

An Incentive Mechanism for Sustainable Blockchain Storage

Yunshu Liu¹, Graduate Student Member, IEEE, Zhixuan Fang², Member, IEEE, Man Hon Cheung³,
Wei Cai⁴, Member, IEEE, and Jianwei Huang⁵, Fellow, IEEE

Abstract—Miners in a blockchain system are suffering from ever-increasing storage costs, which in general have not been properly compensated by the users' transaction fees. This reduces the incentives for the miners' participation and may jeopardize the blockchain security. To mitigate this blockchain insufficient fee issue, we propose a Fee and Waiting Tax (FWT) mechanism, which explicitly considers the two types of negative externalities in the system. Specifically, we model the interactions between the protocol designer, users, and miners as a three-stage Stackelberg game. By characterizing the equilibrium of the game, we find that miners neglecting the negative externality in transaction selection cause they are willing to accept insufficient-fee transactions. This leads to the insufficient storage fee issue in the existing protocol (i.e., deployed in Bitcoin and Ethereum). Moreover, our proposed optimal FWT mechanism can motivate users to pay sufficient transaction fees to cover the storage costs and achieve the unconstrained social optimum. Numerical results show that the optimal FWT mechanism guarantees sufficient transaction fees and achieves an average social welfare improvement of 51.43% or more over the existing protocol. Furthermore, the optimal FWT mechanism reduces the average waiting time of low-fee transactions and all transactions by 68.49% and 61.56%, respectively.

Index Terms—Blockchain storage, game theory, incentive mechanism.

I. INTRODUCTION

WITH the booming of cryptocurrencies, its underlying blockchain protocol imposes ever-increasing and

Manuscript received June 18, 2021; revised December 16, 2021; accepted March 15, 2022; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor K. Jagannathan. This work was supported in part by the Guangdong Basic and Applied Basic Research Foundation under Project 2021B1515120008, in part by the Shenzhen Science and Technology Program under Project JCYJ20210324120011032 and Project JCYJ20210324124205016, in part by the Shenzhen Institute of Artificial Intelligence and Robotics for Society, and in part by the City University of Hong Kong's Start-up Grant under Project 9610493. Part of this paper was presented in ICC 2020 [DOI: 10.1109/ICC40277.2020.9148934]. (Corresponding author: Jianwei Huang.)

Yunshu Liu is with the Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong, and also with the Shenzhen Institute of Artificial Intelligence and Robotics for Society (AIRS), Shenzhen 518129, China (e-mail: ly018@ie.cuhk.edu.hk).

Zhixuan Fang is with the Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing 100084, China, and also with the Shanghai Qi Zhi Institute, Shanghai 200030, China (e-mail: zfang@mail.tsinghua.edu.cn).

Man Hon Cheung is with the Department of Computer Science, City University of Hong Kong, Hong Kong (e-mail: mhcheung@cityu.edu.hk).

Wei Cai and Jianwei Huang are with the School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen 518172, China, and also with the Shenzhen Institute of Artificial Intelligence and Robotics for Society (AIRS), Shenzhen 518129, China (e-mail: caiwei@cuhk.edu.cn; jianweihuang@cuhk.edu.cn).

Digital Object Identifier 10.1109/TNET.2022.3166459

significant storage costs on the solid-state storage drives [2] of the operation nodes (often referred to as *miners* [3]). For example, consider the second-largest cryptocurrency, Ethereum. Its data size grows by nearly 16 folds from 385 gigabytes in Feb. 2017 to 6.0 terabytes in Jan. 2021 [4]. Currently, it costs each miner \$2000 per month to store the entire blockchain [2].

On the other hand, the blockchain users' transaction fee payments to the miners are insufficient to compensate such fast growth and significant storage costs. For example, the Ethereum users paid an average monthly transaction fee of \$7.32 million during the first half of 2020 [5], much smaller than \$20 million monthly costs of all Ethereum nodes storing the entire blockchain (roughly 10,000 nodes [6] and \$2000 monthly cost per node). The gap between the storage cost and transaction fee is filled by block reward, which is designed to gradually decrease over time in many blockchain systems (e.g., Bitcoin [7]).

The insufficient transaction fee issue brings negative impacts on both miners and users. On the one hand, miners will have less incentive to stay in the system, jeopardizing the blockchain system security. For example, the number of miners storing the Ethereum blockchain has declined 66% since 2018, where the high storage costs could have played a major role [8]. The decline of miners may be catastrophic to the blockchain in the long run, as a less decentralized blockchain will become vulnerable to malicious attacks [9]. On the other hand, if the blockchain system is no longer sustainable, users will eventually choose not to use the system and their cryptocurrency will be worthless in the long run. To maintain a healthy decentralized ecosystem, it is critically important to mitigate the issue of insufficient transaction fees (for covering the storage costs).

Mitigating the insufficient fee issue requires the proper mechanism to encourage users to pay sufficient transaction fees to the miners. The protocol designer of the blockchain (e.g., the technical community serves as a leading role in protocol design) needs a proper mechanism to motivate users to pay sufficient transaction fees. However, to the best of our knowledge, there lacks enough *theoretical* mechanism design work aiming at mitigating the insufficient fee issue (although the discussion in the technical community is heated [10]). This motivates us to take the first step in this paper to propose such a mechanism to address the issue.

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

- *Why are miners willing to accept insufficient-fee transactions in the existing blockchain system?*

- *How to design an incentive mechanism to encourage users to pay sufficient transaction fees to the miners?*

To answer the above two questions, we propose a three-stage model to characterize the blockchain system. In Stage III, miners select transactions to include in the blockchain, considering the tradeoff between transaction fees and storage costs. In Stage II, users determine the transaction generation rates at different fees, by considering the tradeoff between paying high transaction fees and bearing high transaction waiting time. The transaction waiting time depends on how miners select transactions. In user-miner fee interaction of Stages II and III, there exist two types of negative externalities, i.e., each miner's transaction selection imposes storage costs on other miners and each user's transaction generation increases the average waiting time of other users. These two types of negative externalities are the reasons behind miners accepting insufficient-fee transactions and users experiencing excessive waiting time, respectively, in the existing protocol (i.e., the fee mechanism currently deployed in Bitcoin and Ethereum). Motivated by such an observation, we propose a Fee and Waiting Tax (FWT) mechanism for the protocol designer in Stage I. The mechanism determines the transaction fee choices for the users and meanwhile imposes waiting tax on the users, in order to motivate users to pay sufficient fees while achieving social welfare maximization.

Our key results and contributions are as follows:

- *Fee mechanism design on blockchain storage:* To the best of our knowledge, this is one of the first theoretical studies on the fee mechanism design aiming at mitigating blockchain insufficient storage fee issue. The goal of the mechanism is to ensure the long-term stability and security of blockchain system, as the ever-increasing storage costs become a significant burden to miners and reduce their incentives to participate in the blockchain operation.
- *Three-Stage Interaction Model:* We propose a three-stage game-theoretical model to characterize the interactions among the protocol designer, users, and miners. The analysis of the model is analytically challenging as the user-miner fee interaction is a two-stage queueing game. Specifically, the state transition of the queue system is stochastically affected by the decision of all users and miners, and each miner faces an integer programming problem which is game-theoretically coupled with other miners' strategies. Nevertheless, we can derive the subgame perfect equilibrium of the model in closed form.
- *Explaining the deficiency of the existing protocol:* Through the analysis of three-party interaction, we find that under the existing protocol each miner is unaware of the negative externality that it imposes on other miners when making transaction selections. This causes the miners to accept transactions with fees not enough to cover the overall system storage costs.
- *Proposing a mechanism to generate sufficient fee and achieve unconstrained social optimum:* We propose an FWT mechanism motivated by two types of negative externalities in the system. We show that the optimal FWT mechanism incentivizes users to pay sufficient

transaction fees for the overall system storage costs while achieving the unconstrained social optimum.

- *Ethereum-based numerical results:* We relax the assumption of two user types and conduct the numerical analysis based on practical Ethereum blockchain parameters. Compared with the existing protocol, our proposed optimal FWT mechanism not only produces sufficient transaction fees but also achieves an average social welfare improvement of 51.43% or more. Moreover, the optimal FWT mechanism reduces the average waiting time of low-fee transactions and all transactions by 68.49% and 61.56%, respectively.

The rest of the paper is organized as follows. Section II reviews the related literature. Section III introduces the system model. We characterize the mathematical details and derive the closed-form subgame perfect equilibrium of the model's three stages in Sections IV, V, and VI, respectively. We evaluate the system performance in Section VII. We conclude this paper in Section VIII.

II. LITERATURE REVIEW

Our work studies the mechanism design on the user-miner fee interaction regarding blockchain storage. Hence, we review the previous literature from two aspects, i.e., analysis of the fee interaction scheme in the existing blockchain and new fee mechanism design in blockchain.

A. Analysis of Fee Interaction Scheme in Existing Blockchain

The first group of literature (e.g., [11]–[15]) analyzing how users set transaction fees regarding the transaction waiting time in the existing blockchain. Huberman *et al.* in [11] found that the desire to shorten the transaction waiting time is the primary reason for users to pay transaction fees. Following this work, several other papers analyzed how waiting time affects users' transaction fees. Easley *et al.* in [12] showed that as the average transaction waiting time increases, the ratio of users who pay fees also increases. Li *et al.* in [13] analyzed the case where users choose between paying a fixed-level fee or not paying the fee. They revealed that an excessive waiting time would discourage low-waiting-time-cost users from paying transaction fees. Several further studies in [14], [15] investigated the factors in blockchain that may impact the waiting time. These works proved that both the block production time and the block propagation time affect the waiting time (hence the transaction fee). However, the previous works did not consider how miners tradeoff between the transaction fees and storage costs. We consider a general model where each miner chooses the transactions considering the tradeoff between the transaction fees and storage costs. This consideration significantly complicates the analysis.

