StableFees: A Predictable Fee Market for Cryptocurrencies

Soumya Basu[*], David Easley[†], Maureen O'Hara[‡] and Emin Gün Sirer[*]

July 11, 2021

Abstract

Blockchain-based cryptocurrencies must solve the problem of assigning priorities to competing transactions. The most widely used mechanism involves each transaction offering a fee to be paid once the transaction is processed, but this first-price mechanism fails to yield stable equilibria with predictable prices. We propose an alternate fee setting mechanism, StableFees, that is based on second-price auctions. We prove that our proposed protocol is free from manipulation by users and miners as the number of users and miners increases and show empirically that gains from manipulation are small in practice. We show that StableFees reduces the fees paid by users and reduces the variance of fee income to miners. Data from December 2017 shows that, if implemented, StableFees could have saved Bitcoin users $272,528,000 USD in transaction fees while reducing the variance of miner's fee income, on average, by a factor of 7.4. We argue that our fee protocol also has important social welfare and environmental benefits.

We thank Phil Daian, Amani Moin, Andrew Morgan, David Yermack and participants at the Georgetown Virtual Seminar Series on Fin Tech and the Crypto and Blockchain Economics Research Conference (CBER) for helpful comments. We also thank our discussant at CBER, Mariana Khapko, for valuable comments and suggestions and Dhruba Basu for assistance with data and graphs.

[*] Department of Computer Science, Cornell University

[†] Departments of Information Science and Economics, Cornell University

[‡] College of Business, Cornell University

1 Introduction

Almost all decentralized cryptocurrencies use the same basic mechanism to prioritize transactions. A user who wants their transaction included in the blockchain attaches a fee to the transaction. This fee serves as a bid for block space. The miner who is building the block then chooses which transactions to include in their block and collects the respective transaction fees from the included users. This mechanism gives the miner an obvious incentive to select the highest fee transactions and it plays a crucial role in rewarding miners for processing transactions.

It is useful to note that although fees were envisioned in the original Bitcoin protocol that protocol did not describe exactly how they would work. The fee mechanism has evolved over time to allow users to signal, and pay for, their desire to have their transactions included in the blockchain before others transactions are included. The initial mechanism took into account a combination of factors including the age of the coins being spent (”bitcoin days destroyed”), but has by now been almost universally replaced by a first price auction mechanism of the kind described above. There is no reason to expect this emergent payment mechanism to result in an efficient, stable or predictable fee market, and that appears to be the case.

The current fee mechanism produces very volatile prices for block space. For example, in Bitcoin, the average daily fee paid in December 2020 ranged from \$2.72 to \$12.05. In December 2017, during a period of heavy trading activity, the average daily fee ranged from \$5.82 to \$61.44. This volatility makes it difficult for users to decide what fee to attach to a transaction and this hampers usability of the crypotcurrency. Users who bid too high have overpaid for their transaction to get confirmed, while users who underbid may not be included in the block even if their transaction is more valuable than some transactions in the block. This can cause block space to be inefficiently allocated; low value transactions may be confirmed while high value transactions are still pending.

To provide insight into why this volatility occurs, we note that the current cryptocurrency fee market shares important features with first-price auctions where users act as bidders and miners act as auctioneers. While there are key differences between the fee market and auctions, prior experience with first price auctions suggests why the current fee market is unstable. In a first price auction for multiple, identical items, the highest bidder pays his bid and gets the first item, the second highest bidder pays his bid and gets the second item, and so on until either items or bidders are exhausted. These auctions do not have a dominant strategy equilibrium. While efficient Bayes-Nash equilibria exist for first price auctions for multiple, identical items with symmetric bidders, these equlibria require users to model accurately the values of other user's bids, which is a difficult, if not impossible, task in the cryptocurrency environment. In Bitcoin, we see this difficulty reflected in the behavior of users who, instead of revealing the full utility of their transaction in the fee they bid, prefer to bid low at first and only increase their bid if they are waiting too long.[1] The strategic bidding induced by the first-price-like nature of the current

[1] The difficulty of determining what to bid has led to the development of bid prediction firms whose models suggest where the bid might be in upcoming blocks. Such predictions, however, can serve as focal points and lead to bidders to try to outbid the predicted bid.

mechanism is an underlying cause of fee instability and inefficiently allocated block space in cryptocurrencies.

Adapting insights from second price auctions to the cryptocurrency setting could potentially ameliorate the above drawbacks. In a second price auction, truthfull bidding is a dominant strategy. So rational users do not need to strategize; instead they can bid a fee equal to their value of having their transaction included in the block. This would make it possible for miners to more efficiently allocate block space, resulting in a more usable and valuable cryptocurrency.

There are several challenges to modifying existing auction theory for application to the cryptocurrency fee market. First, the fee mechanism can only control what each user will pay for their transaction (given their bid) and the reward earned by the miner given the set of transactions they include in the block. Miners are allowed to place any transactions they want in a block, including fee-paying transactions created on the fly after observing users' bids. Thus miners, unlike trusted auctioneers, may be able to manipulate the fee mechanism. Second, users' payments cannot depend on fees attached to transactions not included in the block as the details of these unused transactions are not externally verifiable. Thus, payments must depend only on accepted bids and so truthful bidding cannot be a dominant strategy. These constraints, inherent to most decentralized cryptocurrencies, prevent us from directly employing a multiunit second price auction format.[2]

In this paper, we present StableFees—a mechanism inspired by second price auctions. StableFees provides provable non-manipulation guarantees for both users and miners. We show that as the number of users increases, users' gain from bidding strategically converges to zero. This result demonstrates that in large markets users have a nearly dominant strategy of bidding truthfully. Additionally, we show that miners' gain from manipulating the transactions they include in a block also converges to zero as adoption increases. We demonstrate this result both theoretically and through simulations on real transaction fee distributions. We also show that in the large markets environment, StableFees results in maximal social welfare. An empirical analysis of the Ethereum and Bitcoin blockchains suggests that users could have saved \$13.2 million and \$273 million respectively if StableFees was implemented during December 2017. Empirically comparing StableFees with other fee mechanisms using demand curves based on real Bitcoin data, we show that StableFees provides 49% to 103% more welfare than comparable schemes. Finally, we argue that our mechanism, by reducing unnecessary mining, can be more environmentally sustainable.

1.1 Prior Work

Several recent papers analyze the the current Bitcoin protocol, the games it induces and its efficiency or the lack thereof, and at least two papers propose alternative protocols. Easley et al. (2019) and Huberman et al. (2017) analyze transaction fees, the mining game and waiting times for users in the current Bitcoin protocol. Houy (2014) and Cong et al. (2018) provide analyses of the mining game. Lehar and Parlour (2020) document inefficiencies (non-filled blocks and excess fees) that can be

[2] More precisely, a $K+1$ price auction for K slots on the block.

explained by collusive-price discrimination by miners. Carlsten et al. (2016) describe some consequences of removing the block rewards. Böhme et al. (2015); Harvey (2016); Malinova and Park (2017); Raskin and Yermack (2018); Yermack (2017) and Aune et al. (2017) all examine aspects of the Bitcoin environment. Azevedo et al. (2020) suggest an alternative auction to use to handle transactions bundled off the blockchain and then brought to a block in a batch. Guasoni et al. (2021) develop these (second layer) solutions in more detail and determine when these channels will emerge in cryptocurrencies. Rosenfeld (2011), Eyal and Sirer (2014), Gans and Halaburda (2015), and Gandal and Halaburda (2016) consider various design issues of the Bitcoin protocol, though they mostly focus on security rather than the efficiency of the fee mechanism. Chen et al. (2021) provides an introduction to the economic analysis of blockchains.

Buterin (2018) proposes an alternative mechanism (EIP-1559) and Roughgarden (2020) provides an economic analysis of this mechanism. This alternative mechanism is based on miners estimating, and dynamically adjusting, a single fee that is charged uniformly to all transactions within a block, coupled with dynamically varying the block size to accommodate demand. This approach differs from ours in a few key ways. First, it does not aim to maximize social welfare, and instead adopts heuristics to modify two independent variables, fees and block size. Modifying fees, similar to our work, will maximize transactions cleared subject to any desired block size constraint, determined by any desirable mechanism. However, since block size is a primary determinant of security and centralization, we believe it is prudent to decouple its management from the fee mechanism. Second, Buterin's approach assumes that the demand curve is relatively stable, so that all transactions meeting the fee threshold can be included in the next block. If the demand curve could be inferred accurately such that all transactions whose utility exceeds the block fee can always be accommodated, then Buterin's proposal would have no incentive issues. However, inferring demand curves is difficult in adversarial, Byzantine environments, which is why auction mechanisms are used. Finally, this approach has not been proven to be resistant to manipulation by users and miners. Indeed, there is an uncoordinated attack where users may artificially manipulate the demand curve to simulate demand spikes, a scenario where Buterin's proposal has comparatively weak guarantees, see Tefagh (2021). If it is not resistant to manipulation, then this mechanism will suffer from the same problem as the current first price mechanism, where users have to solve the fee selection problem.

