DYNAMIC TRANSACTION FEE MECHANISM DESIGN

MALLESH M. PAI AND MAX RESNICK

SPECIAL MECHANISMS GROUP

ABSTRACT: Transaction fee mechanism design for blockchains is an active field of research; however, much of the current literature focuses on a static model in which users only care about inclusion in the next block. In contrast, this paper studies the design of transaction fee mechanisms in a more realistic dynamic setting where both transactions and blocks arrive over time and users face a dynamic tradeoff between paying more for immediate inclusion or paying less for delayed inclusion.

In this context we show two main results. First we show that Proof-of-Stake chains, where blocks arrive at deterministic intervals, have lower equilibrium congestion and bids than Proof-of-Work chains with the same demand and throughput. Second, we investigate EIP-1559, a dynamic reserve price mechanism introduced to Ethereum in 2021. We show that, unlike in the static model, EIP-1559 is not Dominant Strategy Incentive Compatible (DSIC) in the dynamic model. Furthermore we show that rapid updates to the reserve price can lead to substantial congestion for low willingness-to-pay users. We use these findings to study how to improve on EIP-1559. In particular, we argue that slower updating rules for EIP-1559 may alleviate the delays for low types.

E-mail address: `mallesh.pai@mechanism.org`, `max.resnick@mechanism.org`.

1. INTRODUCTION

Popular blockchains face a difficult resource allocation problem: the supply of blockspace is limited and inelastic, while demand is highly stochastic and often exceeds supply. Therefore, blockchains must ration blockspace. Early blockchains such as Bitcoin and Ethereum (pre-2021) used a pay-your-bid mechanism, in which users specify a transaction fee that the miner/proposer receives if they include the user's transaction in their block. While easy to describe and execute from the perspective of the chain, this mechanism has drawbacks from the user's perspective.[1] Figuring out how much to bid for inclusion can be difficult for an uninformed user. Furthermore, even when users bid optimally, the efficient equilibrium may lead to high ex-post regret for the user. Motivated by this, Ethereum switched in August 2021 to a mechanism known as EIP-1559, a first-price auction with a dynamically updated reserve price.[2]

A growing literature has studied the design of fee mechanisms under the unique constraints imposed by blockchains. We describe this literature in further detail below in Section 1.2. A majority of this work, however, does not explicitly consider dynamic incentives and their interaction with the mechanism. In particular, a valid signed transaction is eligible for inclusion in any block after it is submitted. In such a setting, a user inherently faces a dynamic tradeoff between faster inclusion and lower fees. In thinking about transaction fee mechanisms, therefore, it is important to think not only about the incentives for the various parties within the context of a single block as has been studied in the literature, but also about how these incentives play out over multiple blocks. This paper carries out precisely such an analysis.

Transaction fee mechanisms aim to balance three core desiderata. The first is User Experience (UX): the mechanism should be easy for a user to participate in. Following the market design literature (see, e.g., Pathak and Sönmez (2008)) this is often modeled as dominant strategy incentive compatibility (DSIC): that is, truthful bidding is a dominant strategy. The second is a throughput constraint: blockchains are computationally constrained environments, and therefore chains often require that the average number of transactions processed not exceed some given amount even if individual blocks can be larger.[3] A third desideratum is implementability: inclusion into a block is determined

[1]Bitcoin and Ethereum are currently the largest blockchains by market capitalization, at $\sim$ \$945 and \$300 billion respectively.

[2]EIP, short for Ethereum Improvement Proposal, is the primary way Ethereum considers and implements upgrades.

[3]In practice, transaction sizes vary and measured in units of *gas*. Ethereum blocks can use up to 30 million gas but the chain targets an average usage of 15M gas per block, a number seen as balancing the needs of validators to keep up with the chain as blocks are produced. We abstract from these complications here in our model and simply consider a count of transactions, with a maximum number that can be included in any single block, and a target average number.

by profit-maximizing miners/ proposers/ producers can enter into off-chain side agreements with users etc., changing the effective mechanism faced by the user—see, e.g., Roughgarden (2020). The goal of the designer therefore is to come up with a mechanism that meets all these three desiderata. This is non-trivial. For example, the pay-your-bid mechanism described above is implementable. However, optimally bidding in such a mechanism may be difficult, and further, if the underlying demand is high, the resulting throughput may be higher than desired. Throttling this throughput can be achieved by setting a well-chosen reserve price: the question then arises—how should such a reserve be set? EIP-1559 implements a dynamic reserve price which can go up or down by up to 12.5% per block depending on the usage in the previous block. Further Roughgarden (2020) shows that it is "typically" DSIC in a single block, while Leonardos et al. (2022) show that the dynamic reserves achieve the stated goal of 15M gas per block. Is transaction-fee mechanism design then solved?

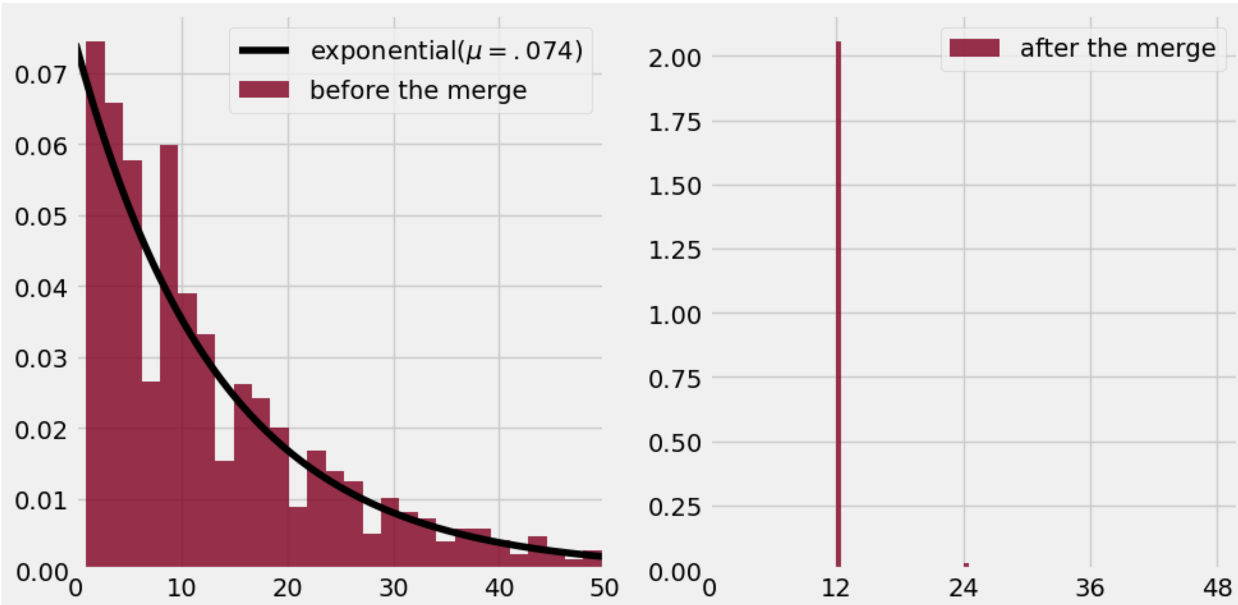
In this paper, we argue that it is not. We describe our model and results in a little more detail in Section 1.1, but prior to that, we wish to discuss the advantages of a dynamic approach to transaction fee mechanism design. First, our results show that the choice of transaction fee mechanism can have subtle interactions with the dynamics of the blockchain. For example, we consider a move from Proof-of-Work (PoW), where blocks arrive at exponentially distributed intervals, to Proof-of-Stake (PoS), where blocks arrive at deterministic, equally spaced intervals— Ethereum undertook precisely this move during the Merge in September 2022 (see Figure 1). We show that even if average block times, block sizes, and demand for blockspace are the same, the transition to proof of stake results in significantly lower congestion.

Second, switching from a static to a more realistic dynamic model exposes new weaknesses for mechanisms that behave well in the static setting. For example, Roughgarden (2020) showed that EIP-1559 is close to dominant strategy incentive compatible (DSIC) for sufficiently high maximum block size in the static setting. Roughly speaking, unless the block is full, the reserve price acts as a posted price, and there is no need to strategize in a static setting. But in the dynamic setting, where users face a tradeoff between fast inclusion and low fees, we show that users once again must behave strategically in order to maximize their welfare.[4]

More surprisingly, we show that EIP-1559 may cause substantial delays for low bids, incentivizing bidders to bid *more* than they would have under the pay-your-bid mechanism! Formally, we compare two mechanisms: the classic pay-your-bid mechanism and a simplified EIP-1559 that is amenable to analytic solutions. We show that even though

[4]Indeed, there is already active research considering how time-insensitive users may optimally time the submission of transactions to the blockchain in the face of such dynamic reserve prices: see, e.g., Mamageishvili and Felten (2022) or Bar-On and Mansour (2023).

