The Cost of Delay: Evidence from the Ethereum Transaction Fee Market

Yinhong Zhao

Professor Campbell Harvey, Primary Faculty Advisor

Professor Michelle Connolly, Secondary Faculty Advisor

Honors Thesis submitted in partial fulfillment of the requirements for Graduation with Distinction in Economics in Trinity College of Duke University

Duke University
Durham, North Carolina
2023

Contents

Acknowledgements

I am incredibly grateful for the invaluable support and guidance from my primary faculty advisor Campbell Harvey during every stage of this project and my undergraduate career. I greatly appreciate the comments from my secondary faculty advisor Michelle Connolly and my peers in Duke Honors Thesis Seminar. I thank Kartik Nayak and Fan Zhang for generously sharing the Ethereum mempool data that is vital for this research; Jaimie Choi, Richard Lombardo, James Roberts, Adam Rosen, Robert Townsend, Vish Viswanathan, Daniel Yi Xu, Shuhan Zou for helpful discussion and feedback; funding support from Duke Bass Connections Student Research Award. All errors are my own.

I would also like to thank the numerous other professors at Duke who have taught, mentored, and supported me at different stages of my undergraduate career, including but not limited to Edward Balleisen, Tim Bollerslev, Jonathan Cummings, Erica Field, Rob Garlick, Duncan Thomas, and Luyao Zhang. Their patient guidance and training prepared me well not only to finish this thesis paper but also to continue pursuing my academic interests in economics in graduate school. A special shout-out goes to my friends who have shared my joy and sorrow and encouraged me all the way along.

Last but not least, I want to thank my family for always believing in me and encouraging my interests. I am eternally grateful for the unconditional support of my parents Dili Zhao and Yuqi Wang and my grandparents Jianping Zhao and Aizhen Gao.

Abstract

Delaying a financial transaction can be costly, but the cost of delay is difficult to estimate in traditional finance. I exploit the unique data offering and market design of the Ethereum blockchain to estimate the cost of delaying financial transactions in decentralized finance (DeFi). I construct a dynamic auction model for the Ethereum transaction fee market that relates users' optimal transaction fee bids to their delay cost functions and network conditions, and I structurally estimate the delay cost functions for different users and transaction types. The average cost of delaying a transaction by one minute is 8.78 US dollars, but the distribution of delay costs is highly skewed to the right. Delay costs are higher for complex transactions and users who trade more frequently. I estimate that welfare loss due to network delay on Ethereum was 14.03 million US dollars per day in July 2021, and I apply the delay cost estimates to evaluate the welfare losses under alternative transaction fee mechanisms.

JEL Codes: G10; L17; D44

Keywords: Delay cost; Blockchain; Auction; Transaction fee

1 Introduction

Time is money. Financial institutions are willing to exhaust a great amount of resources to expedite their transactions. For example, Jump Trading, a high-frequency trading hedge fund, spent 14 million US dollars to acquire land next to the Chicago Mercantile Exchange and build antennas to expedite their transactions by less than a millisecond (Louis (2017)). This is described as a "high-frequency trading arms race" in Budish et al. (2015). On the other hand, individual investors who can afford to wait set "limit orders" on financial platforms in seek of a better price in the future (Handa and Schwartz (1996); Parlour (1998)). These examples illustrate that the cost of delay in the financial system varies for different agents and different types of transactions.

Intuitively, rational agents will wait for potential chances of obtaining better deals in the future if their demand is not urgent, i.e., the delay costs are low. However, it is difficult to estimate the transaction delay costs in traditional finance because a) legacy financial data don't reveal the time when agents initiate new transactions, so transaction delays are difficult to measure; b) financial institutions compete for speed by lump-sum investments in infrastructure, so the transaction-level variance of delay cost is absent.

The emergence of decentralized finance on public blockchains in the past few years presents a unique opportunity to estimate the delay costs in a financial system. Blockchain is a type of distributed ledger technology that consists of a growing list of transaction records securely linked with cryptography, called blocks (Townsend (2020)). It is a computer network that does not rely on a single centralized agent (e.g., a firm or a government) as a trustee or notary to intermediate and settle transactions.

One primary application of blockchains is to execute financial transactions in a decentralized way, the so-called decentralized finance or "DeFi" (Harvey et al. (2021)). Since the DeFi boom in 2020, various applications have been built on blockchains to enable a comprehensive set of operations, including swaps, loans, deposits, token offerings, and derivatives trading. It is the

unique design of the blockchain transaction market and the transparency and immutability of blockchain data that makes the identification and estimation in this paper possible.

Launched in 2015, Ethereum is the most widely used blockchain platform in the world as of January 2023 and home to most major DeFi applications. Due to technological limitations, Ethereum has a limit on transaction throughput. Ethereum blocks arrive every few seconds, and each block has an exogenous limit of the total computation resources that can be consumed by the transactions in the block.[1] Almost always, the demand for transactions is higher than the supply of block spaces, so users must compete for blockchain resources. Ethereum thus uses a mechanism of transaction fees to efficiently allocate the block spaces to users (Roughgarden (2021)).

This paper studies the Ethereum transaction fee market in July 2021, when the transaction fee bidding follows infinitely repeated first-price common-value auctions.[2] Users submit their transaction information and transaction fee bids into the "mempool"or the queue pool. Block builders (or "miners") then select transactions from the mempool and include them in blocks as successful transactions.[3] All transaction fees are collected by block builders, so they prioritize the transactions with higher per-unit transaction fee bids or the so-called "gas price". The transactions not included by block builders in the current block stay in the mempool for consideration in future blocks.

Applying a special case of Afèche and Mendelson (2004) in the Ethereum transaction fee market, I construct a model of dynamic first-price auctions on the demand side of the transaction fee market. Users have private values of their transactions that decay over time. They submit their transactions to the mempool given the blockchain and mempool history they observe, from

[1] Computation resources on Ethereum are counted in units of "gas". In July 2021 (sample period of this paper), each block has a hard-coded "gas limit" of 15 million. See Section 2.3 for a detailed description.

[2] The choice of the study period in July 2021 is driven by data availability. The transaction fee mechanism on Ethereum didn't change between April 15, 2021 and August 5, 2021.

[3] See Easley et al. (2019); Huberman et al. (2021) for detailed analyses of block builders' behavior and the supply side of the market.

which they can fully learn the distribution of delay given any gas price bid. Users then decide whether to submit their transactions and how much fee they bid by maximizing their expected utility. The optimal gas price bid is a function of the delay cost of settling their transactions in later blocks and the network congestion. The optimal gas price bid is strictly lower than the static first-price auction. The key distinction between my framework and the previous literature is that I model user learning and waiting in a dynamic setting.

With the model on optimal bidding, I structurally estimate users' delay costs from their transaction fee bids. A higher transaction fee bid implies a higher delay cost. Users' delay costs are thus identified from the variance of transaction bids under different network conditions. Using a random coefficient model, I show that the average delay cost function of any sample of transactions is identified. The identification relies on two plausible assumptions. First, network conditions impact users' bidding exclusively by changing the waiting time distribution conditional on the gas price bid. Second, the waiting time distribution does not react to the bidding of any infinitesimal user.

Estimation follows a two-stage approach. In the first stage, I fit the waiting time distribution conditional on the bids and network conditions. In the second stage, I use a least square approach to estimate the delay cost function as a polynomial of waiting time. To shed light on the distribution of delay costs, I divide the transaction by delay cost percentiles using the residuals obtained from the main regression and separately estimate the average delay costs for each percentile. The same procedure is replicated in different samples of transactions grouped by user and transaction characteristics. The data are from transaction-level records on the Ethereum blockchain and a data set from Liu et al. (2022a,b) that documents transaction-level waiting times.

The estimation shows that the average cost of delaying a transaction for one minute is \$8.78. This estimate, however, varies widely across different transactions. The complete functional form of the average delay costs is graphed in Fig. 1. The average delay cost would be 94.56 USD for the 10% transactions with the highest delay costs and 1.72 USD for the 10% transactions

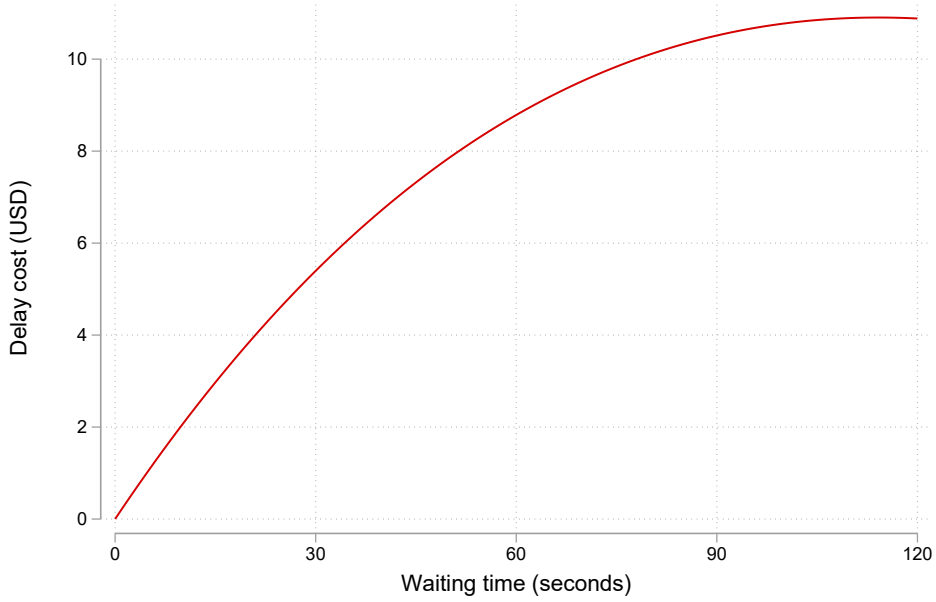


Figure 1: Estimate of the average delay cost function for all Ethereum transactions

with the lowest delay costs. Separately estimating the average delay cost for each percentile of transactions gives a delay costs distribution that is highly skewed to the right.

Using the same method, I further study the heterogeneity of delay costs for different groups of users and transactions. I find that on average, delay costs are higher for complex transactions (i.e., those consuming more computation power) and for users who transact more frequently (i.e., those who complete multiple transactions in the study period). Average delay costs vary at different hours. The analysis can be replicated to estimate the average delay cost for any single user. For example, certain addresses are known to be linked to large centralized exchanges like Binance and Coinbase. I find that Binance displays a higher delay cost than its counterparts.

I apply the delay cost estimates to evaluate the total welfare losses due to transaction delays under different transaction fee mechanisms. Welfare loss is estimated to be 14.03 million US dollars per day under the benchmark mechanism Ethereum adopted in July 2021 that orders transactions by their gas price bids. This, however, performs almost equally well as a counterfactual "socially optimal" mechanism that orders transactions by their delay costs and is much

better than a "naive" mechanism that orders transactions by their time of submission. These results demonstrate a novel welfare criterion for evaluating market designs on blockchains.

Related Literature

This paper contributes to four strands of literature. First, I contribute to the study on the value of time. Since the seminal paper of Becker (1965), the trade-off between time and market goods has been widely studied in the economic literature. Existing literature focus on the value of time for consumers (Deacon and Sonstelie (1985); Goolsbee and Klenow (2006); Aguiar and Hurst (2007); Nevo and Wong (2018)), firms (Lewis and Bajari (2011, 2014)), urban commuters (Buchholz et al. (2020); Goldszmidt et al. (2020)), and self-employed workers (Agness et al. (2022)). Queuing and waiting are also widely studied as a topic in operation research.[4] This paper extends the focus of literature to financial transactions, where the delay cost also plays a very important role.

Second, I contribute to a large literature on strategic behavior in dynamic games. For example, Hendel and Nevo (2006, 2013) study consumer stockpiling, i.e., bringing forward purchases when a good is on sale, and the private cost comes from the storage cost. On the other side, Li et al. (2014) and Papanastasiou and Savva (2017) show that a portion of consumers strategically delays purchases for potential price decreases or additional information. Li et al. (2014) attributes the existence of non-strategic consumers to myopia. This paper endogenizes the extent of strategic delay as a function of users' delay costs.