The most closely related work is the monopolistic miner protocol of Lavi et al. (2017).[3] They propose a protocol in which the winning miner decides how many transactions to put into the block and charges all of them the lowest fee proposed by any transaction he placed in that block. There are fundamental differences between our approaches stemming from our goals and setup. Lavi et al. (2017) assume a single monopolistic miner, and strive to maximize revenue from fees at a cost of lower social welfare. In contrast, our work explicitly targets maximizing social welfare, and operates with many miners. In their system, the monopolistic miner is incentivized to leave transactions offering positive fees out of the block even

[3] Yao (2018) provides proofs of conjectures from Lavi et al. (2017) about the general incentive compatibility and profitability of their monopolistic miner protocol.

if there is space in the block as including them reduces the uniform price he can charge (such behavior is consistent with the findings of Lehar and Parlour (2020)). This behavior is important for maximizing miner revenue, but we believe that the first criterion for a viable protocol should be to use the blockchain efficiently, as otherwise users are discouraged from participating. Their non-manipulation result is stronger than the one we obtain from our mechanism as we only obtain declining gain from manipulation as the system grows, but their result comes at a cost of lower social welfare. Finally, in both our protocol and the protocol proposed by Lavi et al. (2017), users' incentive to behave strategically vanishes as the number of users grows.

In the remainder of the paper, we first discuss the positive and negative aspects of using a multi-unit first price auction for slots on the Bitcoin blockchain and we describe our goals for an improved mechanism. We then present our model of the cryptocurrency fee market and our mechanism, StableFees, which mitigates issues present in the current fee market. To do this, StableFees incorporates and adapts ideas from second price auctions and the VCG (Vickrey-Clarke-Groves) procedure to the trustless, decentralized setting present in today's cryptocurrencies. The key difference between the cryptocurrency fee market and the traditional auction setting is that miners are not trusted auctioneers so they can engage in behavior that manipulates the auction. StableFees accounts for these differences and has provable guarantees about potential manipulation by miners and users. Finally, we provide an analysis of the social welfare achieved by StableFees, EIP-1559 and a monopolist miner.

2 Motivation

To motivate our design, we examine the lessons learned from auction design in the sponsored search market and then discuss challenges unique to the fee market.[4] Overture, the first company to use keyword-based advertising, initially sold ads using a generalized first price auction. In the sponsored search market, advertising slots on the page that appears in response to a search term are sold to advertisers. Slots near the top of the page are preferred to ones down the page and all of these dominate those on the second page, and so on. Auctions are run frequently to determine whose ad appears where. Overture and its advertisers experienced instability: bids in successive auctions would rise as advertisers priced out in one auction tried to get into the next one; and then they would crash once bids reached levels that discouraged bidding at all. Eventually, discouraged advertisers quit and the auction was clearly producing less revenue than should be possible. Figure 1 illustrates that this erratic behavior of bids is also prevalent in the current Bitcoin auction mechanism.

An important aspect of Google's subsequent success in the sponsored search market was its use of a superior auction form: GSP, Google's generalization of the single-unit second price auction to their multi-unit environment. GSP does not have dominant strategies, but it is second-price-like and simple, and it works reasonably

[4] See Easley and Kleinberg (2010), Chapter 15 for a discussion of the sponsored search market and references to the literature on sponsored search.

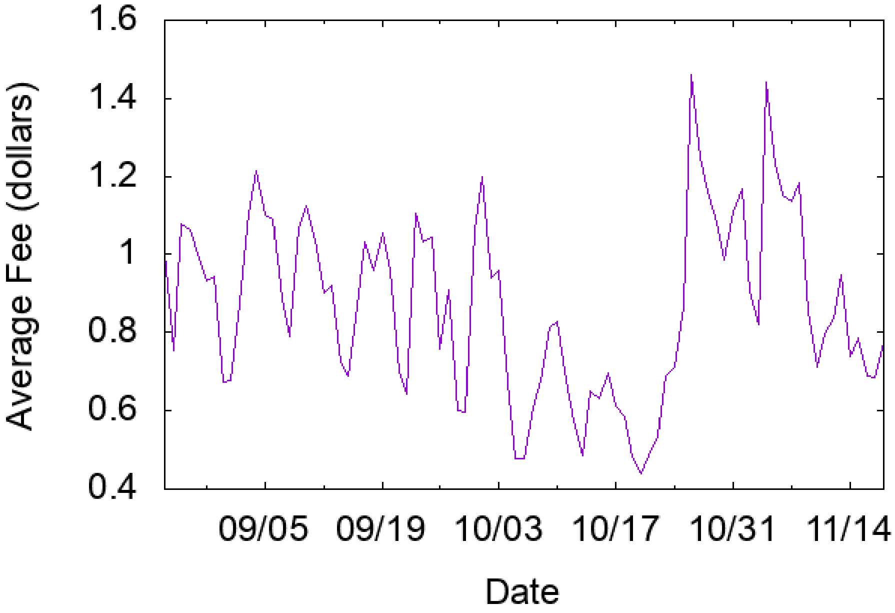


Figure 1: This figure generated from BitInfoCharts (2021) shows a sawtooth pattern for Bitcoin fees over the period August to November 2015. This is similar to the figures and description from Edelman and Ostrovsky (2007) about Overture's first price auction.

well in practice. An earlier generalization of the single item second price auction to multiple items that has dominant strategies is the Vickrey-Clarke-Groves (VCG) procedure, which forms the basis of the auction mechanism used by Facebook[5].

2.1 Lessons From Sponsored Search

Auction theory and the experience of the sponsored search market suggest that some generalization of the second price auction could improve on the Bitcoin protocol. Before modifying a second price auction to fit the bitcoin environment, it is useful to first set out our objectives in designing a protocol and then to describe how multi-unit second price auctions work.

We have three objectives. First, the protocol should result in an efficient assignment of slots on each block to Bitcoin users. So we want to assign slots to users with the highest values, leaving a user out of a block only if there is no user in the block who has a lower true value than the left-out user. An assignment with this property is called **socially optimal**. Second, we want the game induced by the protocol to incentivize non-strategic behavior. Ideally, we would like users' optimal bids, which are the fees they propose to pay, to be their true values for slots and we would like the miner building the block to have no profit motive for deviating from the "rules

[5] See Varian and Harris (2014) for a discussion of the development of GSP and Google and the use of VCG by Facebook

of the auction." Third, we want optimal strategies to be simple and obvious. This last criterion is difficult to quantify, but a protocol that induces a game in which every participant has weakly dominant truth-telling strategies surely satisfies it.

In the standard auction environment a generalized second price auction achieves these goals.[6] A generalized second price auction for K identical items to be sold to $N > K$ bidders who each want at most one item works as follows. Bidders are asked to submit bids to the seller, or to the algorithm running the auction. The bidders who have submitted the K highest bids each win an item and they all pay the $(K + 1)$st highest bid. If the algorithm, or auctioneer, can commit to this auction form, and if bidders private valuations for an item are independent, identically distributed draws from a fixed distribution, then it is a weakly dominant strategy for each bidder to bid truthfully—submit a bid equal to the value for an item. This auction form has another attractive feature—it guarantees that winning bidders place the highest values on the items, so it results in a socially optimal allocation.[7] The following remark summarizes standard results about multi-unit auctions.

Remark: Suppose that the auctioneer has K identical items for sale and can commit to an auction form. Suppose also that each bidder, $i = 1, \ldots, N$ with $N > K$, wants at most one item and that bidders' private values V_i are drawn iid from a distribution on $[0, \bar{V}]$. The auction form chosen by the seller induces a game between the bidders in which each bidder selects a strategy mapping the bidder's private value, V_i, to a bid $b_i \in [0, \bar{V}]$.

1. If the auctioneer runs a generalized second price auction—the K items are sold to the K highest bidders at the $K + 1$st highest bid—then it is a weakly dominant strategy for each bidder to bid truthfully, $b_i = V_i$ for all i, and if each bidder follows this dominant strategy, the assignment induced by the auction is socially optimal.

2. If the number of bidders and the distribution of values is common knowledge, and the auctioneer runs a generalized first price auction—the K items are sold to the K highest bidders and each successful bidder pays his own bid—then there is a Bayes-Nash equilibrium (a list of strategies, one for each bidder, which are mutual best responses) of the game induced by the auction in which the equilibrium assignment is socially optimal.

The observations discussed in this section are formally stated and proved in Appendix A.

For the environment described in the Remark, generalized first price auctions and generalized second price auctions both result in socially optimal assignments. However, in sponsored search the second price-like auction performs better than the first price-like auction. In a second price auction, each bidder only needs to know

[6] We call this a "generalized" second price auction as it is an auction for K items at the $(K+1)$st highest bid rather than an auction for one item at the second highest bid. It is not the GSP procedure used by Google.

[7] This auction is equivalent to the VCG procedure. It is simpler to explain than VCG, but its optimality does depend on the items being identical. VCG does not require that restriction.

his own value and the form of the auction. Bidding truthfully is optimal regardless of who the other bidders are or how they behave. This is not true in the first price auction. Here, the efficiency claim rests on the assumption that play can be described by a Bayes-Nash equilibrium in which each bidder is best responding to each other bidder.

To illustrate the difference in these two auctions, it is useful to examine them in the simplest case in which there is a single item for sale to N bidders with values, V_i, drawn iid from the uniform distribution on $[0,1]$. In a second price auction, it's weakly dominant for each bidder i to simply bid his value V_i. In a first price auction there is an equilibrium in which the optimal strategy for a bidder with value V_i is to bid $(\frac{N-1}{N})V_i$. This first-price result requires knowledge of the number of bidders, depends on distribution of values being uniform, and is optimal only if all other bidders follow the same strategy. However, it does result in a socially optimal allocation because equilibrium bids are increasing in true values.[8]

2.2 Redesigning the Fee Market

Bitcoin is a trustless, decentralized system in which there is no mechanism that can force miners to act as if they are the auctioneer in a generalized second price auction. So any redesigned mechanism has to take into account the incentives of the miners to follow the "rules" of the auction rather than to manipulate it.[9] Furthermore, only the miner knows the transactions and their attached fees in his mempool. Once he writes transactions to the blockchain the details of those transactions are known, but details of the transactions left-out are not known—and the protocol cannot credibly call for payments that depend on those left-out transactions.

