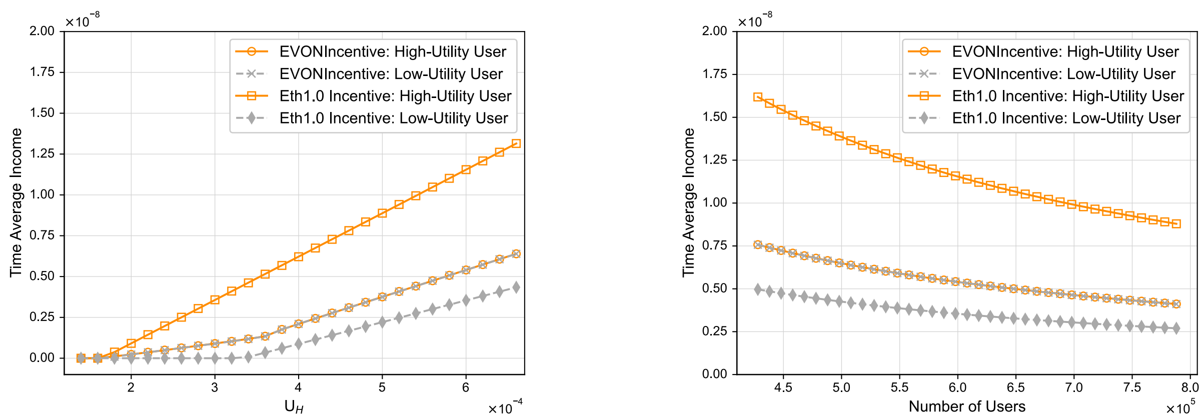


Fig. 6. Variations of system social welfare under different user chain utilities for EVONIncentive and Ethereum 1.0 incentive mechanism.



(a) Levels of income for different types of users under varying transaction utility. $n_c = 6 \times 10^5$

(b) Levels of income for different types of users under varying total number of users. $U_H = 6 \times 10^{-4}$

Fig. 7. Variations in user income for EVONIncentive and Ethereum 1.0 incentive mechanism with changes in transaction utility and total number of users.

formula for the Jain fairness index is as follows:

$$J = \frac{(\sum_{i=1}^{N} \pi_{u_i})^2}{n_u \sum_{i=1}^{N} \pi_{u_i}^2} \tag{24}$$

Fig. 8 displays the differences in the Jain fairness index between EVONIncentive and the Ethereum 1.0 incentive mechanism.

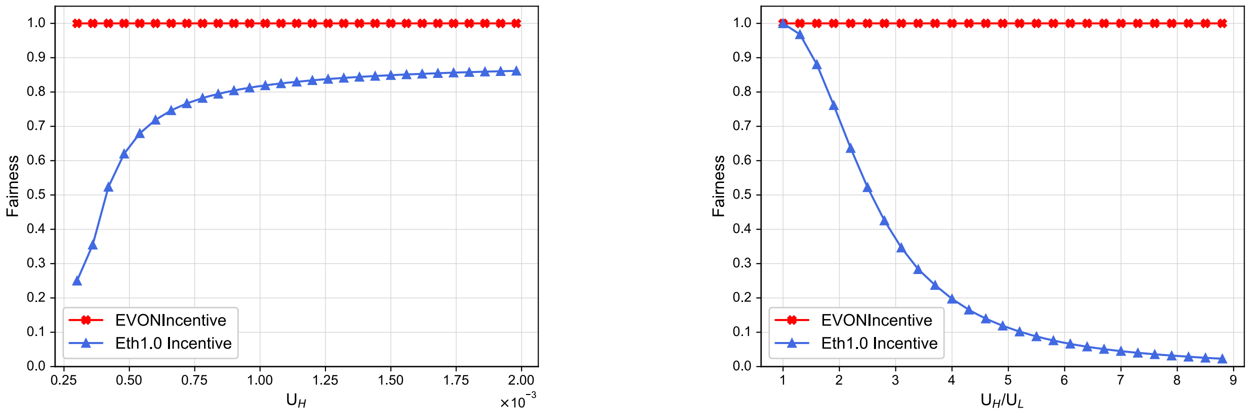
In Fig. 8(a), when the user transaction utility is low, the fairness index of the Ethereum 1.0 incentive mechanism is below 0.3. This is because high-utility users generate transactions with higher transaction fee, which are prioritized by cloud nodes for packaging and inclusion in the blockchain, thereby gaining profits. In contrast, transactions generated by low-utility users are delayed due to lower transaction fee, leading to increased waiting time costs and an inability to profit. Under equivalent conditions, the fairness index of EVONIncentive approaches 1 because the compensation mechanism mitigates the income gap among users.

When the user chain utility is sufficiently high, the fairness index of the Ethereum 1.0 incentive mechanism stabilizes at around 0.86. At this point, the fairness index of EVONIncentive approaches 1, maintaining a close to 15 % advantage.