Third, I contribute to the studies of financial market design in the high-frequency trading setting. Since Demsetz (1968), transaction costs and latency arbitrage trading have been widely studied in the traditional financial markets. To address the problems of front-running, several studies including Farmer and Skouras (2012); Wah and Wellman (2013); Budish et al. (2015)

[4] See, for example, Naor (1969); Holt and Sherman (1982); Hassin (1995); Afèche and Mendelson (2004); Kittsteiner and Moldovanu (2005); Hassin (2016); Che and Tercieux (2020).

propose a frequent batch auction model where time is treated as a discrete variable and orders are processed in batch. Though Budish et al. (2015) investigate the theoretical advantages of this model, it has not been tested in traditional finance. The empirical phenomenon documented in this paper complements the theoretical discussion, and this paper highlights the potential welfare losses due to transaction delays under the frequent batch auction model in Budish et al. (2015).

Fourth, I contribute to a growing literature on blockchain transaction fee markets. Several previous studies including Easley et al. (2019) and Huberman et al. (2021) have studied the Bitcoin payment system proposed in Nakamoto (2008). Other studies model the blockchain transaction fee market as a static auction and study the incentive compatibility under different mechanisms,[5] but this paper shows that the optimal level of bidding in a dynamic setting, which is the true setting, can be significantly smaller than the static setting. Moreover, this paper proposes a novel welfare criterion to evaluate transaction fee mechanisms using counterfactual analysis, contributing to the discussion on optimal transaction fee mechanism designs as in Roughgarden (2020, 2021); Liu et al. (2022a)).

Organization

The rest of the paper proceeds as follows. Section 2 introduces the Ethereum transaction fee market as the setting of this study. Section 3 describes the bidding model in a dynamic first-price auction scenario. Section 4 introduces the data sources. Section 5 discusses the identification and estimation methods. Section 6 present the estimation results. Section 7 discusses an application of the estimates in financial market design. Section 8 concludes.

2 Background: Ethereum Transaction Fee Market

[5]See, for example, Yao (2018); Lavi et al. (2019); Roughgarden (2020, 2021); Chung and Shi (2022).

2.1 Blockchains and DeFi

Since the launch of Bitcoin in 2009 (Nakamoto (2008)), blockchain has emerged as a new technology that is used for a wide range of purposes. Blockchain is a type of distributed ledger technology that consists of a growing list of transaction records securely linked with cryptography. Updates on transaction records are packaged into blocks and are chained together using cryptographic hash functions to allow an audit of the prior history hence the name (Harvey et al. (2021)). Instead of relying on a centralized agent to settle transactions, blockchain is a computer network composed of a group of users usually called "nodes", and they maintain the operation of the ledger collectively through a consensus protocol. One of the most popular applications of blockchain is cryptocurrencies, which are tokens (usually scarce in supply) built on blockchains. The ownership of the tokens is securely recorded on the blockchain, and users can transfer the ownership of these tokens through transactions on the blockchain.

There are several important features of blockchain. First, transactions are immutably recorded. The blocks are linked with each other by cryptography, so any tampering of previous records will lead to inconsistency and will be detected by other users. Second, the entry is permissionless. Any person or entity can operate on public blockchains without permission from any parties or governments. While this improves financial accessibility, it also creates challenges for regulation. Third, users are anonymous, but transactions are fully traceable and transparent. Users are represented with unique 42-character hexadecimal addresses that effectively hide their real identities. However, the transactions between all these addresses are immutably documented on the blockchain, which allows perfect tracking of the relationships between addresses.

DeFi operates on blockchains. Since the DeFi boom in 2020, various decentralized applications (DApps) have been built on blockchains to enable a comprehensive set of financial operations including swaps, loans, deposits, token offerings, stablecoins, and derivatives trading. These applications are built as smart contracts on blockchains, which can be thought of as automated algorithms with open-source code. Compared to traditional finance, DeFi has higher

efficiency, accessibility, transparency, and interoperability (Harvey et al. (2021)). According to data from DefiLlama, the total value locked (TVL) in DeFi as of October 2022 is approximately 54 billion USD.

2.2 The Ethereum Blockchain

Ethereum is the most widely used blockchain in the world. Ether, dubbed as "ETH" in cryptocurrency exchanges, is the native token on Ethereum that supports its operation. Ethereum is a blockchain platform with a built-in Turing-complete programming language, allowing anyone to write smart contracts and decentralized applications where they can create their own arbitrary rules for ownership, transaction formats, and state transition functions (Buterin (2014)). It is a decentralized computation infrastructure that allows free entry and exit of any parties to build and use various (in theory, any) applications.

According to data from Etherscan (2022), Ethereum hosts about 200 million distinct addresses (i.e., accounts, though a user can hold multiple addresses), 580 thousand ERC-20 tokens (i.e., cryptocurrencies built with Ethereum smart contract using a common standard that allows for interoperability), and 1.1 million daily transactions. It is also home to most major decentralized finance protocols (e.g., MakerDAO, Uniswap, Compound, etc.) and NFT marketplaces (e.g., Opensea). Therefore, Ethereum is an important market to study from an industrial organization and market design perspective.

The transactions on Ethereum are processed in blocks, which are batches of simultaneously executed transactions. As a proof-of-work blockchain, Ethereum is secured by a subset of users called "block builders" or "miners". Block builders keep attempting to find roots of a specific cryptographic function, which has proved to be computationally intensive. Each time someone finds a solution, the person can launch a new block and becomes the block builder of that block and thus capture the block reward and transaction fees. Depending on block builders' speed of solving the puzzles, Ethereum blocks arrive following a random process, and each block can

process a fixed amount of computation in the transactions due to technological limitations on scalability.[6] Therefore, there is a limited supply of Ethereum blockchain resources at any point in time, which is in most cases smaller than the total transaction demands. It thus requires a mechanism to efficiently allocate blockchain resources.

2.3 Transaction Fees on Ethereum

Ethereum allocates its blockchain resources through a transaction fee mechanism. While Ethereum occasionally updates its transaction fee mechanism,[7] this paper focuses on the period of our concern in July 2021. In this period, Ethereum largely follows the prototype of the Bitcoin payment system as studied in Easley et al. (2019) and Huberman et al. (2021), which is also similar to the frequent batch auction model proposed in Budish et al. (2015). The following presentation focuses on the Ethereum transaction market in July 2021.

On the supply side, block builders keep attempting to solve the mathematical puzzle that serves as a proof-of-work. On average every 13 seconds, some block builder finds a solution to the mathematical puzzle and claims ownership of the next block. This block builder has full discretion over which pending transactions to include in the block and how they are ordered, and this block builder aims to maximize their own revenues. Block builders are rewarded two newly minted Ether by the Ethereum protocol in addition to all the transaction fees paid by users in their blocks. Therefore, block builders tend to include transactions that generate the highest transaction fees.

Transactions on Ethereum consume the computation power of blockchain operators or the

[6] See Zhou et al. (2020); Gudgeon et al. (2020) for survey of computer science literature. Eyal et al. (2016); Kalodner et al. (2018); Tsabary et al. (2021) discuss potential solutions to the problem.

[7] The three closest transaction fee mechanisms changes are (i) Ethereum adjusted the gas limit for each block from 12.5 million to 15 million in the Berlin Hardfork in April 2021; (ii) Ethereum implemented EIP-1559 in the London Hardfork in August 2021, which changed the transaction fee mechanism from a first-price auction to a mechanism based on a second-price auction; (iii) Ethereum implements the Merge in September 2022 and shifted from a proof-of-work blockchain to a proof-of-stake blockchain. These updates are well anticipated by the public, but users are not likely to react to them when bidding transaction fees because these month-level updates are not relevant for the minute-level transaction waiting time.

"nodes", and the amount of computation that can be accomplished by each block has a hard-coded limit. The computation unit on Ethereum is dubbed "gas", an analogy to gasoline for cars. Each operation on Ethereum consumes a pre-designated amount of gas. For example, a transfer of any amount of Ether, the native token of Ethereum, from one address to another consumes 21,000 units of gas, and a swap of one token with another on Uniswap, a popular decentralized exchange for token swaps, consumes approximately 150,000 units of gas. A detailed gas schedule of the Ethereum blockchain is shown in Fig. B.1. In July 2021, each block on Ethereum held a maximum of 15 million gas units, which can for example execute approximately 700 Ether transfers.

On the demand side, users first submit their transactions to a wait list called "mempool", a queue pool that collects all the pending transactions. Together with the content of their transactions, they also submit gas price bids, which is their willingness to pay the block builder for each unit of gas consumed by their transactions. Block builders monitor the "mempool" and decide which transactions to include into their blocks following the principle of revenue maximization. If a transaction is included into a block, the user then pays a transaction fee that equals to the product of the gas price bid ("pay-as-bid") and the actual amount of gas used for this transaction. The transactions not included into the current block remain in the mempool and keep waiting until either included in a future block or cancelled by the user. Therefore, the Ethereum transaction fee market in the study period forms an infinitely repeated first-price auction.

3 The Bidding Model

The cost of delay comes from a variety of factors on Ethereum. First, other users might compete for the same arbitrage opportunities, and a delay in transaction settlement implies a loss of opportunities. Second, delayed transactions risk being frontrun by malicious parties. The pending transaction might reveal private information on future price movements, so malicious parties

may take advantage the time gap to insert their transactions, which are in many cases illegal. This problem is especially severe in decentralized finance because transactions in the mempool are public and regulation is absent (Daian et al. (2020)). Third, delay might also lead to failure of transaction settlement. For example, Uniswap, the largest decentralized exchange on Ethereum, specifies that transactions must be settled within 20 minutes and would otherwise fail. Capponi et al. (2022) finds that execution risk is the main concern of transaction delays and thus composes the main source of delay costs. Fourth, the design of Ethereum prohibits addresses from submitting another transaction before the earlier transaction was settled or canceled, so delaying a transaction can bear opportunity costs.

I apply a special case of Afèche and Mendelson (2004) in the Ethereum transaction fee market to study how users with a delay cost bid optimally in the dynamic first-price auction of the Ethereum transaction fee market. I adopt a simple delay cost term that includes all the opportunity cost caused by transaction delays. I consider a continuous timeline denoted by t with its unit measured in seconds.

Supply. The supply side of the market is straightforward. Blocks arrive in a homogeneous Poisson arrival process with a shape parameter Λ. Let X_n be the inter-arrival time that follows an exponential distribution $Exp(\Lambda)$. Then the arrival moment of the nth block is $S_n = \sum_{i=1}^{n} X_n$.

User arrivals. Suppose there are a total of N potential users (for example, there are a total of 200 million addresses on Ethereum). Demands of each potential user i comes at a Poisson process with a shape parameter α_i, so the demand from any user comes also following a Poisson process with a shape parameter $A = \sum_{i=1}^{N} \alpha_i$. The gas need of each transaction follows i.i.d. from a known distribution (e.g., Fig. B.3b).

The private value per gas of each submitted transaction, v, is defined as the total welfare of completing the transaction without any delay divided by the total amount of gas used. v depends on the time it is submitted and its own characteristics, i.e., $v = f(\theta, t)$, where θ is a vector of characteristics. Specifically, θ includes the component depending on the user's idiosyncratic

characteristics and other characteristics of the transaction.

Delay cost structure. A user i's utility function for transaction indexed j incorporating transaction delay can be written as

$$u_{ij}(v,t,b) = v_{ij} - C_{ij}(t) - b_{ij}g_{ij}, \tag{1}$$

where v_{ij} is the private valuation, $C_{ij}(t)$ is the delay cost, b_{ij} is the bid price on a unit of gas, and g_{ij} is the total amount of gas used. Assume that C is continuously differentiable and monotonically increasing.