Most importantly, the miner can also act as a user and include his own transactions in the block he is mining, moving money from one of his wallets to another with whatever fee he chooses after observing the fees offered by users. All identities on the blockchain (miners, users, etc) are uniquely identified by a cryptographic key. Thus, it is cheap to create a new identity, but hard to assume the identity of another person. This makes it difficult to enforce roles for each participant since it is possible for a miner to also impersonate other, arbitrarily many, identities which all act as "users".

This ability to act as a "user" or many "users" allows a miner who earns the revenue generated by the block to introduce first-price-like features into a supposedly second price auction at zero cost to himself. To see this in the simplest case, suppose

[8] Both auctions also yield the same expected revenue for the seller (in equilibrium). For the example in the text, a simple calculation shows this, but it is true much more generally according to Myerson's Revenue Equivalence Theorem, Myerson (1981).

[9] Akbarpour and Li (2018) provide an analysis of mechanisms in which the seller can deviate from the rules of the auction. In this case, the mechanism has to be incentive compatible for the seller. They show that a first price auction is the only credible static auction. Essentially, an auctioneer could announce a different auction, such as a second price auction, but then once bids are received, he can submit a false second highest bid just below the actual highest bid—turning the auction into a first price auction. Credibility also matters for our analysis as our miner can submit own bids; but our environment differs as there are multiple items for sale, the mechanism can impose some constraints on the miners, and, most importantly, miners revenue can depend on the fees generated by a sequence of blocks determined by the protocol.

that there is only one transaction per block, a second price auction is announced and all fees that bidders submit can be observed and used by the protocol. The miner can manipulate this auction by including a fictitious transaction paying a fee slightly below the highest offered real fee, so the user with the highest offered fee wins and pays approximately that fee while the miner pays nothing. This makes the single item, "second price" auction with a strategic auctioneer effectively a first price auction. So bidders should place first price bids and in equilibrium, we should see a first price outcome. Nonetheless, we show that with multiple bidders and multiple slots on the block, StableFees achieves second-price-like results.

StableFees starts with the intuition from second price auctions that the fee to be included in a block should be approximately the minimum bid that was included in that block. However, miners are incentivized to manipulate such an auction in order to try to maximize their revenue. To discourage this behavior, StableFees spreads out the fee reward from a block across several blocks so that miners are unable to insert transactions into each block for free. This allows StableFees to provide similar guarantees to second price auctions in realistic conditions while still being deployable in the cryptocurrency ecosystem.

2.3 Additional Design Issues

The potential for side payments and security issues associated with the choice of block size introduce additional limitations and considerations in any redesign of the fee market.

Side payments arise when the issuer of a transaction and the miner of a block arrange a payment outside of the blockchain itself. These payments may be direct, with fiat money or other cryptocurrency tokens changing hands, or indirect, as when a miner pays out rewards to its pool participants. It is difficult for any auction mechanism to effectively deal with side payments as they are inherently more valuable to one particular miner and that value is not shown in the bid for the transaction. We require our design to be resistant to manipulation due to side payments, but we do not see eliminating side payments entirely as either feasible or desirable.

In cryptocurrencies, the hard cap block size is set according to a variety of factors, including security and the fee level. Blocks that are larger than the hard cap are deemed invalid and are not included on the chain. If blocks are small enough, then their size has no effect on the network security—and this secure size is slowly increasing as the underlying network infrastructure improves. However, beyond a certain safe block size, the larger the block, the longer it takes to transmit and the more likely it is to cause adverse effects on the network security. Protocol designers trade off these properties when choosing the hard cap and in our view, the choice of a hard cap should be exogenous to the fee market design.[10]

We note that if a miner is unable to fill a block to a sufficient level, then the price of block space should be approximately zero as space on the block is not scarce. As a result, the auction mechanism should charge transactions in under-filled blocks only some nominal transaction fee.

[10] The block size for a cryptocurrency has implications for the security and performance for a cryptocurrency (see Gencer et al. (2018), Croman et al. (2016)).

We outline a path towards deployment and other considerations in Appendix B.

3 StableFees

We now present the formal model in which StableFees operates, the StableFees protocol and performance guarantees provided by StableFees.

3.1 Model

We denote the fixed number of slots in a block by K. These are the slots that are assigned to transactions by the auction mechanism.[11] We assume that there are $N > K$ users. Users have private values for having their transaction recorded to the current block. These values are denoted V_i, $i = 1, \ldots, N$, and they are drawn iid according to a continuous density g on $[0, \bar{V}]$ with $g(V) \geq \epsilon > 0$ for all $V \in [0, \bar{V}]$.[12] A user who is not included in the current block receives no reward from the current block.[13] Some of these users have transactions waiting in the mempool at the time the current block is being constructed and others are absent. We assume that each user randomly is selected to add a transaction to the pool independently with probability δ where $1 > \delta > 0$.

Users attach transaction fees, or bids, to their transactions. We model users as selecting bids after knowing their own value and knowing how many users are active, but without knowing the realization of values or the choice of bids for other users. We assume that distributions of users and values are common knowledge.

The miner selects which transactions to put into the block after seeing the bids attached to those transactions. We do not address how this miner is selected. However that is done (for example, by proof-of-work or proof-of-stake) our protocol can be applied to determine fees. We consider only blocks for which the number of users in the pool is greater than the number of slots in the block; for other blocks there is no congestion as all users in the pool can be included in the block.

3.2 Protocol Description

1. Any user who wants a transaction recorded in block b can attach a fee to their transaction. Denote the fee attached by user i by f_i.

2. Users whose transactions are included in block b each pay the minimum fee proposed by any user whose transaction is included in block b. The total paid by the users in block b is the revenue generated by block b.

[11] For ease of exposition, our model uses the terminology used in popular proof of work cryptocurrencies, such as Bitcoin or Ethereum. Most cryptocurrencies operate on batches of transactions analogous to blocks, and transaction priority is decided by user's bids.

[12] We take waiting time into account indirectly, through users differing values for block space and we do make use of a sequence of blocks, but we analyze the users' blockchain game only one block at a time.

[13] For simplicity of exposition, we treat all transactions as taking the same amount of space on the blockchain. In practice, the fees we discuss are normalized in some way. For example, in Bitcoin, this would be the fee per byte. Dependent transactions, such as child pays for parent, can be handled by charging the average fee for both transactions.

3. The miner who builds block b is paid the average revenue generated by the B most recently mined blocks, including block b, if and only if the miner fills block b. Otherwise, the block is not included in the blockchain.

    - A block is defined to be filled if it contains K transactions or if the miner pays a *fill penalty.* The necessary fill level, K, is a parameter which can be chosen to be some fraction, say 80%, of the hard cap. The fill penalty is defined to be the difference between K and the number of transactions in the block times the fee paid by each transaction. A miner can also avoid paying the fill penalty by declaring that there were not enough transactions in the mempool to fill the block, in which case each user is charged the minimum allowable fee for a transaction to be included in the mempool and the block is declared filled.
    - A minimum fee required for a transaction to be considered can be included by declaring that transactions are not in the mempool if the proposed fee is below that minimum level.[14]
    - The miner has the option to fill the block to capacity with transactions, but only K of them are priced using this auction mechanism. The other transactions are charged no fees for being included in this block. This allows the miner to, for example, pay pool participants, and is helpful in creating side payment resistance.

This protocol induces a Bayesian game with users and the miner as players. User i's strategy is a mapping from V_i to f_i. The miner chooses which bids to accept (user transactions to place in the block) and any fictitious bids to insert. We analyze Bayes Nash equilibria of this game.

3.3 StableFees Guarantees

If the number of users is small there are incentives for both users and miners to manipulate StableFees. As StableFees uses a K-th price auction for K slots, the marginal sets the price and so has an incentive to under-bid. Ex-ante any user could be the marginal user, so all users have an incentive to underbid. Miners can manipulate by inserting fictitious transactions into a block hoping to increase the minimum fee attached to transactions included in the block without losing too many real transactions. We discuss the small numbers case later in this section and address it empirically and with simulations in Section 4. We next consider the large numbers case and show that the incentives for both users and the miner to manipulate decline to zero as the number of users grows relative to the size of a block.

Truthful User Bidding with a Large Number of Users. Assuming, as we will demonstrate later for the large numbers case, that miners place the K highest fee transactions on the block, and that users whose transactions are placed on the

[14] Including a minimum bid is standard in sponsored search auctions. For example, the minimum bid that Google currently uses is one-cent per click.

block all pay the Kth highest fee, it is not a dominant strategy for users to bid truthfully. However, the incentive to bid strategically is small if the number users is large. To see this, suppose that all other users bid truthfully. Let V_{K-1} and V_K be the $K-1$st highest bid of others and the Kth highest bid of others. If a user's value is below V_K there is no possible gain from bidding strategically as the price will be greater than the user's value for any bid. There is a potential profit from strategic bidding only if the user's value is greater than V_K and this gain is bounded by $V_{K-1} - V_K$. So for user i the expected gain to strategic bidding is bounded by $E[(V_{K-1} - V_K)]$ which converges to 0 in the number of users. For example, with draws of user values according to the uniform distribution on $[0, 1]$ and A active users, the upper bound on gain is $1/A$ and with the exponential with parameter λ, it is $1/\lambda(A - K + 1)$. That is, with a large number of users, the potential gain to strategic user behavior is small and it seems plausible that, rather than attempting to follow a complex manipulation strategy, users will instead follow the simpler, nearly optimal, strategy of truthful bidding.