In Fig. 8(b), as the utility ratio between high-utility and low-utility users increases, the fairness index of the Ethereum 1.0 incentive mechanism continuously decreases, while the fairness index of EVONIncentive remains close to 1, indicating that EVONIncentive maintains high fairness even when the utility ratio between users is large.

The significantly higher Jain's fairness index of EVONIncentive not only indicates a more equitable distribution of income but also contributes to enhanced system security. A system with extreme wealth disparity (low fairness) is more vulnerable to coercion and targeted attacks on a few wealthy participants (e.g., bribery attacks). By promoting a more balanced and fair economic environment, EVONIncentive reduces this systemic vulnerability, making the network more resilient and secure.

The experimental results demonstrate that EVONIncentive has significant advantages over the Ethereum 1.0 incentive mechanism in terms of system social welfare, user income gap, and fairness, validating the effectiveness and scientific nature of EVONIncentive.



(a) Jain fairness index under different transaction utilities. $n_c = 6 \times 10^5$, U_H: $U_L = 2{:}1$

(b) Jain fairness index under different user utility ratios. $n_c = 6{\times}10^5$, $U_H = 6 \times 10^{-4}$

Fig. 8. Fairness of EVONIncentive and Ethereum 1.0 incentive mechanism with respect to variations in transaction utility and user utility ratio.

8. Conclusions

We propose EVONIncentive, a secure hybrid incentive mechanism for the early-stage developed bi-tiered public chain called EVONChain. We model the interactions among system designers, light nodes, miner nodes, and full nodes as two Stackelberg games and propose a miner incentive allocation algorithm based on the Gauss-Seidel iteration method. Through numerical simulation experiments, specific values for parameters related to mining monetary incentives, transaction fee, and compensation are determined in EVONIncentive. Comparative experiments with the Ethereum 1.0 incentive mechanism demonstrate the advantages of EVONIncentive in terms of system social welfare, user income, and fairness.

Our work focuses on innovations in incentive mechanisms within the Proof-of-Work (PoW) consensus paradigm. An important direction for future work involves abstracting the general design principles of EVONIncentive–such as fee selection menus, compensation mechanisms, and reputation-binding schemes–into a more universal economic framework, and exploring its applicability to other consensus mechanisms like Proof-of-Stake (PoS). We will endeavor to conduct comparative studies across consensus mechanisms when models and data conditions permit, so as to further validate the efficacy and robustness of these core ideas in diverse environments.

Besides, we are also willing to refine the design of the reputation mechanism, including designing different reputation deduction mechanisms for specific malicious behaviors of nodes. Additionally, we plan to investigate different designs for system incentives when nodes dynamically enter and exit.

CRediT authorship contribution statement

Jing Li: Writing – original draft, Visualization, Validation, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization; **Tao Xie:** Writing – review & editing, Validation, Supervision, Methodology, Conceptualization; **Yiwei Liu:** Writing – review & editing, Validation, Supervision.