Market structure. Block builders collect all the transaction fees. They select the set of transactions $\mathfrak{S}$ to include in their blocks by solving the profit maximization problem

$$\max_{\mathfrak{S}} \{ \sum_{i,j\in\mathfrak{S}} b_{ij}g_{ij} \} \; s.t. \sum_{i,j\in\mathfrak{S}} g_{ij} \leq G. \tag{2}$$

As a solution to this linear maximization problem, block builders prioritize the users who bid higher gas prices.[8] This makes the market an infinitely repeated first-price open-bid auction. With 200 million addresses as potential users, I may assume that the actions of any infinitesimal user do not affect the distribution of delay in the system.[9]

Information structure. In a dynamic setting, users learn from past auctions. Users observe the past history of blockchain and mempool, from which they can infer the current rate of user arrivals and the distribution of private values, which enables them to form an expectation on the delay given any bid price.[10] Under a specific observable market condition M, which includes

[8] It is true that in marginal cases when the total gas used approaches the gas limit, block builders may deviate from this behavior. However, at most 1 or 2 transactions out of on average 170 transactions in each block would be affected, so the marginal cases are ignored.

[9] In the sample period, the number of transactions done by the most active user accounts for merely 2% of the total number of transactions. It is thus not likely that any single user has market power.

[10] It is reasonable to believe that users can predict future delays even for the less advanced ones because there are plenty of gas price recommendation software on the Internet that people frequently look at. Cryptocurrency wallets widely used for transaction submission like MetaMask show automatic reminders of waiting times when

all the observable information on the blockchain history and current mempool status, a user can predict the probability that the transaction is included in the ith block afterward and thus the full distribution of waiting time conditional on any transaction fee bid b they submit.

Market Equilibrium. By Afèche and Mendelson (2004), there exists a unique symmetric bidding equilibrium in this model, which thus leads to an equilibrium in network waiting time. I do not explicitly endogenize the waiting time equilibrium as in Afèche and Mendelson (2004) but instead estimate it empirically.

Let the conditional waiting time on any gas price bid b under the market condition M be a random variable denoted as $W_M|b$ whose cumulative distributive function denoted as $W_M(t|b)$. I assume that $W_M(t|b) \in [0,1]$ is twice continuously differentiable over both b and t and monotonically increasing on the domain $t \in [0,\infty)$. Let the probability density function of $W_M|b$ be $w_M(t|b)$. Also, the higher the bid is, the larger the probability that the transaction gets included before any time t, which gives $\frac{\partial W_M(t|b)}{\partial b} > 0, \forall b,t > 0$. For tractability purpose, I further assume that $\frac{\partial^2 W_M(t|b)}{\partial b^2} < 0, \forall b,t > 0$.

Since users are atomistic, the distribution of $W_M|b$ is independent of the bid of each single user, so from the perspective of each user, $W_M|b$ follows an exogenous distribution that can be predicted from the history on the blockchain.

Expected utility. The expected utility of a bid gas price b would be

$$\mathbb{E}(u_{ij}(v,b)) = \int_{t=0}^{\infty} u_{ij}(v,t,b) dW_M(t|b) = v_{ij} - E[C_{ij}(W_M|b)] - bg_{ij}. \tag{3}$$

Users will only submit bids if the expected utilities are positive, i.e. $\mathbb{E}(u_{ij}(v,b)) > 0$. If users don't submit bids, denote $b^* = 0$, and the realized utility is zero.

Optimal bids. Users submit the optimal bid (or no submission) that maximizes their expected

users set gas prices. See Fig. B.2 for an example of the fee bidding interface.

utility,

$$b^* = arg\max_b \{\mathbb{E}[u_{ij}(v,b)]\}. \tag{4}$$

The first order condition of Eq. (3) gives $\frac{\partial \mathbb{E}[u_{ij}(v,b)]}{\partial b}|_{b=b^*} = 0$. By Leibniz integral rule, this can be written as

$$g_{ij} + \int_{t=0}^{\infty} C_{ij}(t) \frac{\partial w_M(t|b)}{\partial b}|_{b=b^*} dt = 0. \tag{5}$$

In addition, b^* must also satisfy the second order condition $\frac{\partial^2 \mathbb{E}[u_{ij}(v,b)]}{\partial b^2}|_{b=b^*} < 0$ and the boundary condition that $\mathbb{E}(u_{ij}(v,b^*)) > 0$. Combining these observations, we have the following theorem for users' optimal pricing rule.

Theorem 3.1 (Optimal Bidding)**.** *Consider transaction j submitted by user i with the utility function described in Eq.* (1)*. Suppose the delay cost function C_{ij} is continuously differentiable and monotonically increasing. Suppose $W_M(t|b)$ is the cumulative distribution function of delay time given a bid b, which is twice continuously differentiable over both b and t, and $\partial W_M(t|b)/\partial b > 0, \partial^2 W_M(t|b)/\partial b^2 < 0$ for any $b,t > 0$. Then the optimal bid of this transaction is*

$$b^*_{ij} = \begin{cases} 0 & if \quad \mathbb{E}(u_{ij}(v,\tilde{b_{ij}})) \leq 0, \\ \tilde{b_{ij}} & if \quad \mathbb{E}(u_{ij}(v,\tilde{b_{ij}})) > 0, \end{cases} \tag{6}$$

where $\tilde{b_{ij}}$ is the unique positive solution to Eq. (5)*.*

Proof: See Appendix A.1.

The theorem gives two key observations. First, the optimal bid derived from this model is smaller than that from a static first-price auction model. Intuitively, this is because the transactions have multiple chances of being included in the blocks, whereas in a static setting the transactions only have one chance. Second, the optimal bids are smaller for the users who are more patient, i.e., whose delay costs are lower. These two observations are summarized as the corollaries below. See Appendix A.2 for an example.

Corollary 3.1.1. *The optimal bid in a dynamic first-price auction is strictly smaller than that in a static first-price auction when users are infinitesimal (i.e. $N \to \infty$).*

Proof: All subscripts of ij are omitted in the proof. In a static first price auction, the Bayes-Nash equilibrium bidding strategy is $b^{static} = \frac{N-1}{N}\frac{v}{g} \to \frac{v}{g}$ when $N \to \infty$. By Theorem 3.1, if $\mathbb{E}(u(v,\tilde{b})) \leq 0$, $b^* = 0 \leq b^{static}$; If $\mathbb{E}(u(v,\tilde{b})) > 0$, then $v - C(t) - b^* g > 0$, so $b^* < \frac{v-C(t)}{g} < \frac{v}{g} = b^{static}$.

Corollary 3.1.2. *For two transactions under the same market condition M with the same amount of gas used $g_{ij1} = g_{ij2}$, suppose $C_{ij1}(t) \leq C_{ij2}(t)$ and $C'_{ij1}(t) \leq C'_{ij2}(t)$ for any $t > 0$, then the optimal bids satisfy $b^*_{ij1} \leq b^*_{ij2}$.*

Proof: Using integral by parts, the first-order condition in Eq. (5) *can be rewritten as*

$$\int_{t=0}^{\infty} \frac{\partial W_M(t|b^*_{ij})}{\partial b_{ij}} C'_{ij}(t)dt = g_{ij}.$$

Since $C'_{ij1}(t) \leq C'_{ij2}(t)$ for any $t > 0$, there exists $t_0 > 0$ such that $\frac{\partial W_M(t_0|b^*_{ij1})}{\partial b} > \frac{\partial W_M(t_0|b^*_{ij2})}{\partial b}$. Since $\frac{\partial W_M(t_0|b^*_{ij})}{\partial b}$ is monotonically decreasing for any $t > 0$, $b^*_{ij1} \leq b^*_{ij2}$.

4 Data

4.1 Measurement

I combine data sets from the Ethereum blockchain and mempool. As a ledger itself, the Ethereum blockchain records the whole transaction history, including the addresses involved, the timestamp when block builder started to pack the block (given by block builders), amounts of Ether (the native token of Ethereum) transferred, transaction fees bids, amount of gas used of all transactions that happened in the history. The immutability of the blockchain technology guarantees the accuracy of these information. I access the Ethereum blockchain data through Google Bigquery Platform (2019).

The blockchain, however, does not provide information on the time when each user submits the transaction. To complement this, I use a data set from Liu et al. (2022a,b) that measures the timing of users' submission their transactions to the mempool, $T_{\mathsf{TX}}^{mempool}$, by constantly monitoring the Ethereum mempool. To obtain a reasonable approximation of the mempool data, Liu et al. (2022a,b) operates four geographically-distributed nodes across the globe to get a representative sample of the mempool. Their modified clients store logs of mempool whenever they receive new transactions submitted from the network. The earliest time when any transaction is observed in mempool across all servers is used to estimate the time when users submit their transactions.

To measure the waiting time of each transaction, the timestamp when the transaction is confirmed on blockchain, $T_{\mathsf{TX}}^{blockchain}$, is also necessary. I follow the practices of Liu et al. (2022a,b) to use the timestamp given by the block builder of the *next* block. The reasoning behind this is that the time when the block builder of the next block start to pack the block should be approximately the time when the transactions in the current block are confirmed by the blockchain network. The time that the transaction waits as a pending transaction is then

$$\text{Waiting time of } \mathsf{TX} = T_{\mathsf{TX}}^{blockchain} - T_{\mathsf{TX}}^{mempool}.$$

A distribution of the waiting time collected in the data is shown in Fig. B.3.

4.2 Summary Statistics

The data covers most of the transactions in the 60,000 blocks from block number 12895000 (July 25, 2021) to 12954999 (August 3, 2021) on the Ethereum blockchain. Due to technological issues with mempool monitoring, mempool data is missing from block number 12919573 to block number 12920091 (8:13 - 10:00 +UTC, July 29) and from block number 12924071 to block number 12924777 (1:13 - 4:00 +UTC, July 30). The mempool data is ephemeral, so it

Table 1: Summary Statistics

	mean	SD
gas price (Gwei)	44.04	149.92
waiting time (s)	100.00	310.32
amount of gas used	75,083	149,772
time between two blocks (s)	15.72	14.25
block size (B)	65,209	16,876
block gas limit	14,986,493	19,545
number of transactions submitted per min	915.72	428.83
Observations	11,811,229	

is not possible to recollect the data afterward. However, missing data due to exogenous server breakdown should not be selected and bias the results.

As mentioned in Liu et al. (2022a,b), a small portion (approximately 5%) of the transactions are submitted directly to the block builders to avoid potential frontrunning,[11] so they are not observable in the mempool. As a result, waiting time for these transactions are missing, so I exclude them from the analysis. This might bring downward bias to estimates of average delay cost functions because these transactions are usually high-value and high-delay-cost transactions.

In total, my data include 10462708 transactions from 58000 out of 60,000 blocks at the end of July 2021. Summary statistics of several key variables in the data are presented in Table 1. Distributions of several key variables is presented in Fig. B.3.

4.3 Descriptive Analysis

I run a simple ordinary least square regression to show the correlation between the waiting time of each transaction and gas price bids and market conditions. The results are presented in Table 2. As expected, when the market is more congested (proxied by the number of transactions submitted to the mempool per minute), the waiting time is longer; when the user bids a higher

[11] Transactions in the mempool is public revealed, so other users may take advantage of the information to "frontrun" the transactions by submitting a higher gas price bid or directly colluding with block makers. See more details in Daian et al. (2020).

Table 2: Correlation between waiting time and other variables

Variables	(1) waiting time (s)
num of txs submitted per min	0.169***
	(0.000323)
gas price (Gwei)	-0.0839***
	(0.00120)
Constant	-40.76***
	(0.293)
R-squared	0.026
Observations	10,462,708

OLS regression with standard errors in parentheses
*** p<0.01, ** p<0.05, * p<0.1

gas price, the waiting time is shorter. These results, however, only demonstrate correlations and should not be interpreted as causal because both independent variables can be endogenous.

5 Identification and Estimation

This section discusses identification and estimation methods of the model. The variance of bids in a sample of transactions under different network conditions enables the identification of the delay cost function. Estimation follows a two-stage procedure that first estimates the conditional waiting time distribution and then the delay cost function.