B. New Fee Mechanism Design in Blockchain

The second group of literature focused on transaction fee market design (e.g., [16]–[20]) with different goals. Vitalik *et al.* in [16] proposed a burning base fee mechanism to make the fee prediction easy for the users. Some works aimed to improve the system performance. Hu *et al.* in [17]

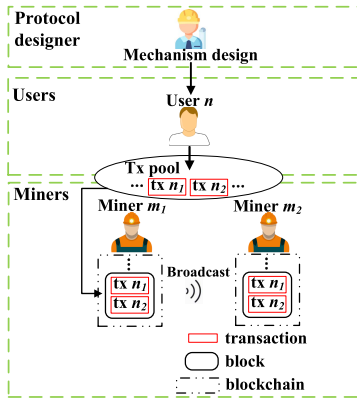


Fig. 1. Blockchain operation.

proposed a correlated-equilibrium-based fee mechanism to achieve both the individual and global optimum. Ai *et al.* in [18] applied the double auction to improve the fairness of the system. The literature [19] used the second prize auction to reduce the variance of transaction fees. Lavi *et al.* in [20] showed that the monopolistic auction is resilient to market manipulation. Overall, the current work on the fee mechanism design did not consider two types of negative externalities in miners' transaction selections and users' transaction generations. Our work is one of the first analytical studies to explicitly consider the mitigation of two types of negative externalities in the blockchain system.

III. SYSTEM MODEL

In this section, we describe the system model of the blockchain. We first introduce the high-level operation process of the blockchain system in Section III-A and then discuss our proposed FWT mechanism in Section III-B. Finally we propose a three-stage Stackelberg game to characterize the blockchain system in Section III-C.

A. Blockchain Operation

In this subsection, we briefly introduce the operation process of blockchain.

Fig. 1 illustrates the typical blockchain operation [21]. The protocol designer first determines the mechanism for users and miners. Then the users generate transactions and choose the transaction fees. Finally, miners select transactions and include them in the blockchain through mining. The details are as follows:

- 1) Protocol designer's mechanism design: The protocol designer determines the consensus protocol for the system. The blockchain online community usually collectively acts as the protocol designer. For example, in 2018, the Ethereum online community¹ proposed Ethereum Improvement Proposal 1234 to decrease block reward by 33%.

¹<https://www.reddit.com/r/ethereum/>

- 2) Users' transaction generation: A user n generates two transactions (tx_{n_1} and tx_{n_2}) and assigns the fee-per-byte value for each transaction.² The transaction fee of a transaction satisfies:

$$\text{transaction fee} = \text{transaction size} \times \text{fee-per-byte}.$$

The transaction fee serves as an incentive for miners to include the transaction into a future *block*.³ Each generated transaction enters the transaction pool (tx pool) and waits for miners to include it in the blockchain.

- 3) Miners' mining: The process of mining a new block (also referred to as one round of mining) contains several steps, as follows:

- First, each miner selects a set of transactions from the transaction pool (e.g., miner m_1 selects both tx_{n_1} and tx_{n_2}).
- Next, miners compete to solve a cryptographic puzzle. Once a miner solves the puzzle (being first among all miners), he will pack his selected transactions, the puzzle solution, along with some auxiliary data into a block. The miner who produces such a new block can get the fees from his selected transactions (e.g., miner m_1 gets fees of tx_{n_1} and tx_{n_2}) and the block reward (for generating this new block) as a bonus. The transactions in the new block are included in the blockchain.
- Finally, the miner who produces the new block broadcasts the block information to his neighbors in the network, and *all* miners need to update the local storage to include the new block.

Next, we will introduce our proposed Fee and Waiting Tax (FWT) mechanism for the protocol designer.

B. FWT Mechanism

There are two types of negative externalities in the existing blockchain protocol, i.e., each miner's transaction selection imposes storage costs on other miners and each user's transaction generation increases the average waiting time of other users. These two types of negative externalities are why miners accept insufficient-fee transactions and users experience excessive waiting time, respectively, in the existing protocol. Motivated by the negative externalities, we propose the FWT mechanism as follows:

- 1) Fee choices: The FWT mechanism offers several fee-per-byte choices for users. Users can choose the transaction generation rates for different fee-per-byte choices. The protocol designer properly optimizes the fee-per-byte choices such that the users pay sufficient fees to cover the total system storage costs.
- 2) Waiting tax: The FWT mechanism imposes a waiting tax on each user. Specifically, each user pays the waiting tax to other users based on the negative impact generated by him (which will be explained in Section V-A). Such

²Users often set the fee-per-byte rather than the transaction fee in Bitcoin [22].

³A block is a container of transactions. In Bitcoin [7], a block contains the cryptographic hash of the previous block, a time-stamp, and the data [23].

TABLE I
KEY NOTATIONS

Variables	
ρ	Fee-per-byte choice
ρ_i	i -th fee-per-byte choice
\mathbf{P}	Waiting tax rate vector
$P_{HH}(P_{HL})$	Waiting tax that a type- H user pays to another type- H (type- L) user
$P_{LH}(P_{LL})$	Waiting tax that a type- L user pays to another type- H (type- L) user
p_{nl}	Waiting tax that user n pays to user l
λ_n	User n 's transaction generation rates
λ_{n_i}	User n 's transaction generation rates at ρ_i
\mathcal{X}_m^k	Miner m 's transaction selection in round k
Parameters	
$N(\mathcal{N})$	Number (set) of users
$N_H(N_L)$	Number of type- H (type- L) users
q	Probability of generating a large transaction in a transaction generation event
$TX_n(t)$	Number of user n 's transactions from time 0 to t
R_n	User n 's transaction on-chain utility
$R_H(R_L)$	A type- H (type- L) user's transaction on-chain utility
$h_B(h_S)$	A type- B (type- S) user's net transaction utility
$\rho_{n,j}$	Fee-per-byte of transaction $tx_{n,j}$
$s_{n,j}$	Size of transaction $tx_{n,j}$
$w_{n,j}$	Waiting time of transaction $tx_{n,j}$
γ	User's impatience level
$M(\mathcal{M})$	Number (set) of miners
C_s	A miner's storage cost per byte
α_m	Miner m 's mining power
μ	Block generation rate
Φ	Block size limit
\mathcal{Q}^k	Transaction pool in round k

a tax will encourage users to be more conservative in generating transactions.⁴ If a user maximizes his payoff without the waiting tax, he will generate too many transactions and all the other users will experience excessive waiting time [24], causing poor user experiences and social welfare reduction. Notice that when we sum up all users' payoffs, the waiting taxes among users actually cancel out. So as a group, the users do not have any extra burden from the waiting tax.

Next, we will present the FWT mechanism in more detail. We summarize key notations of the model in Table I.

C. Three-Stage Stackelberg Game

We model the interactions among the protocol designer, N users, and M miners as a three-stage Stackelberg game, as illustrated in Fig. 2.

⁴Each user can attach some cryptocurrency to his transaction as the waiting tax payment (similar to the attachment of transaction fee). When miners include the user's transaction on blockchain, he pays the waiting tax to the other users.

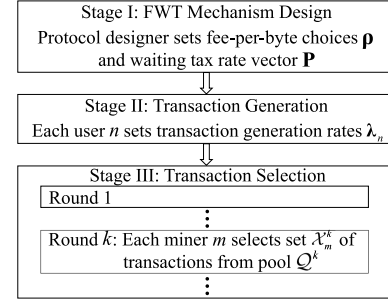


Fig. 2. Three-stage Stackelberg game.

In Stage I, the protocol designer ensures users to pay sufficient fees by setting *fee-per-byte choices* $\rho = (\rho_1, \rho_2, \dots, \rho_I)$. Without loss of generality, we consider that the protocol designer offers users $I \geq 2$ fee-per-byte choices with $\rho_1 > \rho_2 > \dots > \rho_I \geq 0$. Moreover, the protocol designer maximizes the social welfare (a common objective in the literature [25], [26]) by setting *the waiting tax rate vector* \mathbf{P} . The waiting tax rate vector specifies each user's tax payment to all other users, compensating the waiting costs that the user imposes on others.

In Stage II, each user n tradeoffs between paying high transaction fees and bearing long time waiting for transaction inclusion. More specifically, the user chooses the transaction generation rates $\lambda_n = (\lambda_{n_1}, \lambda_{n_2}, \dots, \lambda_{n_I})$, where λ_{n_i} ($i = 1, 2, \dots, I$) denotes the transaction generation rate corresponding to the fee-per-byte choice ρ_i . Such a differentiated generation rate and fee-per-byte choice provide flexibility to meet the requirements of different applications.⁵ Moreover, the waiting tax rate vector \mathbf{P} assigns different taxes to different types of users (the details are in Section V-A) and each user pays the waiting tax to all the others accordingly.

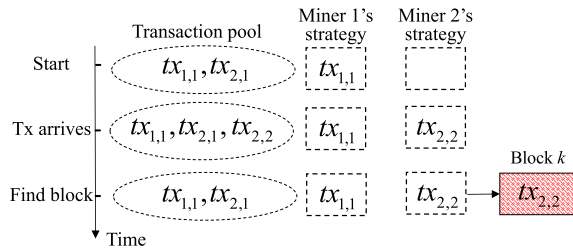
In Stage III, mining proceeds continuously over time. Without loss of generality, we examine the round $k = 1, 2, \dots$ of mining, during which miners mine the block k . The length of each round k (the time between the successful mining of block $k - 1$ and k) follows an exponential distribution.⁶ We further assume that the block propagation delay is zero,⁷ i.e., all miners receive the new block as soon as some miner successfully mines such a block. When determining what to include in a block, each miner m wants to achieve the proper balance between receiving transaction fees and bearing storage costs. The timeline of round k is as follows:

- 1) First, each miner m selects a set of transactions \mathcal{X}_m^k from the transaction pool to include in the new block. The transaction pool is the set of all transactions waiting to be included in a block.
- 2) During round k , users may generate transactions at any time. The newly generated transactions enter the transaction pool and each miner m can change his

⁵For example, in Ethereum, all top-3 users who pay most transaction fees generate transactions with significantly different fee-per-byte (i.e., gas price) for different applications simultaneously [27].

⁶The exponential distribution is confirmed by Bitcoin data analysis [28] and is also commonly done in blockchain analysis [12], [29].

⁷This is a valid assumption because the average block propagation delay in Bitcoin is roughly 2% of block interval time [30].

Fig. 3. Timeline of round k .

transaction selection \mathcal{X}_m^k .⁸ We denote the transaction pool just before miners find block k as \mathcal{Q}^k such that $\mathcal{X}_m^k \subseteq \mathcal{Q}^k$ and notice that finding a new block is a stochastic event.

- 3) When a miner finds block k , the round k ends. The miner who finds the block k receives the transaction fees from the transactions included in his proposed block. All the miners store the block k and bear the costs of storage individually. The transaction pool updates by removing those transactions that have been included in the block k .

Fig. 3 illustrates the above mining process with the case of 2 users and 2 miners. For multiple transactions generated by user n , we will differentiate them in the subscript j , i.e., $tx_{n,j}$ ($j = 1, 2, \dots$).