Data availability

Data will be made available on request.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] S. Nakamoto, Bitcoin: a peer-to-peer electronic cash system, Available at SSRN 3440802 (2008).
[2] V.J. Morkunas, J. Paschen, E. Boon, How blockchain technologies impact your business model, Bus. Horiz. 62 (3) (2019) 295–306.
[3] A. Pal, C.K. Tiwari, A. Behl, Blockchain technology in financial services: a comprehensive review of the literature, J. Global Oper. Strategic Sourcing 14 (1) (2021) 61–80.
[4] S. Trivedi, K. Mehta, R. Sharma, Systematic literature review on application of blockchain technology in E-finance and financial services, J. Technol. Manage. Innovation 16 (3) (2021) 89–102.
[5] L. Bell, W.J. Buchanan, J. Cameron, O. Lo, Applications of blockchain within healthcare, Blockchain in healthcare today 1 (2018) https://www.blockchainhealthcaretoday.com/index.php/journal/article/view/8.
[6] S. Abdallah, N. Nizamuddin, Blockchain-based solution for pharma supply chain industry, Comput. Ind. Eng. 177 (2023) 108997.
[7] M.M. Queiroz, R. Telles, S.H. Bonilla, Blockchain and supply chain management integration: a systematic review of the literature, Supply Chain Manage. Int. J. 25 (2) (2020) 241–254.
[8] S. Abbate, P. Centobelli, R. Cerchione, E. Oropallo, E. Riccio, Blockchain technology for embracing healthcare 4.0, IEEE Trans. Eng. Manage. 70 (8) (2022) 2998–3009.
[9] O. Bychkova, A. Kosmarski, Imagineering a new way of governing: the blockchain and res publica, J. Inf. Technol. Polit. 20 (1) (2023) 34–43.
[10] Á.F. Alcaide, C. Núñez-Gómez, F.M. Delicado, C. Carrión, M.B. Caminero, A blockchain-based e-government service for quantity surveyors, IT Prof. 25 (4) (2023) 61–66.
[11] S. King, S. Nadal, PPCoin: peer-to-peer crypto-currency with proof-of-stake, self published pap. August 19 (1) (2012).
[12] D. Larimer, Dpos consensus algorithm-the missing white paper, Bitshare whitepaper (2017) https://steemit.com/dpos/@dantheman/dpos-consensus-algorithm-this-missing-white-paper.
[13] G. Wood, et al., ETHEREUM: a secure decentralised generalised transaction ledger, Ethereum project yellow pap. 151 (2014) (2014) 1–32.
[14] I. Eyal, A.E. Gencer, E.G. Sirer, R. Van Renesse, {Bitcoin-NG}: A scalable blockchain protocol, in: 13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16), 2016, pp. 45–59.
[15] Y. Sompolinsky, Y. Lewenberg, A. Zohar, SPECTRE: a fast and scalable cryptocurrency protocol, Cryptology ePrint Archive, Paper 2016/1159 (2016) https://eprint.iacr.org/2016/1159.
[16] Y. Sompolinsky, A. Zohar, PHANTOM: A scalable blockDAG protocol, IACR Cryptology ePrint Arch 2018 (2018) 104 http://eprint.iacr.org/2018/104.
[17] C. Li, P. Li, D. Zhou, Z. Yang, M. Wu, G. Yang, W. Xu, F. Long, A.C.-C. Yao, A decentralized blockchain with high throughput and fast confirmation, in: 2020 {USENIX} Annual Technical Conference {USENIX} Aptrddollar; {ATC20)}, 2020, pp. 515–528.

