A Storage Sustainability Mechanism With Heterogeneous Miners in Blockchain

Yunshu Liu, *Member, IEEE*, Shulin Ke, Zhixuan Fang, *Member, IEEE*, Man Hon Cheung,
Wei Cai, *Senior Member, IEEE*, and Jianwei Huang, *Fellow, IEEE*

Abstract—In current blockchain systems, the transaction fee is often not enough to cover the storage cost, jeopardizing blockchain sustainability in the long run. Such a storage sustainability issue is partially due to miners' heterogeneous storage costs and users' low-intensity fee competition. Motivated by these two observations, we propose a Fee and Transaction Expiration Time (FTET) mechanism to alleviate this issue. Specifically, we model the blockchain operation as a three-stage game. In Stage I, the system designer proposes the storage sustainability mechanism. In Stage II, each user decides whether to propose transactions and the corresponding transaction fees. In Stage III, each miner decides which transactions to include in the block. Although the analysis of the heterogeneous miner interaction is technically challenging, we fully solve it in closed-form motivated by how miners select transactions in practice. The equilibrium analysis reveals that high-storage-cost miners admit transactions with fees above a time-increasing threshold. Under the optimal FTET mechanism, the blockchain system can achieve the storage sustainability without any social welfare loss, comparing with the maximum achievable social welfare without the storage sustainability constraint. Moreover, the optimal FTET mechanism achieves a higher social welfare than the fee mechanism in current practice by selectively rejecting some transactions suffering high delays. Finally, we implement a blockchain prototype to compare the performance of the optimal FTET mechanism with the mining round time adjustment (MRTA) mechanism. The optimal FTET mechanism achieves higher social welfare (94.5% on average) and better storage sustainability. We find that more pending transactions may lead to lower transaction fees.

Index Terms—Blockchain storage, transaction fee, incentive mechanism design, game theory.

I. INTRODUCTION

IN THE blockchain protocol, miners (i.e., protocol-running nodes) bear significant yet heterogeneous storage costs. The primary storage cost comes from the use of solid-state drive [1]. Let us take Ethereum as an example, which is the second-largest blockchain system today measured by the market value [2]. Ethereum mainly offers two storage modes with the following storage costs (early 2020 data [1]): the default mode of "full node" costs a miner $300/month and the mode of "archival node" imposes a cost of $2000/month. The heterogeneity comes from the difference of data storage, i.e., 331 GB for a full node and 4.1 TB for an archive node in May 2020, respectively.[1]

To ensure the long-term sustainability of the blockchain system, it is important for the miners to receive enough compensation to cover their storage costs. However, the transactions fees in the current blockchain systems are not enough to cover the storage costs. More specifically, when a blockchain user proposes a transaction (e.g., transfer some cryptocurrency to another user), he will pay the transaction fee to the miner who includes this particular transaction into a block. As almost all data of blockchain are transactions [5], transaction fees play a key role in compensating the miners' storage costs.[2] Nevertheless, transaction fees are insufficient in the current practice. For example, in the first four months of 2020, monthly average transaction fees in Ethereum is $2.90 million, insufficient to cover at least $3.60 million monthly storage costs of all miners (about 12,000 miners in early 2020 [7] and at least $300 monthly storage cost per node).

The imbalance between the user's transaction fee payment and miner's storage cost may lead to the decline of both

[1]For a miner who stores blockchain data locally, the storage cost is still significant. Current SSD drive's price is about $100/TB [3]. Ethereum's data size has grown about 4.15 times in the past three years [4]. If the growth rate continues, the total SSD disk cost of running an archive node for three years is about $4490.

[2]Although the block reward (e.g., in Bitcoin, finding a block can get some Bitcoin cryptocurrency as reward) can also compensate miners' storage costs, it is not a long-term solution since it gradually decreases to give the cryptocurrency anti-inflation property (e.g., Bitcoin [6]).

users and miners. From 2018, about two-thirds of miners have left the Ethereum system [8], where the lack of coverage for storage cost could be an important reason. If such a drop continues, the blockchain system will be difficult to defend the 51% attack [9]. Moreover, as miners leave the system, it will become increasingly difficult to include users' transactions in blockchain and users may eventually choose to stop using the blockchain application. Fewer users mean lower transaction fees, which may cause an even faster decline of miners. Thus, balancing the transaction fee and storage cost is crucially important to ensure the blockchain's long-term sustainability.

There are several key reasons for the storage sustainability issue. One reason is *miners' heterogeneous storage costs*. Some miners have low storage costs such that they are willing to accept low-fee transactions. Another key reason is *users' low-intensity fee competition*. Specifically, users propose transaction fees to compete to be included in blockchain early. As long as the competition is not intense, users can choose to pay low fees and expect that low-cost miners will eventually include their transactions in blockchain. To the best of our knowledge, previous studies generally neglect the above key reasons. In our work, we design a new storage sustainability mechanism by considering the impact of these key issues, to ensure that users pay sufficient transaction fees for storage costs.

In this work, we focus on answering the following two key questions:

(i) *How do miners with heterogeneous storage costs admit transactions in blockchain?*

(ii) *How to control the users' fee competition intensity to ensure the blockchain storage sustainability?*

To answer the above questions, we propose a three-stage model to characterize the interactions in the blockchain system. In Stage I, we propose a *Fee and Transaction Expiration Time* (FTET) mechanism, which ensures the blockchain storage sustainability and maximizes the social welfare. The mechanism computes a minimum fee for users and transaction expiration time for transactions, motivated by two key reasons of storage sustainability issue. In Stage II, we model how each user proposes transactions to balance the transaction's fee payment and delay. In Stage III, we model how each miner includes transactions in blockchain system to balance the income of transaction fee and storage cost.

The key results and contributions of this paper are as follows.

- *Fee mechanism design on blockchain storage*: We focus on how to design a new transaction fee mechanism to address the storage sustainability issue in blockchain, ensuring the long-term stability of the system. Our study is an initial step to address the issue considering miners' heterogeneous storage costs and users' low fee competition intensity.

- *Considering miners with two dimensions of heterogeneity*: We consider miners with two dimensions of heterogeneity: storage cost and mining power. The analysis is technically challenging, as heterogeneous miners face different integer programming problems, which are

game-theoretically coupled. Motivated by how miners select transactions in practice, we derive the equilibrium in closed-form. The analysis reveals that high-storage-cost miners admit transactions with fees above a time-increasing threshold. This result serves as the foundation of the fee mechanism design.

- *Guarantee on storage sustainability with social optimality*: We propose an FTET mechanism motivated by two key reasons of the storage sustainability issue. We show that the optimal FTET mechanism ensures the storage sustainability of the blockchain, while achieving the same maximum social welfare when not considering the storage sustainability constraint. Moreover, the optimal FTET mechanism achieves higher social welfare than today's protocol used in Bitcoin and Ethereum by selectively rejecting some transactions suffering high delays.

- *Extensive real-world blockchain experiments*: We implement a blockchain prototype and conduct experiments to compare the optimal FTET mechanism with the state-of-art mining round time adjustment (MRTA) mechanism. The optimal FTET mechanism achieves better storage sustainability and an average social welfare improvement of 94.5%. The social welfare improvement is due to the transaction expiration rejecting high-delay transactions. Moreover, the transaction fees may decrease in the number of pending transactions.

The rest of the paper is organized as follows. Section II reviews the related literature. Sections III and IV introduce the blockchain system and mathematical model, respectively. We derive the system equilibrium in Section V and evaluate the system performance in Section VI. Section VII concludes the paper.

II. LITERATURE REVIEW

We classify the related works into two categories, i.e., the transaction fee design and storage reduction mechanism.

A. Transaction Fee Mechanism Design

Existing literature [10], [11], [12], [13], [14], [15], [16] proposed different transaction fee mechanisms to improve the blockchain system performance. For example, Liu et al. in [10] and [11] designed the mechanisms to mitigate the storage sustainability issue under homogeneous-storage-cost miners. Ethereum community proposed a burning base fee mechanism [12] to reduce transaction congestion. Hu et al. [13] modeled the users as a coalition and designed a coalition-based fee mechanism to achieve the social optimum. Some other works used the auction model to characterize how users pay transaction fees to compete for the transaction inclusion opportunities. Ai et al. [14] designed a fee mechanism based on the continuous double auction to maximize the fairness level. Lavi et al. [15] proved that in a monopolistic auction, the transaction fee does not decrease with the block size. Basu et al. [16] proved that the miners' revenue variance is lower in the second prize auction.

Comparing with the prior work, this study represents the first work considering the more realistic scenario of how
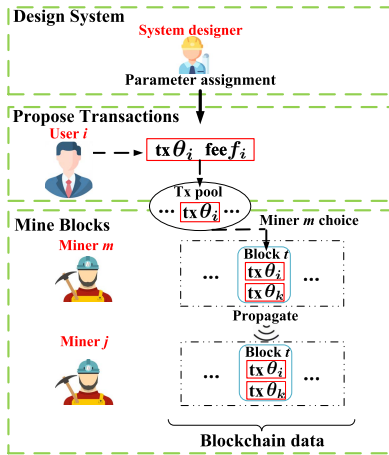
Fig. 1. Blockchain operation.

heterogeneous-storage-cost miners select transactions. Moreover, by carefully designing the minimum fee-per-byte and transaction expiration time, our proposed mechanism is able to control the users' transaction proposing behaviors, hence significantly improve upon the existing schemes in terms of both storage sustainability and social welfare.

B. Storage Reduction Mechanism

The other category of literature [17], [18], [19], [20], [21], [22] tried to ensure the storage sustainability in blockchain by reducing the storage costs. The first set of works [17], [18], [19] adopted the sharding technique. These protocols partition miners into smaller groups and each group only stores a fraction of transactions, such that each transaction imposes lower storage costs on the system. The second set of works [20], [21], [22] adopted the pruning method. These protocols generally reduce the time of storing a transaction, by removing the old transactions.

However, reducing storage costs does not necessarily guarantee the storage sustainability. Based on our analysis in Section V-A, when a miner bares a lower storage cost, he also has a lower fee-per-byte threshold of admitting transactions. Under the lower fee-per-byte threshold, users will pay lower transaction fees, which may still be insufficient to cover the storage cost. Thus, we still need an incentive mechanism to encourage users to pay sufficient transaction fees to guarantee storage sustainability.

III. BLOCKCHAIN OPERATION

In this section, we describe the operation of the blockchain system, focusing on various aspects that are essential to our study.

Fig. 1 shows the operation of the blockchain system, where three key entities interact with each other: the system designer determines the system protocol, users propose transactions (to be included in the blockchain), and miners produce blocks (that include transactions) through mining. The entities interact in three steps:

1) *System designer determines system protocol:*[3] The system designer defines the blockchain protocol and assigns the protocol parameters.[4] Usually, the blockchain community collectively serves as the system designer. The online community comprises of a group of users and miners. Everyone in the community can propose modifications of the existing protocol. When the majority of the community agrees with the protocol, the community will deploy such a protocol proposal. For example, several Ethereum developers proposed the "EIP-1559" protocol in Ethereum online community [23] in April 2019, aiming to alleviate the transaction congestion in Ethereum. This protocol gradually gained enough support and Ethereum community deployed the EIP-1559 protocol in August 2021.

2) *Users propose transactions:* A (generic) user i proposes a transaction (tx θ_i) together with the transaction fee (fee f_i) as an incentive for miners to append the transaction to the blockchain. A common approach is for the user to specify the fee-per-byte [24], and the transaction fee (for a particular transaction) is computed as: transaction fee = fee-per-byte $\times$ size of transaction. Then transactions enter the transaction pool (Tx pool), pending to be included in blockchain.