FIGURE 1. Block Arrival Process Before and After The Merge



Note: Data is from the week before block 15537393 (The Merge block) and the week after. Note that 24 second arrivals occasionally happened after the merge due to missed slots. The exponential distribution plotted on the left has parameter $\mu = \frac{1}{13.5}$, which corresponds to the average arrival rate of blocks before the merge.

both mechanisms would have the same throughput in steady state, i.e., both would result in the same average number of transactions served per block in steady state, users pay more under EIP-1559 and low-paying users face longer delays! The reason is subtle, but akin to the standard monopoly distortion in second-degree price discrimination, or, equivalently, reserve prices in an optimal static auction a la Myerson (1981). There, the seller extracts more revenue from high willingness-to-pay buyers by withholding from low willingness-to-pay buyers. Here, users are time-sensitive and are paying for priority, i.e., to be delayed less. By delaying low willingness-to-pay users' transactions, the mechanism extracts higher revenues analogously. Our results therefore clear up a misunderstanding in the literature— for example several influential papers have argued that the dynamic reserve-price adjustment process in EP-1559 results in the usage of the chain converging to the desired level Leonardos et al. (2021, 2022). A folk argument that is popular in the community therefore suggests that the resulting prices/ revenues must be the "correct" market prices. We show emphatically that this is not the case!

1.1. *Overview of Model and Results*

Our starting point is the following simple dynamic model that well-approximates Bitcoin, and Ethereum pre-2022: users arrive at a Poisson rate λ and choose how much to bid for inclusion of their transaction. A block producer is selected at a fixed exponential rate μ, and selects up to K transactions for inclusion—in particular, the producer selects the (up to) K highest bidders.[5] In the formalism of queueing theory, this is what is known as a $M/M/1$ queue with bulk service. The M/M in M/M/1 signifies that both the arrival and service process are memoryless, while the 1 signifies that there is a single queue. Bulk service corresponds to the fact that multiple transactions may be included in the same block. The analysis of bidding in this auction was first considered in Huberman et al. (2021), and our analysis mirrors theirs.

The techniques developed in the study of these queues therefore allow us to consider how the incentives change if the dynamics of the blockchain change. For example, Ethereum switched in 2022 from a Proof-of-Work chain to a Proof-of-Stake chain. In Proof-of-Stake, blocks are not produced at exponentially distributed intervals but instead (at least to a first approximation) at deterministic time intervals—formally, this is now a $M/D/1$ queue with bulk service (the D stands for deterministic).[6] Our first novel set of results considers bidding for inclusion in such a blockchain. In particular, we show that, even if the deterministic rate of production *equals* the expected rate of block production in the prior Proof-of-Stake world, so that the expected amount of blockspace available per unit time stays constant, the average waiting time, average congestion, and consequently the bids are reduced by a factor of 2.

This replicates intuitions from queueing theory. Under exponential block arrivals, blocks sometimes stochastically arrive early, which typically means few transactions have arrived since the last block resulting in wasted blockspace, or alternatively,stochastically arrive late, which makes it more likely that the number of transactions that have arrived since the last block was mined exceed available blockspace, resulting in delays for (some) users. In contrast, block arrival on proof of stake chains is deterministic leading to more efficient use of the available blockspace and consequently lower wait times. Further, since the incentive to bid is to gain priority, lower delays imply lower equilibrium bids.

Having constructed this model, we focus on the transaction fee mechanism currently used by Ethereum, which is known as EIP-1559. This mechanism was formally studied in the work of Roughgarden (2020), but at its core can be thought of as a first price auction with a dynamic reserve price. This reserve price targets half full blocks in accordance with the long term state growth constraints of the chain. The reserve price depends on the previous block's reserve price and on how many transactions were in the previous block. If the previous block had a reserve r_t and contained k_t transactions ($0 \leq k_t \leq K$)

[5] The left panel of Figure 1 shows that exponential arrivals are a good approximation of the historical PoW Ethereum chain block arrival process.

[6] The right panel of Figure 1 shows visually that the deterministic block arrival process is an accurate model of the Ethereum chain post-switch.

then the new reserve price is given as:

$$r_{t+1} = r_t \times (1+\delta)^{\frac{k_t}{K/2}-1}.$$

In other words, the reserve price in every period increases (by at most a multiplicative factor of $(1+\delta)$) over the reserve price in the previous block if the block is more than half full, and conversely decreases if the block is less than half full.

We model this formally using a simple 2-reserve price system. Blocks are of size 2, i.e., can contain at most 2 transactions. However, there is now a dynamic reserve price: if the previous block is full, the reserve price in the next block is high, while if the previous block is empty it drops to low. If the block contains exactly one transaction, the reserve price stays the same as in the previous block.

We show that low willingness to pay buyers can be *severely* hurt under this mechanism, i.e., their delay can be substantially worse than under a first price auction. This can happen even when the steady state demand is such that the average block is *less than half full*. The intuition is fairly easy to describe: due to the stochastic nature of demand, reserve prices will sometimes rise high and stay there for a while. When reserve prices are high, low types cannot be served even if the blocks are not full but will continue to arrive and accumulate. This can lead to substantial delays for such types. While our model is a (substantial) simplification of EIP-1559, it should be evident that a similar dynamic will apply there too, as discussed in Leonardos et al. (2022).

We then consider the mechanism design question when explicitly accounting for the dynamic considerations listed above. We show that two other mechanisms have good properties under this setting. The first is a tweak to EIP-1559 as designed: we propose that the updating rule associated with the base-fee be substantially slower. This can target the same long-term average number of transactions-per-block, without causing outsized delay for low types which we showed exist in EIP-1559. The second is a substantially more ambitious proposal: essentially we propose the use of a dynamic VCG mechanism. We argue that these do not suffer the miner incentive compatibility issues that static VCG (i.e., uniform price) mechanisms suffer from in this setting.

1.2. *Related Literature*

Initial forays into transaction fee mechanism design considered the Bitcoin protocol. In particular, early work noted that optimal priority fees in Bitcoin's transaction fee mechanism were difficult for users to calculate in practice. However, switching to easier (i.e., strategyproof for users) mechanism such as a uniform price auction where all included transactions pay the same, lowest bid that would have resulted in inclusion in the block, was not implementable in the terminology of the desiderata we introduced above. In blockchains, the planner (i.e., the protocol) does not directly execute the mechanism and instead relies on the miner/ proposer of the block. This miner/ proposer may have incentives therefore to deviate from protocol desired behavior in order to raise their own revenue—for example, a uniform price auction incentivizes miners to potentially include fake transactions in the block to drive up the clearing price. Such points were made and plausible variations studied in Yao (2018), Basu et al. (2023) and Lavi et al. (2022).

A key paper, whose results we rely on, is the work of Huberman et al. (2021). They propose and solve a model of the Bitcoin TFM while explicitly incorporating dynamic considerations: in particular that impatient users face a tradeoff between cost of inclusion, and delay. Our analysis follows and extends theirs to Proof of Stake settings, and also to EIP 1559.

Several other papers do consider a dynamic analysis, using a dynamical systems or control theory perspective, but by necessity given the complexity of solving such models often abstract away from the dynamic incentives of users. Examples of such papers include Diamandis et al. (2022), Leonardos et al. (2021), Crapis et al. (2023), Nisan (2023).

Another key paper in this space is the seminal work of Roughgarden (2020) (see also the associated survey paper of Roughgarden (2021)). This paper was, to our knowledge, the first in the field to explicitly consider a *mechanism design* approach, i.e., formally define desiderata and incentive constraints and then understand which mechanisms are optimal under those constraints. It used this approach to give foundations to certain design choices made in EIP-1559. The paper inspired several subsequent important advances, e.g., Chung and Shi (2023) which proved additional impossibility theorems for such settings, and Chen et al. (2023) which considered relaxing the underlying solution concept to Bayes-Nash equilibrium. More recently, Bahrani et al. (2023) point to additional difficulties faced by EIP-1559 under the current Ethereum market microstructure, formally proving an impossibility result.

Finally, there are several new directions being pursued within this space. The first is an attempt to use additional cryptographic primitives such as secure multi-party computation (MPC) to increase the set of achievable outcomes: see for example Shi et al. (2022). Another direction is multidimensional pricing. Modern blockchains have multiple resources such as computation, storage, bandwidth, etc, and different kinds of transactions use these differently. Some papers therefore consider pricing these resources independently or study related trade-offs—see, e.g., Diamandis et al. (2023).

2. Model

Transactions arrive at random. The number of new arrivals r in any given interval of time v is distributed according to a Poisson distribution with parameter λ:

$$A(r; v) = \exp^{-\lambda v} \frac{(\lambda v)^r}{r!}.$$

Each transaction is associated with a unique user with type $\theta = (\theta_v, \theta_c)$—here θ_v is their willingness to pay for a transaction, whereas their delay cost per unit time is θ_c. The net payoff of a user of type θ whose transaction is included after delay W for a price (bid) of b is:

$$U(W, b, \theta) = \theta_v - \theta_c W - b.$$

We assume that $\theta_c \sim F[0, \bar{c}]$.