Proposition 1: In the model above, if the StableFees mechanism is employed then:

- Truthful bidding is not a dominant strategy for users.
- If all other users bid truthfully, the expected gain to any bidder from strategic bidding converges to 0 as the number of users diverges.

Proofs are given in Appendix C.

Non-Manipulation by Miners with a Large Number of Users. Consider first the case in which miner revenue is not averaged over past blocks; that is, $B = 1$. In this case the miner of block b is paid the revenue generated by block b. Averaging over past blocks reduces incentives to manipulate, so in this subsection, we consider the case that is most demanding for a non-manipulation result.

A miner can manipulate by inserting a fictitious transaction with a fee equal to any of the K highest fees offered.[15] Relabeling the K highest fees from highest to lowest, they are $f_1 \geq f_2 \geq \cdots \geq f_K$. For a miner to not manipulate, we need the revenue generated from K transactions at the Kth highest bid to be greater than the revenue generated from any smaller number of transactions n at the nth highest bid, i.e. $Kf_K > (K-1)f_{K-1}$, $Kf_K > (K-2)f_{K-2}$, and so on. This clearly holds if it holds sequentially, i.e. $Kf_K > (K-1)f_{K-1}$, $(K-1)f_{K-1} > (K-2)f_{K-2}$, and so on. This second collection of inequalities can be written as $nf_n > (n-1)f_{n-1}$ for each $n = 2, \ldots, K$. Or $f_n > (n-1)(f_{n-1} - f_n)$ for each $n = 2, \ldots, K$. If users bid truthfully, then for any fixed n, as the number of users (N) diverges, the left-hand side of this inequality converges to $\bar{V}$ almost surely and the right hand side converges to 0 almost surely. So the miners' gain from manipulation vanishes as the number of users grows.[16]

[15] Inserting a fee between two existing fees is clearly dominated by making the fictitious fee equal to the higher of two nearby fees. Inserting multiple fictitious fees above the Kth highest fee knocks some number of transactions out of the block and sets the price at the lowest fee remaining in block and so is equivalent to a single fictitious fee strategy.

[16] Note that inserting a fictitious transaction with fee greater than the Kth highest real fee is equivalent to choosing to not fill the K slots on the block. So this argument also covers the potential manipulation of restricting the supply of slots.

Proposition 2: In the model above, if the StableFees mechanism is employed and users bid truthfully, then a miners' gain from optimal manipulation declines to 0 as the number of users, N, grows to ∞.

Efficiency with a Large Number of Users The above non-manipulation results for both users and miners suggest that, in the large numbers case, users will approximately bid truthfully, making bids increasing in values, and miners will approximately fill blocks with the users whose bids, and thus their values, are highest. This suggests that in an equilibrium with a large number of users the space on the block will be used approximately optimally. The following proposition formalizes this intuition.

Proposition 3: In the model above, if the StableFees mechanism is employed, then for *any* Bayes-Nash equilibrium of the induced game, the ratio of the equilibrium sum of included user values to the maximal sum of included user values and the ratio of equilibrium miner revenue to maximum miner revenue converges to one with probability one as the number of users, N, grows to ∞.

Miner Incentives with a Small Number of Users. If the number of users is small then users may not bid truthfully, but regardless of how fees are chosen by users, StableFees provides an non-manipulation incentive for miners through averaging over B blocks. In the remainder of this section we discuss the effect of B on this incentive, and in the next section we address it with simulations.

Fee-Based Ordering. Recall that the miner's reward is the average revenue generated over the last B blocks, including the block that the miner has just mined.[17] So the miner receives a fraction of the reward generated by any block that they mine. Thus when choosing which transactions to include in a block the miner has an incentive to order the transactions according to the fees they offer and accept the highest fee transactions first. This force alone does does not imply that the miner will fill the block; averaging over multiple blocks provides that incentive.

Full Blocks. StableFees incentivizes miners not to submit fake transactions to fill blocks. To see why, note that the optimal fake-bid manipulation for a miner is to insert fake bids that are equal to the minimum fee bid by a transaction that the miner wants to include in the block. Each fake transaction from the miner will require the miner to pay the associated fee. Since each block's reward is computed over the past B blocks, the fees collected from a particular block are spread over the next B blocks, including the current block. Thus, the miner can only expect to get a fraction of the increase in reward back.[18] Exactly how many blocks B to average over is an empirical question that depends on how much mining power the largest miner in a blockchain controls.

Note that the fill penalty in StableFees allows miners to perform a manipulation equivalent to inserting fake bids without filling the block with unnecessary transactions. Miners will prefer to use the fill penalty rather than inserting fake bids as

[17] The fill penalty is considered part of the reward for a block.

[18] In a proof of work system his fraction is the fraction of mining power that this miner has.

larger blocks are more likely to get forked and excluded from the blockchain due to random chance.[19]

Side Payment Resistance. StableFees is also designed to resist side payments made outside the auction. StableFees incentivizes miners to place transactions that have value to the miner using the space in the block not under the auction mechanism. This enables the miner to capture the full value of the transaction and include it in the blockchain without paying other miners a fee. This includes transactions that have business value, e.g. payments to members in their mining pool, or side payments from users to include their transaction in the block. This space is limited and should only be available at a premium cost higher than the auction clearing price, making it unattractive to users.

4 Analysis

In this section we evaluate StableFees' properties empirically. First, we use simulations to understand the miner's incentive to manipulate StableFees. Second, we analyze bids during a period where block space was scarce in Bitcoin and Ethereum to estimate how much users are overpaying relative to StableFees and we estimate the reduction in variance of miners revenue achieved by StableFees. Finally, we use simulations to understand the performance of StableFees compared to alternative auction mechanisms that were proposed by Lavi et al. (2017) and Buterin (2018).

4.1 Miner Manipulation

We ran a series of simulations to provide insight into a miner's incentive to manipulate StableFees. We show diminishing benefits to miners from manipulation as the number of miners increases or as the number of blocks we average over in determining miner fees increases. Overall, our results show that for reasonable parameters even optimal manipulation by miners has little benefit.

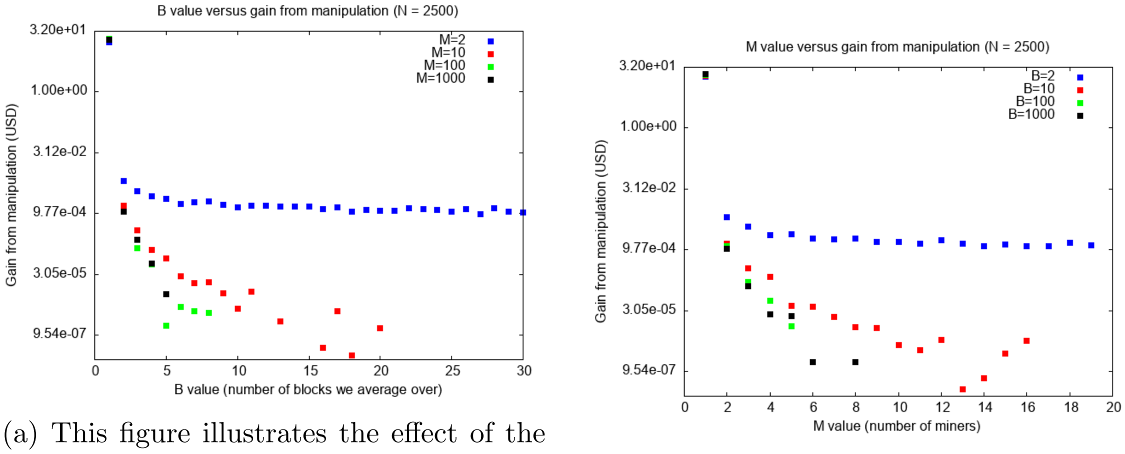
Our simulations take as parameters the number of transactions (N) in the mempool, the maximum number of transactions (K) per block, the number of miners (M) and the number of blocks transaction fees are averaged over (B). We first draw user bids from a power law distribution with a median of 2 cents and a mean of 10 cents, which is similar to the actual transaction fee distribution that appeared on the blockchain in July 2018. The miner then chooses j real transactions to include in a block (where $j \leq K$) and fills the rest of the block with fake transactions. The total fee generated by this block is j times the jth highest bid from the user distribution, which we denote as b_j so the total fee is jb_j. The manipulating miner receives $\frac{jb_j}{B}$ as a reward from fees on this block.

A miner with $\frac{1}{M}$ of the hashpower also expects to receive $\frac{jb_j(B-1)}{MB}$ in fees because this miner is expected to mine $\frac{1}{M}$ of the other $B-1$ blocks that the transaction fees are averaged over. Finally, we subtract the amount that the miner paid to

[19] Forking occurs when two blocks are mined that cannot both be included in the blockchain (e.g. when they have the same height). Any block that is mined has a chance of being forked, but larger blocks have a higher chance of being forked. The probability of a fork occurring depends on the particular chain's properties.

insert fake transactions, which is $(K - j)b_j$. We then take the maximum over all possible values of j and define the gain as the maximum value minus the miner revenue if $j = K$, the case where the miner is honest and has not inserted any fake transactions into the block. We average this gain over 1000 independent trials. In all of our simulations, we keep the number of transactions in a block, K, fixed at $2,000$.