- 1) First, there are two transactions $tx_{1,1}$ and $tx_{2,1}$ in transaction pool in Fig. 3. Miner 1 and 2 adopt strategies $\mathcal{X}_1^k = \{tx_{1,1}\}$ and $\mathcal{X}_2^k = \emptyset$, respectively.
- 2) During round k , user 2 generates a new transaction $tx_{2,2}$ and it enters transaction pool. Miner 1's strategy remains the same while miner 2 changes his strategy to $\mathcal{X}_2^k = \{tx_{2,2}\}$. In this example, $\mathcal{Q}^k = \{tx_{1,1}, tx_{2,1}, tx_{2,2}\}$.
- 3) Miner 2 finds a block k and round k ends. Miner 2 includes $tx_{2,2}$ in blockchain and transaction pool deletes $tx_{2,2}$.

In the next three sections, we will introduce the mathematical detail of each stage of the model and analyze it through backward induction.

IV. STAGE III: TRANSACTION SELECTION EQUILIBRIUM OF MINERS

In this section, we will characterize how miners select transactions in Stage III. We first model miners' transaction selections in round $k = 1, 2, \dots$ of mining as a game in Section IV-A, then we characterize the Nash equilibrium of the game in Section IV-B.

A. Model of Miners Transaction Selection in Round k

We will focus on a particular round k of mining, during which each miner selects a set of transactions to maximize his own payoffs. We formulate the miners' interaction as a non-cooperative game.

- 1) *Miners*: We consider the set of miners as $\mathcal{M} = \{1, \dots, M\}$.⁹ The normalized mining power (e.g., computing

⁸The reason is that the mining process is memoryless and the success rate of finding a block is independent of the included transaction [28].

⁹We analyze the system in a quasi-static state [12], [29] [31]. That is to say, there are no users or miners joining or leaving the system.

power in proof of work) of miner $m \in \mathcal{M}$ is $\alpha_m > 0$, which represents the probability of miner m successfully finding a block. We have $\sum_{m \in \mathcal{M}} \alpha_m = 1$.

- 2) *Miners' Strategies*: Each miner m selects a set $\mathcal{X}_m^k \subseteq \mathcal{Q}^k$ of transactions from the transaction pool \mathcal{Q}^k . The block size limit is Φ , meaning the block can contain up to Φ bytes of transactions, i.e.,

$$\sum_{(n,j) \in \mathcal{X}_m^k} s_{n,j} \leq \Phi, \quad (1)$$

where $s_{n,j}$ denotes the size of transaction $tx_{n,j}$. We generalize the constant transaction size model in [11]–[15] and consider two possible sizes of transactions, i.e., Φ and $\frac{\Phi}{2}$. As transactions vary in both sizes and fees, we adopt a benign assumption where each miner adopts one of the following three strategies:

- Strategy 1: Select the highest fee-per-byte transaction from the transaction pool, i.e.,

$$\mathcal{X}_m^k = \mathcal{H}(\mathcal{Q}^k) \triangleq \arg \max_{(n,j) \in \mathcal{Q}^k} \rho_{n,j}. \quad (2)$$

- Strategy 2: Select two highest fee-per-byte transactions among transactions with the size $\frac{\Phi}{2}$ from the transaction pool, i.e.,

$$\mathcal{X}_m^k = \mathcal{T}(\mathcal{Q}^k) \triangleq \arg \max_{\substack{\mathcal{A} \subseteq \{(n,j) \in \mathcal{Q}^k | s_{n,j} = \frac{\Phi}{2}\} \\ |\mathcal{A}| = 2}} \sum_{(n,j) \in \mathcal{A}} \rho_{n,j}. \quad (3)$$

- Strategy 3: Select no transaction, i.e., $\mathcal{X}_m^k = \emptyset$.

Selecting the highest-fee-per-byte transaction aligns with the empirical studies of blockchain [22], [32]. When there is more than one highest-fee-per-byte transaction, miners will select the earliest generated one, on a first-come-first-serve basis.

- 3) *Miners' Payoff Functions*: Miner m 's payoff depends on both the transaction fee and storage cost.¹⁰

- Transaction fee: For a transaction $tx_{n,j}$, its transaction fee is the product of the transaction size and fee-per-byte, i.e., $s_{n,j} \rho_{n,j}$. Only the miner who successfully finds a block receives the transaction fees from his selection. Thus miner m will get the total transaction fees of $\sum_{(n,j) \in \mathcal{X}_m^k} s_{n,j} \rho_{n,j}$ with probability α_m .
- Storage cost: For analysis, we assume that all miners have homogeneous storage cost of C_s per byte, representing that miners use similar storage technology. Storing a transaction $tx_{n,j}$ with size $s_{n,j}$ imposes a storage cost $s_{n,j} C_s$ to a miner. If any miner $l \in \mathcal{M}$ selects transaction $tx_{n,j}$ (i.e., $\mathcal{X}_l^k = \{(n,j)\}$) and successfully finds a block (with probability α_l), all miners need to store that block and bear the storage costs $s_{n,j} C_s$ for the transaction $tx_{n,j}$ that miner l selects [3]. Overall, miner m 's storage costs in round k is as follows:¹¹

$$C^k(\mathcal{X}_m^k, \mathcal{X}_{-m}^k) = \sum_{l \in \mathcal{M}} \alpha_l \sum_{(n,j) \in \mathcal{X}_l^k} s_{n,j} C_s, \quad (4)$$

¹⁰We do not consider the block reward and the cost of running the mining machine since they are not affected by each miner's transaction selection. Besides, the transaction fees are the key to cover blockchain storage costs as the block reward gradually shrinks.

¹¹We neglect the storage costs of non-transaction data since it is very small, e.g., the fraction of non-transaction data in Bitcoin is about 0.1% [33].

where $\mathcal{X}_{-m}^k = (\mathcal{X}_l^k, \forall l \in \mathcal{M}, l \neq m)$ represents the strategies of all the miners other than m . Miner m 's storage costs in (4) reveals the *negative externality in transaction selection*: when a miner selects a transaction and finds a block, it imposes storage costs to all the other miners.

Combining the transaction fee and storage cost, miner m 's payoff in round k is:

$$v_m^k(\mathcal{X}_m^k, \mathcal{X}_{-m}^k, \rho) = \alpha_m \sum_{(n,j) \in \mathcal{X}_m^k} s_{n,j} \rho_{n,j} - C^k(\mathcal{X}_m^k, \mathcal{X}_{-m}^k). \quad (5)$$

4) *Game Formulation*: We formulate the round k of mining as a non-cooperative game, where miners simultaneously select the transactions (to be included in his block) to maximize their own payoffs.

Game 1 (Stage III: Transaction Selection Game in round k): In Stage III, Transaction Selection Game in round $k = 1, 2, \dots$ is a tuple $\Phi^k = (\mathcal{M}, \mathcal{B}^k, \mathbf{V}^k)$ defined by:

- **Players**: The set of miners \mathcal{M} .
- **Strategies**: Each miner $m \in \mathcal{M}$ selects a set $\mathcal{X}_m^k \in \mathcal{B}_m^k \triangleq \{\mathcal{H}(\mathcal{Q}^k), \mathcal{T}(\mathcal{Q}^k), \emptyset\}$ of transactions. The strategy profiles of all the miners is $(\mathcal{X}_m^k, \forall m \in \mathcal{M})$. The set of feasible strategy profile of all miners is $\mathcal{B}^k = \prod_{m \in \mathcal{M}} \mathcal{B}_m^k$.
- **Payoffs**: The vector $\mathbf{V}^k = (v_m^k, \forall m \in \mathcal{M})$ contains all miners' payoffs as defined in (5).

In Game 1, each miner tradeoffs between the transaction fee and storage cost to maximize his payoff, considering the strategies of other miners. Specifically, on the one hand, miner m gets high revenue for selecting a high-fee transaction and finding a block. Meanwhile, for the highest-fee-per-byte transaction, if its fee is lower than its storage cost, a miner may still select it if all the other miners select it and he will eventually bear the storage cost of it.

B. Nash Equilibrium Analysis

We first define the Nash equilibrium in Definition 1.

Definition 1 (Nash Equilibrium): Given the fee-per-byte choices ρ , a strategy profile $(\mathcal{X}_m^{k,NE}, \forall m \in \mathcal{M})$ constitutes a Nash equilibrium in Game 1 if

$$v_m^k(\mathcal{X}_m^{k,NE}, \mathcal{X}_{-m}^{k,NE}, \rho) \geq v_m^k(\mathcal{X}_m^k, \mathcal{X}_{-m}^{k,NE}, \rho), \quad \forall \mathcal{X}_m^k \in \mathcal{B}_m^k, \forall m \in \mathcal{M}. \quad (6)$$

For the ease of presentation, we define the following function to calculate a strategy's net surplus (fees minus storage costs), i.e.,

$$\eta(\mathcal{X}_m^k) \triangleq \sum_{(n,j) \in \mathcal{X}_m^k} s_{n,j} (\rho_{n,j} - C_s). \quad (7)$$

Then, we summarize the Nash equilibrium as follows, where each miner adopts highest-net-surplus strategy.

Theorem 1 (Miners' Equilibrium in Stage III): The strategy profile $(\mathcal{X}_m^{k,NE}, \forall m \in \mathcal{M})$ constitutes a Nash equilibrium in round k , where

$$\mathcal{X}_m^{k,NE} = \arg \max_{\mathcal{X}_m^k \in \mathcal{B}_m^k} \eta(\mathcal{X}_m^k). \quad (8)$$

Due to the space limit, we leave the proofs of all mathematical results in the online appendix [34].

Corollary 1 reveals an interesting observation from Theorem 1.

Corollary 1: Each miner only accepts $tx_{n,j}$ if its fee-per-byte is higher than a miner's storage cost per byte, i.e., $\rho_{n,j} \geq C_s$. However, a miner's storage cost per byte C_s is insufficient to cover all miners' total storage cost per byte, i.e., MC_s .

Corollary 1 mathematically reveals the *negative externality in transaction selection* introduced in Section IV-A. Each miner only considers his own storage cost when selecting the transaction, without considering the negative impact on all other miners in the system. Hence even if the transaction fee can cover the storage cost of a single miner, it can be far from enough to cover the total storage cost of system. As miners accept insufficient-fee transactions, users may not pay enough transaction fees to cover all miners' total storage costs, causing the storage sustainability issue.

V. STAGE II: TRANSACTION GENERATION EQUILIBRIUM OF USERS

In this section, we will characterize how users generate transactions in Stage II. We first formulate users' transaction generation as a game in Section V-A, then we characterize the Nash equilibrium of the game in Section V-B.