3) *Miners mine blocks:* Miners collectively mine a series of *blocks*[5] in rounds, indexed by $t = 1, 2, \cdots$. Let us consider a particular mining round t:

   a) First, each miner chooses some pending transactions from the transaction pool (e.g., miner m chooses tx θ_i and tx θ_k), to be included in the block (if miner m successfully mines the block in this round).

   b) Then, all miners compete to find the solution of a mathematical puzzle [25]. The miner who finds the solution first will obtain the right to generate the new block (containing his selected transactions) in the blockchain. Such a miner will also collect the corresponding transaction fees (e.g., miner m collects the transaction fees of tx θ_i and tx θ_k). The selected transactions (to be included in the block) will be removed from the transaction pool.

   c) Finally, the successful miner broadcasts the new block to the rest miners. *All* miners will store the block locally and bear the storage cost individually.

[3]The system designer only optimizes the mechanism (i.e., rule of the system) while considering the Nash equilibrium of users and miners. It does not directly determine any user's or miner's decision. At the Nash equilibrium, any unilateral deviation of a user or miner cannot increase his payoff.

[4]In the absence of the system designer, it is possible to implement and enforce our mechanism in a distributed manner. Take the Ethereum as an example. Each user and miner can broadcast his parameter in the form of a transaction. Once receiving these transactions, every user and miner can calculate the mechanism parameters in Theorem 3 of Section V-C in a distributed fashion. The broadcast transaction is generally smaller than 0.5 KB. There are about 12,000 miners and 415,000 daily active users in June 2022, the total data size is about 213.5 MB, which is not a large burden. To enforce the mechanism, miners can reject transactions and blocks that violate the mechanism.

[5]A block is a vehicle of transactions. It consists of the previous block's hash, transactions, the time-stamp, and some the auxiliary data in Bitcoin [6].

Based on the blockchain operation process, we will propose the system model in the next section.

IV. SYSTEM MODEL

In this section, we describe the system model of blockchain. Sections IV-A and IV-B introduce our key modeling assumptions and the basic problem setting, respectively. We then characterize the payoffs of individuals from blockchain and social welfare in Section IV-C. Finally, we formulate a Stackelberg game in Section IV-D.

A. Assumption

To facilitate the later analysis, we make the following assumptions about blockchain:

Assumption 1: All users propose transactions at the beginning of mining round 1 synchronously.

When a miner selects a transaction, he mainly considers the transaction's fee and storage cost, and cares little about the transaction's proposing time [24]. Thus, Assumption 1 allows us to derive insights on how heterogeneous miners select transactions, while simplifies the analysis of how users set transaction fees [14], [16], [26]. We will relax Assumption 1 in Section VI by considering the impact of asynchronous transaction proposals and pending transactions in the transaction pool. Experiment results show that the main insights derived under Assumption 1 are still valid in the more realistic case.

Assumption 2: When a miner selects a total of $k = 1, 2, \cdots$ transactions, his selection is the top-k-fee-per-byte transactions in the transaction pool.

Assumption 2 realistically characterizes miners' behaviors in Bitcoin and Ethereum [24], [27]. Specifically, for the mining pools named "AntPool", "F2Pool", and "BTC.com", their acceptance rates of top-k-fee-per-byte transactions are about 90%, while the acceptance rates of out-of-top-$2k$-fee-per-byte transactions are less than 5%. Theoretically speaking, when a miner selects transactions to maximize his total fee reward, he needs to solve a knapsack problem, as each block can contain a limited size of transactions. An effective heuristic algorithm to solve the knapsack problem is the greedy algorithm as described in Assumption 2. Since the size of the transaction is smaller than 5% of the block size (which is practical in blockchain [28]), the greedy algorithm yields an optimality gap no higher than 5% for the miner, based on the theoretical analysis [29].

Next, we will present the system model. Table I summarizes the key notations of this paper.

B. Problem Setting

In this subsection, we introduce the decisions of the system designer, users, and miners.

1) System Designer's Decision: The system designer designs the Fee and Transaction Expiration Time (FTET) mechanism to achieve blockchain storage sustainability, i.e., for any transaction, its fee will be able to cover its storage costs of all miners.

The FTET mechanism specifies two parameters, motivated by the key reasons behind the storage sustainability issue.

TABLE I
KEY NOTATIONS

Variables	
$\underline{\rho}$	Minimum fee-per-byte
T	Transaction expiration time
z_i	User i's strategy
p_i	User i's transaction proposing probability
ρ_i	User i's fee-per-byte
$\mathcal{X}_m^t$	Miner m's transaction selection in round t
$\sigma_H(t)$	Type H miner's fee-per-byte threshold of admitting transactions in round t
Parameters	
$\mathcal{I}(I)$	Set (number) of users
$\mathcal{M}(M)$	Set (number) of miners
$\mathcal{M}_H(\mathcal{M}_L)$	Set of type H (L) miners
$\mathcal{T}$	Set of mining rounds
θ_i	User i's transaction
s_i	Size of transaction θ_i
$\bar{s}$	Expected size of transaction θ_i
f_i	Fee of transaction θ_i
d_i	Delay of transaction θ_i
$\mathbb{1}(i)$	Indicator function to indicate whether θ_i is in blockchain or not
R	User's utility of one transaction in blockchain
γ	User's impatience level
A	User's transaction expiration loss
K	Number of transactions per block
C_m	Storage costs per byte of miner m
$C_H(C_L)$	Storage costs per byte of type $H(L)$ miner
C^{tot}	Total storage costs per byte of all miners
α_m	Mining power of miner m
Δ	Average block generation time
$\mathcal{Q}^t$	Transaction pool at mining round t

- The *minimum fee-per-byte* $\underline{\rho} \geqslant 0$ for all users and all transactions: As low-storage-cost miners admit low-fee transactions, the FTET mechanism stipulates that no user can propose a fee-per-byte lower than $\underline{\rho}$.
- *Transaction expiration time* $T \in \mathbb{N}^+$ (a positive integer): As users' fee competition intensity is low, the FTET mechanism assigns an expiration time T for transactions to control the fee competition level. After the user proposes a transaction, miners can choose to append it to blockchain within the next T mining rounds. Otherwise, the transaction expires and disappears from the transaction pool. A lower T reduces the maximum delay for the transactions while increasing the fee competition intensity among the users, as there are fewer the transaction inclusion opportunities.

Note that today's protocols in Bitcoin and Ethereum is a special case of the FTET mechanism, with $(\underline{\rho}, T) = (0, \infty)$.

The system designer aims to set the parameters of the FTET mechanism to maximize social welfare, subject to the storage sustainability constraint. Note that social welfare maximization is a reasonable objective for the blockchain online community

as its goal is to select the outcome that the group most prefers [30], [31]. We will elaborate the social welfare function in Section IV-C and the storage sustainability constraint in Section IV-D.

2) Users' Decisions: We consider a set of $\mathcal{I} = \{1, \cdots, I\}$ users,[6] who decide whether to propose transactions considering the number of pending transactions.[7] When proposing a transaction, a user needs to tradeoff between paying a higher fee and bearing a higher delay.[8] Thus, each user $i \in \mathcal{I}$ decides the *transaction proposing probability* $p_i \in [0, 1]$ and *fee-per-byte* $\rho_i \geqslant \underline{\rho}$ for his transaction. Specifically, user i chooses to propose transaction θ_i with a probability p_i, and chooses not propose any transaction with probability $1 - p_i$. For notational convenience, we use $z_i = (p_i, \rho_i)$ to denote each user's strategy.

Each user will choose his strategy to maximize his payoff, which depends on whether his transaction is included in the blockchain, fee payment to the miner, and his transaction's delay. We will elaborate the user's payoff function in Section IV-C.

3) Miners' Decisions: We analyze a total T mining rounds, given all transactions expire afterward. The time duration of each mining round $t \in \mathcal{T} \triangleq \{1, \cdots, T\}$ follows an independent and identically distributed (i.i.d.) distribution with an average of $\overline{\Delta}$ (which depends on the block mining process). We first introduce the miners' decisions in mining round t, and then illustrate the miners' decisions in all T rounds by an example.

There is a set $\mathcal{M} = \{1, \cdots, M\}$ of miners in the blockchain system. At the beginning of mining round t, each miner $m \in \mathcal{M}$ decides to *select a transaction set* $\mathcal{X}_m^t$ to be included in his block, considering the tradeoff between transaction fees and storage costs. Specifically, we assume that each miner can choose up to K transactions from the transaction pool $\mathcal{Q}^t$ (i.e., the set of pending transactions at the beginning of mining round t) due to the block size limit. Thus miner m's strategy $\mathcal{X}_m^t$ satisfies:

$$\mathcal{X}_m^t \in \mathcal{B}_m^t \triangleq \{\mathcal{Y} | \mathcal{Y} \subseteq \mathcal{Q}^t \text{ and } |\mathcal{Y}| \leqslant K\}. \tag{1}$$

From Assumption 2, each miner decides the number of top-fee-per-byte transactions to select to maximize his payoff, which consists of transaction fees and storage costs. We will elaborate the miner's payoff function in Section IV-C.

In Fig. 2, we use an example of $T = 2$ and $M = 2$ to illustrate miners' decisions. At the beginning of mining round 1, the transaction pool $\mathcal{Q}^1$ contains three transactions.

- Mining round 1: Both miners 1 and 2 adopt strategy $\{\theta_2, \theta_3\}$. Miner 2 succeeds in generating the block and appends θ_2 and θ_3 to the blockchain. The transaction
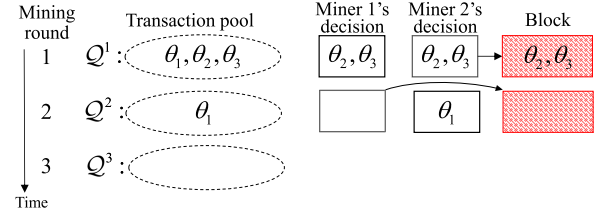

Fig. 2. An example of miners' decisions.

pool deletes the transactions in the blockchain, i.e., $\mathcal{Q}^2 = \mathcal{Q}^1 \setminus \{\theta_2, \theta_3\}$.
- Mining round 2: Miners 1 and 2 adopt strategy $\emptyset$ and $\{\theta_1\}$, respectively. Miner 1 generates the block and appends no transaction to the blockchain. Hence θ_1 expires after mining round 2 as $T = 2$, i.e., $\mathcal{Q}^3 = \emptyset$.

Next we will introduce the miner's payoff, user's payoff, and the social welfare.

C. Payoff Functions and Social Welfare

When miners generate blocks in T mining rounds, miners and users will receive the payoffs from the blocks, as defined below.

1) Miner's Payoff: We consider a generic mining round t. When a miner generates a block, he obtains the transaction fees associated with transactions included in the block. Meanwhile, all miners bear the storage cost of the block. We explain these two terms in details next.[9]

- *Transaction fee*: We denote the normalized mining power of miner m as α_m, satisfying $\sum_{m \in \mathcal{M}} \alpha_m = 1$. When miner m appends a block to blockchain (with a probability α_m), he obtains the transaction fees of all transactions in the block, which is $\sum_{i \in \mathcal{X}_m^t} f_i$. The transaction fee of any θ_i equals to the product of its size and fee-per-byte, i.e., $f_i = s_i \rho_i$.
- *Storage cost*: Miners bear *heterogeneous storage costs* according to their storage modes. Motivated by the practice of Ethereum,[10] we consider miners with two storage types: a set $\mathcal{M}_L$ of m_L low-storage-cost miners (denoted as type-L) and a set $\mathcal{M}_H$ of $m_H = M - m_L$ high-storage-cost miners (denoted as type-H). Specifically, we let C_m denote miner m's storage cost incurred by one byte of transaction:

$$C_m = \begin{cases} C_L, & \text{if } m \in \mathcal{M}_L, \\ C_H, & \text{if } m \in \mathcal{M}_H, \end{cases} \tag{2}$$

where $C_H \geqslant C_L$. The homogeneous cost case of $C_H = C_L$ in [10] is a special case of our model. If a miner m stores a transaction θ_i of size s_i, he will bear the cost of $s_i C_m$. Thus, when any miner $n \in \mathcal{M}$ appends a block to blockchain, miner m (which can be the same or different from n) suffers the cost of $\sum_{i \in \mathcal{X}_n^t} s_i C_m$, since all miners need to store the same block including the transactions

---

[6]We analyze the blockchain system in a quasi-static state [32], [33], [34], where the numbers of users and miners do not change during the time scale of interests.