5.1 Identification

As stated in Theorem 3.1, a higher transaction fee bid implies a higher delay cost. To illustrate this, consider two types of users - one with a low delay cost and one with a high delay cost. Suppose they bid both in a congested market and a non-congested blockchain market. In the non-congested case, a low bid is enough to guarantee quick settlement, so both low and high

delay cost users would bid low. In the congested case, however, a high bid is necessary for quick settlement. In this case, users with low delay cost might still bid low because they do not care about waiting time, but users with high delay costs will bid higher in comparison. Therefore, the bids of the same transaction under different market conditions can identify the delay cost.

While this sheds some light on the delay cost, there are two main identification challenges. First, the full functional form of the delay cost function takes an infinite sample to identify, which is not available in reality. As a compromise, I suppose the delay cost function be a polynomial, which can approximate any continuous function by StoneWeierstrass theorem. Suppose $C_{ij}(t) = \sum_{n=1}^{N} c_{ijn} t^n$, so $C_{ij}(0) = 0$. Second, and more importantly, only bid under one market condition is observed for each transaction, so additional assumptions need to be made. I resolve this by pooling the delay cost functions of multiple transactions with a random coefficient model.

Identification and estimation of the delay cost function are both conducted in two stages. In the first stage, the conditional waiting time distribution $W_M(t|b)$ is identified for any waiting time t conditional on any gas price bid b because the waiting times of all transactions are observed in the data. Specifically, following Hansen (2004),

$$w_M(t|b) = \frac{f(M,b,t)}{f(M,b)},$$

where M is the market condition related to the timing of the transaction. $f(M,b,t)$ is the density of the joint distribution of (M,b,t), and $f(M,b)$ is the density of the marginal distribution.

In the second stage, the average delay cost function is identified for any group of transaction under a random coefficient model. Suppose the delay cost functions of these transactions are independently and identically drawn from a fixed distribution, $C_{ij}(t) \overset{\text{iid}}{\sim} \mathscr{C}_t$. Let $\overline{C}(t) = \mathbb{E}[\mathscr{C}_t]$ and $C_{ij}(t) = \overline{C}(t) + \varepsilon_{ij}(t)$, which gives $\mathbb{E}[\varepsilon_{ij}(t)] = 0$. Let

$$\xi_{ij} = \int_{t=0}^{\infty} \varepsilon_{ij}(t) \frac{\partial w_M(t|b)}{\partial b}|_{b=b^*} dt.$$

With the polynomial form of $C_{ij} = \sum_{n=1}^{N} c_{ijn} t^N$, let $c_{ijn} = \overline{c_n} + \varepsilon_{ijn}$ and

$$P_n(b|W_M) = \int_{t=0}^{\infty} t^n \frac{\partial w_M(t|b)}{\partial b}|_{b=b^*} dt.$$

Then, the first-order condition Eq. (5) can be written as

$$g_{ij} + \sum_{n=1}^{N} \overline{c_n} P_n(b|W_M) + \xi_{ij} = 0. \tag{7}$$

The identification of $\overline{C}(t)$ relies on two plausible assumptions: exclusion and infinitesimal users. First, network conditions impact users' bidding exclusively through changing the distribution of conditional waiting time on gas price bid. Second, the waiting time distribution does not react to the bidding of any single infinitesimal user. Formally,

Assumption 5.1 (Exclusion)**.** *The idiosyncratic delay cost of a transaction satisfies*

$$\varepsilon_{ij} \perp W_M.$$

Assumption 5.2 (Infinitesimal Users)**.** *For any transaction indexed j of user i,*

$$W_M|b_{ij} = W_M.$$

Proposition 5.1. *Under Assumption 5.1 and Assumption 5.2, the average delay cost of a group of transactions $\overline{C(t)}$ is identified.*

Proof: Notice that for any $t_0 > 0$ *and* $0 < n \leq N$

$$\begin{aligned}
\mathbb{E}[\xi_{ij} P_n(b|W_M)] &= \mathbb{E}[\int_{t=0}^{\infty} \varepsilon_{ij}(t)(P_n(b|W_M)\frac{\partial w_M(t|b)}{\partial b})|_{b=b^*} dt] \\
&= \int_{t=0}^{\infty} \mathbb{E}[\varepsilon_{ij}(t)(P_n(b|W_M)\frac{\partial w_M(t|b)}{\partial b})|_{b=b^*}] dt \\
&= \int_{t=0}^{\infty} \mathbb{E}[\varepsilon_{ij}(t)]\mathbb{E}[P_n(b|W_M)\frac{\partial w_M(t|b)}{\partial b})|_{b=b^*}] dt = 0,
\end{aligned}$$

where the second step is by Fubini's Theorem and the third step is by Assumption 5.1. This thus gives the N moments conditions necessary to identify the N coefficients in the polynomial form of $\overline{C_n}$.

5.2 Parameterization

To aid in estimation, I impose a set of parametric assumption on the model. Specifically, I assume the delay cost function to be a polynomial of order N and the conditional waiting time distribution to follow a Gamma distribution.

As stated in Section 5.1, delay cost function is specified as a polynomial with random coefficients. Each coefficient contains a component of average delay cost function $\overline{c_n}$ across the sample and a transaction idiosyncratic delay cost ε_{ijn}. By definition, the idiosyncratic delay cost has zero expectation.

$$\begin{aligned}
C_{ij}(t) &= \sum_{n=1}^{N} c_{ijn} t^N, \\
\overline{c_n} &= \frac{1}{M} \sum_{i,j} c_{ijn}, \\
c_{ijn} &= \overline{c_n} + \varepsilon_{ijn}, \\
\varepsilon_{ijn} &\sim F(\cdot) \quad \mathbb{E}[F] = 0.
\end{aligned}$$

The waiting time distribution is parameterized as a Gamma distribution with both scale and

shape coefficients linear to the transaction fee bid.

$$
\begin{aligned}
W_M(t|b) &\sim Gamma(\alpha_b, \beta_b), \\
\alpha_b &= \alpha_1(b - \alpha_0), \\
\beta_b &= \beta_1(b - \beta_0).
\end{aligned}
$$

α_b is the shape parameter of the Gamma distribution, which primarily determines the skewness of the distribution. β_b is the rate parameter of the Gamma distribution, which primarily determines the length of the overall waiting time. Therefore, α_0 and β_0 can be interpreted as a reserve price for gas price bids at different time periods. α_1 and β_1 capture the marginal response of the shape and rate parameters to a higher bid price.

5.3 Estimation

Estimation follows a two-stage procedure as in Perrigne and Vuong (2019). In the first stage, I estimate the conditional waiting time distribution. I first divide the study period into five-minute intervals, which I consider as separate markets. Then, I sample 100 bid prices in each interval and fit the empirical waiting times within a small interval around these bid prices to a Gamma distribution. After obtaining the distribution parameters, I run a linear regression between the distribution parameters and the bid prices to estimate α_0, α_1, β_0, and β_1 specified in the waiting time distribution for each five-minute interval. An illustrative example of the first-stage result is shown in Fig. B.4.

A shortcoming of this approach is that the linear model occasionally fails to extrapolate to the transactions with extreme values of bid prices. Since α_1 is expected to be negative, α_b might be negative for transactions that bid very high gas prices, which returns an undefined distribution for waiting time. I drop these transactions (about 5% of total data) in the second-stage estimation. This problem will be mitigated with a non-linear model for Gamma distribution coefficients or

estimate the model in a non-parametric way.

In the second stage, I use a least square approach to estimate the polynomial coefficients of the delay cost function. I first match the submission time of the transactions to the distribution parameters estimated in the first stage by five-minute intervals. I then plug the estimates into the moment conditions specified in Eq. (7) to compute the market condition proxies $P_n(b|W_M)$, which can be written in closed forms under the parameter specifications. The polynomial coefficients $\overline{c_n}$ can then be estimated in an ordinary least square regression. I specify the order of polynomial $N = 5$ in the estimation.

The current approach only identifies and estimates the average of the delay costs. The standard error from the least square estimation should be interpreted as the standard error of the average delay cost function estimate instead of that of the delay costs of different transactions. To shed light on the distribution of delay cost functions for different transactions, I use the error term ξ_{ij} from the main specification as a proxy for the idiosyncratic delay costs ε_{ij} and split all transactions to different percentiles by their delay costs. I then separately estimate the average delay costs for different percentiles of transactions. The estimates shed light on the variance of delay costs among the Ethereum transactions.

The same methodology can be used to identify and estimate the average delay cost of any sample of transactions, which enables the study of the heterogeneity of delay costs. I further estimate the average delay cost for transactions of different types (defined by the gas used), by different users (defined by the frequency of transacting), and by different times (defined by the hour of transaction submission). I also estimate the average delay cost for several large centralized exchanges.

6 Results

6.1 Main Results

The full functional forms of delay costs are graphed in Fig. 1 with the coefficient estimates presented in Column (1) of Section 6.1. I use a conversion rate of $ETH/USD = 2500$, which is approximately the average price of ETH in July 2021, to convert fees from ETH to USD. The cost of delaying a transaction for one minute is estimated to be 8.78 US Dollar on average. As expected, the average delay cost function is monotonically increasing. Moreover, the estimated function has a concave structure, which implies a decreasing marginal delay cost. This number, however, varies greatly across different transactions. The average cost of delaying by one minute would be 94.56 USD for the 10% transactions with the highest delay costs and 1.72 USD for the 10% transactions with the lowest delay costs.

To further study the variance of delay costs among Ethereum transactions, I separately estimate the average delay cost functions for every percentile of transactions sorted by the error term ξ_{ij} that I obtain from the main regression. While the sorting might not exactly reflects the idiosyncratic delay cost term ε_{ij}, this exercise still demonstrates the large variance in the delay costs among different transactions. The distribution of these percentile estimates should be identical to the distribution of the delay costs in the whole population, which is graphed in Fig. 2. As expected, the distribution of average delay costs resembles an exponential distribution. The coefficients of the average delay cost function for the 10% transactions with the lowest delay costs, the 25% transactions with the lowest delay costs, the 25% transactions with the highest delay costs, the 10% transactions with the highest delay costs are shown in Columns (2)-(5) in Section 6.1.

6.2 Heterogeneity Analysis

I separately estimate the average delay cost function for several different samples of transactions. First, Fig. 3a shows the average delay costs of simple transactions and complex transactions. A

Table 3: Coefficient estimates for average delay costs of different groups of users

Coefficients	(1) All	(2) Bottom 10%	(3) Bottom 25%	(4) Top 25%	(5) Top 10%
c_1	-86,993***	-17,065***	-26,196***	-473,661***	-912,576***
	(95.30)	(11.08)	(13.78)	(488.7)	(1,306)
c_2	530.8***	104.3***	163.0***	2,771***	5,249***
	(0.795)	(0.0921)	(0.116)	(3.732)	(8.590)
c_3	-1.004***	-0.196***	-0.311***	-5.137***	-9.589***
	(0.00195)	(0.000220)	(0.000282)	(0.00889)	(0.0183)
c_4	$8.82*10^{-4}$***	$1.68*10^{-4}$***	$2.71*10^{-4}$***	$4.48*10^{-3}$***	$8.26*10^{-3}$***
	(2.18e-06)	(2.34e-07)	(3.05e-07)	(9.71e-06)	(1.85e-05)
c_5	-2.11e-07***	-3.81e-08***	-6.22e-08***	-1.09e-06***	-1.98e-06***
	(6.62e-10)	(6.38e-11)	(8.56e-11)	(2.93e-09)	(5.26e-09)
Obs.	9937603	993708	2484396	2484400	993760
R^2	0.157	0.884	0.799	0.360	0.366

*** p<0.01, ** p<0.05, * p<0.1

Standard errors in parentheses. The variables are the coefficients in the delay cost function form $C(t) = c_1 t + c_2 t^2 + c_3 t^3 + c_4 t^4 + c_5 t^5$. The unit of $C(t)$ here is in Gwei (10^{-9} ETH). Column (1) shows the polynomial coefficients for the average delay cost function among all Ethereum transactions. Column (2) - (5) shows the polynomial coefficients for average delay cost functions among the 10% transactions with the lowest delay costs, the 25% users with the lowest delay costs, the 25% users with the highest delay costs, and the 10% users with the highest delay costs. The quantiles are divided with the error terms ξ_{ij} obtained from the main regression in Column (1), which are used as proxies for the idiosyncratic delay costs ε_{ij}.