Figure 2b shows the effect of the number of miners on a miner's gain from optimal manipulation for various values of B. As the number of miners increases, the dominant term in the block reward is the reward from transaction fees from the freshly mined block, which decreases as B increases. Thus, the gain from manipulation declines as the number of miners increases, but the decrease is most pronounced with higher values of B.



(a) This figure illustrates the effect of the number of blocks we average over (B) on the miner's gain from manipulation. The simulation shows that increasing the number of blocks we average over decreases the gain from manipulation with enough miners.

(b) This figure illustrates the effect of the number of miners on the miner's gain from manipulation. The simulation shows that increasing the number of miners decreases the gain from manipulation as long as we average over enough blocks.

Figure 2: These figures show the miner's gain from manipulation as we modulate different parameters. The mempool size, N, is fixed at 2000 for both experiments.

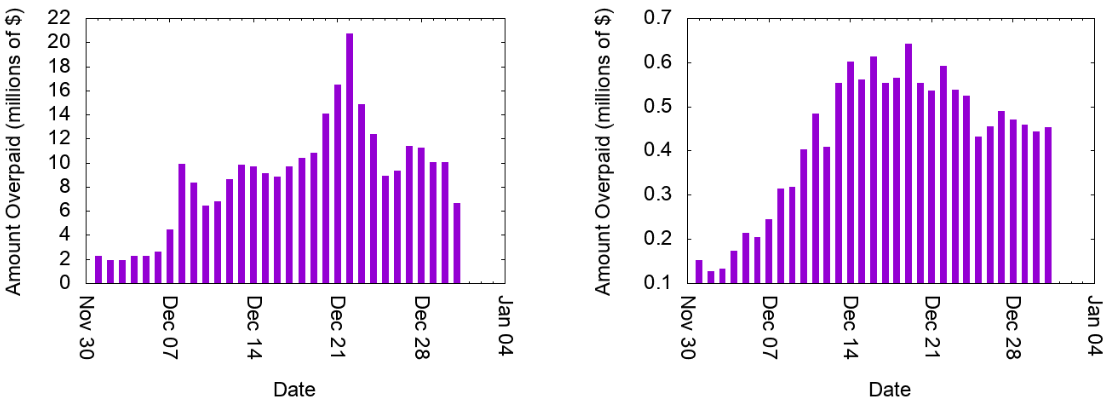
Figure 2a shows that the gain from optimal manipulation declines as the number of blocks averaged over increases and that it is uniformly lower for high numbers of miners. This is due to the fact that as B increases, the fee rewards obtained from mining a block decreases.

Our simulations show that miners gain little from optimal manipulation for reasonable numbers of miners, users and blocks averaged over. Miner revenue ranges from \$15 to \$40 over these simulations and the gain from manipulation never exceeded 3 cents when $B > 2$ and $M > 2$, so the gain from manipulation relative to total revenue is negligible. Of course, our analysis excludes the miner's incentive to make Bitcoin succeed so as to maintain the value of their Bitcoin holdings and the ongoing value of the mining operation. Including this dimension would further reduce miner manipulation incentives. We believe that taken together these results suggest that miners are not likely to manipulate StableFees.

4.2 Blockchain Bid Analysis

To provide insight into the benefits of StableFees we analyzed the actual bids that appeared on the blockchain during a period of high demand in Bitcoin and Ethereum (December 2017). We can compute the total fees paid by users in the current system, but we do not observe what bids would be if StableFees was to be adopted. The bid distribution would be different, but fortunately we do not need the entire distribution of bids to determine the revenue that would be generated by StableFees. This revenue is determined by the number of transactions in the block and minimum fee bid among these transactions. So for our revenue comparison we will assume that the same transactions would be included in the block and that the minimum accepted bid would be approximately the same in the two scenarios.

We first examine how much users could save if StableFees was used. To apply our mechanism to each day, we take all of the blocks that were mined that day and look at the transactions in each block. We then calculate the fee per byte for each transaction to normalize the fees paid, and then we have every transaction pay the smallest fee per byte that appears on the block per our protocol. We plot the difference between the actual fees paid and the fees that users would pay under StableFees. Figure 3a shows that users in this time period could have saved 273 million USD if StableFees was used in Bitcoin. Figure 3b shows a similar trend in Ethereum, but the dollar amounts are significantly smaller, which is to be expected as Ethereum has a higher processing capacity than Bitcoin. Even so, users in this time period could have saved 13.2 million USD if StableFees was used in Ethereum.



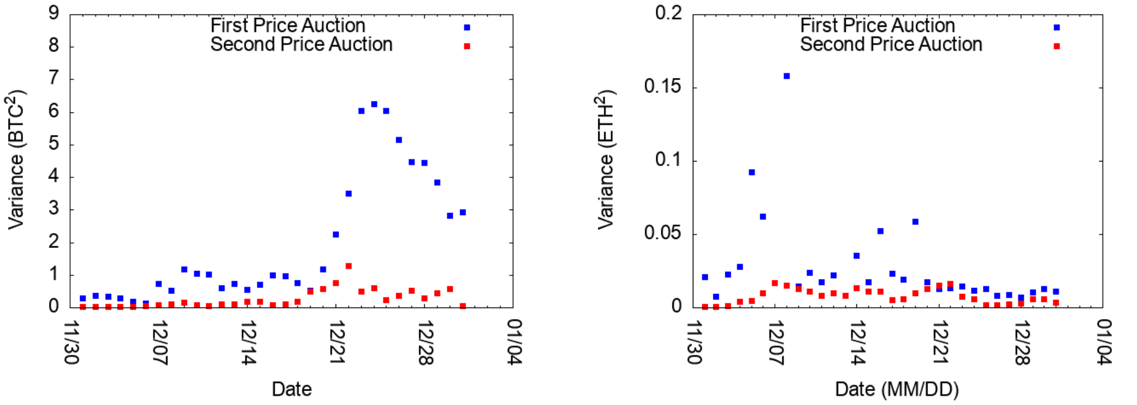
(a) Figure showing savings in Bitcoin.

(b) Figure showing savings in Ethereum.

Figure 3: These figures show how much (in millions of USD) that users could have saved if the transactions were using StableFees instead of the current scheme in December 2017. We see that the savings in Ethereum are lower due to its higher throughput.

StableFees has the potential to improve predictability by reducing the variance of miner's rewards. To quantify this, we use the same calculation as before to obtain the fees that miners would receive under StableFees. We then compute the variance of the transaction fees on each block using StableFees and the current first price mechanism. Figure 4a shows that the variance is lower in Bitcoin when using StableFees, by up to a factor of 20 on some days, with an average of 7.4.

Figure 4b shows that the same trend holds in Ethereum, with a max of 75 times and an average factor of 7.9. In StableFees, payouts are averaged over B blocks, which would further decrease the variance by an additional factor of B^2 relative to Figure 4a and Figure 4b.



(a) Figure showing variance in Bitcoin.

(b) Figure showing variance in Ethereum.

Figure 4: These figures illustrate the variance in payouts from transactions fees to each miner in December 2017. We see that the current mechanism has a significantly higher variance than StableFees, resulting in less stable payouts.

4.3 Social Welfare

We now ask how much social welfare StableFees generates compared to alternate auction schemes. For this calculation we assume that there is a fixed number of users all of whom have utility functions that are quasi-linear in money. If a user's transaction is placed on the block the user gains the value of the transaction and pays the fee specified by the mechanism. If it is not placed on the block the user gets no value from the transaction and pays no fee. Thus, the value from potential transactions which are not on the block is zero. The sum of user utilities is then the sum of values placed on the block minus the total amount paid by users. Miner payoff is the amount the miner is paid minus the cost of mining. In the analysis in this section we keep the total number of miners and thus the cost of mining fixed so the total (variable) payment to miners is the total amount paid by users. This implies that social welfare (the sum of user and miner payoffs) is the total user value of transactions placed on the block. Our notion of social welfare is thus equivalent to efficiency; welfare is maximized if and only if the highest value transactions are placed on the block and the block is filled.

To answer the social welfare question, we ran an experiment where we simulate users and miners, run the auction protocol, and analyze what happened in the trace.

In our experiment, we set number of transactions allocated by the auction mechanism to $2,000$ and we have $20,000$ users in the system[20]. Each user's value is

[20] In order to have a fair comparison for each protocol, we assume a fixed number of transactions per block. Protocols that fluctuate the block size slightly are parameterized to target the same block size to ensure a fair comparison. The block size has implications for the security of the protocol

randomly selected from the distribution specified in Section 4.1 and then users bid according to the auction mechanism. The miners follow the auction protocol rationally in order to decide which transactions get confirmed. We repeat the process until $1,000$ blocks have been mined. We report results from the last 900 blocks, as the Buterin (2018) auction mechanism is sensitive to the initial parameters for the first few blocks.

To set optimal user and miner behavior, we use the results or recommendations found in each proposed mechanism. For each scheme, we assume that behavior is determined by the results of the large numbers analysis for that scheme Although we are testing four schemes, StableFees and Buterin's mechanism Buterin (2018), which we call Eth, have two interesting parameterizations. So we consider six auctions.