Transactions are served at intervals in batches of at most K transactions (blocks). For most of our analytical results, we consider $K = 2$. While our theorems apply for $K > 2$, analytic solutions as a function of other primitives cannot be achieved for $K \geq 4$, and the model needs to be solved numerically in that case.[7] We assume that the intervals of time v between blocks are independent of each other and have the same probability distribution B, which has density $dB(v)$ equal to a χ^2-distribution with an even number $2p$ degrees of freedom, i.e.,

$$dB(v) = \frac{\mu^p}{\Gamma(p)} v^{p-1} e^{-\mu v} dv.$$

For the special case of $p = 1$, therefore, this reduces to a standard exponential distribution with parameter μ. Conversely, a deterministic arrival process can be approximated by letting both p and μ tend to infinity keeping their ratio constant.

We assume all participants know the system parameters and steady-state distribution, and focus on time-invariant stationary strategies for the users. Formally, the congestion queuing game we study can be described as follows: users arrive and choose whether to participate and submit a bid, or opt-out and receive their outside option of 0. When bidding they do not observe the current queue, i.e., we consider a steady state analysis where a bidder of type θ upon opting in submits a transaction of bid b and receives an expected payoff that equals:

$$\theta_v - b - \theta_c W^\star(b).$$

Here $W^\star(\cdot)$ is the *equilibrium* waiting time as a function of the bid given that all (other) users are bidding according to a steady state bidding function $b^\star(\cdot)$.

Straightforwardly, then, an agent's incentive of how much to bid on the intensive margin does not depend on θ_v, and is solely a function of θ_c: θ_v only determines whether a user chooses to bid at all or takes their outside option. In the majority of what follows, therefore, we ignore θ_v and consider an agent's type as only their impatience, i.e. θ_c. Implicitly, therefore, we are either assuming users have no outside option, or, alternately, assuming that the support of user's types is such that even the lowest value/ most impatient buyer prefers to submit a bid than take their outside option.

[7]At a high level, the reason for no analytic solution stems from the fact that polynomials of degree larger than 4 do not have analytic solutions.

By the revelation principle, we can consider an equivalent direct revelation mechanism implementation of this, where users directly report their type θ and receive an expected delay and make a payment as a function of their bids. Formally, if the user reports type θ_c, they receive a delay of $W^\star(b^\star(\theta_c))$, and pays $b^\star(\theta_c)$. For $b^\star$ to constitute an equilibrium, therefore, it must be the case that, for every $\theta_c \in [0, \bar{c}]$, we have:

$$\theta_c \in \operatorname*{argmax}_{\theta'_c \in [0,\bar{c}]} \theta_v - b^\star(\theta'_c) - \theta_c W^\star(b^\star(\theta'_c)).$$

Straightforward arguments from single dimensional mechanism design (see, e.g., Myerson (1981)) then give us the following.

THEOREM 1. *Fix any pay-your-bid transaction fee mechanism such that there exists a steady-state bidding equilibrium $b^\star$ and corresponding delay as a function of bids $W^\star$. Then the following must be true:*

(1) $W^\star(b^\star(\theta_c))$ is non-increasing in θ_c.

(2) The equilibrium bids and delay function must jointly satisfy:

$$b^\star(\theta_c) = -\theta_c W^\star(b^\star(\theta_c)) + \int_0^{\theta_c} W^\star(b^\star(c))dc + b^\star(0). \tag{1}$$

In particular equation (1) is the celebrated revenue equivalence theorem in this setting. This theorem already hints at the simple economics of bidding for priority in transaction fee mechanisms: users bid to avoid delays. The higher is the equilibrium delay suffered by types who bid lower, holding everything else fixed, the larger is a user's incentive to bid to try and get priority. The proof is fairly straightforward and we reproduce it briefly below.

PROOF. Let us define $V(\theta'_c, \theta_c) = \theta_v - b^\star(\theta'_c) - \theta_c W^\star(b^\star(\theta'_c))$, i.e., it is the expected utility of a type θ_c that reports itself to be type θ'_c. Incentive compatibility implies that for any two types θ_c and θ'_c, we have that:

$$\begin{aligned} V(\theta_c, \theta_c) &\geq \theta_v - b^\star(\theta'_c) - \theta_c W^\star(b^\star(\theta'_c)), \\ &= \theta_v - b^\star(\theta'_c) - \theta'_c W^\star(b^\star(\theta'_c)) - (\theta_c - \theta'_c) W^\star(b^\star(\theta'_c)), \\ &= V(\theta'_c, \theta'_c) - (\theta_c - \theta'_c) W^\star(b^\star(\theta'_c)), \\ \implies V(\theta_c, \theta_c) - V(\theta'_c, \theta'_c) &\geq -(\theta_c - \theta'_c) W^\star(b^\star(\theta'_c)) \end{aligned}$$

Combining with the reverse inequality from incentive compatibility for θ'_c, we have that

$$-(\theta_c - \theta'_c) W^\star(b^\star(\theta'_c)) \leq V(\theta_c, \theta_c) - V(\theta'_c, \theta'_c) \leq -(\theta_c - \theta'_c) W^\star(b^\star(\theta_c))$$

Since this is true for any θ_c, θ'_c it must be the case that for $\theta_c \geq \theta'_c$ we have that $W^\star(b^\star(\theta'_c)) \geq W^\star(b^\star(\theta_c))$ otherwise the inequality above cannot be satisfied.

Taking limits as $\theta_c' \to \theta_c$ we have that $U(\theta_c) \equiv V(\theta_c, \theta_c)$ is almost everywhere differentiable in θ_c, and has derivative $W^\star(b^\star(\theta_c))$. Equation (1) follows. ■

3. Pure Priority Fee Mechanisms

We first study the classic pay-your-bid, no reserve price mechanisms such as those used by Bitcoin and early Ethereum. In this case, the bid establishes a straightforward form of priority queueing Hassin and Haviv (2003). When a block is formed, it does not contain the earliest arriving transactions, but instead the highest priority transactions (those that pay the highest tips). Intuitively, therefore, users' delays given a bid depend only on the arrival rate of transactions that bid higher. Since bids must be increasing in θ_c, we have straightforwardly that delays in equilibrium depend only on the mass of transactions with a higher priority. Therefore the right-hand side of (1) is purely a function of the primitives of the problem (transaction and block arrival dynamics, distribution of θ_c). This allows us to determine the payments in equilibrium using Equation (1).

More formally, consider a standard first come first served queue. Given arrival rate $\hat{\lambda}$ of transactions, block arrival rate μ and blocks of size K transactions, let us denote the *effective load* as:

$$\hat{\rho} \equiv \frac{\hat{\lambda}}{K\mu}.$$

Denote by $\bar{Q}(\hat{\rho})$ as a function of the steady state expected queue length of transactions given effective load $\hat{\rho}$. This needs to be calculated for the underlying block and transaction arrival dynamics. We can now straightforwardly show the following theorem:

Theorem 2. *Consider a transaction and let $\hat{\lambda}$ be the arrival rate of higher-priority transactions (i.e., transactions that offer greater fees. The expected time until the transaction is processed is a function of the block size K, the arrival rate μ and the load parameter corresponding to higher-priority transactions, $\hat{\rho} \equiv \hat{\lambda}/(\mu K)$ and is equal to:*

$$W_K(\hat{\rho}) = \frac{1}{K\mu}\frac{\partial \bar{Q}(\hat{\rho})}{\partial \hat{\rho}} \tag{2}$$

where $\bar{Q}$ is the steady state expected queue length of transactions given effective load $\hat{\rho}$. Here $W_K(\cdot)$ refers to the expected waiting time for a transaction which has $\hat{\rho}$ effective load of higher-priority transactions.

Proof. By Little's law, we have that:

$$\lambda \bar{W}_k(\hat{\rho}) = \bar{Q}$$

where $\bar{W}_K$ is the average delay, measured in number of blocks, for transactions when the effective load is $\hat{\rho}$.