[7]For example, Ethereum application cryptokitties show that users' transaction proposing rate drops if there are too many pending transactions [35].

[8]In practice, a user generally uses the fee recommendation software to set the transaction fee, where he enters his delay requirement, and the software recommends the corresponding transaction fee [24]. Such a decision process is how a user tradeoffs between transaction fee and delay.

[9]We treat the block reward and mining cost as the sunk revenue and sunk cost for each miner (and do not include them in our model), because they are independent of each miner's transaction selection. Moreover, the long-term blockchain storage sustainability mainly depends on transaction fees, as the block reward decreases over time.

[10]Ethereum offers two storage modes for a full node: default and archive. On Feb. 2021, a full node's default mode requires 650 GB of disk space, while an archival mode requires 6.3 TB [4].

that miner n selects [36].[11] As each miner n appends a block with a probability α_n, miner m's storage cost in mining round t is follows:

$$\psi_m^t(\mathcal{X}_m^t, \boldsymbol{\mathcal{X}}_{-m}^t) = \sum_{n \in \mathcal{M}} \alpha_n \sum_{i \in \mathcal{X}_n^t} s_i C_m, \qquad (3)$$

where $\boldsymbol{\mathcal{X}}_{-m}^t = (\mathcal{X}_n^t, \forall n \in \mathcal{M}, n \neq m)$ denotes other miners' strategies (except miner m) in mining round t.

Overall, miner m's *payoff function* in mining round t is as follows:

$$v_m^t(\mathcal{X}_m^t, \boldsymbol{\mathcal{X}}_{-m}^t) = \alpha_m \sum_{i \in \mathcal{X}_m^t} s_i \rho_i - \psi_m^t(\mathcal{X}_m^t, \boldsymbol{\mathcal{X}}_{-m}^t). \quad (4)$$

2) User's Payoff: We first define the delay faced by any transaction θ_i, then we present user i's surplus from θ_i, and finally we formulate user i's payoff function.

The delay of transaction θ_i is the time span from θ_i being proposed to being included in blockchain:

- *Proposing time*: We denote the start time of mining round t as $\text{time}(t)$. When user i proposes θ_i at the beginning of mining round 1, the proposing time of θ_i is $\tau_i^{\text{pro}} = \text{time}(1)$.
- *In-blockchain time*: When a miner generates a block $t \in \mathcal{T}$ containing θ_i, the mining round t ends and round $t + 1$ starts. Thus, we define θ_i's in-blockchain time as $\tau_i^{\text{in}} = \text{time}(t + 1)$.
- The *delay* is the difference between the in-blockchain time and the proposing time, i.e. $d_i = \tau_i^{\text{in}} - \tau_i^{\text{pro}}$. The delay d_i is a random variable, since time duration of each mining round is not deterministic (e.g., Ethereum). Moreover, it is a function of the strategies of all users and all miners in all T mining rounds, i.e., $d_i(\boldsymbol{z}, \boldsymbol{\mathcal{X}})$ with $\boldsymbol{z} = (z_i, \forall i \in \mathcal{I})$ and $\boldsymbol{\mathcal{X}} = (\boldsymbol{\mathcal{X}}^t = (\mathcal{X}_m^t, \forall m \in \mathcal{M}), \forall t \in \mathcal{T})$, because it depends on the number of transactions, transaction fee-per-byte values, and miners' transaction selection decisions. Lemma 1 in Section V-A presents the expression of expected delay $\mathbb{E}[d_i]$.

Then we introduce user i's surplus from his transaction θ_i, based on whether or not θ_i is eventually included in a block:

- If θ_i is included in a block: The surplus derived from θ_i includes the following three aspects:
  - Satisfaction level: User i will experience the satisfaction level $R > 0$ when transaction θ_i is included in the blockchain. For instance, a user can successfully purchase a crypto artwork [37] and experience some happiness once the corresponding transaction is approved and included in the blockchain.
  - Transaction fee: User i pays the transaction fee of $s_i \rho_i$ to a miner who includes the transaction in the block. We assume that the size of a transaction θ_i follows an i.i.d. distribution with an expected value of $\mathbb{E}[s_i] = \bar{s}$.
  - Delay cost: We assume that the user bears a cost of $\gamma > 0$ for each unit of delay. The factor γ reflects

the impatience level of the user, where a higher γ indicates a higher user impatience level. Thus, transaction θ_i incurs a delay cost of $\gamma d_i(\boldsymbol{z}, \boldsymbol{\mathcal{X}})$. Such a cost reveals the *fee competition* among users: as each user wants to bear a low delay cost, he needs to set a higher fee-per-byte than others to encourage the miners to include his transaction in the blockchain earlier.

To sum up, user i's surplus derived from in-blockchain transaction θ_i is

$$\delta_i(\boldsymbol{z}, \underline{\rho}, \boldsymbol{\mathcal{X}}) = R - s_i \rho_i - \gamma d_i(\boldsymbol{z}, \boldsymbol{\mathcal{X}}). \qquad (5)$$

The surplus is a function of minimum fee-per-byte $\underline{\rho}$ because user i's fee-per-byte satisfies $\rho_i \geqslant \underline{\rho}$.

- The transaction θ_i expires if it is not included in any block after T mining rounds from its proposing time. In this case, user i does not obtain the satisfaction level R or pay fee $s_i \rho_i$. However, user i still needs to wait until expiration and suffers a loss of

$$\delta_i(\boldsymbol{z}, \underline{\rho}, \boldsymbol{\mathcal{X}}) = -A. \qquad (6)$$

We assume that $A \geqslant \gamma T \overline{\Delta} > 0$, as the amount of penalty A includes both the delay cost and loss due to the transaction expiration.

The following indicate function will help simplify the expression:

$$\mathbb{1}(i) = \begin{cases} 1, & \text{if } \theta_i \text{ is included in blockchain,} \\ 0, & \text{if } \theta_i \text{ is not included in blockchain.} \end{cases} \qquad (7)$$

Hence user i's surplus derived from θ_i can be written as

$$\delta_i(\boldsymbol{z}, \underline{\rho}, \boldsymbol{\mathcal{X}}) = \mathbb{1}(i)[R - s_i \rho_i - \gamma d_i(\boldsymbol{z}, \boldsymbol{\mathcal{X}})] - [1 - \mathbb{1}(i)]A. \qquad (8)$$

Finally, we present the user's payoff function. When user i proposes transaction θ_i, we have defined his surplus from θ_i in (8). When user i proposes no transaction, we define his surplus as 0. Then user i's *payoff function* can be written as:

$$u_i(\boldsymbol{z}, \underline{\rho}, \boldsymbol{\mathcal{X}}) = p_i \mathbb{E}[\delta_i(\boldsymbol{z}, \underline{\rho}, \boldsymbol{\mathcal{X}})], \qquad (9)$$

where p_i is the transaction proposing probability and $\mathbb{E}[\delta_i(\boldsymbol{z}, \underline{\rho}, \boldsymbol{\mathcal{X}})]$ is user i's expected surplus from transaction θ_i. The expectation is taken in terms of random variables delay d_i and transaction size s_i.

3) Social Welfare: The *social welfare* is the summation of all entities' payoffs over T mining rounds, i.e.,

$$sw(\underline{\rho}, T, \boldsymbol{z}, \boldsymbol{\mathcal{X}}) = \sum_{i \in \mathcal{I}} u_i(\boldsymbol{z}, \underline{\rho}, \boldsymbol{\mathcal{X}}) + \sum_{t \in \mathcal{T}} \sum_{m \in \mathcal{M}} v_m^t(\boldsymbol{\mathcal{X}}^t). \quad (10)$$

D. Stackelberg Game Formulation

We model the strategic interactions among the system designer, users, and miners as a three-stage Stackelberg game as shown in Fig. 3.

---

[11]We neglect the storage cost of non-transaction data since it is very small. For example, on Feb 1st, 2020, the average block size of Bitcoin is 1.313 MB, while non-transaction data in a block is less than 1 KB [5].
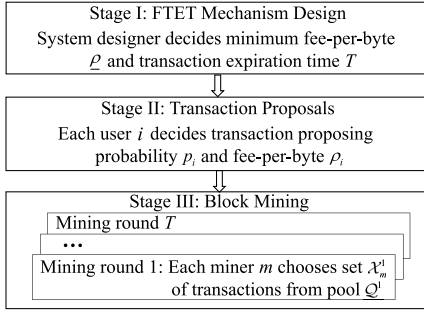
Fig. 3. Three-stage Stackelberg game.

1) Stage I: FTET Mechanism Design: In Stage I, the system designer designs the FTET mechanism, including setting the minimum fee-per-byte $\underline{\rho}$ and transaction expiration time T. The mechanism design should satisfy the *storage sustainability constraint* defined as follows:

Definition 1 (Storage Sustainability Constraint): A mechanism satisfies the storage sustainability constraint *if (11) holds for each user i who may propose a transaction (i.e., $p_i > 0$).*

$$\mathbb{E}[\rho_i] \geqslant C^{\text{tot}} \triangleq m_L C_L + m_H C_H. \tag{11}$$

In other words, the storage sustainability constraint means that each user's expected fee-per-byte is higher than all miners' total storage costs per byte.

Moreover, the system designer's goal is to maximize the social welfare, such that we can formulate the *FTET mechanism design problem* as follows:

$$\max \quad sw(\underline{\rho}, T, \boldsymbol{z}, \boldsymbol{\mathcal{X}})$$
$$\text{s.t.} \quad (11)$$
$$\text{var.} \quad \underline{\rho} \geqslant 0, \quad T \in \mathbb{N}^+. \tag{12}$$

2) Stage II: Transaction Proposals: In Stage II, users propose their transactions in a game theoretical fashion as follows:

Game 1 (Users' Transactions Proposing Game in Stage II): The users' transaction proposing game is a tuple $\Omega = (\mathcal{I}, \mathcal{Z}, \boldsymbol{U})$ defined by:

- *Players: The set $\mathcal{I}$ of users.*
- *Strategies: Each user i chooses an action $z_i = (p_i, \rho_i) \in \mathcal{Z}_i = [0, 1] \times [\underline{\rho}, \infty)$, i.e., transaction proposing probability and fee-per-byte. The strategy profile of all users is $\boldsymbol{z} = (z_i, \forall i \in \mathcal{I})$ and the set of feasible strategy profile of all users is $\mathcal{Z} = \prod_{i \in \mathcal{I}} \mathcal{Z}_i$.*
- *Payoffs: The vector $\boldsymbol{U} = (u_i, \forall i \in \mathcal{I})$ contains all users' payoffs as defined in (9).*

3) Stage III: Block Mining: In each mining round $t \in \mathcal{T}$ of Stage III, the miners interact with each other in a game theoretical fashion as follows:

Game 2 (Miners' Transaction Selection Game in Mining Round t in Stage III): The miners' transaction selection game in mining round t is a tuple $\Gamma^t = (\mathcal{M}, \mathcal{B}^t, \boldsymbol{V}^t)$ defined by:

- *Players: The set $\mathcal{M}$ of miners.*
- *Strategies: Each miner m chooses a transaction set $\mathcal{X}_m^t \in \mathcal{B}_m^t$, where $\mathcal{B}_m^t$ is defined in (1). The strategy profile of all miners is $\boldsymbol{\mathcal{X}}^t = (\mathcal{X}_m^t, \forall m \in \mathcal{M})$ and the set of feasible strategy profile of all miners is $\mathcal{B}^t = \prod_{m \in \mathcal{M}} \mathcal{B}_m^t$.*

- *Payoffs: The vector $\boldsymbol{V}^t = (v_m^t, \forall m \in M)$ contains all miners' payoffs as defined in (4).*

Next, we use backward induction to solve the three-stage game.