StableFees and *StableFees-Opt* are both instantiations of the StableFees auction mechanism. We set B to be 10 in both instantiations. However, in StableFees, we pessimistically assume a large miner with 20% of the total hashpower whereas, for StableFees-Opt, we assume that each miner has a very small hashpower and consequently will not win any further blocks outside of the one they mined. StableFees is a pessimistic but realistic parameterization of the StableFees algorithm, while StableFees-Opt is what the best case performance of StableFees would look like. In both instantiations, the miner attempts to manipulate the auction to maximize their total revenue.

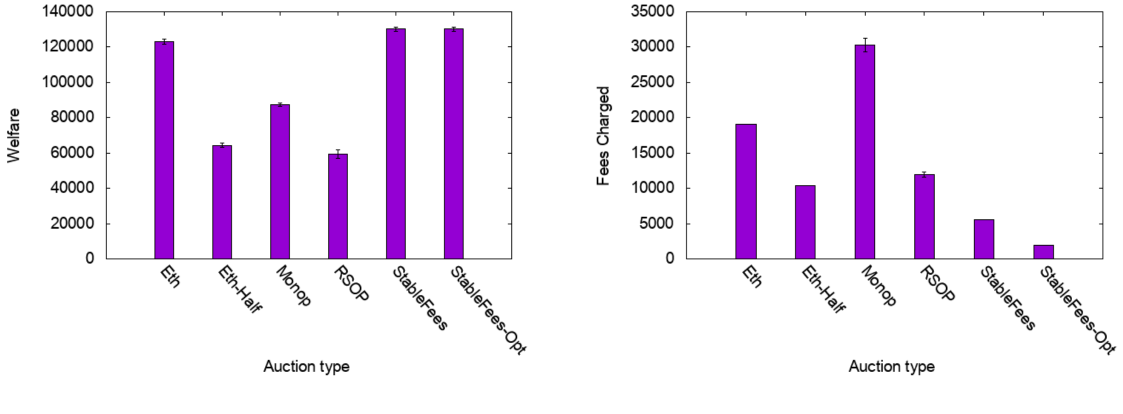
RSOP (Lavi et al. (2017)) is a randomized sampling auction mechanism. To initialize RSOP, a parameter α is set similarly to our B value. However, to prevent our choice of α from skewing the results, we assume that miners do not misbehave and simply include the $2,000$ highest pending bids in the block. Note that in RSOP, not all transactions in a block are actually confirmed since they simply exist to set the price. Users simply bid their true value since this is proven to be the optimal strategy.

In *Monop* (Lavi et al. (2017)), miners choose the number of transactions to maximize their revenue. Additionally, under the large user assumption, users bid truthfully in this auction mechanism.

The auction mechanism proposed in Buterin (2018) uses the number of transactions in each block to estimate the demand for block space and consequently does not have a fixed block size. Blocks have a minFee parameter, which is the price that each transaction included in the block is charged. All transactions in the mempool that offer to pay more than the minFee are included in the block. There is a hard cap on the block size, but the protocol sets the price aiming for a block size of half of the hard cap. To compare alternative fee setting mechanisms with Buterin's proposal, we instantiate his proposal in two ways: *Eth* and *Eth-Half.* In *Eth*, we set the maximum block size at $4,000$ transactions, which implies that the protocol will aim for a block size of $2,000$ transactions. However, some blocks will have more than $2,000$ transactions and might result in degraded security. In *Eth-Half*, we choose the most conservative block size for Buterin's mechanism, which in our simulations is $2,000$ transactions. In this parameterization, no blocks will have more than $2,000$ transactions, but the target block size is only half the available block space. Given

as larger blocks take longer to propagate through the network and should be fairly independent of the fee mechanism used.

a safe block size of $2,000$ transactions, all sensible choices for the target block size will be between $2,000$ and $4,000$ transactions, with higher choices resulting in more unsafe blocks but higher welfare.

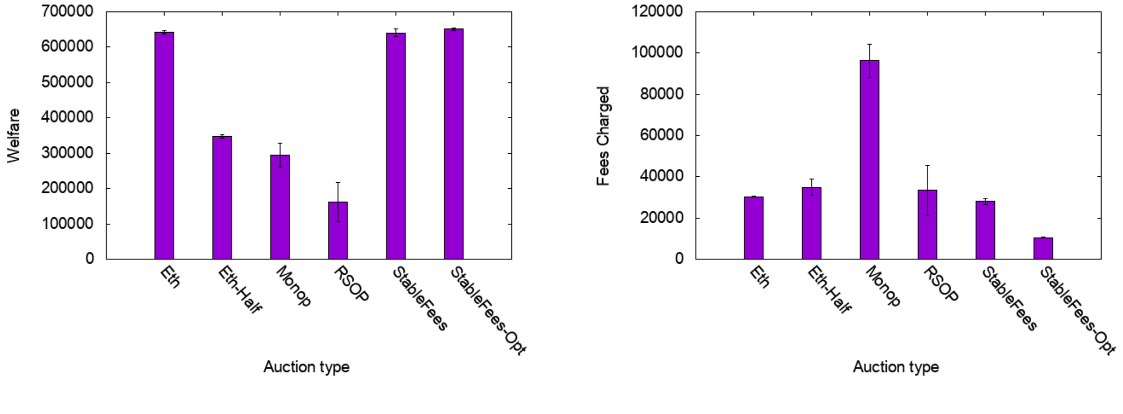


(a) Welfare.

(b) User fees paid.

Figure 5: Welfare and user fees paid under a constant demand curve

In the simulation used to generate Figure 5, the demand curve is constant and the same as the demand curve we used to study miner manipulation—values are drawn from a power law with a median of 2 cents and a mean of 10 cents. Among the most realistic parameterizations (StableFees, Eth-Half, RSOP, Monop), we see that StableFees produces 49% more welfare than the second best parameterization, Monop. Among all parameterizations, we see that both variations of StableFees and Eth produce the greatest amount of social welfare. However, StableFees charges much lower fees to the user. Most importantly, 33.6% of blocks in the Eth auction contained more than $2,000$ transactions so the security guarantees are weakened.



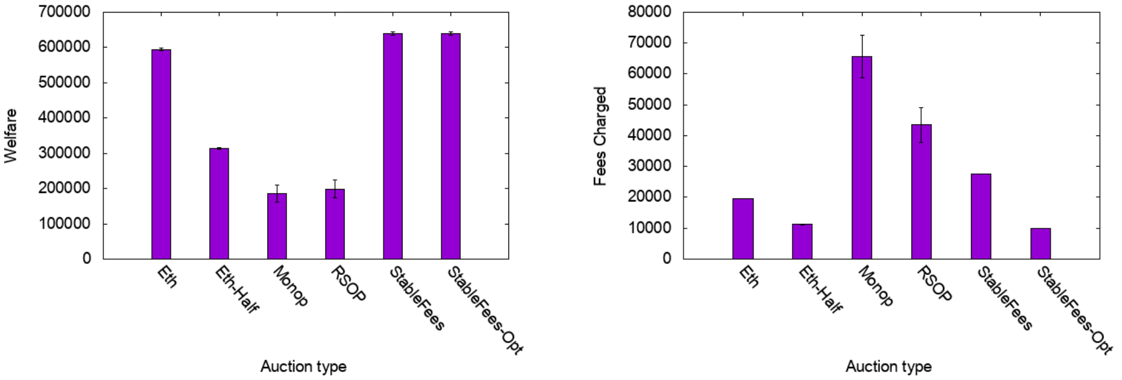
(a) Welfare.

(b) User fees paid.

Figure 6: Welfare and user fees paid under a fluctuating demand curve

In the simulation used to generate Figure 6, we take the demand curve from above and multiply the bids by a fluctuating factor from 2 to 10 to randomly increase and decrease demand throughout the experiment. Again, StableFees performs better than all other realistic parameterizations by producing 85% more welfare than

Eth-Half, which was the second best realistic parameterization. Among all parameterizations, we see that both variations of StableFees and Eth produce the greatest amount of social welfare. However, the fees charged to the user are now comparable between the three auction mechanisms due to Eth undercharging users for block space. This comes at a cost of decreased security, as 64.9% of blocks contained more than 2000 transactions in the Eth auction.



(a) Welfare.

(b) User fees paid.

Figure 7: Welfare and user fees paid under a demand spikes

In the simulation used to generate Figure 7, we either take the bid from the power law directly or multiply it by 10. We fluctuate between these two extremes throughout the experiment in order to simulate demand spikes. StableFees is again the best performing realistic parameterization with 103% more welfare produced than Eth-Half. Similarly, both variations of StableFees and Eth produced the greatest amount of social welfare at the cost of 46.1% of blocks containing more than 2000 transactions. Notice that the Monopolistic miner and RSOP produce the lowest amount of overall welfare while charging the largest amount of user fees. This is likely due to the higher chance of manipulation with demand spikes and the larger fee spreads.

Taking a step back, we see that StableFees consistently produces the largest social welfare among these mechanisms. Moreover, StableFees does comparatively better when the demand fluctuates more, with the relatively best results for StableFees coming when auctions were tested using large demand spikes. Additionally, StableFees, StableFees-Opt and Eth are the three mechanisms that produce the largest amount of welfare, though Eth does it at the cost of the security of the cryptocurrency. Finally, one interesting point to note is that StableFees is very close to StableFees-Opt, with the largest discrepancy being in Figure 6, where StableFees-Opt produces 1.6% more welfare than StableFees. This means that, even in a cryptocurrency with large miners, StableFees performs well.