Consider now the priority queueing scenario were transactions of a higher priority are served ahead of transactions of lower priority. Letting $W_K(\hat{\rho})$ denote the expected waiting time of a transaction when $\hat{\rho}$ is the effective load of transactions with high priority, we have, by definition:

$$\bar{W}_K(\hat{\rho}) = \frac{1}{\hat{\rho}} \int_0^{\hat{\rho}} W_K(r) dr.$$

Substituing in the definition of $\hat{\rho}$, this implies that:

$$K\mu \int_0^{\hat{\rho}} W_K(r) dr = \bar{Q}(\hat{\rho}).$$

Differentiating with respect to $\hat{\rho}$ we arrive at.

$$K\mu W_k(r) = \frac{\partial \bar{Q}(\hat{\rho})}{\partial \hat{\rho}} \tag{3}$$

After dividing both sides by $k\mu$ the theorem follows. ■

Therefore, straightforwardly, if we can determine the expected queue length of transactions in a system as a function of expected load, we can determine the wait times for a given priority via Theorem 2, and then calculate the equilibrium bid via Theorem 1. The latter can be done leveraging standard techniques of these queues. In particular, assuming that the type $\theta_c = 0$ submits an equilibrium bid of 0, we have the following straightforward corollary of Theorems 1 and 2:

COROLLARY 1. *Suppose the transaction fee mechanism is a pay-your-bid mechanism. Suppose further in a priority queue, the wait time for a transaction measured in blocks is $W_K(\hat{\rho})$ given effective load $\hat{\rho}$ of higher priority transactions. Assuming all users participate (i.e. do not take the outside option, equivalently that θ_v is high enough), we have that:*

$$b^\star(\theta_c) = -\theta_c W_K\left(\frac{\lambda(1 - F(\theta_c))}{K\mu}\right) + \int_0^{\theta_c} W_K\left(\frac{\lambda(1 - F(c))}{K\mu}\right) dc$$

This corollary follows straightforwardly from Theorem 1, noting that the lowest cost type $\theta_c = 0$ must submit a bid $b^\star(0) = 0$ in equilibrium and substituting in $W^\star(b) = W_K(\hat{\rho})$ where $\hat{\rho}$ is the effective load of transactions with higher costs (and therefore a higher bid).

3.1. *Exponential Block arrivals*

Let us begin with the simplest model: blocks can contain at most K transactions and arrive at an exponential rate μ. We assume that $\mu > K\lambda$, otherwise the queue will grow unbounded in steady state.

Define the parameter ρ, which represents the average load of the system, as $\rho := \lambda/(K\mu)$. Note that $\rho \in [0, 1)$. The following standard result from the analysis of bulk service systems will be useful, see e.g., Section 4.6 of Kleinrock (1975). Our results in what follows recover the results of Huberman et al. (2021).

LEMMA 1 (Kleinrock 1975)**.** *Consider a queue system of a single queue, with arrivals according to a Poisson process of rate $\lambda \geq 0$ and bulk service in batches of up to $K \geq 1$ with service times exponentially distributed with parameter $\mu > 0$. Suppose that the effective load $\rho \equiv \lambda/(\mu K) \in [0, 1)$. Then the queueing system is stable, and the steady state queue length Q has the geometric distribution:*

$$P(Q = \ell) = (1 - z_0)z_0^\ell \ \ \ell = 0, 1, \dots$$

where $z_0 = z_0(\rho, K)$ is the unique solution of

$$z^{K+1} - (K\rho + 1)z + K\rho = 0 \tag{4}$$

in the interval $[0, 1)$.

Note that the expected number of entries in a geometrically distributed queue with parameter $z_0(\rho, K)$ is $\frac{z_0(\rho,K)}{1-z_0(\rho,K)}$. Therefore we have that

$$\bar{Q}(\hat{\rho}) = \frac{z_0(\hat{\rho}, K)}{1 - z_0(\hat{\rho}, K)}$$

THEOREM 3 (Huberman et al. 2021)**.** *Suppose inter-block times are exponentially distributed with rate μ. Consider a transaction and let $\hat{\lambda}$ be the arrival rate of higher-priority transactions (i.e., transactions that offer greater fees. The expected time until the transaction is processed is a function of the block size K, the arrival rate μ and the load parameter corresponding to higher-priority transactions, $\hat{\rho} \equiv \hat{\lambda}/(\mu K)$ and is equal to:*

$$W_K^E(\hat{\rho}) = \frac{1}{\mu} \frac{1}{(1 - z_0)(1 + K\hat{\rho} - (K + 1)z_0^K)} \tag{5}$$

where z_0 is the polynomial root of (4) defined in Lemma 1, and $W_K^E(\cdot)$ is the expected wait time (the E in the superscript here refers to exponential).

PROOF. From Theorem 2, we have that

$$\begin{aligned} W_K^E(\hat{\rho}) &= \frac{1}{K\mu} \frac{\partial \bar{Q}(\hat{\rho})}{\partial \hat{\rho}} \\ &= \frac{1}{K\mu} \frac{\partial}{\partial \rho} \frac{z_0(\hat{\rho}, K)}{1 - z_0(\hat{\rho}, K)} \\ &= \frac{1}{K\mu} \frac{1}{(1 - z_0)^2} \frac{\partial z_0(\hat{\rho}, K)}{\partial \hat{\rho}} \end{aligned}$$

Calculating $\frac{\partial z_0(\hat{\rho},K)}{\partial \hat{\rho}}$ from equation (4), the result follows. ■

Some straightforward applications now follow from this characterization in the case of $K = 1, 2$.

3.1.1. *One Transaction Per Block.* First, suppose that $K = 1$, i.e., blocks can contain only 1 transaction. In this case, (4) reduces to

$$z^2 - (\rho + 1)z + \rho = 0$$

In this case, we straightforwardly have that roots are ρ and 1, and therefore, the unique root in $[0, 1)$ must be $z_0(\rho, K) = \rho$. Plugging into (5), we have that

$$W_1^E(\hat{\rho}) = \frac{1}{\mu(1-\hat{\rho})^2},$$

and therefore applying Corollary 1,

$$b^\star(\theta_c) = -\theta_c \frac{1}{\mu(1 - \frac{\lambda(1-F(\theta_c))}{\mu})^2} + \int_0^{\theta_c} \frac{1}{\mu(1 - \frac{\lambda(1-F(c))}{K\mu})^2} dc \tag{6}$$

3.1.2. *Two Transactions Per Block.* Now, suppose that $K = 2$, i.e., blocks can contain up to 2 transactions. In this case, (4) reduces to

$$z^3 - (2\hat{\rho} + 1)z + 2\hat{\rho} = 0$$

In this case, we have that the unique root in $[0, 1)$ must be

$$z_0(\hat{\rho}) = \frac{1}{2}(\sqrt{1 + 8\hat{\rho}} - 1).$$

Plugging into (5), we have that

$$W_2^E(\hat{\rho}) = \frac{1}{(1 - z_0(\hat{\rho}))(1 + 2\hat{\rho} - 3z_0(\hat{\rho})^2)}$$

where z_0 is as given above, and again we have that:

$$b^\star(\theta_c) = -\theta_c W_2^E\left(\frac{\lambda(1 - F(\theta_c))}{K\mu}\right) + \int_0^{\theta_c} W_2^E\left(\frac{\lambda(1 - F(c))}{K\mu}\right) dc \tag{7}$$

3.2. *Deterministic Block Arrivals*

As we described earlier, deterministic block arrivals are a feature of Proof-of-Stake chains, where a random proposer of a given block is selected by the protocol to propose a block at deterministic time intervals, rather than via the inherently stochastic process in Proof-of-Work protocols where the proposer is the first to find a solution to a particular "cryptopuzzle." Figure 1 shows that after the switch to Proof-of-Stake, the Ethereum protocol block arrival process is well-approximated by a deterministic process. Let $1/\mu$ now denote the deterministic inter-arrival time between blocks. The following result from Bailey (1954) and is the analogue to Lemma 1 in the deterministic block arrival setting:

LEMMA 2 (Bailey 1954). *Consider a queue system of a single queue, with arrivals according to a Poisson process of rate $\lambda \geq 0$ and bulk service in batches of up to $K \geq 2$ with service times determinstical with interarrival times $1/\mu$. Suppose that the effective load $\rho \equiv \lambda/(\mu K) \in [0, 1)$. Then the queueing system is stable, and the steady state queue length Q has a generating function Π given as:*

$$\Pi(z) = \frac{\sum_{j=0}^{K-1} \pi_j (z^s - z^j)}{z^K \exp\{K\rho(1-z)\} - 1}.$$

Here π_j for $j = 0, \ldots K-1$ are the solution to the system of equations:

$$\sum_{j=0}^{K-1} (K-j)\pi_j = K(1-\rho),$$

$$\forall i \in [K-1] : \sum_{j=0}^{K-1} \pi_j (z_i^s - z_i^j) = 0$$

where z_i, for $i = 1, \ldots, K-1$ are the $K-1$ roots of $z^K \exp\{K\rho(1-z)\} - 1 = 0$ with $|z_i| < 1$.

Given the probability generating function $\Pi(z)$ is known, note that the expected queue length can simply be calculated as $\Pi'(1)$. Calculating and simplifying, we have that the expected queue length $\bar{Q}$ is:

$$\bar{Q} = \frac{1 - K(1-\rho)^2}{2(1-\rho)} + \sum_{i=1}^{K-1} \frac{1}{1-z_i} \tag{8}$$

As in the previous section, we can now apply our theorem to deliver explicit results for $K = 1, 2$.