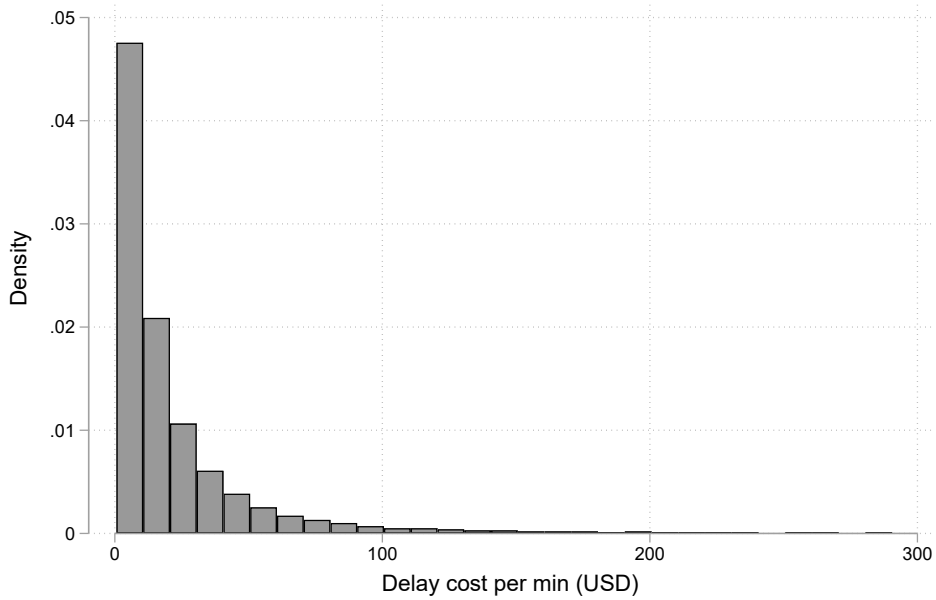


Figure 2: Distribution of estimates of average delay costs per minute among Ethereum transactions. Estimates of average delay costs are sampled from different percentiles of delay costs. The percentiles are divided with the error terms ξ_{ij} obtained from the main regression, which are used as proxies for the idiosyncratic delay costs ε_{ij}. Costs of delay vary significantly for different transactions.

simple transaction is defined to be one that uses exactly 21,000 units of gas, which is the amount of gas used to transfer ETH tokens from one address to another on Ethereum. Simple transactions account for about 35% of the transactions on Ethereum. Complex transactions make up all the other transactions. As expected, complex transactions have higher delay costs on average than simple transactions. The coefficient estimates are presented in Table B.1.

Second, Fig. 3b shows the average delay cost function for users with different trading frequencies. I group users by the number of transactions they made in the study period. Low-frequency users trade 1-5 times in the sample period (33.3% of all transactions), who can be retail investors or individual blockchain users. Mid-frequency users trade between 6-100 times in the period (29.9% of all transactions), who can be experienced investors or institutions. High-frequency users trade 101-10000 times in the period (19.0% of all transactions), which should

mainly be institutions. Ultra-high-frequency users trade more than 10000 times in the period (17.9% of all transactions), which are mainly the centralized exchanges serving different blockchain customers (e.g., the exchanges analyzed in Fig. 3d). The result shows that low-frequency and ultra-high-frequency users have lower delay costs on average than mid- and high-frequency users. The delay costs of ultra-high-frequency users are slightly higher than low-frequency users because the centralized exchanges are mainly used by retail investors, who are essentially the same cohort as low-frequency users. The delay costs of mid- and high-frequency users are not distinguishable from each other. The coefficient estimates are presented in Table B.2.

Third, Fig. 3c demonstrates the cost of delaying one minute for the 227 different hours in the sample. I group transactions by the hour in which they are submitted. The delay costs demonstrate a strong autocorrelation across hours. Since both centralized and decentralized crypto exchanges operate 24-7, the "opening effects" in traditional finance are not obvious here. There are two peaks in the sample period that might be related to shocks in the market. Overall, the delay costs are stable during the study period.

Fourth, Fig. 3d demonstrates the average delay costs for several large centralized exchanges, including Binance, Coinbase, Crypto.com, and Gemini. Most transactions from centralized exchanges are related to user deposits and withdrawals of cryptocurrencies. It is surprising to see that the delay costs of Binance transactions are higher than the three counterparts, which are not distinguishable from each other. This might reflect the demand for settlement speed from Binance users, but this may also be attributed to a more aggressive fee bidding strategy adopted by Binance. The coefficient estimates are presented in Table B.3. A similar analysis can be done for any addresses on Ethereum if the addresses perform enough transactions.

6.3 Discussion

This section discusses the potential issues that might bias the estimation of average delay costs in this paper.

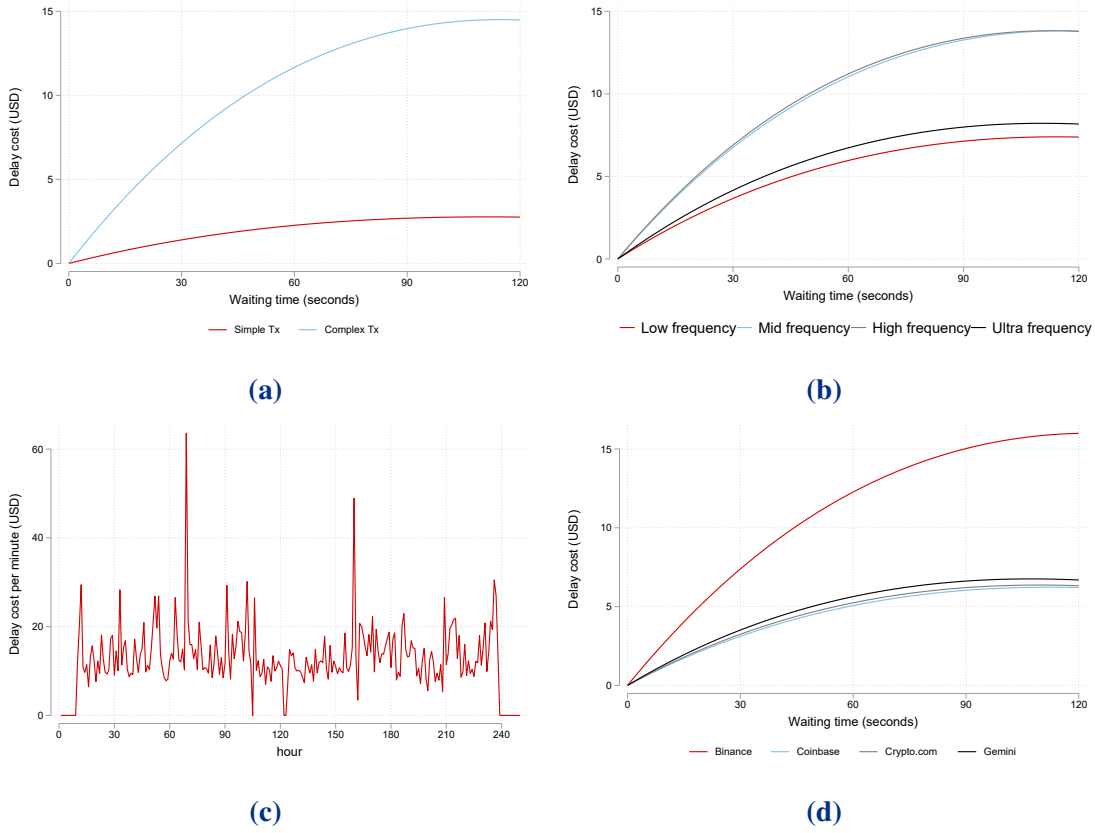


Figure 3: Panel (a) shows the average delay cost estimates by transaction types. Simple transactions are the ones that use exactly 21,000 gas (35% of all transactions), which transfers some amount of ETH from one address to another. Complex transactions are the ones that use more than 21,000 gas (65% of all transactions). Panel (b) shows the average delay cost estimates by trading frequency. Low-frequency users trade 1-5 times in the sample period (33.3% of all transactions); mid-frequency users trade between 6-100 times in the period (29.9% of all transactions); high-frequency users trade 101-10000 times in the period (19.0% of all transactions); ultra-high-frequency users trade more than 10000 times in the period (17.9% of all transactions). Panel (c) shows the estimates of the average costs of delaying transactions by a minute. Transactions are grouped by submission hour. Hour 0 is defined to be 12 a.m. July 25, 2021 UTC+0. Panel (d) shows the average delay cost estimates of several large centralized exchanges.

First, can the variance of bids reflect factors other than delay costs? For example, Li et al. (2014) attributes the lack of strategy of a part of consumers to behavioral myopia. Liu et al. (2022a) argues that Ethereum users sometimes overbid because of bounded rationality. However, if behavioral biases were the main drivers of the variance in gas price bids, amateur users would overbid more due to lack of experience and thus be estimated to have higher delay costs. Also, behavioral biases are supposed to be constant over time. These predictions from behavioral narratives are contradictory with the findings in Fig. 3b and Fig. 3c. Users who trade more frequently have higher average delay cost estimates, and the average delay cost estimates of users vary across time. Moreover, Capponi et al. (2022) analyzes the Ethereum transaction fees in a reduced-form way and finds that users bid high fees to reduce the execution risk of their orders due to blockchain congestion, which also supports the narrative of delay costs.

Second, can the assumption of the exogeneity of market conditions be violated? For example, users with lower delay costs may select to submit their transactions when the network is less congested. However, I argue that this is not likely to happen. While users can hide their private information by concealing their transactions, they waste the potential chances to settle their transactions when they conceal their transactions by submitting them later. The transactions that have incentives to conceal are the ones with low delay costs and high private information, but these two conditions are contradictory in nature. Thus, the subjective selection of users is not likely to bias the estimates.

Third, the non-linear nature of the waiting time distribution might limit the extrapolation of the estimates. This paper adopts relatively strong assumptions on the functional form of conditional waiting time distribution, which is modeled as a Gamma distribution with both parameters linear to gas price bids. The smooth delay cost function obtained from the estimation might be attributed to the parametric assumptions, and it should not be interpreted as a nature of delay cost functions. Moreover, because most transactions wait for less than a minute as shown in Fig. B.3, estimation for delay cost when time is large might be inaccurate due to limited sample size in

both the first and second stages of the estimation. Therefore, I only present the delay cost functions for waiting times up to 120 seconds. As a next step, both the first-stage and the second-stage estimation can be conducted in a non-parametric way as in Athey and Haile (2007) to improve the accuracy of the results.

Lastly, the estimation suffers from the missing data issue as introduced in Section 4. Waiting time of about 5% of the transactions is missing, and these transactions might be submitted directly to the block makers without entering the mempool. These transactions might have higher private values and delay costs than the average transactions. Hence, the estimates presented in this paper apply to the transactions publicly submitted to the mempool and serve as a lower bound for all transactions on Ethereum.

7 Application: Financial Market Design

7.1 Welfare Estimation

Ethereum uses the transaction fee mechanism introduced in Section 2.3 to allocate its block spaces during the study period. However, alternative mechanisms have been proposed and implemented to reduce transaction delays and improve user experience. Are these mechanisms optimal in minimizing the cost of transaction delays? How far away are they from the socially optimal case? The estimation of transaction delay costs sheds light on these questions. With the delay cost estimates for each user and transaction group, I conduct counterfactual simulation on alternative arrangements of Ethereum transactions and compare the welfare loss under different mechanisms.

To obtain heterogeneity in delay cost estimation, I group transactions by their transaction type (simple or complex), user experience (low, medium, high or ultra-high), and delay cost percentile estimated from the baseline regression. I then separately estimate the average delay

costs for each of these 800 groups of transactions. With knowledge of the waiting time,[12] the total welfare loss can be written as

$$L = \sum_{i \in \mathscr{J}} C(t_i).$$

This gives that delays on Ethereum trigger a daily welfare loss of 14.03 million USD.[13]

7.2 Counterfactual Analysis

I then conduct simulation exercises to estimate welfare losses under alternative transaction fee mechanisms. I compute the counterfactual waiting time for each transaction under two alternative transaction fee mechanisms and then compute the counterfactual welfare loss using the same methodology as in Section 7.1.[14]

First, I study the "socially optimal" case where transactions are settled according to their private delay costs. Transactions with higher delay costs are settled first. This is not a feasible mechanism in the real world because delay costs can only be measured ex-post, so this case serves as a theoretical optimum for comparison. Second, I study a "naive" first-in-first-out (FIFO) mechanism, which means that transactions are settled in the order of submission to the mempool.