A. Model of Users Transaction Generation

In Stage II, users set the *transaction generation rates* to maximize their own payoffs.

1) *Transaction Generation Event*: We denote the set of users as $\mathcal{N} = \{1, \dots, N\}$. For a user $n \in \mathcal{N}$, we model his transaction generation event as follows:

- generate one transaction with a size Φ with probability q , and
- generate two transactions with a size $\frac{\Phi}{2}$ with probability $1 - q$,

where probability $q \in [0, 1]$ and Φ is the block size limit. The setting is reasonable as the user can partition a large transaction into several small ones [35].

2) *Users' Strategies*: At each fee-per-byte choice $\rho_i \in \{\rho_1, \rho_2, \dots, \rho_I\}$, user n 's transaction generation events follow a Poisson process. The strategy of user n is to set the *transaction generation rates* $\lambda_n = (\lambda_{n_1}, \lambda_{n_2}, \dots, \lambda_{n_I})$, where λ_{n_i} ($i = 1, 2, \dots, I$) is the arrival rate of user n 's transaction generation event at ρ_i , which satisfies the following constraints:

$$\sum_{i=1}^I \lambda_{n_i} \leq \frac{\mu}{N}. \quad (9)$$

where μ is the system average block generation rate and each user's maximum transaction generation rate is $\frac{\mu}{N}$. Constraint (9) ensures that it is feasible to include all generated transactions from all users in the blockchain (if the miners choose to do so in Stage III).

3) *Transaction Waiting Time*: Here we define the waiting time of any transaction $tx_{n,j}$ as the time lapse between the generation time and the on-chain time.

- Generation time of transaction $tx_{n,j}$ is denoted as $t_{n,j}^{\text{gen}}$.
- On-chain time: When a miner selects transaction $tx_{n,j}$ and finds a block in round $k = 1, 2, \dots$, then round k ends, and transaction $tx_{n,j}$ is included in the blockchain. Thus, round k 's ending time $t^{\text{end}}(k)$ is the transaction on-chain time, i.e.,

$$t_{n,j}^{\text{on}} = \begin{cases} t^{\text{end}}(k), & \text{if } tx_{n,j} \text{ is included in block } k, \\ \infty, & \text{if } tx_{n,j} \text{ is not included in any block.} \end{cases} \quad (10)$$

- Waiting time is the difference between the on-chain time minus and the generation time, i.e., $w_{n,j} = t_{n,j}^{\text{on}} - t_{n,j}^{\text{gen}}$. Waiting time $w_{n,j}$ is a random variable as the block generation is stochastic. The rate of transactions entering the transaction pool affects the waiting time $w_{n,j}$, thus it is a function of all users transaction generation rates, i.e., $\lambda = (\lambda_n, \forall n \in \mathcal{N})$. We will compute the expectation of $w_{n,j}$ in Lemma 1 of Section V-B.

a) *Negative externality in transaction generation*: When user n generates a transaction and miners include it in the blockchain, other transactions in the transaction pool have to wait. Thus, user n 's transactions increase the average waiting time of all the other users' transactions. If a user maximizes his own payoff without considering the negative externality, all the other users will experience excessive waiting time, reducing the social welfare. This motivates us to propose the waiting tax to let each user internalize such a negative externality, increasing social welfare.

4) *User n 's Surplus Obtained From One Transaction $tx_{n,j}$* : User n 's surplus obtained from one transaction $tx_{n,j}$ depends on whether $tx_{n,j}$ is included in the blockchain.

- If $tx_{n,j}$ is included in the blockchain: The surplus depends on the on-chain utility from one transaction, transaction fee, waiting cost, and waiting tax.
- User n 's on-chain utility from $tx_{n,j}$: When $tx_{n,j}$ is included in the blockchain, user n will obtain utility of R_n . For example, a user gets a certain level of utility when successfully purchasing a kitty in Ethereum-based game cryptokitties. To model the users' heterogeneity of utilities, we consider two user types: with N_H high-utility users (type- H) and $N_L = N - N_H$ low-utility users (type- L). Notice that we will consider the case of more types in Section VII-B.¹² Thus, user n 's on-chain utility from one transaction is

$$R_n = \begin{cases} R_H, & \text{if user } n \text{ is type-}H, \\ R_L, & \text{if user } n \text{ is type-}L, \end{cases} \quad (11)$$

where $R_H \geq R_L$. Our model generalizes the homogeneous utility model in [11]–[13].

¹²The theoretical analysis of multiple user types is equivalent to solving the cubic equations with multiple variables, which is challenging. We will consider it in our future work.

- Transaction fee of $tx_{n,j}$: User n pays the transaction fee $f_{n,j} = s_{n,j}\rho_{n,j}$ to the miner who includes $tx_{n,j}$ in the blockchain. The size of transaction $tx_{n,j}$ satisfies $s_{n,j} \in \{\Phi, \frac{\Phi}{2}\}$. The fee-per-byte $\rho_{n,j}$ belongs to the protocol designer's assigned fee-per-byte choices, i.e., $\rho_{n,j} \in \{\rho_1, \rho_2, \dots, \rho_I\}$.
- Waiting cost of $tx_{n,j}$: The transaction waiting time $w_{n,j}(\lambda)$ imposes a cost to user n , which we assume to be a linear function with the impatience coefficient γ , i.e., $\gamma w_{n,j}(\lambda)$. A higher γ means users are less patient.
- Waiting tax of $tx_{n,j}$: Since user n 's transaction generation increases the expected transaction waiting time of any other user $l \neq n$, we introduce the waiting tax p_{nl} to let users internalize this negative externality. Specifically, user n pays user l the amount of $s_{n,j}p_{nl}$ to compensate the waiting costs n imposes. The payment is proportional to the transaction's size, as a larger-size transaction occupies more space in a block and more transactions need to wait for the future block. Depending on the types of users n and l , the possible waiting tax has four different values $\mathbf{P} = (P_{HH}, P_{HL}, P_{LH}, P_{LL})$:

$$p_{nl} = \begin{cases} P_{HH}, & \text{if both user } n \text{ and } l \text{ are type-}H, \\ P_{HL}, & \text{if user } n \text{ is type-}H \text{ and user } l \text{ is type-}L, \\ P_{LH}, & \text{if user } n \text{ is type-}L \text{ and user } l \text{ is type-}H, \\ P_{LL}, & \text{if both user } n \text{ and } l \text{ are type-}L. \end{cases} \quad (12)$$

To sum up, when $tx_{n,j}$ is included in the blockchain, user n 's surplus is

$$\theta_{n,j}(\lambda, \rho, \mathbf{P}) = R_n - s_{n,j}\rho_{n,j} - \gamma w_{n,j}(\lambda) - s_{n,j} \sum_{l \in \mathcal{N}, l \neq n} p_{nl}. \quad (13)$$

- If $tx_{n,j}$ is not included in the blockchain (i.e., not in any block), user n will not get the transaction on-chain utility R_n or pay the fee $f_{n,j}$. He also does not need to pay the waiting tax, as the transaction does not occupy any space in a block and does not increase the waiting time of other users. However, user n still experiences the (possibly infinite) waiting time to know that the transaction will not be included. In this case, user n 's surplus is

$$\theta_{n,j}(\lambda, \rho, \mathbf{P}) = -\gamma w_{n,j}(\lambda). \quad (14)$$

To simplify the formulation, we define the indicator function to indicate whether $tx_{n,j}$ is included in the blockchain as follows

$$\mathbf{1}(n, j) = \begin{cases} 1, & \text{if } tx_{n,j} \text{ is included in blockchain,} \\ 0, & \text{if } tx_{n,j} \text{ is not included in blockchain.} \end{cases} \quad (15)$$

Hence user n 's surplus obtained from $tx_{n,j}$ can be written as

$$\theta_{n,j}(\lambda, \rho, \mathbf{P}) = \mathbf{1}(n, j)(R_n - s_{n,j}\rho_{n,j} - s_{n,j} \sum_{l \in \mathcal{N}, l \neq n} p_{nl}) - \gamma w_{n,j}(\lambda). \quad (16)$$

5) *Users' Time-Average Payoff*: User n 's payoff is the summation of the surplus from all his transactions and the waiting tax paid to him by other users.¹³ For user n , we denote the number of all his generated transactions in time interval $[0, t]$ as $TX_n(t)$. His time-average payoff is

$$u_n(\boldsymbol{\lambda}, \boldsymbol{\rho}, \mathbf{P}) = \lim_{t \rightarrow \infty} \frac{\sum_{j=1}^{TX_n(t)} \mathbb{E}[\theta_{n,j}(\boldsymbol{\lambda}, \boldsymbol{\rho}, \mathbf{P})] + \sum_{l \in \mathcal{N}, l \neq n} \sum_{j=1}^{TX_l(t)} \mathbf{1}(l, j) s_{l,j} p_{ln}}{t}, \quad (17)$$

where $\mathbb{E}[\theta_{n,j}(\boldsymbol{\lambda}, \boldsymbol{\rho}, \mathbf{P})]$ is user n 's expected surplus from transaction $tx_{n,j}$. The expectation is taken in terms of the random variable waiting time $w_{n,j}$.

6) *Game Formulation*: We formulate users' transaction generation as a non-cooperative game, where users set transaction generation rates simultaneously to maximize their own payoffs.¹⁴

Game 2 (Stage II: Transaction Generation Game):

In Stage II, Transaction Generation Game is a tuple $\Omega = (\mathcal{N}, \Lambda, \mathbf{U})$ defined by:

- *Players*: The set of users \mathcal{N} .
- *Strategies*: Each user n sets transaction generation rate λ_n , where the strategy space is $\Lambda_n = \{\lambda_n = (\lambda_{n_1}, \lambda_{n_2}, \dots, \lambda_{n_I}) | \lambda_n \text{ satisfies (9)}\}$. The strategy profiles of all the users is $\boldsymbol{\lambda} = (\lambda_n, \forall n \in \mathcal{N})$ and the set of all feasible strategy profiles is $\Lambda = \Lambda_1 \times \dots \times \Lambda_N$.
- *Payoffs*: The vector $\mathbf{U} = (u_n, \forall n \in \mathcal{N})$ contains all users' payoffs as defined in (17).

In Game 2, each user faces a *tradeoff* between paying a high fee and suffering a high transaction waiting time. Since miners prefer to include transactions with high fees, user n will experience a lower average waiting time by generating more high-fee transactions. However, if paying a high fee is too costly, user n would be better off by generating more low-fee transactions and bearing a higher average waiting time.