V. ANALYSIS OF THE THREE-STAGE MODEL

In this section, we derive the equilibrium of the three-stage game via backward induction. We derive the miners' equilibrium of Stage III in Section V-A, the users' equilibrium of Stage II in Section V-B, and the solution of FTET mechanism design problem of Stage I in Section V-C, respectively.

A. Stage III: Miners' Transaction Selection Game

We first solve the miners' equilibrium strategies in Game 2 in each mining round t, given the FTET mechanism designed in Stage I and the users' strategies in Stage II. Then we analyze the transaction's delay based on miners' equilibrium.

1) Miners' Equilibrium in Stage III: The Stage III contains T mining rounds. As miners' decisions in an early mining round will impact their payoffs in later mining rounds,[12] we will derive the subgame perfect Nash equilibrium (SPNE) in Stage III. We first define the Nash equilibrium for the game in each mining round t, and then we define the subgame and SPNE.

Definition 2 (Nash Equilibrium): A strategy profile $(\mathcal{X}_m^{t,\text{NE}}, \forall m \in \mathcal{M})$ constitutes a Nash equilibrium *of Game 2 in mining round $t \in \mathcal{T}$ if*

$$v_m^t(\mathcal{X}_m^{t,\text{NE}}, \boldsymbol{\mathcal{X}}_{-m}^{t,\text{NE}}) \geqslant v_m^t(\mathcal{X}_m^t, \boldsymbol{\mathcal{X}}_{-m}^{t,\text{NE}}),$$
$$\forall \mathcal{X}_m^t \in \mathcal{B}_m^t, \quad \forall m \in \mathcal{M}. \tag{13}$$

At a Nash equilibrium, no miner can increase his payoff by unilaterally changing his strategy.

Definition 3 (Subgame): A subgame is a subset of all games in Stage III that includes an initial game and all its successor games.

In other words, the game from any mining round $t \in \mathcal{T}$ to the last mining round T constitutes a subgame.

Definition 4 (Subgame Perfect Nash Equilibrium): A strategy profile in Stage III constitutes a subgame perfect Nash equilibrium *if and only if it is a Nash equilibrium in every subgame.*

At SPNE (to be shown in Theorem 1), a miner selects all the top-fee-per-byte transactions (up to K transactions due to the block size) with fee-per-byte no lower than a threshold. When the threshold is ρ_j, the corresponding transaction selection strategy in mining round t is as follows:

$$\mathcal{Q}^t(\rho_j) \triangleq \arg \max_{\substack{\mathcal{Q}' \subseteq \{i \in \mathcal{Q}^t | \rho_i \geqslant \rho_j\} \\ |\mathcal{Q}'| \leqslant K}} \sum_{i \in \mathcal{Q}'} \rho_i. \tag{14}$$

Moreover, we define two functions to characterize miners' thresholds in mining round t as follows:

$$\sigma_H(t) \triangleq \max\{(\sum_{m \in \mathcal{M}_H} \alpha_m)^{T-t} C_H, C_L, \underline{\rho}\}. \tag{15}$$
$$\sigma_L(t) \triangleq \max\{C_L, \underline{\rho}\}. \tag{16}$$

[12] When a miner includes a transaction in blockchain, all miners cannot include the transaction and get the transaction fee afterwards.

Then we summarize the SPNE of Stage III in Theorem 1.

Theorem 1 (Miners' SPNE in Stage III): The strategy profile $(\mathcal{X}_m^{t,\text{SPNE}}, \forall m \in \mathcal{M}, \forall t \in \mathcal{T})$ *constitutes a SPNE in Stage III, where*

$$\mathcal{X}_m^{t,\text{SPNE}} = \begin{cases} Q^t(\sigma_H(t)), & \text{if } m \in \mathcal{M}_H, \\ Q^t(\sigma_L(t)), & \text{if } m \in \mathcal{M}_L. \end{cases} \tag{17}$$

We leave all the proof of lemmas, propositions, and theorems in the online appendix [38] due to the space limit.

Here we elaborate the intuition behind Theorem 1. In mining round t, type-H and type-L miners' *fee-per-byte thresholds* for admitting any transaction are $\sigma_H(t)$ and $\sigma_L(t)$, respectively. That is to say, if $\rho_i < \sigma_H(t)$ ($\rho_i < \sigma_L(t)$, respectively), none of the type-H (type-L, respectively) miners will select transaction θ_i.

Moreover, we summarize an interesting insight of type-H miner's fee-per-byte threshold as follows.

Corollary 1: As T increases, the type-H miners' fee-per-byte threshold $\sigma_H(t)$ decreases.

To understand Corollary 1, consider a transaction θ_i that type-L miners will admit but type-H miners will not. As the expiration time T increases, if type-H miners still do not plan to admit the transaction, type-L miners will be more likely to append θ_i to the blockchain, causing type-H miners to receive no transaction fee but eventually bear the storage cost. Then type-H miners would rather accept θ_i, bearing the storage cost but getting some fee.

Corollary 1 demonstrates one of the key reasons causing storage sustainability issue, i.e., low-storage-cost miners cause the blockchain to admit low-fee transactions. In today's blockchain protocol (i.e., $(\rho, T) = (0, \infty)$), a type-H miner's fee-per-byte threshold for accepting a transaction is the same as a type-L miner's. Thus, a type-H miner will admit all transactions (including low-fee ones) that a type-L miner admits.

2) Transaction's Delay: Based on miners' equilibrium strategies, we will derive the transaction θ_i's expected delay $\mathbb{E}[d_i]$. The delay of transaction θ_i depends on whether the transaction expires and the numbers of transactions with a higher or equal fee-per-byte. To facilitate the discussions, we define the indicator function comparing value against T as

$$\mathbb{I}_T(t) \triangleq \begin{cases} 1, & \text{if } t \leqslant T, \\ 0, & \text{if } t > T. \end{cases} \tag{18}$$

The sets of other users (excluding i) who assign fee-per-byte strictly higher than ρ_i and equaling ρ_i are $\mathcal{H}_i(\rho_i)$ and $\mathcal{E}_i(\rho_i)$, respectively:

$$\mathcal{H}_i(\rho_i) \triangleq \{j \in \mathcal{I} \setminus \{i\} | \rho_j > \rho_i\}, \tag{19}$$

$$\mathcal{E}_i(\rho_i) \triangleq \{j \in \mathcal{I} \setminus \{i\} | \rho_j = \rho_i\}. \tag{20}$$

Lemma 1 characterizes the expected delay of transaction θ_i.

Lemma 1: If transaction θ_i's fee-per-byte satisfies $\rho_i \geqslant \sigma_H(\min\{T, \lceil \frac{I}{K} \rceil\})$,[13] then the transaction will experience an expected delay of $\mathbb{E}[d_i]$ in (21), as shown at the bottom of the next page.[14]

Lemma 1 does not include $\rho_i < \sigma_H(\min\{T, \lceil \frac{I}{K} \rceil\})$ because it does not happen at the equilibrium of Stage II (as we will show in Theorem 2).

Here we explain the intuition of Lemma 1 by considering transaction θ_i in mining round t. When $\rho_i \geqslant \sigma_H(\min\{T, \lceil \frac{I}{K} \rceil\})$, transaction θ_i's fee-per-byte is above any miner's fee-per-byte threshold in any mining round $t \in \mathcal{T}$.

- If transaction θ_i is among the top-K-fee-per-byte transactions, then all the miners will admit θ_i in mining round t. Hence its expected delay is $t\overline{\Delta}$.
- If transaction θ_i is not the among top-K-fee-per-byte transactions, then none of the miners will admit θ_i in mining round t. Hence we need to check θ_i in mining round $t + 1$ (or θ_i will expire if $t = T$).

B. Stage II: Users' Transaction Proposing Game

In this subsection, we solve the users' equilibrium strategies in Game 1, given the FTET mechanism designed in Stage I while anticipating the SPNE in Stage III.

We characterize the non-existence of pure strategy Nash equilibrium (i.e., Definition 2) in Proposition 1.

Proposition 1 (Non-existence of Pure Strategy Nash Equilibrium in Stage II): When $\sigma_H(\min\{T, \lceil \frac{I}{K} \rceil\}) < \frac{R - \gamma \overline{\Delta}}{\overline{s}}$, there does not exist a pure strategy Nash equilibrium in Stage II.

Proposition 1 shows that when the miner's fee-per-byte threshold is low (i.e., $\sigma_H(\min\{T, \lceil \frac{I}{K} \rceil\}) < \frac{R - \gamma \overline{\Delta}}{\overline{s}}$), there does not exist a pure strategy equilibrium. This is due to the payoff function's discontinuity at fee-per-byte equal to other users'. We illustrate this point by a simple 2-user-1-transaction-per-block example (i.e., $I = 2$ and $K = 1$). When user 1's fee-per-byte is not high, user 2 can set a fee-per-byte slightly higher than user 2's to reduce the delay, since miners choose the top-fee-per-byte transaction first. User 1 will follow the same logic, and the fee-per-byte may end up to be high. However, if any user sets a high fee-per-byte, the other user can set a low fee-per-byte (i.e., the fee-per-byte threshold that type-H miners accept transactions) to reduce the fee payment. As a result, there is no pure strategy equilibrium where users achieve mutual best responses.

As the pure strategy Nash equilibrium does not exist if $\sigma_H(\min\{T, \lceil \frac{I}{K} \rceil\}) < \frac{R - \gamma \overline{\Delta}}{\overline{s}}$, next we will focus on characterizing the *mixed strategy Nash equilibrium* in Stage II. To facilitate the analysis, we use $\mu_{i,p}$ ($\mu_{i,\rho}$, respectively) to denote a probability measure[15] over user i's strategy space of transaction proposing probability $[0, 1]$ (fee-per-byte $[\underline{\rho_i}, \infty)$, respectively). Thus, we can denote user i's mixed strategy as $\boldsymbol{\mu}_i = (\mu_{i,p}, \mu_{i,\rho})$, i.e., a two-dimension probability measure.

Definition 5 (Mixed Strategy Nash Equilibrium): A vector of probability measure $(\boldsymbol{\mu}_i^{\text{NE}}, \forall i \in \mathcal{I})$ constitutes a mixed strategy Nash equilibrium *if (22), as shown at the bottom of the next page, holds for each $i \in \mathcal{I}$ and each $\boldsymbol{\mu}_i$.*

Furthermore, we consider the symmetric mixed strategy Nash equilibrium (SMNE), where all users use the same mixed strategy. The symmetric equilibrium has been widely used in

---

[13]Notice that $\lceil \cdot \rceil$ is the ceiling function.
[14]We denote the set of all subsets of any set $\mathcal{L}$ as $2^{\mathcal{L}} = \{\mathcal{J} | \mathcal{J} \subseteq \mathcal{L}\}$.

[15]A probability measure is a real-valued function that assigns a probability to each event in a probability space.

cases that players of the game have i.i.d. random parameters (i.e., the transaction size in our model) [32], [39]. For the ease of exposition, we define some terminology related to the SMNE.