[18] H. Yu, I. Nikolić, R. Hou, P. Saxena, OHIE: blockchain scaling made simple, in: 2020 IEEE Symposium on Security and Privacy (SP), IEEE, 2020, pp. 90–105.
[19] T. Xiong, T. Xie, J. Xie, X. Luo, ORIC: A self-adjusting blockchain protocol with high throughput, in: 2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom), IEEE, 2021, pp. 1422–1434.
[20] H. Huang, Z. Huang, X. Peng, Z. Zheng, S. Guo, MVCom: Scheduling most valuable committees for the large-scale sharded blockchain, in: 2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS), IEEE, 2021, pp. 629–639.
[21] H. Huang, X. Peng, J. Zhan, S. Zhang, Y. Lin, Z. Zheng, S. Guo, BrokerChain: a cross-shard blockchain protocol for account/balance-based state sharding, in: IEEE INFOCOM 2022-IEEE Conference on Computer Communications, IEEE, 2022, pp. 1968–1977.
[22] I. Eyal, E.G. Sirer, Majority is not enough: bitcoin mining is vulnerable, Commun. ACM 61 (7) (2018) 95–102.
[23] M. Rosenfeld, Analysis of bitcoin pooled mining reward systems, arXiv:1112.4980 (2011).
[24] Y. Kong, J. Li, T. Xiong, T. Xie, EVONChain: A bi-tiered public blockchain network architecture, Peer-to-Peer Netw. Appl. 16 (6) (2023) 2892–2914.
[25] R. Han, Z. Yan, X. Liang, L.T. Yang, How can incentive mechanisms and blockchain benefit with each other? a survey, ACM Comput. Surv. 55 (7) (2022) 1–38.
[26] J. Huang, K. Lei, M. Du, H. Zhao, H. Liu, J. Liu, Z. Qi, Survey on blockchain incentive mechanism, in: Data Science: 5th International Conference of Pioneering Computer Scientists, Engineers and Educators, ICPCSEE 2019, Guilin, China, September 20–23, 2019, Proceedings, Part I 5, Springer, 2019, pp. 386–395.
[27] Z. Yu, X. Liu, G. Wang, A survey of consensus and incentive mechanism in blockchain derived from P2P, in: 2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS), IEEE, 2018, pp. 1010–1015.
[28] Z. Liu, N.C. Luong, W. Wang, D. Niyato, P. Wang, Y.-C. Liang, D.I. Kim, A survey on blockchain: a game theoretical perspective, IEEE Access 7 (2019) 47615–47643.
[29] Y. Huang, J. Tang, Q. Cong, A. Lim, J. Xu, Do the rich get richer? fairness analysis for blockchain incentives, in: Proceedings of the 2021 International Conference on Management of Data, 2021, pp. 790–803.
[30] G. Huberman, J.D. Leshno, C. Moallemi, Monopoly without a monopolist: an economic analysis of the bitcoin payment system, Rev. Econ. Stud. 88 (6) (2021) 3011–3040.
[31] I. Tsabary, I. Eyal, The gap game, in: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, 2018, pp. 713–728.
[32] S. Jiang, J. Wu, Bitcoin mining with transaction fees: a game on the block size, in: 2019 IEEE International Conference on Blockchain (Blockchain), IEEE, 2019, pp. 107–115.
[33] J. Ma, J.S. Gans, R. Tourky, Market Structure in Bitcoin Mining, Technical Report, National Bureau of Economic Research, 2018.
[34] E. Altman, D. Menasché, A. Reiffers-Masson, M. Datar, S. Dhamal, C. Touati, R. El-Azouzi, Blockchain competition between miners: a game theoretic perspective, Front. Blockchain 2 (2020) 26.
[35] A. Aldweesh, Analysis, evaluation and benchmark the ethereum incentive mechanism, in: 2021 IEEE 11th International Conference onElectronics Information and Emergency Communication (ICEIEC) 2021 IEEE 11th International Conference on Electronics Information and Emergency Communication (ICEIEC), IEEE, 2021, pp. 1–4.
[36] A.C.-C. Yao, An incentive analysis of some bitcoin fee designs, arXiv:1811.02351 (2018).
[37] G. Xue, J. Xu, H. Wu, W. Lu, L. Xu, Incentive mechanism for rational miners in bitcoin mining pool, Inf. Syst. Front. 23 (2021) 317–327.
[38] P.R. Rizun, A transaction fee market exists without a block size limit, Block Size Limit Debate Working Pap. (2015) https://api.semanticscholar.org/CorpusID:19451777.
[39] R. Zhang, B. Preneel, On the necessity of a prescribed block validity consensus: analyzing bitcoin unlimited mining protocol, in: Proceedings of the 13th International Conference on Emerging Networking EXperiments and Technologies, 2017, pp. 108–119.
[40] C. Jonathan, K.V. Thorsten, The economics of cryptocurrencies-bitcoin and beyond, JEL Classi_cation E 4 (2018) E5.
[41] E. Pagnotta, Bitcoin as decentralized money: prices, mining rewards, and network security, SSRN Electron. J. (2018) https://doi.org/10.2139/ssrn.3264448.
[42] P.-O. Goffard, Fraud risk assessment within blockchain transactions, Adv. Appl. Probab. 51 (2) (2019) 443–467.
[43] Z. Wang, Q. Wu, Incentive for historical block data sharing in blockchain, in: 2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), IEEE, 2019, pp. 0913–0919.
[44] Y. Sompolinsky, A. Zohar, Bitcoin's underlying incentives, Commun. ACM 61 (3) (2018) 46–53.
[45] S. Fu, X. Huang, L. Liu, Y. Luo, BFCRI: A blockchain-based framework for crowdsourcing with reputation and incentive, IEEE Trans. Cloud Comput. 11 (2) (2022) 2158–2174.
[46] Y. Zhang, M. Van der Schaar, Reputation-based incentive protocols in crowdsourcing applications, in: 2012 Proceedings IEEE INFOCOM, IEEE, 2012, pp. 2140–2148.
[47] T. Burnham, R. Sami, A reputation system for selling human computation, in: Proceedings of the ACM SIGKDD Workshop on Human Computation, 2009, pp. 54–57.
[48] C. Ihle, D. Trautwein, M. Schubotz, N. Meuschke, B. Gipp, Incentive mechanisms in peer-to-peer networks–a systematic literature review, ACM Comput. Surv. 55 (14s) (2023) 1–69.
[49] J. Chang, Z. Pang, W. Xu, H. Wang, G. Yin, An incentive compatible reputation mechanism for P2P systems, J. Supercomput. 69 (3) (2014) 1382–1409.
[50] R. Zeng, C. Zeng, X. Wang, B. Li, X. Chu, Incentive mechanisms in federated learning and a game-theoretical approach, IEEE Netw. 36 (6) (2022) 229–235.
[51] R. Zeng, M. Zhao, J. Han, Y. Bi, X. Wang, LLMCatalyst: A novel incentive mechanism for client-assisted foundation model training, in: 2025 IEEE/ACM 33rd International Symposium on Quality of Service (IWQoS), IEEE, 2025, pp. 1–10.
[52] D. Bertsekas, J. Tsitsiklis, Parallel and Distributed Computation: Numerical Methods, 2015.
[53] D. Easley, M. O'Hara, S. Basu, From mining to markets: the evolution of bitcoin transaction fees, J. Financ. Econ. 134 (1) (2019) 91–109.
[54] D. Easley, M. O'Hara, S. Basu, From mining to markets: the evolution of bitcoin transaction fees, J. Financ. Econ. 134 (1) (2019) 91–109.
[55] 2024, https://developer.aliyun.com/article/1179462.
[56] 2024, https://www.amazon.com/ssd/s?k=ssd.