4.4 Welfare with an Endogenous Number of Miners

The number of miners should adjust in response to changes in the fees they earn. In an equilibrium with free entry, expected profit to mining must be zero and this condition, along with expected revenue and costs per miner, determines the equilibrium number of miners. For our purposes here the details of that determination are not important; see Easley et al. (2019) for details. What matters is that, holding cost per miner fixed, an increase in revenue per miner will increase the equilibrium number of miners and similarly a decrease will reduce the equilibrium number of miners.

Some number of miners, say $\bar{M}$, is needed for secure posting of transactions to the blockchain. Let $BR + F \geq 0$ be the minimum per block revenue necessary to have $\bar{M}$ miners in equilibrium where BR is the exogenous block reward and F is total fees collected by the miner who writes a block. If the equilibrium number of miners is less than $\bar{M}$ then the blockchian fails and the social welfare it generates is 0. Clearly, any fee payments in excess of F are socially wasteful. They are paid by users but they are then wasted in competition by the excessive number of miners necessary to drive expected profit to zero. So social welfare is

$$\sum_{i \in Block} (V_i - f_i) + \min\{BR + F, BR + \sum_{i \in Block} f_i\}$$

if the equilibrium number of miners is at least $\bar{M}$ and 0 otherwise.

As we have shown, Stablefees reduces fees relative to current levels and relative to the fees proposed by alternative mechanisms. This will increase social welfare if the resulting equilibrium number of miners is at least $\bar{M}$; that is if the rewards induced by StableFees are at least $BR + F$. It could create an issue with security if revenue is reduced below $BR + F$. We suspect that currently this is not an issue as most (currently approximately 90 percent) of BitCoin miner revenue comes from the block reward, BR, rather than from fees.[21] As Bitcoin block rewards decline over time, having miner rewards remain sufficient for security could become an issue—and then a minimum fee (as is allowed in StableFees) may be needed. A force in the other direction (arguing for fewer miners than we have now) is that mining uses a large amount of electricity in what is clearly a socially wasteful competition. See Benetton et al. (2021) for more discussion of these environmental issues. One advantage of StableFees is that miner revenue is generally lower and so our protocol is environmentally friendly.

5 Conclusion

Cryptocurrencies cannot go mainstream if constructing a transaction imposes a cognitive load or requires complex strategic bidding behavior. We show that the fee mechanism currently used in a variety of coins encourages users to employ complex bidding strategies. We then present StableFees, an alternative that obviates this need and offers a more stable, predictable fee market.

Both the generalized first price auction and StableFees work well in equilibrium if there is a large number of users relative to the capacity of blocks. However, in StableFees, the transaction fee offered by a user only affects what a successful user

[21] As of January 25, 2021 the 30-day averages were \$3.62 for fees and \$35.85 for total revenue according to Blockchain.com

pays if the user has the, potentially unique, Kth highest bid. Otherwise the fee only affects whether the user is in the block or not in it. So the gain to strategic bidding is small if there are many users. In a first price auction, every user pays their bid if his transaction is in the block. Here, strategic bidding is inescapable although its gain does converge to zero as the number of users grows. Also we see in our simulations that miner revenue will have lower variance under StableFees. But what happens to the actual payout with real users and miners is unclear, at least in part because of the non-robustness of the first price procedure.

Our argument is that StableFees offers an improvement over the current Bitcoin protocol; not that it is an optimal mechanism. If StableFees is adopted then with a large number of users and miners there are simple non-manipulative strategies that are nearly optimal. In our model, if participants follow these strategies, then StableFees provides a social welfare maximizing allocation. We view this as a step in the direction of more broadly optimal mechanisms which could take into account the impact of the fee setting mechanism on aspects of the environment that we hold constant, particularly security issues and the number of users.

Finally, we note that our analysis applies to proof-of-work protocols such as those used in Bitcoin, Ethereum and many others. Alternative protocols are being considered and used in a variety of different digital currencies. Most notably, Ethereum is considering a switch to proof of stake. Regardless of the protocol, cryptocurrencies will need to prioritize transactions somehow. Most cryptocurrencies charge fees to use the network and induce a first price auction. Thus, they face the same problems described above.

References

Mohammad Akbarpour and Shengwu Li. 2018. Credible Mechanisms. In *Proceedings of the 2018 ACM Conference on Economics and Computation (EC '18)*. ACM, New York, NY, USA, 371–371. `https://doi.org/10.1145/3219166.3219169`

Rune Tevasvold Aune, Adam Krellenstein, Maureen O'Hara, and Ouziel Slama. 2017. Footprints on a Blockchain: Trading and Information Leakage in Distributed Ledgers. *The Journal of Trading* 12, 3 (2017), 5–13. `https://doi.org/10.3905/jot.2017.12.3.005`

E. M. Azevedo, D. M. Pennock, B. Waggoner, and E. G. Weyl. 2020. Channel Auctions. *Management Science* 66, 5 (2020), 2071–2082.

M. Benetton, G. Compiani, and A. Morse. 2021. When Cryptomining Comes to Town: High Energy-Use Spillovers to the Local Economy. *SSRN* (2021). `https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3779720`

BitInfoCharts. 2021. Bitcoin Avg. Transaction Fee historical chart. `https://bitinfocharts.com/comparison/bitcoin-transactionfees.html`. (2021). Accessed: 2021-05-11.

Rainer Böhme, Nicolas Christin, Benjamin Edelman, and Tyler Moore. 2015. Bitcoin: Economics, Technology, and Governance. *Journal of Economic Perspectives* 29, 2 (May 2015), 213–238. `https://doi.org/10.1257/jep.29.2.213`

Vitalik Buterin. 2018. Blockchain Resource Pricing. `https://ethresear.ch/uploads/default/original/2X/1/197884012ada193318b67c4b777441e4a1830f49.pdf`. (2018). Accessed: 2019-02-10.

Miles Carlsten, Harry Kalodner, S Matthew Weinberg, and Arvind Narayanan. 2016. On the instability of bitcoin without the block reward. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 154–167.

Long Chen, Lin William Cong, and Yizhou Xiao. 2021. A Brief Introduction to Blockchain Economics. In *Information for Efficient Decision Making: Big Data, Blockchain and Relevance*. World Scientific Publishing Company, 1–40.

Lin William Cong, Zhiguo He, and Jiasun Li. 2018. Decentralized mining in centralized pools. *SSRN* (2018). `https://ssrn.com/abstract=3143724`

Kyle Croman, Christian Decker, Ittay Eyal, Adem Efe Gencer, Ari Juels, Ahmed Kosba, Andrew Miller, Prateek Saxena, Elaine Shi, Emin Gün Sirer, et al. 2016. On scaling decentralized blockchains. In *International conference on financial cryptography and data security*. Springer, 106–125.

D. Easley and J. Kleinberg. 2010. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World.* Cambridge University Press, New York, NY, USA.

David Easley, Maureen O'Hara, and Soumya Basu. 2019. From Mining to Markets: The Evolution of Bitcoin Transaction Fees. *Journal of Financial Economics* 134, 1 (2019), 91–109.

Benjamin Edelman and Michael Ostrovsky. 2007. Strategic bidder behavior in sponsored search auctions. *Decision support systems* 43, 1 (2007), 192–198.

Ittay Eyal and Emin Gün Sirer. 2014. Majority Is Not Enough: Bitcoin Mining Is Vulnerable. In *Proceedings of the 18th International Conference of Financial Cryptography and Data Security (FC '18)*. Springer, 436 – 454.

Neil Gandal and Hanna Halaburda. 2016. Can We Predict the Winner in a Market with Network Effects? Competition in Cryptocurrency Market. *Games* 7, 3 (July 2016), 1–21. https://ideas.repec.org/a/gam/jgames/v7y2016i3p16-d73475.html

Joshua S Gans and Hanna Halaburda. 2015. Some economics of private digital currency. In *Economic Analysis of the Digital Economy*. University of Chicago Press, 257–276.

Adem Efe Gencer, Soumya Basu, Ittay Eyal, Robbert van Renesse, and Emin Gün Sirer. 2018. Decentralization in Bitcoin and Ethereum Networks.. In *Proc. of the Financial Cryptography and Data Security Conference*.

Paolo Guasoni, Gur Huberman, and Clara Shikelman. 2021. Lightning Network Economics: Channels. *SSRN* (2021). https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3840374

Campbell Harvey. 2016. Cryptofinance. *SSRN* (2016). http://ssrn.com/abstract=2438299

Nicolas Houy. 2014. The Bitcoin mining game. *SSRN* (2014). https://ssrn.com/abstract=2407834

Gur Huberman, Jacob D Leshno, and Ciamac C Moallemi. 2017. Monopoly without a monopolist: An economic analysis of the bitcoin payment system. *Columbia Business School, No. 17-92* (2017).

Matthew O. Jackson and Ilan Kremer. 2006. The Relevance of a Choice of Auction Format in a Competitive Environment. *The Review of Economic Studies* 73, 4 (10 2006), 961–981. https://doi.org/10.1111/j.1467-937X.2006.00404.x

Ron Lavi, Or Sattath, and Aviv Zohar. 2017. Redesigning Bitcoin's fee market. *arXiv preprint arXiv:1709.08881* (2017).

Alfred Lehar and Christine Parlour. 2020. Miner Collusion and the BitCoin Protocol. *Working paper, University of Calgary* (2020).