Definition 6 (Stage II SMNE Types): The probability measure vector $(\boldsymbol{\mu}_i^{\mathrm{NE}} = (\mu_{i,p}^{\mathrm{NE}}, \mu_{i,\rho}^{\mathrm{NE}}), \forall i \in \mathcal{I})$ denotes a SMNE, which has the following several types:

(i) *An All-SMNE is that each user proposes a transaction, i.e., $\mu_{i,p}^{\mathrm{NE}}(\{p_i = 1\}) = 1$.*

(ii) *A Partial-SMNE is that each user proposes a transaction with a probability $\pi_1^* \in (0,1)$, i.e., $\mu_{i,p}^{\mathrm{NE}} (\{p_i = \pi_1^*\}) = 1$.*

(iii) *A None-SMNE is that no users propose any transaction, i.e., $\mu_{i,p}^{\mathrm{NE}}(\{p_i = 0\}) = 1$.*

Theorem 2 summarizes the conditions under which different types of SMNEs emerge in Stage II. Here we define the intermediate function $g(p_i, \rho_i, T)$ in (23), as shown at the bottom of this page, to characterize the equilibrium strategy of users. "CDF" is the abbreviation for "cumulative distribution function".

Theorem 2 (Users' SMNE in Stage II):

1) *When $T \geqslant \lceil \frac{I}{K} \rceil$, Table. II summarizes the SMNE of Stage II.*

2) *When $T < \lceil \frac{I}{K} \rceil$, Table. III summarizes the SMNE of Stage II.*

As the miner's fee-per-byte threshold $(\sigma_H(\lceil \frac{I}{K} \rceil)$ or $\sigma_H(T))$ increases, the user's transaction proposing probability decreases and eventually equals zero (i.e., None-SMNE), because the higher fee-per-byte payment reduces each user's incentive to propose a transaction. Moreover, we notice that the All-SMNE does not exist when $KT < I$. This is because when $KT < I$, if each user proposes one transaction for all I users, miners at most append KT transactions to the blockchain and some users must suffer an expiration loss $-A$, which is worse than not proposing any transaction and getting a zero payoff. Hence users have an incentive to lower the transaction proposing probability. Notice that the user's fee-per-byte strategy depends on the system parameters like the number of transactions per block K and block generation time $\overline{\Delta}$, as function g in (23) depends on these parameters. We illustrate these factors' impact on the user's fee-per-byte strategy in online appendix [38] due to space limit.

C. Stage I: System Designer's FTET Mechanism Design

In this subsection, we first derive the optimal FTET mechanism and then analyze the corresponding social welfare. Finally, we discuss the impact of the transaction expiration time.

1) The Optimal FTET Mechanism: By solving Problem (12), we derive the optimal FTET mechanism in Theorem 3, where $\lfloor \cdot \rfloor$ denotes the floor function.

Theorem 3 (The Optimal FTET Mechanism): The optimal FTET mechanism (characterized by the optimal solution of Problem (12)) satisfies

$$(\underline{\rho}^*, T^*) = \left(C^{\mathrm{tot}}, \lfloor \frac{R - \bar{s}C^{\mathrm{tot}} + A}{\gamma \overline{\Delta}} \rfloor \right). \quad (27)$$

Theorem 3 shows that under the optimal FTET mechanism, the minimum fee-per-byte is $\underline{\rho}^* = C^{\mathrm{tot}}$, ensuring the storage sustainability. The transaction expiration time is a finite value (i.e., $T^* = \lfloor \frac{R - \bar{s}C^{\mathrm{tot}} + A}{\gamma \overline{\Delta}} \rfloor$). Before mining round T^*, a transaction's delay cost is small compared with the expiration loss, such that it is better off to include it in blockchain. After mining round T^*, the delay cost is large and the transaction expires.

2) Social Welfare of the Optimal FTET Mechanism: We will compare the social welfare of the optimal FTET mechanism with the *unconstrained social optimum*, which is the maximum social welfare of the FTET mechanism without consideration of storage sustainability constraint (11).

Formally, the unconstrained social optimum sw^{opt} is the optimal value of Problem (28).

$$sw^{\mathrm{opt}} \triangleq \max sw(\rho, T, \boldsymbol{z}, \boldsymbol{\mathcal{X}}) \quad \text{var. } \rho \geqslant 0, T \in \mathbb{N}^+. \quad (28)$$

$$\mathbb{E}[d_i] = \frac{\sum_{\mathcal{J}_1 \in 2^{\mathcal{H}_i(\rho_i)}} \sum_{\mathcal{J}_2 \in 2^{\mathcal{E}_i(\rho_i)}} \prod_{j_1 \in \mathcal{J}_1 \cup \mathcal{J}_2} \prod_{j_2 \in (\mathcal{H}_i(\rho_i) \setminus \mathcal{J}_1) \cup (\mathcal{E}_i(\rho_i) \setminus \mathcal{J}_2)} \sum_{j_3=0}^{|\mathcal{J}_2|} \frac{p_{j_1}(1-p_{j_2})\overline{\Delta}}{|\mathcal{J}_2|+1} \lceil \frac{|\mathcal{J}_1|+j_3+1}{K} \rceil \mathbb{I}_T(\lceil \frac{|\mathcal{J}_1|+j_3+1}{K} \rceil)}{\sum_{\mathcal{J}_1 \in 2^{\mathcal{H}_i(\rho_i)}} \sum_{\mathcal{J}_2 \in 2^{\mathcal{E}_i(\rho_i)}} \prod_{j_1 \in \mathcal{J}_1 \cup \mathcal{J}_2} \prod_{j_2 \in (\mathcal{H}_i(\rho_i) \setminus \mathcal{J}_1) \cup (\mathcal{E}_i(\rho_i) \setminus \mathcal{J}_2)} \sum_{j_3=0}^{|\mathcal{J}_2|} \frac{p_{j_1}(1-p_{j_2})}{|\mathcal{J}_2|+1} \mathbb{I}_T(\lceil \frac{|\mathcal{J}_1|+j_3+1}{K} \rceil)}. \quad (21)$$

$$\int_{\mathcal{Z}} \mathbb{E}[u_i(z_i, \boldsymbol{z}_{-i}, \underline{\rho}, \boldsymbol{\mathcal{X}})] d(\boldsymbol{\mu}_i^{\mathrm{NE}}(z_i) \times \boldsymbol{\mu}_{-i}^{\mathrm{NE}}(\boldsymbol{z}_{-i})) \geqslant \int_{\mathcal{Z}} \mathbb{E}[u_i(z_i, \boldsymbol{z}_{-i}, \underline{\rho}, \boldsymbol{\mathcal{X}})] d(\boldsymbol{\mu}_i(z_i) \times \boldsymbol{\mu}_{-i}^{\mathrm{NE}}(\boldsymbol{z}_{-i})). \quad (22)$$

$$g(p_i, \rho_i, T) = \begin{cases} \sum_{n=0}^{I-1} \binom{I-1}{n} p_i^n (1-p_i)^{I-1-n} (R - \bar{s}\rho_i - \lceil \frac{n+1}{K} \rceil \gamma \overline{\Delta}), & \text{if } T \geqslant \lceil \frac{I}{K} \rceil, \\ \sum_{n=0}^{KT-1} \binom{I-1}{n} p_i^n (1-p_i)^{I-1-n} (R - \bar{s}\rho_i - \lceil \frac{n+1}{K} \rceil \gamma \overline{\Delta}) - \sum_{n=KT}^{I-1} \binom{I-1}{n} p_i^n (1-p_i)^{I-1-n} A, & \text{if } T < \lceil \frac{I}{K} \rceil. \end{cases} \quad (23)$$

TABLE II

THE SMNE OF STAGE II WHEN $T \geqslant \lceil \frac{I}{K} \rceil$

SMNE condition	SMNE type	$\mu_{i,p}^{\mathrm{NE}}$	$\mu_{i,\rho}^{\mathrm{NE}}$
$\sigma_H(\lceil \frac{I}{K} \rceil) \leqslant \frac{R - \lceil \frac{I}{K} \rceil \gamma \overline{\Delta}}{\bar{s}}$	All-SMNE	$\mu_{i,p}^{\mathrm{NE}}(\{p_i = 1\}) = 1$	$\mu_{i,\rho}^{\mathrm{NE}}$'s CDF $G_1^{\mathrm{NE}}(\rho_i)$ is in (24)
$\sigma_H(\lceil \frac{I}{K} \rceil) \in (\frac{R - \lceil \frac{I}{K} \rceil \gamma \overline{\Delta}}{\bar{s}}, \frac{R - \gamma \overline{\Delta}}{\bar{s}})$	Partial-SMNE	$\mu_{i,p}^{\mathrm{NE}}(\{p_i = \pi_2^*\}) = 1$ and $\pi_2^* \in (0,1)$ satisfies $g(\pi_2^*, \sigma_H(\lceil \frac{I}{K} \rceil), T) = 0$	$\mu_{i,\rho}^{\mathrm{NE}}$'s CDF $G_2^{\mathrm{NE}}(\rho_i)$ is in (25)
$\sigma_H(\lceil \frac{I}{K} \rceil) \geqslant \frac{R - \gamma \overline{\Delta}}{\bar{s}}$	None-SMNE	$\mu_{i,p}^{\mathrm{NE}}(\{p_i = 0\}) = 1$	$\mu_{i,\rho}^{\mathrm{NE}}(\{\rho_i = \sigma_H(\lceil \frac{I}{K} \rceil)\}) = 1$

TABLE III

THE SMNE OF STAGE II WHEN $T < \lceil \frac{I}{K} \rceil$

SMNE condition	SMNE type	$\mu_{i,p}^{\mathrm{NE}}$	$\mu_{i,\rho}^{\mathrm{NE}}$
$\sigma_H(T) < \frac{R - \gamma \overline{\Delta}}{\bar{s}}$	Partial-SMNE	$\mu_{i,p}^{\mathrm{NE}}(\{p_i = \pi_3^*\}) = 1$ and $\pi_3^* \in (0,1)$ satisfies $g(\pi_3^*, \sigma_H(T), T) = 0$	$\mu_{i,\rho}^{\mathrm{NE}}$'s CDF $G_3^{\mathrm{NE}}(\rho_i)$ is in (26)
$\sigma_H(T) \geqslant \frac{R - \gamma \overline{\Delta}}{\bar{s}}$	None-SMNE	$\mu_{i,p}^{\mathrm{NE}}(\{p_i = 0\}) = 1$	$\mu_{i,\rho}^{\mathrm{NE}}(\{\rho_i = \sigma_H(T)\}) = 1$

Although Problem (12) has more constraints than Problem (28), we can show that the optimal FTET mechanism actually achieves the unconstrained social optimum in Proposition 2.

Proposition 2 (Achieving Unconstrained Social Optimum): The optimal solutions of Problems (12) and (28) are the same. The social welfare of the optimal FTET mechanism equals the unconstrained social optimum.

At the unconstrained social optimum, the corresponding minimum fee-per-byte is $\rho^{\mathrm{opt}} = C^{\mathrm{tot}}$, i.e., the total storage cost per byte of all miners. This value lets each user internalize his storage cost imposed on the system and encourages the user to propose transaction in a social-optimum way. Under such a minimum fee-per-byte, each user's equilibrium strategy is to pay a fee-per-byte no lower than C^{tot}, guaranteeing storage sustainability without the constraint.

3) Impact of Transaction Expiration Time: We summarize the impact of transaction expiration time T^* on the blockchain system in the following Proposition 3.

Proposition 3: Under the optimal FTET mechanism, if $K \lfloor \frac{R - \bar{s} C^{\mathrm{tot}} + A}{\gamma \overline{\Delta}} \rfloor < I$ and $R > \bar{s} C^{\mathrm{tot}} + \gamma \overline{\Delta}$, then the probability for a transaction θ_i to expire is non-zero (i.e., $\mathbb{P}(\mathbb{1}(i) = 0) > 0$).

Propositions 2 and 3 imply that the optimal mechanism may let transactions expire to achieve the unconstrained social optimum, i.e., higher social welfare than today's protocol without transaction expiration (i.e., $(\rho, T) = (0, \infty)$). The rationale behind this surprising result is as follows. Although the transaction expiration time T^* intensifies the fee competition and causes the expiration loss to users, it also sets an upper bound on the transaction's delay and reduces delay costs. Thus, the optimal FTET mechanism will let some transactions expire when the delay cost is high.