Table 4 compares the outcomes under the benchmark mechanism with these two alternative mechanisms. The daily welfare loss due to transaction delays in the benchmark mechanism is only slightly larger than that in the "socially optimal" case that ranks transactions using ex-post information on delay cost. This is consistent with the theory in Theorem 3.1 that gas price bids

[12] Transactions waiting more than five minutes are usually beyond the effective domain of the delay cost function I estimate. Using the original function might result in a negative delay cost estimate. In these cases, I use the maximum value on the delay cost function as an alternative. For example, if a transaction waits for fifteen minutes but the delay cost function peaks at three minute, I adopt the peak value of the delay cost function as the delay cost for this function.

[13] In the sample period, the average delay cost for each transaction is 11.84 USD (though the median delay cost is much lower). 823 transactions are settled each minute on average. These estimates give the daily welfare loss of 14.03 million USD.

[14] Due to the limits in computation, I only simulate the 1,678,247 transactions seen by the network before July 27, 2021. This sample size should be large enough to deliver a reliable estimate of welfare loss due to delay costs.

Table 4: Simulated results of three transaction fee mechanisms

Variables		(1) Benchmark (by gas price)	(2) Counterfactual 1 (by delay cost)	(3) Counterfactual 2 (by time seen)
Waiting time (s)	(Quantiles)			
	10%	3.70	2.98	338.16
	25%	7.31	9.11	696.21
	50%	15.71	36.20	1401.16
	75%	37.46	202.51	2091.06
	90%	116.95	1401.61	2451.65
Observations		1,678,247	1,678,247	1,678,247
Delay cost (USD)	(Quantiles)			
	10%	0.60	0.97	3.02
	25%	1.48	2.66	6.17
	50%	3.83	5.77	13.55
	75%	10.10	11.91	31.55
	90%	24.17	23.20	66.81
Observations		1,678,247	1,678,247	1,678,247
Daily welfare loss (million USD)		14.03	13.83	41.17

Simulated quantiles of waiting time, quantiles of delay cost, and daily welfare loss under three different transaction fee mechanisms. Simulation is run for transactions in the sample but seen before July 27, 2021. Daily welfare loss is calculated assuming that 823 transactions are settled each minute, so daily welfare loss is the average daily transaction number times mean delay cost under each specific mechanism. Column (1) presents the benchmark transaction fee mechanism used in reality as introduced in Section 2.3, where transactions are ordered by their gas price bids. Column (2) presents the counterfactual mechanism where transactions are ordered by their ex-post delay costs. Transactions with higher estimated delay costs are settled first. Column (3) presents the counterfactual mechanism where transactions are ordered by their time of submission. Transactions submitted first are settled first.

reflect users' intrinsic delay costs. Controlling market conditions, the benchmark mechanism that orders transactions by gas prices performs almost equally well as the socially optimal case in terms of minimizing welfare loss. Moreover, the benchmark mechanism also performs better regarding waiting times for most transactions.

On the other side, the benchmark mechanism performs three times better than the "naive" first-in-first-out mechanism. Column (3) in Table 4 shows that substantial delays can emerge due to congestion in the network, which can cause significant welfare loss. This comparison underscores the importance of an effective transaction fee mechanism on blockchain efficiency.

7.3 Policy Implications

The welfare analysis in this section has several important implications for market design both in traditional finance and on blockchains.

First, Budish et al. (2015) proposes the mechanism of frequent batch auctions in the settlement of financial transactions. They propose to divide the trading day into frequent but discrete time intervals. All trade requests received during the same interval are treated as having arrived at the same (discrete) time. At the end of each interval, all outstanding orders are processed in batches, using a uniform-price auction. They believe that the introduction of a frequent batch auction can stop the socially wasteful arms race for network speed and transforms the competition on speed into competition on price.

The frequent batch auction model is very similar to the Ethereum transaction fee mechanism studied in this paper, except that the gaps between two blocks on Ethereum are random and much longer. Our analysis shows that frequent batch auction also introduces unnecessary delays to the transaction settlement, which can cause significant welfare loss for all users. While the problem might be mitigated in traditional finance with a much higher transaction throughput and lower transaction delays, the exact welfare implication of the frequent batch auction model needs further investigation.

Second, the welfare estimates show partial optimality of the benchmark transaction fee mechanism, which is still widely used on many other blockchain platforms including Bitcoin.[15] Existing literature like Roughgarden (2021); Chung and Shi (2022) evaluates different mechanism designs using criteria like incentive compatibility. This paper proposes a novel criterion - welfare loss due to transaction delay. I use counterfactual analysis to show that the benchmark transaction fee mechanism performs almost as well as the socially optimal mechanism. From this, I empirically hypothesize that a mechanism cannot perform significantly better if it differs from the benchmark mechanism only on demand-side features. This preliminary hypothesis would need support from theoretical proof, which is left to future works.

The benchmark mechanism might not be optimal in minimizing welfare loss due to transaction delays compared to other mechanisms that change supply-side features. For example, EIP-1559, a recent transaction fee mechanism reform on Ethereum implemented in August 2021, makes two changes: a) users bid gas prices in a mechanism similar to a second price auction instead of a first price auction and b) block gas limit becomes variable between 0 to 30 million instead of fixed at 15 million (though the target average is still 15 million). Liu et al. (2022a) finds that the reform significantly reduces the waiting time on Ethereum. This mechanism changes the supply-side feature of Ethereum by making the block size variable, so it might also bring significant welfare gains.

8 Conclusion

This paper estimates the average delay costs of the DeFi transactions on Ethereum. With a dynamic auction model that links the users' per unit gas price bids to the delay costs and market conditions, the average delay cost functions of any sample of transactions can be identified and estimated. The estimation is first conducted on all transactions on Ethereum and then on different

[15] In fact, this mechanism was first proposed in the Bitcoin white paper (Nakamoto (2008)) and also called Bitcoin Payment System (BPS) as analyzed in Easley et al. (2019); Huberman et al. (2021).

groups of transactions by user and transaction characteristics. The results are then applied to empirically evaluate different blockchain market designs by the welfare losses.

This paper shows that delays are costly for financial transactions. One natural direction of future work is to study whether better market designs can reduce the waiting time in financial transactions. For example, Liu et al. (2022a) finds that EIP-1559 causally reduces the waiting time on Ethereum and improves user experiences, but the reason behind this is not well understood. More theoretical works are needed to analyze the transaction fee mechanisms from a dynamic perspective. In addition, while the dynamic auction model in this paper involves the conditional waiting time distribution, it is taken as an exogenous variable related to market conditions and empirically estimated from the data. I leave the work of endogenizing the equilibrium gas price and waiting time for future studies.

The cost of delay in the Ethereum transaction fee market, while not directly comparable, resembles that in the general financial market. The risk of being frontrun and execution failure compose the main source of delay cost on the Ethereum blockchain, and these risk factors widely exist in any financial market. This paper thus sheds light on the general understanding of delay costs in finance.

While the previous studies in blockchain economics mostly apply economics and finance theories to understand phenomena in decentralized finance, this paper demonstrates that observations and insights gained on decentralized finance can have general economic implications. As an important market from an industrial organization and market design perspective, the decentralized finance on Ethereum provides a novel setting to study various phenomena that are otherwise difficult in economics.

References

Afèche, Philipp and Haim Mendelson (2004) "Pricing and Priority Auctions in Queueing Systems with a Generalized Delay Cost Structure," *Management Science*, 50, 869–882, 10.1287/mnsc.1030.0156.

Agness, Daniel, Travis Baseler, Sylvain Chassang, Pascaline Dupas, and Erik Snowberg (2022) "Valuing the Time of the Self-Employed," 10.3386/w29752.

Aguiar, Mark and Erik Hurst (2007) "Life-Cycle Prices and Production," *American Economic Review*, 97, 1533–1559, 10.1257/aer.97.5.1533.

Athey, Susan and Philip A. Haile (2007) "Chapter 60 Nonparametric Approaches to Auctions," *Handbook of Econometrics*, 3847–3965, 10.1016/s1573-4412(07)06060-6.

Becker, Gary S (1965) "A Theory of the Allocation of Time," *The Economic Journal*, 75, 493–517, 10.2307/2228949.

Buchholz, Nicholas, Laura Doval, Jakub Kastl, Filip Matjka, and Tobias Salz (2020) "The Value of Time: Evidence From Auctioned Cab Rides," 10.3386/w27087.

Budish, Eric, Peter Cramton, and John Shim (2015) "The High-Frequency Trading Arms Race: Frequent Batch Auctions as a Market Design Response," *The Quarterly Journal of Economics*, 130, 1547–1621, 10.1093/qje/qjv027.

Buterin, Vitalik (2014) "Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform. By Vitalik Buterin (2014)."

Capponi, Agostino, Ruizhe Jia, and Shihao Yu (2022) "The Information Content of Blockchain Fees," *SSRN Electronic Journal*, 10.2139/ssrn.4236993.

Che, Yeon-Koo and Olivier Tercieux (2020) "Optimal Queue Design," *SSRN Electronic Journal*, 10.2139/ssrn.3743663.

Chung, Hao and Elaine Shi (2022) "Foundations of Transaction Fee Mechanism Design."

Daian, Philip, Steven Goldfeder, Tyler Kell, Yunqi Li, Xueyuan Zhao, Iddo Bentov, Lorenz Breidenbach, and Ari Juels (2020) "Flash Boys 2.0: Frontrunning in Decentralized Exchanges, Miner Extractable Value, and Consensus Instability," 05, 10.1109/SP40000.2020.00040.

Deacon, Robert T. and Jon Sonstelie (1985) "Rationing by Waiting and the Value of Time: Results from a Natural Experiment," *Journal of Political Economy*, 93, 627647.

Demsetz, Harold (1968) "The Cost of Transacting," *The Quarterly Journal of Economics*, 82, 33, 10. 2307/1882244.

Easley, David, Maureen O'Hara, and Soumya Basu (2019) "From mining to markets: The evolution of bitcoin transaction fees," *Journal of Financial Economics*, 134, 91–109, 10.1016/j.jfineco.2019.03.004.

Etherscan (2022) "Ethereum Charts and Statistics | Etherscan."

Eyal, Ittay, Adem Efe Gencer, Emin Gun Sirer, and Robbert Van Renesse (2016) "Bitcoin-NG: A Scalable Blockchain Protocol," in *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*, 45–59, Santa Clara, CA: USENIX Association, March.

Farmer, J. Doyne and Spyros Skouras (2012) "UK Governments Foresight Project," 03.

Goldszmidt, Ariel, John List, Robert Metcalfe, Ian Muir, V. Kerry Smith, and Jenny Wang (2020) "The Value of Time in the United States: Estimates from Nationwide Natural Field Experiments," 10.3386/ w28208.

Goolsbee, Austan and Peter J Klenow (2006) "Valuing Consumer Products by the Time Spent Using Them: An Application to the Internet," *American Economic Review*, 96, 108–113, 10.1257/ 000282806777212521.

Gudgeon, Lewis, Pedro Moreno-Sanchez, Stefanie Roos, Patrick McCorry, and Arthur Gervais (2020) "SoK: Layer-Two Blockchain Protocols," in Bonneau, Joseph and Nadia Heninger eds. *Financial Cryptography and Data Security*, 201–226, Cham: Springer International Publishing.

Handa, Puneet and Robert A. Schwartz (1996) "Limit Order Trading," *The Journal of Finance*, 51, 1835–1861, 10.1111/j.1540-6261.1996.tb05228.x.

Hansen, Bruce (2004) "Nonparametric Conditional Density Estimation."