B. Nash Equilibrium Analysis

Based on the equilibrium of Stage III, we analyze the equilibrium of Stage II in this subsection. We first compute the transaction waiting time, then we present the users' equilibrium in Stage II.

1) *Transaction Waiting Time*: According to miners' equilibrium strategies in Stage III, the process of transaction arriving (i.e., users' transaction generation event) and leaving (i.e., miners' block generation) is an M/M/1 queue [36], where transactions with higher fee-per-byte have priority over transactions with lower fee-per-byte. We summarize user n 's time-average transaction waiting time in following Lemma 1.

Lemma 1 (Users' Transaction Waiting Time): The time-average transaction waiting time of each user $\forall n \in \mathcal{N}$ is in (18), as shown at the bottom of the next page.

¹³When user n 's transaction is not included in the blockchain, other users with on-chain transactions still pay the waiting tax to compensate user n 's waiting cost, because other users' on-chain transactions will delay the process that user n finds out that miners do not select his transaction.

¹⁴Here we assume that each user does not consider the influence of his strategic decision on other users (i.e., each user is a price taker). This assumption holds for a blockchain system with many users.

TABLE II
TYPES B AND S AND CORRESPONDING TYPES H AND L

If $h_H \geq h_L$	$B = H, \mathcal{N}_B = \mathcal{N}_H$	$S = L, \mathcal{N}_S = \mathcal{N}_L$
If $h_H < h_L$	$S = H, \mathcal{N}_S = \mathcal{N}_H$	$B = L, \mathcal{N}_B = \mathcal{N}_L$

Equation (18a) corresponds to the case where miners (eventually) choose to include the transaction as the fee-per-byte values of all user n 's transactions are higher than C_s . Equation (18b) corresponds to the case where the transaction waiting time is infinity, as no miner chooses to include the transaction with fee-per-byte strictly lower than C_s .

2) *Users' Equilibrium in Stage II*: Here we characterize the users' equilibrium strategies. Similar to prior blockchain literature [1], [12], we consider the symmetric Nash equilibrium (SNE) where the same type of users adopt the same strategy.

For the ease of exposition, we first define some terminology related to the users' equilibrium.¹⁵

Definition 2 (Stage II Equilibrium): At a ρ_i -SNE ($i = 1, 2, \dots, I$), all users only generate transactions with the fee-per-byte ρ_i .

At an equilibrium, each user n 's *net transaction utility* h_n plays an important role in his transaction generation rate, which defined as follows

$$h_n = \begin{cases} h_H = (2 - q)R_H - \Phi[(N_H - 1)P_{HH} + N_L P_{HL}], & \text{if } n \in \mathcal{N}_H, \\ h_L = (2 - q)R_L - \Phi[N_H P_{LH} + (N_L - 1)P_{LL}], & \text{if } n \in \mathcal{N}_L, \end{cases} \quad (19)$$

where \mathcal{N}_H and \mathcal{N}_L are the set of types H and L users, respectively.

Notice that h_H may not be larger than h_L due to the waiting time tax rate vector $(P_{HH}, P_{HL}, P_{LH}, P_{LL})$. We define type- B as the bigger net transaction utility user type (i.e., $B = \arg \max_{i \in \{L, H\}} h_i$) and type- S as the smaller net transaction utility user type (i.e., $S = \arg \min_{i \in \{L, H\}} h_i$). We illustrate the connections between types B and S as well as types H and L in Table II.

Next, we characterize the types B and S users' equilibrium strategies at the ρ_i -SNE ($i = 1, 2, \dots, I$) in Proposition 1. To facilitate the analysis, we denote \mathbf{e}_i as the I -dimension vector with all entries being zero except i -th being 1.

Proposition 1 (Stage II Equilibrium Strategy): The following strategy profile $(\boldsymbol{\lambda}_n^{\text{NE}} = \pi_B(h_B, h_S, \rho_i)\mathbf{e}_i, \forall n \in \mathcal{N}_B, \boldsymbol{\lambda}_l^{\text{NE}} = \pi_S(h_B, h_S, \rho_i)\mathbf{e}_i, \forall l \in \mathcal{N}_S)$ constitutes a ρ_i -SNE, where $\pi_B(h_B, h_S, \rho_i)$, $\pi_S(h_B, h_S, \rho_i)$, and the intermediate variables $A_1(h_B, \rho_i)$ and $A_2(h_B, h_S, \rho_i)$ are shown in (20)-(23), at the bottom of the next page, respectively.

Here we explain the intuition of the ρ_i -SNE. When both net transaction utilities h_B and h_S are small (i.e., conditions in (20a) and (21a)), users do not generate transactions. When h_B is large but h_S is small (i.e., conditions in (20b) and (21b)),

¹⁵There can be other SNE but we pay attention to the Pareto-dominant one, where each user achieves no smaller payoff compared to other possible SNEs [12], [37].

only type- B users generate transactions. When both h_B and h_S are large (i.e., conditions in (20c) and (21c)), all users generate transactions.

We define Δ function in (24), as shown at the bottom of the page, to characterize the boundary of equilibrium. Based on the equilibrium characterized in Proposition 1, we summarize users' equilibria in Theorem 2.

Theorem 2 (Users' Equilibria in Stage II):

- If $\Delta(h_B, h_S, \rho_2) > \Phi\rho_1$, then there exists a ρ_1 -SNE.
- If $\Delta(h_B, h_S, \rho_i) \leq \Phi\rho_{i-1}$, $\Delta(h_B, h_S, \rho_{i+1}) > \Phi\rho_i$, and $i = 2, 3, \dots, I-1$, then there exist a ρ_i -SNE.
- If $\Delta(h_B, h_S, \rho_I) \leq \Phi\rho_{I-1}$, then there exist a ρ_I -SNE.

We illustrate the SNE of Stage II with a three-fee-choice example. Fig. 4 shows the SNE against ρ_1 and ρ_2 with

$$\lim_{t \rightarrow \infty} \frac{\sum_{j=1}^{TX_n(t)} \mathbb{E}[w_{n,j}(\boldsymbol{\lambda})]}{t} = \begin{cases} \frac{(2-q)\lambda_{n_1}}{\mu - \sum_{l \in \mathcal{N}} \lambda_{l_1}} + \sum_{i=2}^I \frac{(2-q)\mu\lambda_{n_i}}{(\mu - \sum_{x=1}^{i-1} \sum_{l \in \mathcal{N}} \lambda_{l_x})(\mu - \sum_{x=1}^i \sum_{l \in \mathcal{N}} \lambda_{l_x})}, & \text{if } \forall i = 1, 2, \dots, I, \rho_i \geq C_s \text{ or} \\ \left(\rho_i < C_s \text{ and } \lambda_{n_i} = 0 \right), & (18a) \\ \infty, & \text{otherwise.} \end{cases} \quad (18b)$$

$$\pi_B(h_B, h_S, \rho_i) = \begin{cases} 0, & \text{if } h_B \leq \Phi\rho_i + \frac{(2-q)\gamma}{\mu}, \\ A_1(h_B, \rho_i), & \text{if } h_B > \Phi\rho_i + \frac{(2-q)\gamma}{\mu} \text{ and} \\ & h_S \leq \Phi\rho_i + \frac{(2-q)\gamma}{\mu - N_B A_1(h_B, \rho_i)}, \\ \min\left\{ \frac{\mu}{N_B + N_S}, \right. & \\ \left. \frac{[\mu - A_2(h_B, h_S, \rho_i)][(h_B - \Phi\rho_i)A_2(h_B, h_S, \rho_i) - (2-q)\gamma]}{N_B[(h_B - \Phi\rho_i)A_2(h_B, h_S, \rho_i) - (2-q)\gamma] + N_S[(h_S - \Phi\rho_i)A_2(h_B, h_S, \rho_i) - (2-q)\gamma]} \right\}, & \\ & \text{if } h_S > \Phi\rho_i + \frac{(2-q)\gamma}{\mu - N_B A_1(h_B, \rho_i)}. \end{cases} \quad (20a) \quad (20b) \quad (20c)$$

$$\pi_S(h_B, h_S, \rho_i) = \begin{cases} 0, & \text{if } h_B \leq \Phi\rho_i + \frac{(2-q)\gamma}{\mu}, \\ 0, & \text{if } h_B > \Phi\rho_i + \frac{(2-q)\gamma}{\mu} \text{ and} \\ & h_S \leq \Phi\rho_i + \frac{(2-q)\gamma}{\mu - N_B A_1(h_B, \rho_i)}, \\ \frac{\mu - N_B \pi_B(h_B, h_S, \rho_i)}{N_S} & \\ - \frac{(2-q)\gamma(N_S - 1) + \sqrt{(2-q)^2\gamma^2(N_S - 1)^2 + 4N_S(h_S - \Phi\rho_i)(2-q)\gamma[\mu - N_B \pi_B(h_B, h_S, \rho_i)]}}{2(h_S - \Phi\rho_i)N_S^2}, & \\ & \text{if } h_S > \Phi\rho_i + \frac{(2-q)\gamma}{\mu - N_B A_1(h_B, \rho_i)}. \end{cases} \quad (21a) \quad (21b) \quad (21c)$$

$$A_1(h_B, \rho_i) = \min\left\{ \frac{\mu}{N_B} - \frac{(2-q)\gamma(N_B - 1) + \sqrt{(2-q)^2\gamma^2(N_B - 1)^2 + 4(2-q)\gamma\mu N_B(h_B - \Phi\rho_i)}}{2N_B^2(h_B - \Phi\rho_i)}, \frac{\mu}{N_B + N_S} \right\}. \quad (22)$$

$$A_2(h_B, h_S, \rho_i) = \frac{(2-q)\gamma(N_B + N_S - 1) + \sqrt{(2-q)^2\gamma^2(N_B + N_S - 1)^2 + 4(2-q)\gamma\mu[N_B(h_B - \Phi\rho_i) + N_S(h_S - \Phi\rho_i)]}}{2[N_B(h_B - \Phi\rho_i) + N_S(h_S - \Phi\rho_i)]}. \quad (23)$$

$$\Delta(h_B, h_S, \rho_i) = \max\left\{ \Phi\rho_i + \frac{(2-q)\gamma[(N_B - 1)\pi_B(h_B, h_S, \rho_i) + N_S\pi_S(h_B, h_S, \rho_i)]}{\mu[\mu - N_B\pi_B(h_B, h_S, \rho_i) - N_S\pi_S(h_B, h_S, \rho_i)]}, \right. \\ \left. h_S - \frac{(2-q)\gamma}{\mu} - \frac{(2-q)\gamma\pi_S(h_B, h_S, \rho_i)[2\mu - N_B\pi_B(h_B, h_S, \rho_i) - N_S\pi_S(h_B, h_S, \rho_i)]}{\mu[\mu - N_B\pi_B(h_B, h_S, \rho_i) - N_S\pi_S(h_B, h_S, \rho_i)]^2} \right\}. \quad (24)$$