VI. EXPERIMENTAL RESULTS

The analysis so far has assumed that users propose their transactions synchronously at the beginning of mining round 1. In practice, however, users often propose their transactions asynchronously and face pending transactions proposed earlier. To validate the performance of the proposed mechanism in a realistic environment, we implement a complete Bitcoin-based blockchain testbed. We first compare our proposed mechanism with the state-of-art mining round time adjustment (MRTA) mechanism proposed in [10] under the asynchronous transaction proposing in Sections VI-A to VI-C. Then we analyze how pending transactions affect the user's strategy in Section VI-D.

A. Experiment Setup

We implement a Bitcoin-based blockchain system on a cluster of three cloud instances in [40]. Each instance contains

$$
G_1^{\mathrm{NE}}(\rho_i) = \begin{cases} 0, & \text{if } \rho_i < \sigma_H(\lceil \frac{I}{K} \rceil), \\ \text{the solution of equation } g(1 - G_1^{\mathrm{NE}}(\rho_i), \rho_i, T) = g(1, \sigma_H(\lceil \frac{I}{K} \rceil), T), & \text{if } \sigma_H(\lceil \frac{I}{K} \rceil) \leqslant \rho_i \text{ and } \\ & \rho_i \leqslant \sigma_H(\lceil \frac{I}{K} \rceil) + \frac{\gamma \overline{\Delta}(\lceil \frac{I}{K} \rceil - 1)}{\bar{s}}, \\ 1, & \text{if } \rho_i > \sigma_H(\lceil \frac{I}{K} \rceil) + \frac{\gamma \overline{\Delta}(\lceil \frac{I}{K} \rceil - 1)}{\bar{s}}. \end{cases} \quad (24)
$$

$$
G_2^{\mathrm{NE}}(\rho_i) = \begin{cases} 0, & \text{if } \rho_i < \sigma_H(\lceil \frac{I}{K} \rceil), \\ \text{the solution of equation } g(\pi_2^*[1 - G_2^{\mathrm{NE}}(\rho_i)], \rho_i, T) = 0, & \text{if } \sigma_H(\lceil \frac{I}{K} \rceil) \leqslant \rho_i \leqslant \frac{R - \gamma \overline{\Delta}}{\bar{s}}, \\ 1, & \text{if } \rho_i > \frac{R - \gamma \overline{\Delta}}{\bar{s}}. \end{cases} \quad (25)
$$

$$
G_3^{\mathrm{NE}}(\rho_i) = \begin{cases} 0, & \text{if } \rho_i < \sigma_H(T), \\ \text{the solution of equation } g(\pi_3^*[1 - G_3^{\mathrm{NE}}(\rho_i)], \rho_i, T) = 0, & \text{if } \sigma_H(T) \leqslant \rho_i \leqslant \frac{R - \gamma \overline{\Delta}}{\bar{s}}, \\ 1, & \text{if } \rho_i > \frac{R - \gamma \overline{\Delta}}{\bar{s}}. \end{cases} \quad (26)
$$

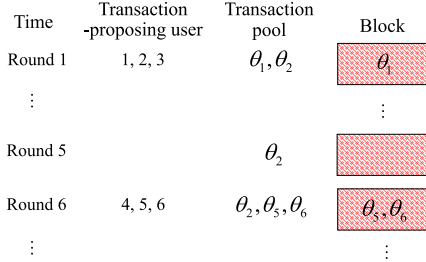|	
Average transaction size [41]	$\overline{s} = 150$ bytes
Maximum number of transactions per block [42]	$K = 200$
Average time of one mining round [43]	$\overline{\Delta} = 13$ seconds
Number of miners [7]	$M = 5 \times 10^4$
Ratio of two types of miners	$m_H : m_L = 1 : 1$
Heterogeneity of miners' storage costs (archival node:full node) [4]	$C_H : C_L = 10 : 1$
Storage cost per byte (set based on price of Amazon's SSD disk [3])	$C_L = 10^{-10}$ USD/byte

Fig. 4. An example of asynchronous transaction proposing.

an AMD EPY$^{\text{TM}}$ Rome CPU, 16GB of RAM, and a 5-Mbps communication link. We simulate a docker swarm network with a predefined overlay network and give a unique IPv4 address for each miner.

We set the blockchain parameters based on Ethereum as summarized in Table IV. The corresponding total storage cost per byte is 2.75×10^{-5}/byte and storage cost per transaction is 4.13×10^{-3}/transaction, serving as the benchmark in the experiment.

Besides our proposed optimal FTET mechanism, we will also implement the benchmnark MRTA mechanism. The MRTA mechanism works as follows [10]. If the default Ethereum protocol (i.e., $(\underline{\rho}, T) = (0, \infty)$) cannot satisfy the storage sustainability constraint, the MRTA mechanism will increase the mining round time $\overline{\Delta}$ until satisfying the constraint. When the transaction's delay increases, users' competition for short delays drives up the transaction fee.

We evaluate the optimal FTET mechanism under the *synchronous setting* (where all users propose transactions at the beginning of mining round 1) and compare two mechanisms under the *asynchronous setting* as follows:

Under the asynchronous setting, the strategic interactions among users are more complicated. We divide the users into two batches of equal sizes. We present an illustrative example of how users propose transactions in Fig. 4 with $N = 6$.

- The first batch of users 1, 2, and 3 determine their transaction proposing probabilities and fee-per-byte in mining round 1. Specifically, users 1 and 2 propose θ_1 and θ_2, respectively. User 3 proposes no transaction. Miners only append θ_1 in blockchain from mining rounds 1 to 5 and θ_2 remains in the transaction pool in mining round 5.
- The second batch of users 4, 5, and 6 make their decisions in mining round 6 (i.e., roughly 1 minute later). Specifically, user 4 proposes no transaction. Users 5 and 6 propose θ_5 and θ_6, respectively.

The second batch of users can observe the transaction history before their proposals and make their decisions accordingly. The first batch of users will take the second batch of users' future strategic behaviors into consideration when making their decisions. It is very challenging to fully characterize the equilibrium of such a dynamic game in closed-form.

To numerically solve the new asynchronous model, we discretize each user's two-dimensional strategy space into a 50×50 mesh, such that each user chooses the strategy from a point in the mesh. We still consider the symmetric Nash equilibrium where the first batch of users make the same decision and so do the second batch. Then we derive the subgame perfect Nash equilibrium by the Zermelo's algorithm [44].

We run each experiment 500 times with randomized miners' mining power and show the average result.

B. Expected Fee-Per-Byte

To compare two mechanisms in terms of storage sustainability, we study the impact of the number of users I, the user's impatience level γ, and user's satisfaction level R on the expected fee-per-byte.

When studying the impact of the number of users I, we set $I \in [500, 5000]$, because the blockchain system can include about 500-5000 transactions (in multiple blocks) during the general range of the transaction's delay in Ethereum (i.e., 30 seconds to 5 minutes) [45]. When studying the impact of users' parameters γ and R, we choose to fix $I = 2000$. We will further specify other system parameters in the figure captions.

Fig. 5 illustrates the impact of various parameters on the expected fee-per-byte $\mathbb{E}[\rho_i^{\text{NE}}]$. All three subfigures show the same curves with the same legends. We will explain Fig. 5(a) in details. To facilitate the explanation, we further plot the corresponding average number of transactions in Fig. 6, as the number of transactions affects the fee-per-byte (to be explained).

1) Fig. 5(a): Impact of the Number of Users I on $\mathbb{E}[\rho_i^{\text{NE}}]$: The benchmark storage cost curve is the total storage cost per byte (i.e., C^{tot}). When a mechanism's fee-per-byte curve is higher than the storage cost, the mechanism meets the storage sustainability constraint. The FTET-syn and FTET-asyn curves correspond to the expected fee-per-byte values of the optimal FTET mechanism under synchronous and asynchronous transaction proposing, respectively. The MRTA-asyn curves show the expected fee-per-byte of MRTA protocol under the asynchronous setting.

We first focus on the optimal FTET mechanisms under the synchronous transaction proposing. Its expected fee-per-byte (FTET-syn) is higher than C^{tot}, ensuring the storage sustainability. This result also holds for both Figs. 5(b) and (c). This observation validates the result in Theorem 3. Moreover, the expected fee-per-byte first increases in the number of users I and then roughly remains constant when I is large enough. Initially, the number of proposed transactions increases with I (as illustrated in Fig. 6(a)) and users' competition for experiencing a short delay drives up the transaction fee. However, when I is large, the fee competition is very intense. Further
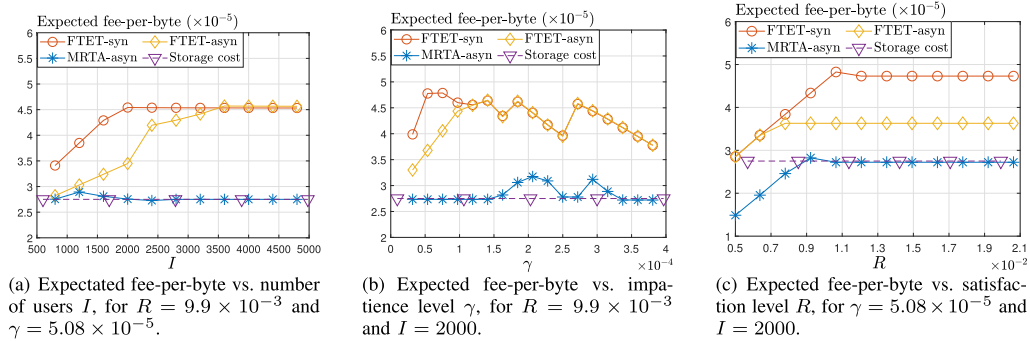
Fig. 5. Impact of number of users I, impatience level γ, and satisfaction level R on expected fee-per-byte $\mathbb{E}[\rho_i^{\mathrm{NE}}]$.
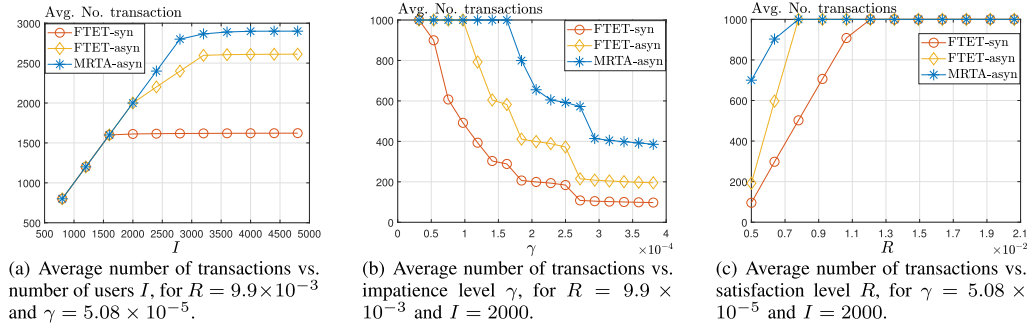
(a) Expectated fee-per-byte vs. number of users I, for $R = 9.9 \times 10^{-3}$ and $\gamma = 5.08 \times 10^{-5}$.

(b) Expected fee-per-byte vs. impatience level γ, for $R = 9.9 \times 10^{-3}$ and $I = 2000$.

(c) Expected fee-per-byte vs. satisfaction level R, for $\gamma = 5.08 \times 10^{-5}$ and $I = 2000$.

Fig. 6. Impact of number of users I, impatience level γ, and satisfaction level R on average number of transactions.

(a) Average number of transactions vs. number of users I, for $R = 9.9 \times 10^{-3}$ and $\gamma = 5.08 \times 10^{-5}$.