3.2.1. *One Transaction Per Block.* Note that in this case, we have that (8) reduces to

$$\begin{aligned}
\bar{Q} &= \frac{1 - (1-\rho)^2}{2(1-\rho)} \\
&= \frac{\rho}{2} + \frac{\rho}{2(1-\rho)} \\
&= \frac{\rho}{2} + \frac{1}{2(1-\rho)} - \frac{1}{2} \\
\implies \frac{\partial \bar{Q}}{\partial \rho} &= \frac{1}{2(1-\rho)^2}.
\end{aligned}$$

and therefore we have that:[8]

$$W_1^D(\hat{\rho}) = \frac{1}{2\mu(1-\hat{\rho})^2},$$

[8]Here the D superscript refers to deterministic block times

and therefore applying Corollary 1,

$$b^{\star}(\theta_c) = -\theta_c \frac{1}{2\mu(1-\frac{\lambda(1-F(\theta_c)}{\mu})^2} + \int_0^{\theta_c} \frac{1}{2\mu(1-\frac{\lambda(1-F(c))}{K\mu})^2} dc \tag{9}$$

Comparing with (6), we have that equilibrium bids *halves* relative to exponential block arrivals, even though the demand process and the expected block production rate of the chain is the same in both cases! As we discussed earlier, the reason is related to the variance in block arrival times: with exponential arrivals, blocks sometimes arrive very soon after the previous block, and therefore go empty (since no new transactions have arrived), while other times blocks arrive late, during which period substantial demand has built up.

This straightforward observation is nevertheless useful, given our earlier stated desideratum of User Experience. Admittedly, first price auctions are not dominant strategy incentive compatible. Nevertheless, having a smaller range of equilibrium bids, and less variance, means that the user's problem of figuring out how much to bid is simpler under Proof-of-Stake than Proof-of-Work, holding all else constant!

3.2.2. *Two transactions Per Block.* Substituting $K = 2$ into (8), we have

$$\bar{Q} = \frac{1-2(1-\rho)^2}{2(1-\rho)} + \frac{1}{1-z_1}$$

where z_1 is the solution of

$$z^2 \exp\{2\rho(1-z)\} - 1 = 0,$$

i.e.,

$$z_1(\rho) = --\frac{W_L\left(\sqrt{e^{-2\rho}\rho^2}\right)}{\rho}, \ \frac{dz_1}{d\hat{\rho}} = \frac{2}{\sqrt{1+8\rho}}.$$

where W_L is the standard product log/ Lambert W function.[9] Further recall that we have that:

$$\mu W_2^D(\hat{\rho}) = \frac{1}{2(1-\rho)^2} + \frac{1}{(1-z_1)^2}\frac{dz_1}{d\hat{\rho}},$$

where $z_1(\cdot)$, $\frac{dz_1}{d\hat{\rho}}$ are as given above.

$$b^{\star}(\theta_c) = -\theta_c W_2^D\left(\frac{\lambda(1-F(\theta_c))}{K\mu}\right) + \int_0^{\theta_c} W_2^D\left(\frac{\lambda(1-F(c))}{K\mu}\right) dc \tag{10}$$

3.3. *Numerical Comparisons*

We are now in a position to use our results above to do some simple numerical comparisons. Formally, suppose $\lambda = K$ and $\mu = 2$. Therefore the effective load overall under

[9] We apologize for the notation overload.

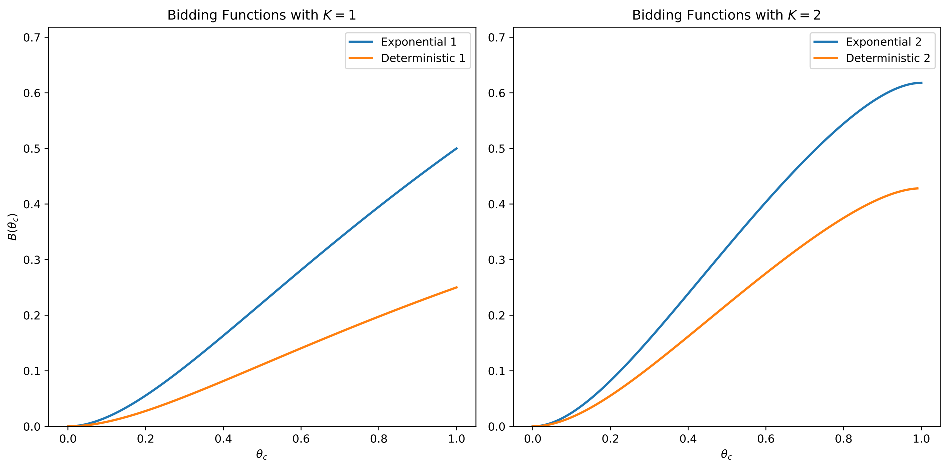


FIGURE 2. Caption

both $K = 1$ and $K = 2$ is $\frac{1}{2}$. Suppose further that θ_c is distributed uniformly over $[0, 1]$. The bid functions for the 4 cases are plotted below in Figure 2.

4. EIP-1559

We can now formally consider EIP-1559 under the same framework. In particular, Theorem 1 allows us to compute the equilibrium bids as a function of equilibrium delays. In what follows below we first provide a formal description of EIP-1559 for the reader, and provide straightforward evidence that it is *not* dominant strategy incentive compatible in a dynamic setting.[10]

4.1. *A Brief Summary of EIP-1559 and Some Empirical Evidence*

Recounting our discussion from earlier, EIP-1559 essentially introduces a dynamic reserve price (within the Ethereum community, this is called a base fee) to the first price auction.[11] The dynamic reserve price in a block is computed as a function of the previous block's reserve price, and the "fullness" of the previous block: as we described earlier, in our context where all transactions are equally sized, if the previous block had a reserve r_t, contained k_t transactions ($0 \leq k_t \leq K$) then the new reserve price is given as:

$$r_{t+1} = r_t \times \exp\left\{\left(\frac{k_t}{K/2} - 1\right)\ln(1+\delta)\right\}.$$

[10]We hasten to add that this is not an "error" in the existing literature. Roughgarden (2020) for example, explicitly considers a static setting, and therefore our dynamic considerations do not apply.

[11]In order to ensure what we term as implementability, i.e., so that the proposer does not have an incentive to undercut the proposed reserve price, the reserve price for any included transaction is burned, i.e., it accrues to the protocol rather than to the proposer. The proposer keeps the portion of included bids that exceed the reserve, incentivizing them to include the highest bids above the reserve price up to the maximum block size. This was formally argued in Roughgarden (2020).

The explicit parameter chosen is $\delta = 12.5\%$, i.e., a full block increases the reserve price over the previous block by 12.5% while an empty block reduces the reserve price by the same factor. Implicitly therefore, assuming some elasticity of demand, the dynamic reserve price mechanism targets blocks to be half-full on average. As we described earlier, this attempts to tradeoff the fact that in the short-run, the Ethereum validator networks are able to process larger blocks, but the long-run average must be roughly half-full blocks at the current max block size (of 30M gas).[12]

Note the reserve price of a block is known in advance of the block (since it only depends on the previous block's reserve price and fullness). A straightforward argument then delivers that in a static context, this mechanism is "almost" dominant strategy. In particular, consider a user interested only in inclusion in a *given* block. The reserve price at that block is known. If the maximum possible block size is large enough relative to the stochastic arrival rate of transactions, then a block will rarely be completely full, and therefore the user can a) guarantee inclusion at a bid slightly higher than the reserve price and b) has no incentive to bid any higher than that.

It is instructive to look to the data to see how well this performs. We collected the base fee (i.e., reserve price) and the gas used for 200,000 blocks from block numbers 18,700,000 through 18,899,999.[13] These blocks span roughly a month's worth of Ethereum blocks from Dec-02-2023 through Dec-30-2023 (at 12 seconds per block, Ethereum produces 216000 blocks per 30-day month). Firstly, on our data, the average gas used per block is 15.107 million, i.e., fairly close to the desired 15 million gas target. Further, only 4219 blocks in the data are more than 99% full. Since the average block contains 150–200 transactions, this means that for the remaining $\sim$ 196,000 or 98% of blocks in our sample, a bid of slightly higher than the known base fee would guarantee inclusion. In that sense, indeed, bidding the current base fee is an "almost" dominant strategy for an impatient user since it guarantees inclusion with high probability.

When considering the dynamic setting, however, the story is different. Gas fees vary quite substantially from block to block. In our data, the average absolute percentage change between successive blocks was 3.25%. An example of these fluctuations can be seen in Figure 4.

A user who is willing to wait, therefore may achieve substantial savings by waiting for a later block with a lower base fee. Since future base fees are unknown from the perspective of a current user, there is therefore a non-trivial tradeoff: achieve inclusion (with high

[12]Specifically, transactions increase the size of the state that the underlying virtual state machine must track. The 15M target average therefore is intended to limit the average rate of growth of the state, in order to be manageable for the underlying hardware.

[13]Our data and code are available upon request, Ethereum has a standard API for gathering this data from a node, but a reader interested in gathering the data for themselves will need an API key from a provider such as Infura or Alchemy.