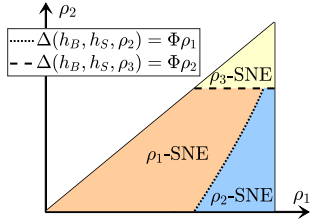


Fig. 4. A three-fee-choice example of SNEs VS ρ_1 and ρ_2 .

fixed ρ_3 , reflecting users' tradeoff between paying the low fee and bearing low waiting time. When fee choices ρ_1 and ρ_2 are small, then $\Delta(h_B, h_S, \rho_2) > \Phi\rho_1$ and a ρ_1 -SNE exists. In other words, all users choose the highest fee-per-byte ρ_1 , because ρ_1 is not high enough and hence the consideration of low waiting time dominates the consideration of paying the low fee (i.e., choosing ρ_2 or ρ_3). As ρ_1 increases such that $\Delta(h_B, h_S, \rho_2) \leq \Phi\rho_1$ and $\Delta(h_B, h_S, \rho_3) > \Phi\rho_2$, the ρ_2 -SNE emerges, where all users choose the medium fee-per-byte ρ_2 . This is because when ρ_1 is high compared to ρ_2 , the consideration of paying the relatively low fee (i.e., choosing ρ_2) dominates the consideration of low waiting time. Moreover, the fee choice ρ_2 is not high compared with ρ_3 and hence the consideration of low waiting time dominates the consideration of paying the relatively low fee (i.e., choosing ρ_3). As both ρ_1 and ρ_2 increase such that $\Delta(h_B, h_S, \rho_3) \leq \Phi\rho_2$, the ρ_3 -SNE emerges, where all users choose the lowest fee-per-byte ρ_3 .

VI. STAGE I: OPTIMAL FWT MECHANISM OF PROTOCOL DESIGNER

In this section, we will characterize the protocol designer's optimal FWT mechanism in Stage I. We first formulate the FWT mechanism design as an optimization problem in Section VI-A, then we compute its optimal solution in Section VI-B.

A. FWT Mechanism Design of Protocol Designer

In Stage I, the protocol designer optimizes the FWT mechanism to encourage users to pay sufficient fees while maximizing the social welfare.

1) *Decision Variables*: The protocol designer's decision variables are the fee-per-byte choices $\rho = (\rho_1, \rho_2, \dots, \rho_I)$ (with $\rho_1 > \rho_2 > \dots > \rho_I \geq 0$) and the waiting tax rate vector $\mathbf{P} = (P_{HH}, P_{HL}, P_{LH}, P_{LL})$. The fee-per-byte choices encourages users to pay sufficient transaction fees to mitigate the negative externality in transaction selection in Stage III. The waiting tax rate vector let each user internalize the waiting costs imposed on others, dealing with the negative externality in transaction generation in Stage II.

2) *Sufficient Fee Condition*: For any user n with a positive transaction generation rate (i.e., $\sum_{i=1}^I \lambda_{n_i} > 0$), the FWT mechanism aims at inducing an average fee-per-byte value that can cover the total storage cost per byte of all miners, i.e.,

$$\rho_n^{\text{avg}} = \frac{\sum_{i=1}^I \lambda_{n_i} \rho_i}{\sum_{i=1}^I \lambda_{n_i}} \geq MC_s, \quad \forall n \in \{l \in \mathcal{N} \mid \sum_{i=1}^I \lambda_{l_i} > 0\}. \quad (25)$$

3) *Social Welfare*: The social welfare equals the sum of users' and miners' time-average payoffs.

Based on miner m 's payoff v_m^k in round k in (5), the miner m 's time-average payoffs as

$$v_m(\mathcal{X}, \rho) = \lim_{t \rightarrow \infty} \frac{\sum_{k=1}^{\text{Round}(t)} v_m^k(\mathcal{X}_m^k, \mathcal{X}_{-m}^k, \rho)}{t}, \quad (26)$$

where $\mathcal{X} = (\mathcal{X}_m^k, \forall m, \forall k)$ is the strategy profile of all miners in Stage III and $\text{Round}(t)$ is the number of rounds completed in time interval $[0, t]$.

The social welfare is as follows

$$sw(\rho, \mathbf{P}, \lambda, \mathcal{X}) = \sum_{n \in \mathcal{N}} u_n(\lambda, \rho, \mathbf{P}) + \sum_{m \in \mathcal{M}} v_m(\mathcal{X}, \rho). \quad (27)$$

4) *FWT Mechanism Design*: We formulate the FWT mechanism design problem in (28), which aims at maximizing the social welfare subject to sufficient transaction fee covering the storage cost.

$$\max sw(\rho, \mathbf{P}, \lambda, \mathcal{X})$$

$$\text{s.t. (25), } \rho_1 > \rho_2 > \dots > \rho_I \geq 0,$$

$$\text{var. } \rho = (\rho_1, \rho_2, \dots, \rho_I), \quad \mathbf{P} = (P_{HH}, P_{HL}, P_{LH}, P_{LL}).^{16} \quad (28)$$

B. Optimal Solution of FWT Mechanism Design

In this subsection, we will solve Problem (28) and discuss the property of its optimal solution.

1) *Optimal Solution of FWT Mechanism Design Problem*: The optimal solution to Problem (28) is as follows.

Theorem 3 (Optimal Solution of FWT Mechanism Design Problem): The optimal FWT mechanism corresponds to the optimal solution of Problem (28) as follows:

- If $R_H \leq \frac{M\Phi C_s}{2-q} + \frac{\gamma}{\mu}$, then
 - the fee-per-byte choices are $(\rho_i^*, \forall i = 1, 2, \dots, I)$ where $\rho_i^* = MC_s + \frac{(I-i)(2-q)\gamma}{\Phi\mu}$,
 - the waiting tax rate vector $(P_{HH}^*, P_{HL}^*, P_{LH}^*, P_{LL}^*) \in \mathbb{R}^4$ satisfies the following conditions:

$$\begin{cases} (N_H - 1)P_{HH}^* + N_L P_{HL}^* = 0, \\ N_H P_{LH}^* + (N_L - 1)P_{LL}^* = 0. \end{cases} \quad (29)$$

- If $R_H > \frac{M\Phi C_s}{2-q} + \frac{\gamma}{\mu}$, then
 - the fee-per-byte choices $(\rho_i^*, \forall i = 1, 2, \dots, I)$ where ρ_i^* satisfies following condition:

$$\rho_i^* = \frac{(I-i)(2-q)(\mu R_H - \gamma)}{\Phi\mu} - (I-i-1)MC_s. \quad (30)$$

- the waiting tax rate vector $(P_{HH}^*, P_{HL}^*, P_{LH}^*, P_{LL}^*) \in \mathbb{R}^4$ satisfies the conditions in (31), as shown at the bottom of the next page, and the intermediate variables g_1 and g_2 are shown in (32) and (33), as shown at the bottom of the next page, respectively.

¹⁶The waiting tax can be negative, which motivates users to generate transactions by compensating them. This makes the mechanism more flexible.

The insights of Theorem 3 are as follows: If $R_H \leq \frac{M\Phi C_s}{2-q} + \frac{\gamma}{\mu}$, both types of users have low transaction on-chain utilities, which are insufficient to cover a transaction's total storage costs plus waiting costs. Thus the optimal FWT mechanism prevents both types of users from generating any transactions, such that the sum of any user's waiting tax payment is 0.

If $R_H > \frac{M\Phi C_s}{2-q} + \frac{\gamma}{\mu}$, type- H users have high transaction on-chain utility. Thus the optimal FWT mechanism allows users to generate transactions. The protocol designer sets the lowest fee-per-byte $\rho_I^* = MC_s$ to guarantee the sufficient fee condition. Since users will generate transactions at SNE, the sum of a type- H user's (or type- L user's, respectively) waiting tax payment is non-zero as shown in (31a) (or (31b), respectively).

2) *Property of Optimal FWT Mechanism*: To characterize the property of the optimal FWT mechanism, we first establish the benchmark of *unconstrained social optimum*, which is the maximum social welfare that can be achieved without considering the sufficient fee condition (25), i.e., the maximum value of the objective function of Problem (34).

$$\begin{aligned} \max \quad & sw(\boldsymbol{\rho}, \mathbf{P}, \boldsymbol{\lambda}, \boldsymbol{\mathcal{X}}) \\ \text{s.t.} \quad & \rho_1 > \rho_2 > \dots > \rho_I \geq 0, \\ \text{var.} \quad & \boldsymbol{\rho} = (\rho_1, \rho_2, \dots, \rho_I), \quad \mathbf{P} = (P_{HH}, P_{HL}, P_{LH}, P_{LL}). \end{aligned} \quad (34)$$

Then we characterize the property of the optimal FWT mechanism in Proposition 2.

Proposition 2 (Guarantee on Unconstrained Social Optimum): The social welfare of the optimal FWT mechanism equals the unconstrained social optimum.

Proposition 2 shows through our careful design of the FWT mechanism, imposing the sufficient condition of (25) does not lead to any loss of social welfare.

VII. PERFORMANCE EVALUATIONS

So far, we have studied the FWT mechanism design under *two types of users*. In this section, we will relax such an assumption and conduct some numerical analysis under *four types of users*. We will evaluate the performance of the optimal FWT mechanism ("FWT") by comparing it with the existing blockchain protocol ("Existing", i.e., the fee mechanism currently deployed in Bitcoin and Ethereum). We study the impact

TABLE III
BLOCKCHAIN PARAMETERS

Blockchain throughput [5]	$\mu = 15$
The number of miners [5]	$M = 10^4$
Transaction size (bytes) [38]	$\Phi = 150$
Number of fee-per-byte choices	$I = 10$
Storage cost per byte (USD/byte) [39]	$C_s = 5 \times 10^{-10}$
Ratio of transaction on-chain utilities	$R^1 : R^2 : R^3 : R^4$ $= 4 : 3 : 2 : 1$

of various system parameters on the social welfare, fee-per-byte payment, and waiting time on both schemes.