(b) Average number of transactions vs. impatience level γ, for $R = 9.9 \times 10^{-3}$ and $I = 2000$.

(c) Average number of transactions vs. satisfaction level R, for $\gamma = 5.08 \times 10^{-5}$ and $I = 2000$.

increasing I does not increase the number of transactions and the transaction fee.

Then we discuss the asynchronous settings. The expected fee-per-byte of the optimal FTET mechanism under synchronous transaction proposing is higher than under the asynchronous case. Notice that this relation also appears in both Figs. 5(b) and (c). Thus, we make the following observation:

Observation 1: In the optimal FTET mechanism, users choose higher fee-per-byte under the synchronous setting than under the asynchronous setting.

The reason behind Observation 1 is that synchronous transaction proposals intensify the user competition and lead to higher transaction fees. In the asynchronous setting, the competition among users is less, which is reflected through the lower transaction fees.

Moreover, the key insights that the optimal FTET mechanism guarantees the storage sustainability still holds in the asynchronous setting.

Observation 2: The optimal FTET mechanism guarantees the storage sustainability under the asynchronous case.

The optimal FTET mechanism specifies the minimum fee-per-byte as the total storage costs per byte (i.e., $\underline{\rho}^* = C^{\mathrm{tot}}$) in Theorem 3. Thus, it can always guarantee storage sustainability, which has nothing to do with synchronous or asynchronous settings.

On the other hand, the MRTA mechanism cannot always guarantee the storage sustainability (i.e., in Fig. 5(c) when $R < 9 \times 10^{-3}$). The reason will be explained in Fig. 5(c).

2) Fig. 5(b): Impact of the Impatience Level γ on $\mathbb{E}[\rho_i^{\mathrm{NE}}]$: We make an interesting observation as follows.

Observation 3: In the optimal FTET mechanism, a higher impatience level γ does not always lead to a higher expected fee-per-byte $\mathbb{E}[\rho_i^{\mathrm{NE}}]$.

The reason behind Observation 3 is that as γ increases, the increased delay cost leads to the lower number of transactions (as illustrated in Fig. 6(b)), which may lower the intensity of fee competition and hence a lower expected fee-per-byte.

3) Fig. 5(c): Impact of the Satisfaction Level R on $\mathbb{E}[\rho_i^{\mathrm{NE}}]$: We observe that all the expected fee-per-byte values initially increase with satisfaction level R and then remain constant. The reason is that the number of transactions initially increases and then remains constant (as illustrated in Fig. 6(b)). Hence the intensity of fee competition follows the same trend.

When $R < 9 \times 10^{-3}$, the MRTA mechanism cannot meet the storage sustainability constraint. This result is similar to Observation 3: The increase of mining round time and delay cost may not necessarily increase the expected fee-per-byte to meet the storage sustainability constraint.

C. Social Welfare

The social welfare measures the mechanism's overall benefit to the system. In this subsection, we study how the number of users I, the user's impatience level γ, and the user's satisfaction level R from a transaction affect the social welfare sw under two mechanisms. We set the parameters the same as those in Section VI-B. Fig. 7 illustrates the results. All three subfigures show the same curves with the same legends. We will explain Fig. 7(a) in details.

1) Fig. 7(a): Impact of the Number of Users I on Social Welfare: Considering the left red vertical axis of Fig. 7(a), the FTET-syn and FTET-asyn curves plot the social welfares of the optimal FTET mechanism under synchronous and asynchronous transaction proposing, respectively. The MRTA-asyn curves show the social welfares of the MRTA mechanism under the asynchronous setting.
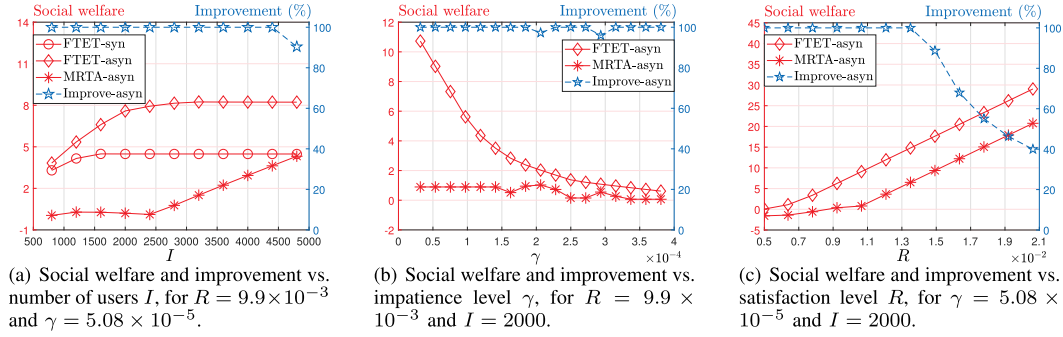
(a) Social welfare and improvement vs. number of users I, for $R = 9.9 \times 10^{-3}$ and $\gamma = 5.08 \times 10^{-5}$.

(b) Social welfare and improvement vs. impatience level γ, for $R = 9.9 \times 10^{-3}$ and $I = 2000$.

(c) Social welfare and improvement vs. satisfaction level R, for $\gamma = 5.08 \times 10^{-5}$ and $I = 2000$.

Fig. 7. Impact of number of users I, impatience level γ, and satisfaction level R on social welfare sw.

We first observe the optimal FTET mechanism under the synchronous setting: the social welfare first increases in I and then roughly remains constant. When I is small, the number of transactions is small (i.e., Fig. 6(a)) and a transaction's delay cost is small. Thus, a transaction's satisfaction level R is higher than its storage and delay costs. As the number of transactions increases with I, the social welfare is higher. However, when I is large, the transaction expiration time T^* prevents users from proposing more transactions (i.e., Fig. 6(a)) and suffering large delay costs. Then the social welfare remains constant.

Then we focus on the asynchronous settings. The social welfare of the optimal FTET mechanism under the asynchronous setting is higher than under the synchronous setting. This is because half of the users in the asynchronous setting propose their transactions at the beginning of mining round 6, hence experiencing smaller delay costs than the synchronous case.

Considering the right blue vertical axis of Fig. 7(a), the Improve-asyn curve corresponds to the social welfare improvement of the optimal FTET mechanism over the MRTA mechanism under the asynchronous setting. We observe that the social welfare improvement is significant with an average value of 99.1%. Moreover, the average social welfare improvements in Figs. 7(b) and (c) are 99.6% and 83.2%, respectively. Thus, we make the following observation:

Observation 4: Under the asynchronous transaction proposing, the optimal FTET mechanism achieves an average social welfare improvement of 94.5% over the MRTA mechanism.

The optimal FTET mechanism introduces the transaction expiration to reject the high-delay transactions. Hence the mechanism avoids the high delay cost and increases the social welfare, which has nothing to do with synchronous or asynchronous settings.

2) Fig. 7(b): Impact of the Impatience Level γ on Social Welfare: By observing two red curves on the left red vertical axis, we find that the optimal FTET mechanism's social welfare roughly decreases in γ, due to the increased delay cost. On the other hand, the MRTA's social welfare does not significantly change with the impatience level. The MRTA mechanism adjusts the delay cost to a level to meet the storage sustainability constraint, neutralizing the impatience level's impact on delay cost.

3) Fig. 7(c): Impact of the Satisfaction Level R on Social Welfare: For the left red vertical axis, we notice that the social welfares of both mechanisms increase in R, due to each user's
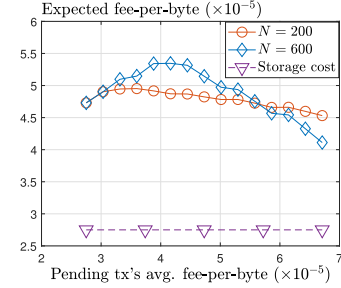

Fig. 8. Average fee-per-byte of pending transactions' impact on users' fee-per-byte.

higher satisfaction level from a transaction. For the right blue vertical axis, we observe that the social welfare improvement decreases in R. As the social welfare increases, the large delay cost caused by the MRTA mechanism becomes less significant.

D. Pending Transactions' Impact

We further conduct some numerical analysis on how pending transactions affect users' strategies. Specifically, when a user makes a decision, he needs to consider the pending transactions proposed earlier and other users' transaction proposing strategies in the same time slot.

We analyze the users' expected fee-per-byte under the optimal FTET mechanism. We consider two cases where the numbers of pending transactions are $N = 200$ and $N = 600$, respectively. We assign the fee-per-byte of pending transactions according to the fee-per-byte distribution of Bitcoin's pending transactions on June 7th, 2022 [46].

Fig. 8 illustrates how the average fee-per-byte of pending transactions affects users' expected fee-per-byte $\mathbb{E}[\rho_i^{\text{NE}}]$. We observe that the user's fee-per-byte first increases and then decreases with the fee-per-byte of pending transactions. This observation reflects how users tradeoff between fee and delay. At first, more high-fee pending transactions intensify the fee competition, forcing users to pay higher fees. However, when there are too many high-fee pending transactions, the fee competition is too intense and users would rather pay low fees and bear long delays. Thus, we have the following observation:

Observation 5: Fees may decrease in the number of pending transactions.

We then observe that users' fee-per-byte has a larger variance under $N = 600$ than under $N = 200$. The reason is as follows: For $N = 600$, there are more pending transactions

than $N = 200$. When pending transactions' average fee-per-byte is low, more pending transactions lead to more intense fee competition and users pay higher fees. When pending transactions' average fee-per-byte is high, more pending transactions cause the fee competition to become too intense (and more intense than $N = 200$), such that users prefer to pay lower fees than $N = 200$.

The key insight that the optimal FTET mechanism guarantees the storage sustainability still holds in Fig. 8 when there are pending transactions proposed earlier.

Observation 6: The optimal FTET mechanism guarantees storage sustainability when there are pending transactions proposed earlier.

The optimal FTET mechanism specifies the minimum fee-per-byte as the total storage costs per byte (i.e., $\varrho^* = C^{\text{tot}}$) in Theorem 3. Thus, it can always guarantee storage sustainability, which has nothing to do with the pending transactions.

VII. CONCLUSION

In this paper, we studied the blockchain storage sustainability issue. We noticed that the key reasons for the issue are miners' heterogeneous storage costs and users' low-intensity fee competition. Then we proposed a Fee and Transaction Expiration Time (FTET) mechanism to incentivize users to pay sufficient transaction fees for the storage costs.

We concluded that high-storage-cost miners admit transactions with fees above a time-increasing threshold. We showed that the optimal FTET mechanism ensures the storage sustainability, while achieving the same maximum social welfare when not considering the storage sustainability constraint. Moreover, the system can improve its social welfare by selectively rejecting some transactions suffering high delays. Furthermore, we implemented a blockchain testbed to compare the performance of the optimal FTET mechanism with the state-of-art mining round time adjustment (MRTA) mechanism. The optimal FTET mechanism achieves better storage sustainability and an average social welfare improvement of 94.5% under asynchronous transaction proposing. The social welfare improvement is due to the transaction expiration rejecting high-delay transactions. We find that more pending transactions may lead to lower transaction fees.

In the future work, we will consider a more practical system model where users enter the system following a stochastic process. We will also consider the case where each miner selects transactions according to the optimal solution of the knapsack problem.