TABLE 1. How often can a user save the discount percentage, in hindsight, within the given window, for N=10,000 random start times in our dataset.

Window	1% discount	5% discount	10% discount	20% discount
5 blocks	8009	4564	1183	28
50 blocks	9555	8290	5570	1267
300 blocks	9887	9527	8294	4150

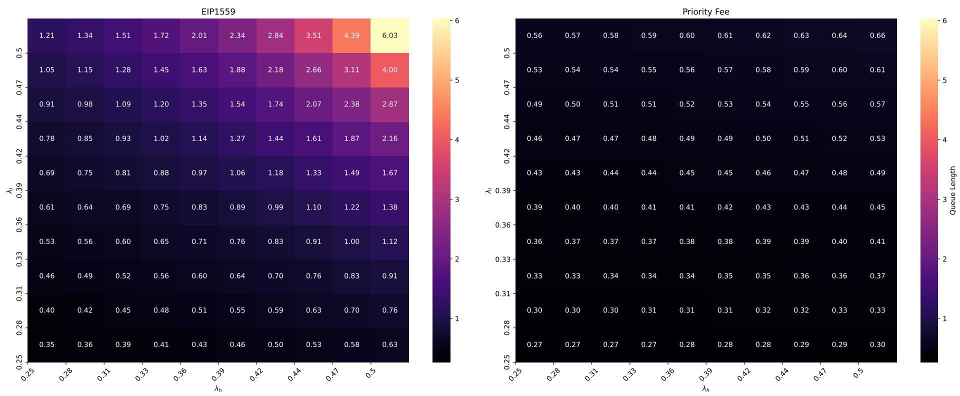
probability) at the current base fee or wait and possibly save money in a future block. To demonstrate this we considered the following simple exercise: randomly draw N starting blocks (corresponding to the "birth" of the user's desire to transact). Consider a window in the corresponding K blocks, where we considered $K = 5, 50$ and 300 corresponding to the idea of waiting for 1 minute, 10 minutes and 1 hour respectively. How often can the user transact at a certain discount given this window? Table 1 summarizes what we found in our data. For example, in hindsight, given a random starting point and a 5 block window, there exists with high ($\sim$ 80%) probability a block in the window with an at least 1% lower base fee, and $\sim$ 46% probability a block with 5% lower base fees. Given a longer willingness to wait, of 50 blocks, even more substantial savings are possible, with a $\sim$ 56% probability of a block in the window with a 10% lower base fee and $\sim$ 12.7% probability of a 20% base lower! The latter rises to $\sim$ 41.5% given a one hour (300 block) window. This of course is only indicative since these calculations are in hindsight: nevertheless it suggests a substantial role for strategizing if the user is willing to tradeoff delay with cost. Of course the user's problem is harder still: they face an optimal stopping problem which requires them to estimate the distribution of base fees in order to know when to submit a bid!

4.2. *A Simplified Model of EIP-1559*

We can now explicitly consider EIP-1559. We restrict attention to $K = 2$ transactions per block. For the purposes of analytic expression, we consider a simplification: there are two possible reserve prices, a low reserve r_L normalized to 0 and a high reserve, r_H. The reserve price in a block is determined as a function of the previous block: if the previous block is full, the reserve price moves up to r_H, if the previous block is empty, the reserve price moves down to 0, and the reserve price stays unchanged if the block is half full. We continue to analyze this within the context of the steady-state model — i.e., a user upon arriving does not observe the current reserve price or state of the queue, they simply submit a bid that maximizes their expected net payoff in the steady state.

Suppose λ_H is the rate of arrival of transactions that are willing to pay a fee that exceeds the high reserve price r_H, and λ_L is the rate of arrival of transactions that are not. Note that by assumption we must have $\lambda = \lambda_L + \lambda_H$.

FIGURE 3. Expected queue length for L type transactions under EIP-1559



It is advantageous to analyze this system via the embedded markov-chain prior to a block arrival. Formally, consider a 3 dimensional state: (r, q_L, q_H) where q_L and q_H are the number of L and H type transactions in the queue respectively. Letting a_L and a_H be the random number of transaction arrivals of each type in that period, we can describe the state transitions as:

$$\begin{aligned}
(r_H, q_L, q_H) &\to (r_L, q_L + a_L, a_H) \text{ if } q_H = 0 \\
&\to (r_L, q_L + a_L, (q_H - 2, 0)^+ + a_H) \text{ otherwise,} \\
(r_L, q_L, q_H) &\to (r_L, a_L, a_H) \text{ if } q_L + q_H \leq 1 \\
(r_L, q_L, q_H) &\to (r_H, q_L + a_L, q_H - 2 + a_H) \text{ if } q_H \geq 2 \\
(r_L, q_L, 1) &\to (r_H, q_L - 1 + a_L, a_H) \\
(r_L, q_L, 0) &\to (r_H, q_H - 2 + a_L, a_H).
\end{aligned}$$

Using standard techniques, we can write down the recurrence equations that define the steady-state distribution over these states in this setting. In the appendix, we analyze precisely this object. In the body of the text here however let us satisfy ourselves with simulations. Figure 3 shows the expected wait time of L types as a function of $\lambda_L, \lambda_H \in [0.25, 0.5]$. In the simulation we have assumed that $\mu = 1$ and $K = 2$. Note that this means that the maximum effective load in the simulation, which occurs when we have the highest possible $\lambda_L = \lambda_H = 0.5$, is 0.5, i.e., substantially smaller than 1. Nevertheless, by visually inspecting Figure 3, we see that low types' expected queue length is ~ 6, and by an application of Little's law, this corresponds to a waiting period of 12 blocks! For comparison, the queue length for these transactions under a no-reserve first price auction is $\sim .66$.

These queue lengths imply two things:

(1) Firstly, low-type transactions are severely disadvantaged under this simplified model of EIP-1559 relative to the pure priority mechanism.
(2) Secondly, applying the pricing formula from Theorem 1, r_H can be substantially higher than what those types would have bid under a first price auction.

The reason for the former is easy to explain: once prices move high, an empty block is needed to bring prices back down. Even though there may be available blockspace in the interim, it is priced out of reach of the low type, leading to the low type needing to wait until an empty block. The resulting delay is indeed shown to be substantial, *even when* the effective load is such that the chain is uncongested. Formally, from the perspective of transactions that submit bids below the high reserve price r_H, it is as if the block production process takes a *vacation* whenever the chain moves into the high reserve price state. During this period (of length a stochastic number of blocks), low-bid transactions continue to arrive, but none are processed.

The latter then follows from a standard economic monopoly distortion intuition we discussed in Theorem 1 under EIP-1559 the low type is served inefficiently (with large delays). This makes the high type willing to pay more to avoid being delayed.

Part of the underlying rationale for the dynamic price in EIP 1559 was to limit blocks to being half full on average in order to limit state growth, while still allowing for full blocks during periods of high demand. However, in our simulations, all transactions are served in steady state regardless of the choice of mechanism so there is no difference in state-growth: our parameters are selected so that the average block is less than half-full in steady state. However, EIP-1559 raises substantially more revenue (which was not a design goal of the mechanism), while, or more precisely in part due to the increased delays for low willingness-to-pay users.

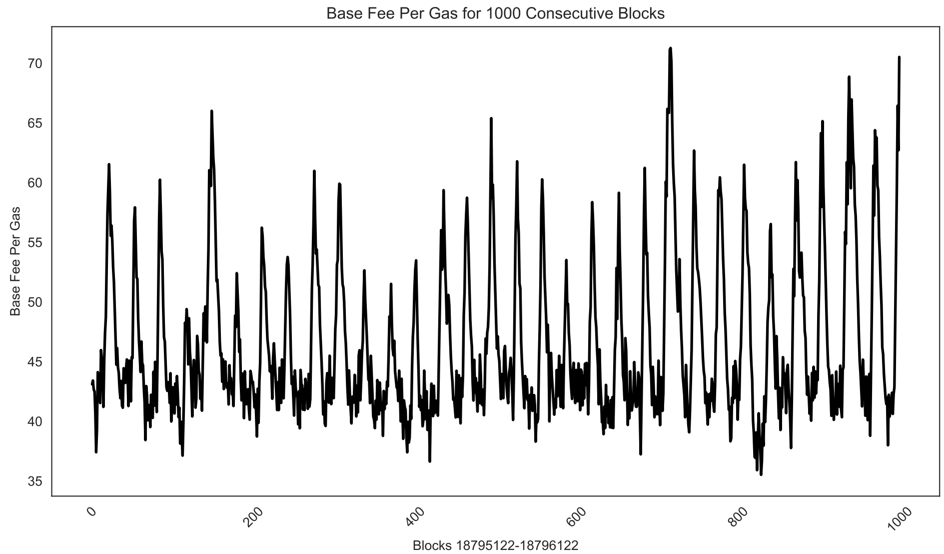
4.3. *Slower Updating Reserve Prices*

So far we have analyzed our model in the context of steady-state under a fixed arrival rate of transactions. In doing so, we have deliberately hamstrung EIP-1559's dynamic reserve price mechanism to make the starkest possible reserve. In particular, the dynamic reserve price was meant to find the correct reserve price to achieve a targeted long-run average use of the chain (in order to limit state growth etc). The dynamic process needs to do able to do this even if the underlying demand characteristics, i.e., arrival rates are changing.[14]

Since arrival rates here in our model are fixed, in steady state the reserve price should be as well. However, the point remains well-made: it appears that the current implementation of EIP-1559 has dynamic reserve prices that are far too sensitive to the previous

[14]Blockchain usage famously follows cycles of boom-and-bust periods that last several months each.