Harvey, Campbell R, Ashwin Ramachandran, and Joey Santoro (2021) *Defi And The Future Of Finance.*: John Wiley.

Hassin, Rafael (1995) "Decentralized Regulation of a Queue," *Management Science*, 41, 163–173, 10.1287/mnsc.41.1.163.

Hassin, Refael (2016) *Rational queueing*: CRC Press.

Hendel, Igal and Aviv Nevo (2006) "Measuring the Implications of Sales and Consumer Inventory Behavior," *Econometrica*, 74, 1637–1673, 10.1111/j.1468-0262.2006.00721.x.

——— (2013) "Intertemporal Price Discrimination in Storable Goods Markets," *American Economic Review*, 103, 2722–2751, 10.1257/aer.103.7.2722.

Holt, Charles A. and Roger Sherman (1982) "Waiting-Line Auctions," *Journal of Political Economy*, 90, 280294.

Huberman, Gur, Jacob D Leshno, and Ciamac Moallemi (2021) "Monopoly without a Monopolist: An Economic Analysis of the Bitcoin Payment System," *The Review of Economic Studies*, 88, 10.1093/restud/rdab014.

Kalodner, Harry, Steven Goldfeder, Xiaoqi Chen, S. Matthew Weinberg, and Edward W. Felten (2018) "Arbitrum: Scalable, private smart contracts," in *27th USENIX Security Symposium (USENIX Security 18)*, 1353–1370, Baltimore, MD: USENIX Association, August.

Kittsteiner, Thomas and Benny Moldovanu (2005) "Priority Auctions and Queue Disciplines That Depend on Processing Time," *Management Science*, 51, 236248.

Lavi, Ron, Or Sattath, and Aviv Zohar (2019) "Redesigning Bitcoin's fee market," *The World Wide Web Conference on - WWW '19*, 10.1145/3308558.3313454.

Lewis, G. and P. Bajari (2014) "Moral Hazard, Incentive Contracts, and Risk: Evidence from Procurement," *The Review of Economic Studies*, 81, 1201–1228, 10.1093/restud/rdu002.

Lewis, Gregory and Patrick Bajari (2011) "Procurement Contracting With Time Incentives: Theory and Evidence," *The Quarterly Journal of Economics*, 126, 11731211.

Li, Jun, Nelson Granados, and Serguei Netessine (2014) "Are Consumers Strategic? Structural Estimation from the Air-Travel Industry," *Management Science*, 60, 2114–2137, 10.1287/mnsc.2013.1860.

Liu, Yulin, Yuxuan Lu, Kartik Nayak, Fan Zhang, Luyao Zhang, and Yinhong Zhao (2022a) "Empirical Analysis of EIP-1559: Transaction Fees, Waiting Time, and Consensus Security," *arXiv.org*, 10.48550/arXiv.2201.05574.

——— (2022b) "Replication Data for: "Empirical Analysis of EIP-1559: Transaction Fees, Waiting Time, and Consensus Security"," *Harvard Dataverse*, doi:10.7910/DVN/K7UYPI.

Louis, Brian (2017) "Trading Fortunes Depend on a Mysterious Antenna in an Empty Field," 05.

Nakamoto, Satoshi (2008) "Bitcoin: a Peer-to-Peer Electronic Cash System," 10.

Naor, P. (1969) "The Regulation of Queue Size by Levying Tolls," *Econometrica*, 37, 15, 10.2307/1909200.

Nevo, Aviv and Arlene Wong (2018) "The Elasticity of Substitution Between Time and Market Goods: Evidence from the Great Recession," *International Economic Review*, 60, 25–51, 10.1111/iere.12343.

Papanastasiou, Yiangos and Nicos Savva (2017) "Dynamic Pricing in the Presence of Social Learning and Strategic Consumers," *Management Science*, 63, 919–939, 10.1287/mnsc.2015.2378.

Parlour, Christine A. (1998) "Price Dynamics in Limit Order Markets," *Review of Financial Studies*, 11, 789–816, 10.1093/rfs/11.4.789.

Perrigne, Isabelle and Quang Vuong (2019) "Econometrics of Auctions and Nonlinear Pricing," *Annual Review of Economics*, 11, 27–54, 10.1146/annurev-economics-080218-025702.

Platform, Google Cloud (2019) "Big Query Public Data - crypto_ethereum," 01.

Roughgarden, Tim (2020) "Transaction Fee Mechanism Design for the Ethereum Blockchain: An Economic Analysis of EIP-1559."

——— (2021) "Transaction Fee Mechanism Design," *Proceedings of the 22nd ACM Conference on Economics and Computation*, 10.1145/3465456.3467591.

Townsend, Robert M (2020) *Distributed ledgers : design and regulation of financial infrastructure and payment systems*: The Mit Press.

Tsabary, Itay, Matan Yechieli, Alex Manuskin, and Ittay Eyal (2021) "MAD-HTLC: Because HTLC is Crazy-Cheap to Attack," in *2021 IEEE Symposium on Security and Privacy (SP)*, 1230–1248, 10.1109/SP40001.2021.00080.

Wah, Elaine and Michael P. Wellman (2013) "Latency arbitrage, market fragmentation, and efficiency," *Proceedings of the fourteenth ACM conference on Electronic commerce - EC '13*, 10.1145/2492002.2482577.

Wood, Gavin (2022) "Ethereum: A Secure Decentralised Generalised Transaction Ledger," 10.

Yao, Andrew Chi-Chih (2018) "An Incentive Analysis of some Bitcoin Fee Designs."

Zhou, Qiheng, Huawei Huang, Zibin Zheng, and Jing Bian (2020) "Solutions to Scalability of Blockchain: A Survey," *IEEE Access*, 8, 16440–16455, 10.1109/access.2020.2967218.

A Mathematical Appendix

A.1 Proof of Theorem 3.1

For simplicity purpose, all subscript of ij are omitted here. The case when for any $b > 0$, $\mathbb{E}(u(v,b)) < 0$ is trivial because the only choice is to not submit any bids for a zero realized utility.

Suppose there exists $b > 0$ such that $\mathbb{E}[u(v,b)] \geq 0$. Recall that $\mathbb{E}[U(v,b)] = v - \int_{t=0}^{\infty} C(t) dW_M(t|b) - bg$. Using integral by parts, we have

$$\int_{t=0}^{\infty} C(t) dW_b(t) = C(t) W_M(t|b)|_{t=0}^{t=\infty} - \int_{t=0}^{\infty} W_M(t|b) C'(t) dt = - \int_{t=0}^{\infty} W_M(t|b) C'(t) dt$$

because $C(t) = 0$ when $t = 0$ and $W_M(t|b) = 0$ when $t \to \infty$. Then, the first-order condition in Eq. (5) can be rewritten as

$$\int_{t=0}^{\infty} \frac{\partial W_b(t)}{\partial b} C'(t) dt = g$$

Let $f(b) = \int_{t=0}^{\infty} \frac{\partial W_b(t)}{\partial b} C'(t) dt$. Then by Leibniz's Rule, $f'(b) = \int_{t=0}^{\infty} \frac{\partial^2 W_b(t)}{\partial b^2} C'(t) dt$. Knowing that $\frac{\partial^2 W_b(t)}{\partial b^2} < 0$ and $C(t)$ is monotonically increasing in t, which gives $C'(t) > 0$, so $\frac{\partial^2 W_b(t)}{\partial b^2} C'(t) < 0$. Therefore, $f'(b) < 0$ for any $b, t > 0$. This means that $f(b)$ is monotonically decreasing, which guarantees the uniqueness of the solution to Eq. (5).

Since $\frac{\partial W_b(t)}{\partial b} > 0$ and $C'(t) > 0$, $f(b) > 0$ for any $b > 0$. Notice that $f(b) \to 0$ when $b \to \infty$ because $\frac{\partial W_b(t)}{\partial b} \to 0$, and $f(b) \to \infty$ when $b \to 0$ because $\frac{\partial W_b(t)}{\partial b} \to \infty$. By intermediate value theorem, there exists $\tilde{b} > 0$ such that $f(b) = g$. Therefore, the first-order condition Eq. (5) has a unique positive solution $\tilde{b}$.

It has been shown that $f'(b) < 0$ for any $b > 0$, the $\frac{\partial^2 \mathbb{E}[U(v,\tilde{b})]}{\partial b^2} = f'(\tilde{b}) < 0$, which means that $\tilde{b}$ is a global maximum of $\mathbb{E}[U(v,b)]$. However, the user would make the bid if and only if the expected utility is positive. Therefore, Eq. (6) holds, which concludes the proof.

A.2 Example of Theorem 3.1

Suppose the delay cost function is linear $C_{ij}(t) = c_{ij}t$. Suppose $W_M|b$ follows a exponential distribution $W_M|b \sim Exp(\lambda_b)$, where $\lambda_b = mb$, m being a constant depending on the network conditions. So $w_M(t|b) = mbe^{-mbt}$. Then the expected utility function in Eq. (3) can be written as

$$\mathbb{E}(u_{ij}(v,b)) = \int_{t=0}^{\infty}(v_{ij} - c_{ij}t)mbe^{-mbt}dt - b = v_{ij} - \frac{c_{ij}}{mb} - b. \tag{8}$$

The first order condition is then $\frac{c_{ij}}{mb^2} = 1$, which gives

$$\tilde{b_{ij}} = \sqrt{\frac{c}{m}} > 0. \tag{9}$$

The second order condition is

$$\frac{d^2\mathbb{E}(u_{ij}(v,b))}{db^2} = -\frac{c_{ij}}{2mb^3} < 0,$$

so $\tilde{b_{ij}}$ is the global maximum of $\mathbb{E}(u_{ij}(v,b))$ when $b > 0$. $b^*_{ij} = \tilde{b_{ij}}$ when $\mathbb{E}(u_{ij}(v,b^*_{ij})) > 0$, which is equivalent to $v_{ij} > 2\sqrt{\frac{c_{ij}}{m}}$. Therefore,

$$b^*_{ij} = \begin{cases} 0 & \text{if } v_{ij} \leq 2\sqrt{\frac{c_{ij}}{m}}, \\ \sqrt{\frac{c_{ij}}{m}} & \text{if } v_{ij} > 2\sqrt{\frac{c_{ij}}{m}}. \end{cases} \tag{10}$$

Notice that the optimal bid b^*_{ij} strictly increases as the delay cost c_{ij} increases or if the network parameter m decreases, which means the network becomes more congested. Also notice that the optimal bid b^*_{ij} is strictly smaller than the level of $O(v)$ in a static first-price auction.

B Additional Graphs and Tables

APPENDIX G. FEE SCHEDULE

The fee schedule G is a tuple of 31 scalar values corresponding to the relative costs, in gas, of a number of abstract operations that a transaction may effect.