A. Setup of Numerical Analysis

We summarize the parameters of the numerical analysis in Table III, where we set the blockchain-related parameters based on Ethereum, and R^1 to R^4 represent the transaction on-chain utilities for four types of users, respectively.

To solve the new model, we discretize each user's transaction generation rate at each fee-per-byte choice into ten equal-interval levels, and each user's strategy is to choose one level. We still focus on the symmetric Nash equilibrium where each type of user adopts the same strategy. Then we apply Zermelo's algorithm [40] to derive the subgame perfect Nash equilibrium of the model.

For the existing protocol of blockchain, the lowest and highest fee-per-byte choices correspond to 20% to 80% percentile of Ethereum's fee-per-byte (i.e., also referred to as gas price) on Oct. 2020.

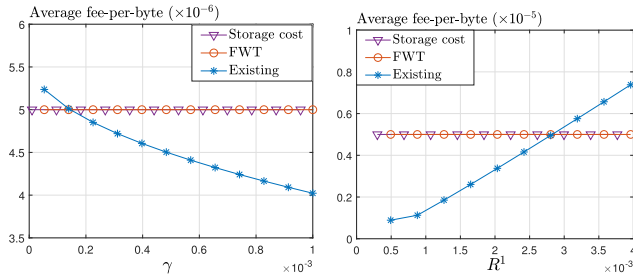
B. Fee-per-Byte and Social Welfare

In this subsection, we study how users' parameters (impatience level and transaction on-chain utility) affect both schemes in terms of fee-per-byte and social welfare. For users' parameters, we set the user's highest transaction on-chain utility as $R^1 \in [3 \times 10^{-4}, 4 \times 10^{-3}]$ and the user's impatience level as $\gamma \in [10^{-5}, 10^{-3}]$. Under such a setting, the daily number of transactions of the existing protocol is between 0.95 to 1.25 millions. This range aligns well with the daily number of transactions in Oct. 2020 that is between 0.96 to 1.25 millions [41].

$$\begin{cases} (N_H - 1)P_{HH}^* + N_L P_{HL}^* = \frac{(2-q)R_H}{\Phi} - \rho_I^* - \frac{(2-q)\gamma[\mu - (N_H - 1)g_1 - N_L g_2]}{\Phi(\mu - N_H g_1 - N_L g_2)^2}, & (31a) \\ N_H P_{LH}^* + (N_L - 1)P_{LL}^* = \frac{(2-q)R_L}{\Phi} - \rho_I^* - \frac{(2-q)\gamma[\mu - N_H g_1 - (N_L - 1)g_2]}{\Phi(\mu - N_H g_1 - N_L g_2)^2}. & (31b) \end{cases}$$

$$g_1 = \min \left\{ \frac{\mu}{N_H + N_L}, \frac{1}{N_H} \left[\mu - \sqrt{\frac{(2-q)\gamma\mu}{(2-q)R_H - M\Phi C_s}} \right] \right\}. \quad (32)$$

$$g_2 = \begin{cases} 0, & \text{if } R_L \leq \frac{M\Phi C_s}{2-q} + \frac{\gamma(N_H + N_L)^2}{N_L^2 \mu}, \\ \frac{\mu}{N_H + N_L} - \frac{1}{N_L} \sqrt{\frac{(2-q)\gamma\mu}{(2-q)R_L - M\Phi C_s}}, & \text{otherwise.} \end{cases} \quad (33)$$



(a) Users' average fee-per-byte vs. impatience level γ for $R^1 = 2 \times 10^{-3}$. (b) Users' average fee-per-byte vs. transaction on-chain utility R^1 for $\gamma = 5 \times 10^{-5}$.

Fig. 5. Impact of users' parameters on average fee-per-byte ρ_n^{avg} .

1) *Fee-per-Byte*: Fig. 5 (a) and (b) illustrate the impact of impatience level γ and transaction on-chain utility R^1 on the average fee-per-byte ρ_n^{avg} , respectively.

- Fig. 5(a): Storage cost in the figure corresponds to all miners' total storage cost per byte (i.e., MC_s) and serves as a benchmark for the other two curves. Under the optimal FWT mechanism (FWT in the figure), we observe that the average fee-per-byte can cover the total storage cost, satisfying the sufficient fee condition. However, under the existing protocol (Existing in the figure), the sufficient fee condition does not hold when $\gamma \geq 1.45 \times 10^{-3}$. Moreover, we make an interesting observation as follows:

Observation 1: As the impatience level γ increases, users pay lower average fee-per-byte ρ_n^{avg} in the existing protocol.

We explain the reason behind Observation 1 as follows. When the users become more impatient, they generate fewer transactions to reduce waiting costs. Fewer transactions lead to lower incentives to pay high transaction fees and compete for short waiting time.

- Fig. 5(b): The correspondences of curves and the legend are the same as Fig. 5(a). Under the optimal FWT mechanism, the system always satisfies the sufficient fee condition. Under the existing protocol, users increase the average fee-per-byte with R^1 and the sufficient fee condition only holds when $R^1 \geq 2.8 \times 10^{-3}$. As the user's on-chain utility R^1 increases, users can afford higher transaction fees, such that they pay high transaction fees to reduce waiting time.

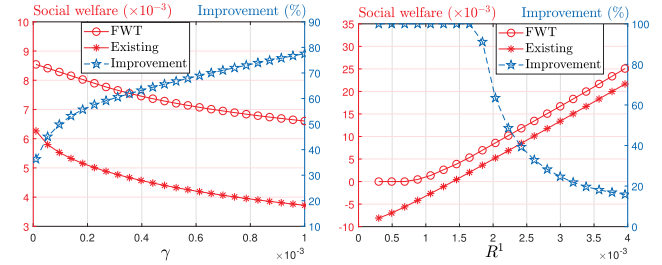
From Fig. 5, we make the following observation:

Observation 2: The optimal FWT mechanism can satisfy the sufficient fee condition under four types of users.

Although we propose the FWT mechanism based on the two-type-user model, Fig. 5 demonstrates that the optimal FWT mechanism still ensures the sufficient fee condition and outperforms the existing protocol.

2) *Social Welfare*: Fig. 6 (a) and (b) illustrate the impact of impatience level γ and transaction on-chain utility R^1 on the social welfare sw , respectively.

- Fig. 6(a): On the left axis, two red curves plot the social welfares of the optimal FWT mechanism (FWT in the figure) and the existing protocol (Existing in the figure). We notice that the social welfares of both



(a) Social welfare and improvement vs. impatience level γ for $R^1 = 2 \times 10^{-3}$. (b) Social welfare and improvement vs. transaction on-chain utility R^1 for $\gamma = 5 \times 10^{-5}$.

Fig. 6. Impact of users' parameters on social welfare sw .

schemes decrease in γ , due to the increased waiting cost with the increasing impatience level γ . On the right axis, the blue curve marked in stars plots the optimal FWT mechanism's social welfare improvement over the existing protocol (Improvement in the figure) and average improvement is 51.43%. Such an improvement is due to the optimal FWT mechanism addresses the negative externality in transaction generation and reduces the transaction waiting time.

- Fig. 6(b): The correspondences of curves and the axes are similar as Fig. 6(a). On the left axis, we observe that the social welfares of both schemes increase in R^1 , due to the increased on-chain utility. On the right axis, the social welfare improvement decreases in R^1 with an average value of 61.04%. The reason for such a decrease in improvement is as follows. In the existing protocol, the average fee-per-byte increases with R^1 (i.e., Fig 5(b)), preventing users from generating too many transactions and causing excessive waiting costs on others.

An observation from Fig. 6 is as follows:

Observation 3: The optimal FWT mechanism achieves an average social welfare improvement of 51.43% or more compared with the existing protocol.

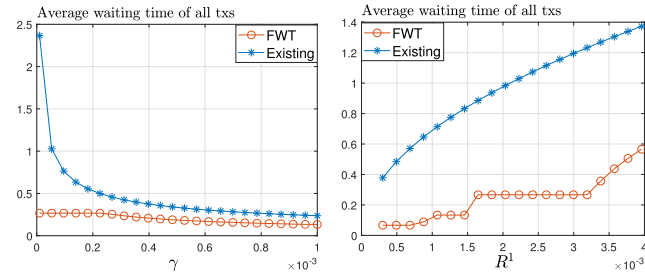
Although we propose the FWT mechanism based on the two-type-user model, Observations 2 and 3 demonstrate that the optimal FWT mechanism dominates the existing protocol in both social welfare and the satisfaction of the sufficient fee condition. As the existing protocol is a feasible choice of FWT mechanism, the FWT mechanism's domination has nothing to do whether it is two or four user types.

C. Transaction's Waiting Time

In this subsection, we study how parameters affect both schemes in terms of transaction's waiting time, which reflects users' benefits received from the optimal FWT mechanism.

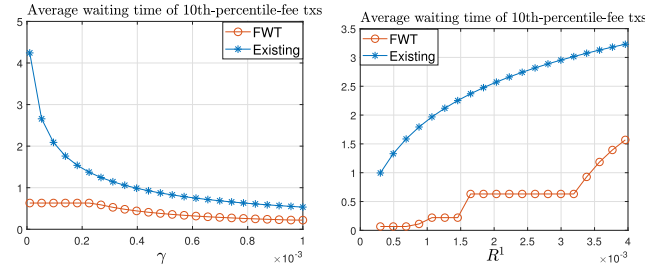
Fig. 7 (a) and (b) illustrate the impact of impatience level γ and transaction on-chain utility R^1 on all transactions' average waiting time, respectively.

- Fig. 7(a): We observe that the average waiting time of both schemes decrease in γ , as users generate fewer transactions due to the increased waiting cost.
- Fig. 7(b): The average waiting time of both schemes increase in R^1 , as users propose more transactions due to the increased transaction on-chain utility.



(a) All transactions' average waiting time vs. impatience level γ for $R^1 = 2 \times 10^{-3}$. (b) All transactions' average waiting time vs. transaction on-chain utility R^1 for $\gamma = 5 \times 10^{-5}$.

Fig. 7. Impact of users' parameters on all transactions' average waiting time.



(a) 10th-percentile-fee transactions' average waiting time vs. impatience level γ for $R^1 = 2 \times 10^{-3}$. (b) 10th-percentile-fee transactions' average waiting time vs. transaction on-chain utility R^1 for $\gamma = 5 \times 10^{-5}$.

Fig. 8. Impact of users' parameters on 10th-percentile-fee transactions' average waiting time.