FIGURE 4. Fluctuations in Ethereum Base Fee over 1000 blocks (3.5 hours)



period's block size, and the inherent stochasticity of transaction arrivals means that reserve prices are also fairly volatile (recall that we showed above that the average absolute percent change in the base fee is $\sim 3.2\%$).

This volatile base-fee has 2 downsides: first, when users are willing to trade-off delay and inclusion, the randomness in the base-fee turns the user's bidding problem into a fairly complex stopping problem. Secondly, the base-fee can move up for short periods explicitly due to a large number of arrivals in some interval while the underlying arrival process is stable. These periods produce delays for low willingness-to-pay users even without changing the steady-state usage of the chain!

Motivated by this, we propose a simple remedy: slow down the base-fee updating process so that updates to the base-fee reflect changes in the underlying arrival parameters rather than short-term stochastic bursts. Formally, consider an additional parameter N and let $\bar{k}_{t,N}$ be the average usage/ number of transactions in the previous N blocks from period t, i.e., $t-N+1, t-N+2, \ldots t$. Define the update rule as:

$$r_{t+1} = r_t \times \exp\left\{ \left(\frac{\bar{k}_{t,N}}{K/2} - 1 \right) \ln(1+\delta) \right\}$$

There is now an additional parameter that needs to be selected: N. Choosing this N depends on additional considerations for example how many consecutive blocks can the chain stably process blocks that are fuller than the long-run target size. However, for a well-chosen N,

(1) Over windows of size $\ll N$, base fees will be roughly stable. Users can therefore bid according to the first-price auction logic described in Section 3, and even guarantee reasonable inclusion probabilities at or near the base fee.
(2) Over windows of size $\gg N$, average usage of the chain will converge to the desired target of $K/2$.

5. Conclusions

In this paper we built on the (small) literature that explicitly considers the dynamic aspects of transaction fee mechanisms. We have two major takeaways:

(1) Proof-of-Stake chains have lower delays and therefore lower bids relative to Proof-of-Work chains. In particular, to the extent that transaction fee mechanisms were designed to improve user experience (UX) and make fees more predictable, Proof-of-Stake already goes a long way towards achieving this.
(2) The dynamic reserve price, at least in the context of our model does not opitmally serve the stated goal (of limiting throughput/ state growth): in particular it raises more revenue than is socially efficient, by inefficiently delaying low types and causing high types to pay higher fees.

The latter finding lends itself to an easy tweak to the EIP-1559 mechanism: if the goal is to limit long-term state growth, this may be better achieved by having a dynamic reserve price that adjusts slowly. Under the EIP-1559 process, random variation in arrivals (due to stochasticity of the arrival process) can lead to excessive increases in the reserve price, and corresponding delays for the low types. In other words the mechanism may be fooled by randomness.

We conclude with a final possibility— implementing Dynamic VCG mechanisms. As we discussed earlier, a desideratum that has attracted much attention is DSIC, i.e, dominant strategy incentive compatible mechanisms. In the single-block context, this reduces to a uniform price auction (or if K transactions are included, a $K+1^{\text{th}}$-bid auction). However, multiple authors have pointed out that this is not incentive compatible for the miner/ validator: a validator will have an incentive to include fake transactions to increase its own revenue. This is akin to the concept of shill bidders in traditional auctions.

However, in the dynamic setting, we may be able to design DSIC mechanisms, or at least, mechanisms that are DSIC for most users, that also satisfy miner incentive compatibility.[15]

[15] An alternate plausible model is one in which each user has a private value v, a time of arrival $\underline{t}$ and a deadline $\bar{t}$. All 3 components are private information to the user: v is interpreted as before, $\underline{t}$ is interpreted as the time at which the user learns they need to transact, and the deadline is the time after which inclusion has no value to the user. The latter captures a binary form of discounting, i.e., transacting either has a fixed value or none at all. This model has been considered in revenue management settings, see, e.g., Pai and Vohra (2013).

In this setting, consider the dynamic VCG mechanism:

(1) Users report (θ_v, θ_c) at or after their arrival.
(2) When constructing a block, the miner selects the expected efficient (highest value) K transactions among those available.
(3) For any transaction that is included, the stated willingness to pay $\theta_v - D\theta_c$ is escrowed by the blockchain, where D is the number of blocks that have passed since the user reported.
(4) At the deadline that equals θ_v/θ_c at which the user's willingness to pay drops to 0 the chain calculates the lowest price that would have allowed for inclusion, all else fixed. The remainder is refunded from the escrow.

The mechanism described above is DSIC for users—essentially, this is a dynamic VCG mechanism (see e.g., Bergemann and Said 2010). Further the miner incentive compatibility violations of uniform price auctions will apply here only for transactions that expire in the block that they are included. Assuming most transactions are sufficiently patient that they are filled well before expiry, we have a mechanism that is DSIC for users and has no shill-bidder incentives for miners. Of course, the incentives of miners/ proposers/ builders to efficiently pack blocks is not clear here— they may be incentivized to prefer transactions that are at or near expiry, since these transactions will have more certainty in terms of what revenues they yield upon inclusion. We leave this topic for further investigation.

References

Maryam Bahrani, Pranav Garimidi, and Tim Roughgarden. 2023. Transaction Fee Mechanism Design with Active Block Producers. *arXiv preprint arXiv:2307.01686* (2023).

Norman TJ Bailey. 1954. On queueing processes with bulk service. *Journal of the Royal Statistical Society: Series B (Methodological)* 16, 1 (1954), 80–87.

Yogev Bar-On and Yishay Mansour. 2023. Optimal Publishing Strategies on a Base Layer. *arXiv preprint arXiv:2312.06448* (2023).

Soumya Basu, David Easley, Maureen O'Hara, and Emin Gün Sirer. 2023. StableFees: A predictable fee market for cryptocurrencies. *Management Science* (2023).

Dirk Bergemann and Maher Said. 2010. Dynamic Auctions. *Wiley Encyclopedia of Operations Research and Management Science* (2010).

Xi Chen, David Simchi-Levi, Zishuo Zhao, and Yuan Zhou. 2023. Bayesian Mechanism Design for Blockchain Transaction Fee Allocation. *Available at SSRN 4413816* (2023).

Hao Chung and Elaine Shi. 2023. Foundations of transaction fee mechanism design. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 3856–3899.

Davide Crapis, Ciamac C Moallemi, and Shouqiao Wang. 2023. Optimal dynamic fees for blockchain resources. *arXiv preprint arXiv:2309.12735* (2023).

Theo Diamandis, Alex Evans, Tarun Chitra, and Guillermo Angeris. 2022. Dynamic pricing for non-fungible resources. *arXiv preprint arXiv:2208.07919* (2022).

Theo Diamandis, Alex Evans, Tarun Chitra, and Guillermo Angeris. 2023. Designing Multidimensional Blockchain Fee Markets. In *5th Conference on Advances in Financial Technologies (AFT 2023)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

Refael Hassin and Moshe Haviv. 2003. *To queue or not to queue: Equilibrium behavior in queueing systems*. Vol. 59. Springer Science & Business Media.

Gur Huberman, 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, 6 (2021), 3011–3040.

L. Kleinrock. 1975. *Queueing Theory. Volume 1: Theory*. Wiley-Interscience.

Ron Lavi, Or Sattath, and Aviv Zohar. 2022. Redesigning Bitcoin's fee market. *ACM Transactions on Economics and Computation* 10, 1 (2022), 1–31.

Stefanos Leonardos, Barnabé Monnot, Daniël Reijsbergen, Efstratios Skoulakis, and Georgios Piliouras. 2021. Dynamical analysis of the eip-1559 ethereum fee market. In *Proceedings of the 3rd ACM Conference on Advances in Financial Technologies*. 114–126.

Stefanos Leonardos, Daniël Reijsbergen, Barnabé Monnot, and Georgios Piliouras. 2022. Optimality Despite Chaos in Fee Markets. arXiv:2212.07175 [cs.GT]

Akaki Mamageishvili and Edward W Felten. 2022. Efficient L2 Batch Posting Strategy on L1. *arXiv preprint arXiv:2212.10337* (2022).

Roger B Myerson. 1981. Optimal auction design. *Mathematics of operations research* 6, 1 (1981), 58–73.