REFERENCES

[1] (Nov. 2018). *Launch an Ethereum Archive Node Now on Quiknode.* [Online]. Available: https://medium.com/quiknode/welcoming-the-newest-member-of-the-quiknode-family-the-eth-archive-node-ac66201e0793

[2] *Cryptocurrency Market Value.* (Feb. 2022). [Online]. Available: https://coinmarketcap.com/all/views/all/

[3] *Price of SSD.* (Feb. 2022). [Online]. Available: https://www.amazon.com/ssd/s?k=ssd

[4] *Ethereum Archive Node Size.* (Feb. 2022). [Online]. Available: https://etherscan.io/chartsync/chainarchive

[5] *Block Size Data.* (Feb. 2022). [Online]. Available: https://www.blockchain.com/charts/avg-block-size

[6] S. Nakamoto. *Bitcoin: A Peer-to-Peer Electronic Cash System.* (2008). [Online]. Available: https://bitcoin.org/bitcoin.pdf

[7] *Ethereum Full Node.* (Feb. 2022). [Online]. Available: https://www.ethernodes.org/

[8] *Node Decline.* (Dec. 2020). [Online]. Available: https://thebitcoin.pub/t/ethereum-essentials-node-nuances/52785

[9] *Bitcoinwiki. Weaknesses.* (Jun. 2020). [Online]. Available: https://en.bitcoin.it/wiki/Weaknesses

[10] Y. Liu, Z. Fang, M. H. Cheung, W. Cai, and J. Huang, "Economics of blockchain storage," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–6.

[11] Y. Liu, Z. Fang, M. H. Cheung, W. Cai, and J. Huang, "An incentive mechanism for sustainable blockchain storage," *IEEE/ACM Trans. Netw.*, vol. 30, no. 5, pp. 2131–2144, Oct. 2022, doi: 10.1109/TNET.2022.3166459.

[12] *Eip-1559.* (Apr. 2019). [Online]. Available: https://github.com/ethereum/EIPs/blob/master/EIPS/eip-1559.md

[13] Q. Hu, Y. Nigam, Z. Wang, Y. Wang, and Y. Xiao, "A correlated equilibrium based transaction pricing mechanism in blockchain," in *Proc. IEEE Int. Conf. Blockchain Cryptocurrency (ICBC)*, May 2020, pp. 1–7.

[14] Z. Ai, Y. Liu, and X. Wang, "ABC: An auction-based blockchain consensus-incentive mechanism," in *Proc. IEEE 26th Int. Conf. Parallel Distrib. Syst. (ICPADS)*, Dec. 2020, pp. 609–616.

[15] R. Lavi, O. Sattath, and A. Zohar, "Redesigning Bitcoin's fee market," in *Proc. World Wide Web Conf. (WWW)*, 2019, pp. 2950–2956.

[16] S. Basu, D. Easley, M. O'Hara, and E. Sirer, "Towards a functional fee market for cryptocurrencies," 2019, *arXiv:1901.06830.*

[17] H. Dang, T. T. A. Dinh, D. Loghin, E.-C. Chang, Q. Lin, and B. C. Ooi, "Towards scaling blockchain systems via sharding," in *Proc. Int. Conf. Manage. Data*, Jun. 2019, pp. 123–140.

[18] E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, E. Syta, and B. Ford, "OmniLedger: A secure, scale-out, decentralized ledger via sharding," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2018, pp. 583–598.

[19] M. Zamani, M. Movahedi, and M. Raykova, "RapidChain: Scaling blockchain via full sharding," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2018, pp. 931–948.

[20] E. Palm, O. Schelén, and U. Bodin, "Selective blockchain transaction pruning and state derivability," in *Proc. Crypto Valley Conf. Blockchain Technol. (CVCBT)*, Jun. 2018, pp. 31–40.

[21] R. Dennis, G. Owenson, and B. Aziz, "A temporal blockchain: A formal analysis," in *Proc. Int. Conf. Collaboration Technol. Syst. (CTS)*, Oct. 2016, pp. 430–437.

[22] W. K. Chan, J.-J. Chin, and V. T. Goh, "Simple and scalable blockchain with privacy," *J. Inf. Secur. Appl.*, vol. 58, pp. 102700–102710, May 2021.

[23] *Ethereum Community.* (Sep. 2015). [Online]. Available: https://ethereum.org/en/community/

[24] *Fee-Per-Byte.* (Oct. 2021). [Online]. Available: https://metamug.com/article/security/blockchain-transaction-fee-satoshi-per-byte

[25] W. Cai, Z. Wang, J. B. Ernst, Z. Hong, C. Feng, and V. C. Leung, "Decentralized applications: The blockchain-empowered software system," *IEEE Access*, vol. 6, pp. 53019–53033, 2018.

[26] M. Ul Hassan, M. Husain Rehmani, and J. Chen, "DEAL: Differentially private auction for blockchain-based microgrids energy trading," *IEEE Trans. Services Comput.*, vol. 13, no. 2, pp. 263–275, Apr. 2019.

[27] C. Wang, X. Chu, and Y. Qin, "Measurement and analysis of the bitcoin networks: A view from mining pools," in *Proc. 6th Int. Conf. Big Data Comput. Commun. (BIGCOM)*, Jul. 2020, pp. 180–188.

[28] *Bitcoin Transaction Size.* (Oct. 2015). [Online]. Available: https://tradeblock.com/blog/analysis-of-bitcoin-transaction-size-trends

[29] *Approximation Algorithm.* (Jan. 2009). [Online]. Available: https://courses.engr.illinois.edu/cs598csc/sp2009/lectures/lecture_4.pdf

[30] Y. Jiao, P. Wang, D. Niyato, and K. Suankaewmanee, "Auction mechanisms in cloud/fog computing resource allocation for public blockchain networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 9, pp. 1975–1989, Sep. 2019.

[31] C. Chen, J. Wu, H. Lin, W. Chen, and Z. Zheng, "A secure and efficient blockchain-based data trading approach for Internet of Vehicles," *IEEE Trans. Veh. Technol.*, vol. 68, no. 9, pp. 9110–9121, Sep. 2019.

[32] D. Easley, M. O'Hara, and S. Basu, "From mining to markets: The evolution of Bitcoin transaction fees," *J. Financial Econ.*, vol. 134, no. 1, pp. 91–109, Oct. 2019.

[33] I. Tsabary and I. Eyal, "The gap game," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Jun. 2018, pp. 713–728.

[34] R. Pass and E. Shi, "FruitChains: A fair blockchain," in *Proc. ACM Symp. Princ. Distrib. Comput.*, Jul. 2017, pp. 315–324.

[35] *Congestion of Cryptokitties*. (Feb. 2018). [Online]. Available: https://consensys.net/blog/news/the-inside-story-of-the-cryptokitties-congestion-crisis/

[36] W. Wang et al., "A survey on consensus mechanisms and mining strategy management in blockchain networks," *IEEE Access*, vol. 7, pp. 22328–22370, 2019.

[37] M. Franceschet et al., "Crypto art: A decentralized view," *Leonardo*, vol. 54, no. 4, pp. 402–405, Aug. 2021. [Online]. Available: https://doi.org/10.1162/leon_a_02003

[38] *Online Appendix*. (Feb. 2022). [Online]. Available: https://www.dropbox.com/s/zrg1bs34k19fsdu/JSAC-App.pdf?dl=0

[39] J. Li, Y. Yuan, S. Wang, and F.-Y. Wang, "Transaction queuing game in Bitcoin blockchain," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 114–119.

[40] *Blockchain Prototype*. (Feb. 2022). [Online]. Available: https://github.com/ksl20200108/StorageSustainabilityMechanism

[41] *Average Transaction Size*. (Nov. 2017). [Online]. Available: https://ethereum.stackexchange.com/questions/30175/what-is-the-size-bytes-of-a-simple-ethereum-transaction-versus-a-bitcoin-trans

[42] *Number of Transactions Per Block*. (Feb. 2022). [Online]. Available: https://etherscan.io/

[43] *Block Time*. (Feb. 2022). [Online]. Available: https://etherscan.io/chart/blocktime

[44] M. J. Osborne and A. Rubinstein, *A Course Game Theory*. Cambridge, MA, USA: MIT Press, 1994.

[45] *Ethereum Transaction Delay*. (Jun. 2019). [Online]. Available: https://legacy.ethgasstation.info/blog/ethereum-transaction-how-long/

[46] *Bitcoin Pending Transactions' Fee-Per-Byte Distribution*. (Jun. 2022). [Online]. Available: https://bitcoinfees.earn.com/

Yunshu Liu (Member, IEEE) received the B.Eng. and M.Eng. degrees from Shanghai Jiao Tong University, Shanghai, China, in 2015 and 2018, respectively. He is currently pursuing the Ph.D. degree with the Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong. His research interests include the field of network economics, game theory, and control theory, with current emphasis on blockchain and its applications.

Shulin Ke is currently pursuing the bachelor's degree in financial engineering with The Chinese University of Hong Kong, Shenzhen. His research interests include the field of decentralized finance, data analysis of game finance, and physiology basis for traditional Chinese medicine.

Zhixuan Fang (Member, IEEE) received the B.S. degree in physics from Peking University, China, in 2013, and the Ph.D. degree in computer science from Tsinghua University, Beijing, China, in 2018. He is currently a Tenure-Track Assistant Professor at the Institute for Interdisciplinary Information Sciences (IIIS), Tsinghua University. He mainly focuses on the design and analysis of multi-agent systems and networked systems.

Man Hon Cheung received the B.Eng. and M.Phil. degrees in information engineering from The Chinese University of Hong Kong (CUHK) in 2005 and 2007, respectively, and the Ph.D. degree in electrical and computer engineering from the University of British Columbia (UBC) in 2012. He is currently an Assistant Professor with the Department of Computer Science, City University of Hong Kong. Previously, he was a Research Assistant Professor at the Department of Information Engineering, CUHK. He was awarded the Graduate Student International Research Mobility Award by UBC and the Global Scholarship Programme for Research Excellence by CUHK. He serves as a Technical Program Committee Member in IEEE INFOCOM, WiOpt, ICC, GLOBECOM, and WCNC. He is an Associate Editor of IEEE COMMUNICATIONS LETTERS.

Wei Cai (Senior Member, IEEE) received the B.Eng. degree from Xiamen University in 2008, the M.Sc. degree from Seoul National University in 2011, and the Ph.D. degree from The University of British Columbia (UBC) in 2016. He is currently an Assistant Professor in computer engineering with the School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen. He is also working as the Director of the CUHK(SZ)-White Matrix Joint Metaverse Laboratory. He has coauthored more than 70 journals and conference papers in the area of interactive multimedia and distributed/decentralized systems. His recent research interests are mainly in the topic of human-centered computing for metaverse, including blockchain, digital game, web 3.0, and computational art. He serves as a Technical Program Committee Member of ACM MM, MMSys, and NOSSDAV. He was a recipient of the 2015 Chinese Government Award for the Outstanding Self-Financed Students Abroad, the UBC Doctoral Four-Year-Fellowship from 2011 to 2015, and the Brain Korea 21 Scholarship. He also received the Best Student Paper Award from ACM BSCI2019 and the best paper awards from CCF CBC2018, IEEE CloudCom2014, SmartComp2014, and CloudComp2013. He is an Associate Editor of IEEE TRANSACTIONS ON CLOUD COMPUTING.

Jianwei Huang (Fellow, IEEE) received the Ph.D. degree in ECE from Northwestern University in 2005. He is currently a Presidential Chair Professor and Associate Vice President, The Chinese University of Hong Kong, Shenzhen. He is working as the Vice President with the Shenzhen Institute of Artificial Intelligence and Robotics for Society and the Editor-in-Chief of IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING. He worked as a Post-Doctoral Research Associate at Princeton University from 2005 to 2007. From 2007 to 2018, he was as a Faculty Member of the Department of Information Engineering, The Chinese University of Hong Kong. He has published more than 300 papers in leading venues, with a Google Scholar citation of more than 14900 and an H-index of 61. He has coauthored 11 best paper awards, including the 2011 IEEE Marconi Prize Paper Award in Wireless Communications. He has coauthored seven books, including the textbook on wireless network pricing. His research interests are in the area of network optimization, network economics, and network science, with applications in communication networks, energy networks, data markets, and crowd intelligence. He is an IEEE ComSoc Distinguished Lecturer and a Clarivate Web of Science Highly Cited Researcher.