Moreover, we make the following observation from Fig. 7:

Observation 4: The optimal FWT mechanism reduces all transactions' average waiting time by an average of 61.56%.

Observation 4 demonstrates that the users bear significantly lower waiting time in the optimal FWT mechanism, as the waiting tax makes users more conservative in generating transactions.

In Fig. 8, we plot the average waiting time of 10th-percentile-fee transactions (i.e., 90% transactions have higher fee-per-byte than the transactions we consider). We make the following observation.

Observation 5: The optimal FWT mechanism reduces 10th-percentile-fee transactions' average waiting time by an average of 68.49%.

We notice that the waiting time reduction for a 10th-percentile-fee transaction (i.e., 68.49%) is higher than the average waiting time reduction for all transactions (i.e., 61.56%). The reason is as follows. The optimal FWT mechanism can reduce the number of transactions because of the waiting tax. For a top-fee transaction, other transactions do not delay its waiting time. Thus, reducing the number of transactions in the optimal FWT mechanism does not affect the top-fee transaction's waiting time. On the other hand, the optimal FWT mechanism provides more benefit to the low-fee transactions, as the mechanism significantly reduces the number of high-priority transactions for low-fee ones.

VIII. CONCLUSION

In this paper, we proposed an FWT mechanism to mitigate the issue of insufficient storage fee in blockchain. We noticed

two types of negative externalities in the system: a miner's transaction selection imposes storage costs on other miners and a user's transaction generation imposes waiting costs on other users. Motivated by the negative externalities, the FWT mechanism offers fee choices to users and imposes waiting tax on them. We modeled the interactions among the protocol designer, users, and miners as a three-stage Stackelberg game. We derived the subgame perfect equilibrium of the game in closed-form. Based on the equilibrium, we found that miners neglecting the negative externality in transaction selection cause the insufficient fee issue in the existing blockchain. We showed that the optimal FWT mechanism achieves the unconstrained social optimum and guarantees that users pay sufficient transaction fees for storage costs. Ethereum-based numerical results showed that the optimal FWT mechanism guarantees sufficient transaction fees and achieves an average social welfare improvement of 51.43% or more over the existing protocol. Moreover, the optimal FWT mechanism reduces the average waiting time of low-fee transactions and all transactions by 68.49% and 61.56%, respectively.

In the future work, we will further consider users with multi-dimensional heterogeneity, where both their transaction on-chain utility and impatience level are heterogeneous. We also plan to study the problem under the dynamic setting, where users enter and leave the system following a stochastic process.

REFERENCES

- [1] 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.
- [2] *Ethereum Archive Node*. Accessed: Nov. 2018. [Online]. Available: <https://medium.com/quiknode/welcoming-the-newest-member-of-the-quiknode-family-the-eth-archive-node-ac66201e0793>
- [3] W. Wang *et al.*, "A survey on consensus mechanisms and mining strategy management in blockchain networks," *IEEE Access*, vol. 7, pp. 22328–22370, 2019.
- [4] *Ethereum Archive Node Size*. Accessed: Mar. 2022. [Online]. Available: <https://etherscan.io/chartsync/chainarchive>
- [5] *Ethereum Transaction Fee*. Accessed: Jul. 2020. [Online]. Available: <https://studio.glassnode.com/compare?a=ETH&c=usd&e=&m=fees.VolumeSum&mAvg=0&mMedian=0&mScl=lin&miner=&resolution=1month&s=1440896401&sameAxis=true&u=1601510400&z=zoom>
- [6] *Number of Full Node*. Accessed: Mar. 2022. [Online]. Available: <https://www.ethernodes.org/history>
- [7] S. Nakamoto. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. [Online]. Available: <https://bitcoin.org/bitcoin.pdf>
- [8] *Node Decline*. Accessed: Dec. 2020. [Online]. Available: <https://thebitcoin.pub/t/ethereum-essentials-node-nuances/52785>
- [9] *Bitcoinwiki*. Accessed: Jun. 2020. [Online]. Available: <https://en.bitcoin.it/wiki/Weaknesses>
- [10] *Storage Discussion*. Accessed: Dec. 2018. [Online]. Available: <https://ethereum-magicians.org/search?q=state%20rent>
- [11] G. Huberman, J. D. Leshno, and C. Moallemi, "Monopoly without a monopolist: An economic analysis of the bitcoin payment system," *Rev. Econ. Stud.*, vol. 88, no. 6, pp. 3011–3040, Nov. 2021.
- [12] 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.
- [13] J. Li, Y. Yuan, S. Wang, and F.-Y. Wang, "Transaction queuing game in bitcoin blockchain," in *Proc. Intell. Vehicles Symp.*, 2018, pp. 114–119.
- [14] P. R. Rizun, "A transaction fee market exists without a block size limit," Bitcoin Unlimited, Tech. Rep., 2015. [Online]. Available: <https://www.bitcoinunlimited.info/> and <https://www.bitcoinunlimited.info/resources/feemarket.pdf>
- [15] R. Zhang and B. Preneel, "On the necessity of a prescribed block validity consensus: Analyzing bitcoin unlimited mining protocol," in *Proc. ACM Int. Conf. Emerg. Netw. Exp. Technol.*, 2017, pp. 108–119.
- [16] *EIP-1559*. Accessed: Apr. 2019. [Online]. Available: <https://github.com/ethereum/EIPs/blob/master/EIPS/eip-1559.md>

- [17] 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.
- [18] 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.
- [19] S. Basu, D. Easley, M. O'Hara, and E. Sirer, "Towards a functional fee market for cryptocurrencies," Jan. 2019, *arXiv:1901.06830*. [Online]. Available: <https://arxiv.org/pdf/1901.06830.pdf>
- [20] R. Lavi, O. Sattath, and A. Zohar, "Redesigning bitcoin's fee market," in *Proc. World Wide Web Conf.*, 2019, pp. 2950–2956.
- [21] A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder, *Bitcoin Cryptocurrency Technologies: A Comprehensive Introduction*. Princeton, NJ, USA: Princeton Univ. Press, 2016.
- [22] *Fee-Per-Byte*. Accessed: Oct. 2021. [Online]. Available: <https://metamug.com/article/security/bitcoin-transaction-fee-satoshi-per-byte>
- [23] 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.
- [24] A. H. Barnett, "The Pigouvian tax rule under monopoly," *Amer. Econ. Rev.*, vol. 70, no. 5, pp. 1037–1041, 1980.
- [25] 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.
- [26] 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.
- [27] *Top 50 Gas Spenders*. Accessed: Mar. 2022. [Online]. Available: <https://etherscan.io/gastracker#gassender>
- [28] R. Bowden, H. P. Keeler, A. E. Krzesinski, and P. G. Taylor, "Block arrivals in the bitcoin blockchain," 2018, *arXiv:1801.07447*.
- [29] I. Tsabary and I. Eyal, "The gap game," in *Proc. 11th ACM Int. Syst. Storage Conf.*, Jun. 2018, pp. 713–728.
- [30] C. Decker and R. Wattenhofer, "Information propagation in the bitcoin network," in *Proc. IEEE P2P Proc.*, Sep. 2013, pp. 1–10.
- [31] R. Pass and E. Shi, "FruitChains: A fair blockchain," in *Proc. ACM Symp. Princ. Distrib. Comput.*, Jul. 2017, pp. 315–324.
- [32] 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.
- [33] *Block Size Data*. Accessed: Mar. 2022. [Online]. Available: <https://www.blockchain.com/charts/avg-block-size>
- [34] *Online Appendix*. Accessed: Dec. 2021. [Online]. Available: <https://share.weiyun.com/7LX12bs7>
- [35] *Pay Less Fee*. Accessed: Feb. 2021. [Online]. Available: <https://blog.makerdao.com/four-ways-defi-users-can-pay-less-in-ethereum-gas-fees/>
- [36] D. Gross, *Fundamentals Queueing Theory*. Hoboken, NJ, USA: Wiley, 2008.
- [37] C. Huang, H. Yu, J. Huang, and R. A. Berry, "Crowdsourcing with heterogeneous workers in social networks," in *IEEE Global Commun. Conf.*, Dec. 2019, pp. 1–6.
- [38] [Online]. Available: <https://ethereum.stackexchange.com/questions/30175/what-is-the-size-bytes-of-a-simple-ethereum-transaction-versus-a-bitcoin-trans>
- [39] *Price of SSD*. Accessed: Mar. 2022. [Online]. Available: <https://www.amazon.com/ssd/s?k=ssd>
- [40] M. J. Osborne and A. Rubinstein, *A Course Game Theory*. Cambridge, MA, USA: MIT Press, 1994.
- [41] *Ethereum Daily Transaction*. Accessed: Oct. 2020. [Online]. Available: <https://etherscan.io/chart/tx>



Yunshu Liu (Graduate Student 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.



Zhixuan Fang (Member, IEEE) received the B.S. degree in physics from Peking University, Beijing, China, in 2013, and the Ph.D. degree in computer science from Tsinghua University, Beijing, in 2018. He is currently a Tenure-Track Assistant Professor at the Institute for Interdisciplinary Information Sciences (IIIS), Tsinghua University, China. 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 Program for Research Excellence by CUHK. He serves as a Technical Program Committee Member for IEEE INFOCOM, WiOpt, ICC, GLOBECOM, and WCNC. He is an Associate Editor of IEEE COMMUNICATIONS LETTERS.



Wei Cai (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 of computer engineering with the School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen. He is serving as the Director of CUHK(SZ)-White Matrix Joint Metaverse Laboratory. He has coauthored more than 70 journal 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 for 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 worked as a Post-Doctoral Research Associate at Princeton University from 2005 to 2007. From 2007 until 2018, he was on the Faculty of the Department of Information Engineering, The Chinese University of Hong Kong, Shenzhen. Since 2019, he has been on the Faculty at The Chinese University of Hong Kong, where he is currently a Presidential Chair Professor and an Associate Dean of the School of Science and Engineering. He also serves as a Vice President of the Shenzhen Institute of Artificial Intelligence and Robotics for Society. His research interests are in the area of network optimization, network economics, and network science, with applications in communication networks, energy networks, data markets, crowd intelligence, and related fields. He has published more than 300 papers in leading venues, with a Google Scholar citation of 14000 and an H-index of 61. He has coauthored ten 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*. He was an IEEE ComSoc Distinguished Lecturer and a Clarivate Web of Science Highly Cited Researcher. He is the Editor-in-Chief of IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING and was the Associate Editor-in-Chief of IEEE OPEN JOURNAL OF THE COMMUNICATIONS SOCIETY.