Noam Nisan. 2023. Serial Monopoly on Blockchains. arXiv:2311.12731 [cs.GT]

Mallesh M Pai and Rakesh Vohra. 2013. Optimal dynamic auctions and simple index rules. *Mathematics of Operations Research* 38, 4 (2013), 682–697.

Parag A Pathak and Tayfun Sönmez. 2008. Leveling the playing field: Sincere and sophisticated players in the Boston mechanism. *American Economic Review* 98, 4 (2008), 1636–1652.

Tim Roughgarden. 2020. Transaction fee mechanism design for the Ethereum blockchain: An economic analysis of EIP-1559. *arXiv preprint arXiv:2012.00854* (2020).

Tim Roughgarden. 2021. Transaction fee mechanism design. *ACM SIGecom Exchanges* 19, 1 (2021), 52–55.

Elaine Shi, Hao Chung, and Ke Wu. 2022. What can cryptography do for decentralized mechanism design. *arXiv preprint arXiv:2209.14462* (2022).

Andrew Chi-Chih Yao. 2018. An incentive analysis of some Bitcoin fee designs. *arXiv preprint arXiv:1811.02351* (2018).

Appendix A. A Steady-State Analysis of EIP-1559

We now turn our attention to analytically studying the wait-times for transactions that do not bid up to the high-reserve price in our simplified 2-reserve-price model of EIP-1559. As we discussed earlier, from the perspective of such transactions, the block arrival process randomly takes "vacations": in particular, whenever a block with a low reserve price takes 2 transactions (which, in turn, occurs when there are at least 2 transactions waiting), the reserve price switches to high and does not return to low until there is an empty block.

Let us therefore consider the following discrete markov chain where the state is the number of *low* bid transactions (i.e., lower than the high reserve price) right before a block with a low reserve price. If n is the number of such transactions, denote by p_n the steady state probability of such a state occurring. Observe that if $n \geq 2$ then the next block necessarily has a high-reserve price. The corresponding next state in our markov chain will correspond to the state of the queue at the random first future block where reserve price is low. During this period some stochastic number of low bid transactions will arrive and be added to the queue. Let us denote the probability generating function of arrivals in this case by $A(z)$, i.e.,

$$A(z) = \sum_{i=0}^{\infty} a_i z^i, \tag{11}$$

where a_i is the probability of exactly i arrivals in this period. Conversely, if $n < 2$, it is possible conditional on few enough arrivals that the very next block has a low reserve price. Denote the corresponding probability generating function by $B(z)$, i.e.,

$$B(z) = \sum_{i=0}^{\infty} b_i z^i. \tag{12}$$

Finally, we will need 3 numbers c_0, c_1 and c_2. The first is the probability of 0 high-type arrivals during the period between blocks, the second 1 arrival and finally the latter 2 or more arrivals. Note that $c_0 + c_1 + c_2 = 1$ by definition. All of A, B, and c can be determined given the arrival process of low- and high-type transactions, we do this in the sequel.

Let us now write down the following:

$$x_B = (1 - c_2)p_0 + c_0 p_1,$$
$$x_A(n) = c_2 p_n + c_1 p_{n+1} + c_0 p_{n+2}.$$

We now have that in steady state, the recurrence equation that describes steady-state probabilities is given by:

$$p_n = \sum_{j=0}^{n} a_{n-j} x_A(j) + b_n x_B \tag{13}$$

Let us describe how this was derived in words. How can a queue in steady state have n low-type transactions prior to the arrival of a low-reserve price block. The first way is that the previous low reserve-price block had 0 low-type transactions and at most 1 high-type transaction arrived in the block interval preceding that, or 1 low-type transaction and 0 high-type transactions in the block preceding that. That way, the previous block ends with 0 transactions waiting in queue, but the price stays low.[16] Then b_n transactions arrive. This is the latter term. The other way is similarly that the previous low-block ended with j transactions but the price was switching up to high, and then a_{n-j} transactions arrive before the price switches back down to low. The former term sums over all such possibilities.

We can now calculate these steady state probabilities using the standard probability generating function method. Formally, multiply each (13) corresponding to p_n by z^n on

[16] If 2 or more high type transactions arrived, then the price must be rise to high.

both sides, and sum.

$$\begin{aligned}\sum_{n=0}^{\infty} p_n z^n &= \sum_{n=0}^{\infty}\sum_{j=0}^{n} a_{n-j} x_A(j) z^n + \sum_{n=0}^{\infty} b_n x_B z^n \\ &= \sum_{n=0}^{\infty}\sum_{j=0}^{n} a_{n-j} z^{n-j}\left(c_2 p_j + c_1 p_{j+1} + c_0 p_{j+2}\right) z^j + B(z) x_B\end{aligned}$$

Note that the left hand side is the probability generating function for this markov chain, which in keeping with our convention we will denote as $P(z)$ so we can write this as:

$$\begin{aligned}P(z) &= \sum_{n=0}^{\infty}\sum_{j=0}^{n} a_{n-j} z^{n-j}\left(c_2 p_j + c_1 p_{j+1} + c_0 p_{j+2}\right) z^j + B(z) x_B \\ &= \sum_{n=0}^{\infty}\sum_{j=0}^{n} a_{n-j} z^{n-j}\left(c_2 p_j z^j + \frac{c_1 p_{j+1} z^{j+1}}{z} + \frac{c_0 p_{j+2} z^{j+2}}{z^2}\right) + B(z) x_B \\ &= A(z) c_2 P(z) + \frac{A(z) c_1 (P(z) - p_0)}{z} + \frac{A(z) c_0 (P(z) - p_0 - p_1 z)}{z^2} + B(z) x_B.\end{aligned}$$

Collecting terms we therefore have that:

$$\begin{aligned}P(z) &= \frac{B(z)\left((1-c_2)p_0 + c_0 p_1\right) - A(z)\left(\frac{p_0 c_1}{z} + \frac{p_0 c_0 + p_1 c_0 z}{z^2}\right)}{1 - c_2 A(z) - c_1 \frac{A(z)}{z} - c_0 \frac{A(z)}{z^2}} \\ \implies P(z) &= \frac{z^2 B(z)\left((1-c_2)p_0 + c_0 p_1\right) - A(z)\left(p_0 c_1 z + p_0 c_0 + p_1 c_0 z\right)}{z^2 - c_2 A(z) z^2 - c_1 A(z) z - c_0 A(z)}.\end{aligned}$$

To determine p_0 and p_1 we can use the following facts. First, $P(1) = 1$. Note that the numerator and denominator both evaluate to zero at $z = 1$ since $A(1) = B(1) = c_0 + c_1 + c_2 = 1$. We can use L'Hospital's rule. Note that the derivative of the denominator with respect to z is

$$-c_0 A'(z) - c_1 z A'(z) - c_2 z^2 A'(z) - c_1 A(z) - 2c_2 z A(z) + 2z.$$

Evaluated at $z = 1$, observing that $A(1) = 1$ by definition, we have

$$= 2 - c_1 - 2c_2 - A'(1)$$

Similarly, the derivative of the numerator with respect to z equals:

$$-((c_1 p0 + c_0 p1) A(z)) + 2((1-c_2)p_0 + c_0 p_1) z B(z) - (c_0 p_0 + c_1 p_0 z + c_0 p_1 z) A'(z) + ((1-c_2)p_0 + c_0 p_1) z^2$$

and substituting in $z = 1$, we have:

$$-((c_1 p0 + c_0 p1)) + 2((1-c_2)p_0 + c_0 p_1) - (c_0 p_0 + c_1 p_0 + c_0 p_1) A'(1) + ((1-c_2)p_0 + c_0 p_1) B'(1)$$

Therefore we must have that:

$$\frac{-((c_1p0 + c_0p1)) + 2((1-c_2)p_0 + c_0p_1) - (c_0p_0 + c_1p_0 + c_0p_1)A'(1) + ((1-c_2)p_0 + c_0p_1)B'(1)}{2 - c_1 - 2c_2 - A'(1)} = 1$$

As an example, note that the arrival process of low type transactions is known to be Poisson with arrival rate λ_L, so $B(z) = \exp(\lambda_L(1-z))$. To simplify further, suppose that only 0, 1, or 2 high type transactions arrive in each period. In this case, upon a switch to the high price, there can be no pending high-type transactions, and the number of periods of high prices is geometrically distributed with stopping probability c_0, i.e. probability of there being k periods of high-prices is $(1-c_0)^{k-1}c_0$. Let us denote this probability generating function of this distribution by $H(z)$, where we know that

$$H(z) = \frac{c_0 z}{1-(1-c_0)z}$$

Therefore we must have that $A(z) = H(B(z))$, i.e.,

$$A(z) = \frac{c_0 \exp(\lambda_L(1-z))}{1-(1-c_0)\exp(\lambda_L(1-z))}.$$

These can be substituted into the probability generating function to find an analytic expression for the expected queue length, and therefore the average wait times via Little's law.