Name	Value	Description*
G_{zero}	0	Nothing paid for operations of the set W_{zero}.
G_{base}	2	Amount of gas to pay for operations of the set W_{base}.
$G_{verylow}$	3	Amount of gas to pay for operations of the set $W_{verylow}$.
G_{low}	5	Amount of gas to pay for operations of the set W_{low}.
G_{mid}	8	Amount of gas to pay for operations of the set W_{mid}.
G_{high}	10	Amount of gas to pay for operations of the set W_{high}.
$G_{extcode}$	700	Amount of gas to pay for operations of the set $W_{extcode}$.
$G_{balance}$	400	Amount of gas to pay for a BALANCE operation.
G_{sload}	200	Paid for a SLOAD operation.
$G_{jumpdest}$	1	Paid for a JUMPDEST operation.
G_{sset}	20000	Paid for an SSTORE operation when the storage value is set to non-zero from zero.
G_{sreset}	5000	Paid for an SSTORE operation when the storage value's zeroness remains unchanged or is set to zero.
R_{sclear}	15000	Refund given (added into refund counter) when the storage value is set to zero from non-zero.
$R_{suicide}$	24000	Refund given (added into refund counter) for suiciding an account.
$G_{suicide}$	5000	Amount of gas to pay for a SUICIDE operation.
G_{create}	32000	Paid for a CREATE operation.
$G_{codedeposit}$	200	Paid per byte for a CREATE operation to succeed in placing code into state.
G_{call}	700	Paid for a CALL operation.
$G_{callvalue}$	9000	Paid for a non-zero value transfer as part of the CALL operation.
$G_{callstipend}$	2300	A stipend for the called contract subtracted from $G_{callvalue}$ for a non-zero value transfer.
$G_{newaccount}$	25000	Paid for a CALL or SUICIDE operation which creates an account.
G_{exp}	10	Partial payment for an EXP operation.
$G_{expbyte}$	10	Partial payment when multiplied by $\lceil \log_{256}(exponent) \rceil$ for the EXP operation.
G_{memory}	3	Paid for every additional word when expanding memory.
$G_{txcreate}$	32000	Paid by all contract-creating transactions after the *Homestead transition*.
$G_{txdatazero}$	4	Paid for every zero byte of data or code for a transaction.
$G_{txdatanonzero}$	68	Paid for every non-zero byte of data or code for a transaction.
$G_{transaction}$	21000	Paid for every transaction.
G_{log}	375	Partial payment for a LOG operation.
$G_{logdata}$	8	Paid for each byte in a LOG operation's data.
$G_{logtopic}$	375	Paid for each topic of a LOG operation.
G_{sha3}	30	Paid for each SHA3 operation.
$G_{sha3word}$	6	Paid for each word (rounded up) for input data to a SHA3 operation.
G_{copy}	3	Partial payment for *COPY operations, multiplied by words copied, rounded up.
$G_{blockhash}$	20	Payment for BLOCKHASH operation.

Figure B.1: Amount of gas consumed by each operation on Ethereum. Figure sources from Ethereum Yellow Paper (Wood (2022)).

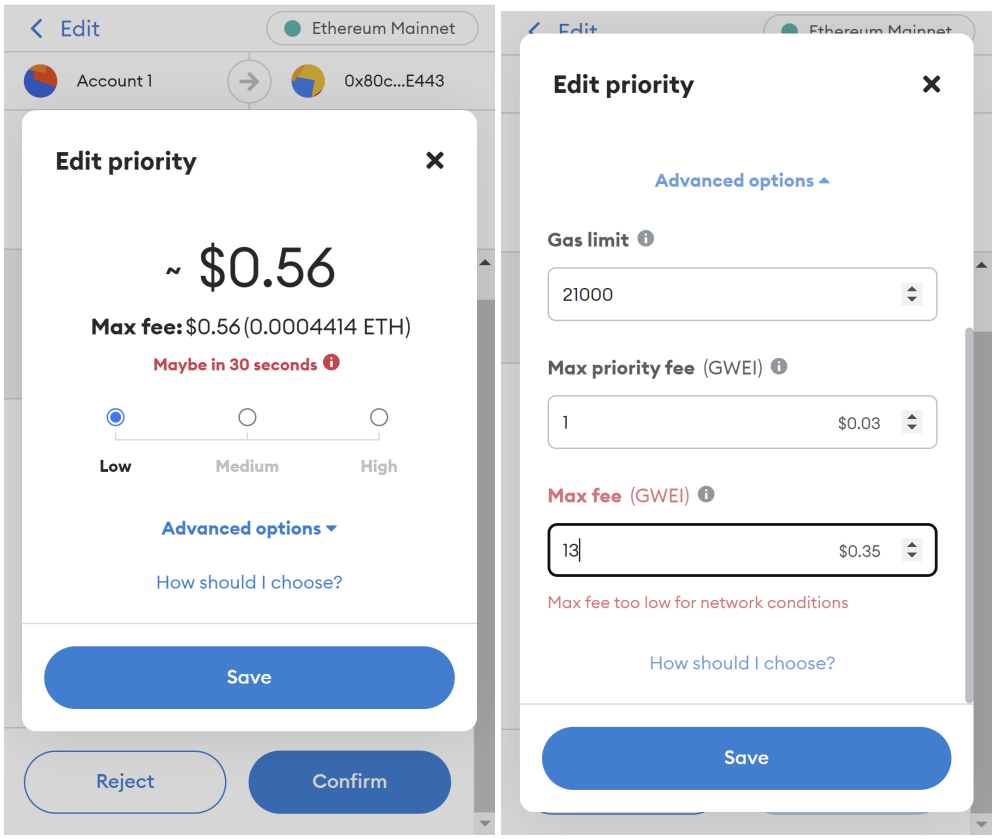
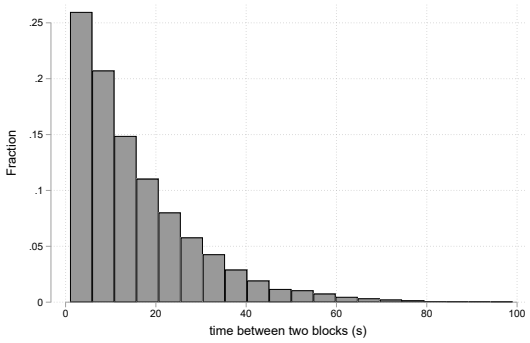
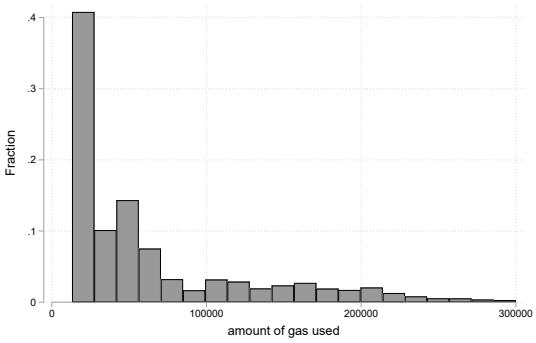


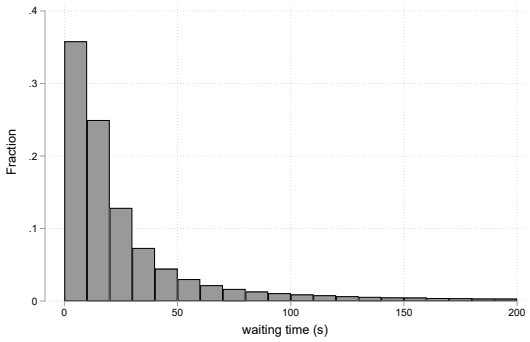
Figure B.2: Example of fee bidding in MetaMask. The graph on the left shows the basic fee bidding interface, where the software offers three levels of options for preliminary users to choose. Expected waiting time is explicitly shown on the top. Users who don't understand the mechanism of fee bidding can click on "How should I choose?" to learn more. The graph on the right show the interface after more advanced users click on "Advanced options" in the left graph. Users can bid exact number of per unit gas price here. If the gas prices users input are too low, the software will show reminders. Therefore, even less experienced users should be able to form reliable expectations about waiting time with the assistance of software.



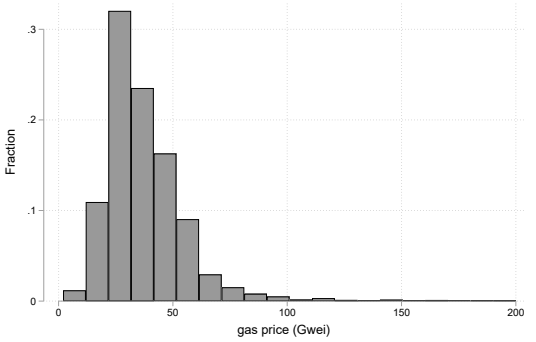
(a) Time between two blocks



(b) Gas used by each transaction



(c) Waiting time of each transaction



(d) Gas price paid of each transaction

Figure B.3: Distribution of several variables

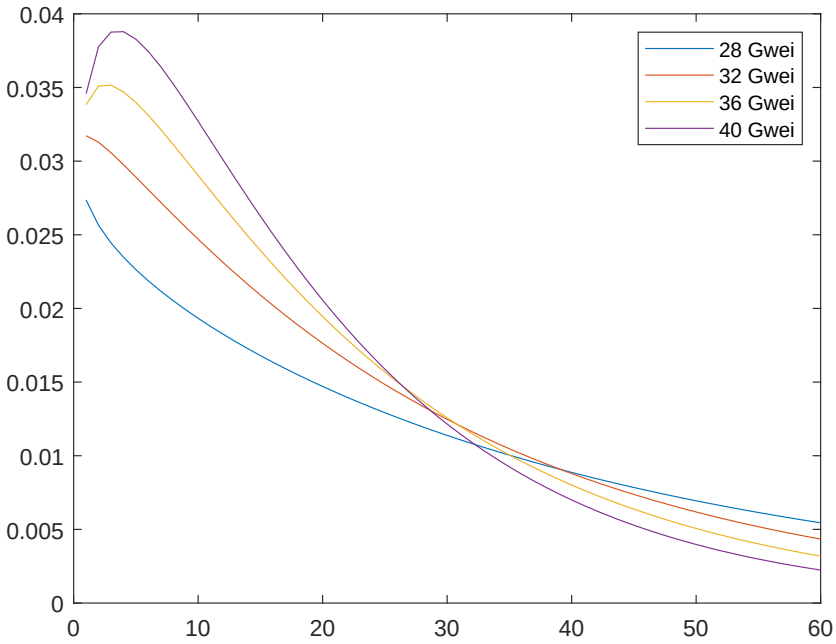


Figure B.4: Example of first-stage density estimates - result obtained by averaging over the density estimates over all five-minute periods

Table B.1: Coefficient estimates for average delay costs of different types of transactions

	(1)	(2)
VARIABLES	Simple Tx	Complex Tx
c_1	-22,746***	-115,255***
	(16.33)	(143.4)
c_2	142.9***	700.0***
	(0.136)	(1.201)
c_3	-0.280***	-1.317***
	(0.000337)	(0.00295)
c_4	0.000255***	0.00115***
	(3.84e-07)	(3.27e-06)
c_5	-6.32e-08***	-2.72e-07***
	(1.21e-10)	(9.83e-10)
Observations	3486309	6275903
R-squared	0.520	0.194

Standard errors in parentheses
*** p<0.01, ** p<0.05, * p<0.1

Table B.2: Coefficient estimates for average delay costs of users with different trading frequencies

VARIABLES	(1) Low frequency	(2) Mid frequency	(3) High frequency
c_1	-38,881***	-95,902***	-95,000***
	(89.43)	(152.2)	(164.7)
c_2	244.0***	575.5***	577.3***
	(0.740)	(1.267)	(1.415)
c_3	-0.482***	-1.072***	-1.076***
	(0.00184)	(0.00309)	(0.00354)
c_4	0.000448***	0.000928***	0.000923***
	(2.11e-06)	(3.42e-06)	(3.99e-06)
c_5	-1.15e-07***	-2.19e-07***	-2.13e-07***
	(6.72e-10)	(1.02e-09)	(1.22e-09)
Observations	1538488	4726526	3672589
R-squared	0.216	0.171	0.153

Standard errors in parentheses
*** p<0.01, ** p<0.05, * p<0.1

Table B.3: Coefficient estimates for average delay costs of different centralized exchanges

VARIABLES	(1) Binance	(2) Coinbase	(3) Crypto.com	(4) Gemini
c_1	-116,979***	-50,401***	-52,604***	-57,290***
	(211.0)	(105.7)	(282.1)	(519.9)
c_2	647.9***	310.4***	329.6***	370.8***
	(2.202)	(0.985)	(2.584)	(4.600)
c_3	-1.075***	-0.583***	-0.617***	-0.739***
	(0.00569)	(0.00265)	(0.00648)	(0.0118)
c_4	0.000823***	0.000513***	0.000510***	0.000684***
	(6.20e-06)	(3.26e-06)	(7.15e-06)	(1.36e-05)
c_5	-1.73e-07***	-1.26e-07***	-1.06e-07***	-1.76e-07***
	(1.75e-09)	(1.11e-09)	(2.18e-09)	(4.34e-09)
Observations	654706	523020	71180	33607
R-squared	0.441	0.457	0.518	0.443

Standard errors in parentheses
*** p<0.01, ** p<0.05, * p<0.1