Eric Lombrozo, Johnson Lau, and Pieter Wuille. 2017. Segregated Witness. BIP 141, https://github.com/bitcoin/bips/blob/master/bip-0141.mediawiki, retrieved Jun. 2017. (2017).

Katya Malinova and Andreas Park. 2017. Market Design with Blockchain Technology. *SSRN* (2017). https://ssrn.com/abstract=2785626

Paul R Milgrom. 1985. The economics of competitive bidding: a selective survey. *Social goals and social organization: Essays in memory of Elisha Pazner* (1985), 261–292.

Roger B. Myerson. 1981. Optimal Auction Design. *Mathematics of Operations Research* 6, 1 (1981), 58–73. http://www.jstor.org/stable/3689266

Max Raskin and David Yermack. 2018. *Digital currencies, decentralized ledgers and the future of central banking.* Technical Report. Edward Elgar Publishing.

Meni Rosenfeld. 2011. Analysis of bitcoin pooled mining reward systems. *arXiv preprint arXiv:1112.4980* (2011).

Tim Roughgarden. 2020. Transaction Fee Mechanism Design for the Ethereum Blockchain: An Economic Analysis of EIP-1559. *Working paper, Columbia University* (2020).

Jeroen M. Swinkels. 2001. Efficiency of Large Private Value Auctions. *Econometrica* 69, 1 (2001), 37–68. http://www.jstor.org/stable/2692185

Mojtaba Tefagh. 2021. Path-dependence of EIP-1559 and the simulation of the resulting permanent loss. https://ethresear.ch/t/path-dependence-of-eip-1559-and-the-simulation-of-the-resulting-permanent-loss/8964. (2021). Accessed: 2021-05-11.

Hal R. Varian and Christopher Harris. 2014. The VCG Auction in Theory and Practice. *American Economic Review* 104, 5 (May 2014), 442–45. https://doi.org/10.1257/aer.104.5.442

Robert J. Weber. 1983. Multi-Object Auctions. In *Auctions, Bidding, and Contracting: Uses and Theory*, Richard Engelbrecht-Wiggans, Martin Shubik, and Robert M. Stark (Eds.). New York University Press, New York City, 165–191.

Andrew Chi-Chih Yao. 2018. An Incentive Analysis of Some Bitcoin Fee Designs. *arXiv:1811.02351* (2018).

David Yermack. 2017. Corporate Governance and Blockchains. *Review of Finance* 21, 1 (01 2017), 7–31. https://doi.org/10.1093/rof/rfw074

A Formal Observations about the Current Mechanism

Users can attach any fee they like, or no fee, to their transaction. If the miner of the current block places a user's transaction in the block, then the miner keeps the fee. A profit maximizing miner clearly selects the K highest bids, or all bids if there are less than K bids, places those transactions on the block, and earns those bids. The miner has no incentive to manipulate by entering fictitious transactions as by doing so, he simply removes real transactions and the fees they generate without changing the fees paid by other transactions. So our model of the current protocol is equivalent to a generalized first price auction run by an auctioneer who can commit to running this auction type.

Remark: Our model of bidding for slots in current cryptocurrency protocols induces a generalized first price auction for the K slots on the current block. The users with the K highest bids (proposed fees) win slots on the block and each winning user pays his bid (fee).

Thus, from the users' point of view the fee setting game is a generalized first price auction for K identical items. This auction has a symmetric Bayes-Nash equilibrium in which equilibrium bids are increasing in values and so, in this equilibrium, the highest value transactions are placed on the block.[22] If all participants have quasi-linear utility then social surplus is the sum of the values of users whose transactions are placed in the block, and so it is maximized. Miner revenue and the total payment by users net out and so do not affect social surplus. Of course, this is an interim notion of social surplus as it takes the winning miner as fixed and does not consider any externalities produced by the mining industry. Proofs are included in Appendix C.

Proposition 4: Our model of current cryptocurrency protocols applied to a single block of size K induces a game which has a symmetric Bayes-Nash equilibrium which is efficient.

If miners could commit to an auction form and the protocol could use all bids to determine the assignment and payments, then it could be modified to induce a generalized second price auction for the K slots. In this auction, the K highest bidders would have their transactions placed on the block and they would pay a uniform price equal to the $K + 1$st highest bid. In this auction it is a (weakly) dominant strategy for each bidder to bid his true value. To see this, note that each bidders' bid only affects whether or not his transaction is placed on the block; it does not affect the price he will pay if he gets on the block, as that price is the bid of a bidder who is not successful. A bidder wants to be on the block if the price is no more than his value and he does not want to be on the block otherwise. A bid

[22] Symmetry of users matters for this claim. With asymmetric distributions, equilibria are not symmetric, and efficiency need not occur. It also depends on our assumption that each user has only one transaction in the mempool. If users are interested in multiple slots on the block, then efficiency can also be lost.

equal to his true value ensures this. Most importantly, note that this reasoning does not depend on how many other bidders are present or on what they do; so bidding truthfully is a dominant strategy. Bidders using their dominant strategies result in maximum realized social surplus.

Proposition 5: If miners could commit to use a generalized second price auction and the protocol could use all bids to determine the outcome, then the protocol could be modified to induce a generalized second price auction. In this auction, truth-telling would be a (weakly) dominant strategy for users and the assignment would be efficient.

Because miners cannot commit to an auction form and the $K + 1$st bid is not observable, this generalized second price auction is not feasible. Most importantly, the miner's ability to submit a fake transaction after observing the fees in the mempool destroys the "truth-telling is a dominant strategy" result.[23] For a block with K slots the optimal fictitious transaction(s) by the miner are not as simple as just matching the Kth highest bid; the equivalent of matching the highest bid in one-unit "second price auction" thereby turning it into a first price auction. For example, suppose that $K = 2$ and there are three transactions in the mempool with attached fees of $f_1 > f_2 > f_3$. Then there are two possible manipulations by the miner: (1) Insert a transaction with fee equal to f_2 and earn total fee of $2f_2$, or (2) Insert a transaction with fee equal to f_1 and earn total fee of f_1. Manipulation (1) is better than the generalized second price auction (for the miner) and (2) is better than (1) if $f_1/2 > f_2$. Thus, the miner always has an incentive to manipulate unless there are $K + 1$ bids of equal highest value.

B Deployment of StableFees

In this section, we discuss how to deploy StableFees and issues that may arise when picking parameters for each cryptocurrency. Additionally, we show how to deploy a version of StableFees as a soft fork that confers most of the benefits of StableFees without requiring everyone to upgrade.

B.1 Parameterizing StableFees

Deploying StableFees requires us to set multiple parameters: the fill level, the minimum fee and the number of blocks we average the rewards over (B). We briefly discuss some issues around how to set these parameters for Bitcoin as a case study. The fill level should be set to something very high, such as 80 or 90 percent, as miner transactions should not take up a large amount of block space. Our work assumes that all transactions are equally sized, so the fill level refers to the resource being used to determine if a block is full and bids are per unit resource. In Bitcoin today, this is the block weight rather than the block size Lombrozo et al. (2017). Similarly,

[23] If the miner could only submit fictitious transactions before observing the fees in the mempool, then truth-telling would remain a dominant strategy for users. From the point of view of users the miner is just acting as another user or users and this has no effect on any user's incentive to bid truthfully.

Ethereum uses gas rather than block size. To set the minimum fee, one can use the default minimum relay fee or any other similarly small fee. As Bitcoin often operates at full capacity, the minimum fee does not matter much. For cryptocurrencies that are not operating at their full capacity, the minimum fee should be set to the amount of resources consumed by a transaction. Finally, to determine B, we note that for Bitcoin, the largest mining pool has about 21% of the hashpower Gencer et al. (2018) implying that $M = 5$ for the largest miner. Thus, from Figure 2b, we see that setting $B = 10$ allows the miner very little gain from manipulation and setting B higher does not make StableFees significantly more robust.

B.2 Soft Fork Deployment

Our protocol allows for incremental deployment without changing the core cryptocurrency code through a mechanism called a *soft fork*. A soft fork allows miners to enforce additional rules on the existing blocks in a cryptocurrency. The chief benefit of a soft fork is that users who have not upgraded their code can participate in the amended protocol and transparently benefit from the new rules being introduced to the blockchain. The primary drawback of a soft fork is that the existing transaction format and payoff structure must be used. This implies that not every change can be accommodated with a soft fork and that a supermajority of miners must enforce the new rules for them to go into and remain in effect. Such a supermajority may exist even though StableFees may lower miner fees since the loss of the fee revenue can be compensated by improved adoption and a corresponding rise in price. Additionally, in many cryptocurrencies today, the miner revenue is dominated by a fixed block reward rather than the revenue generated from fees. We describe how StableFees can be implemented as a soft fork on top of a legacy coin based on first price auctions.

The soft fork rule that we impose is based on the observation that, given a set of bids, a first price auction will cause each user to overpay compared to StableFees and will select the same transactions. Those overpayments accrue to the miner who creates the block. Consequently, StableFees can be encoded on top of a first price substrate by simply adding an additional rule which requires the miner to refund the excess fees. Specifically, in our soft fork, every block contains a *refund transaction*, where the miner pays the users the difference between their payments under StableFees and their payments under a first price auction. Each transaction participating in the StableFees mechanism can simply include an additional address field that denotes who to pay any excess fees to. The miner can then pay out any excess fees for this transaction to the specified address, and any block not including such a refund transaction can be treated as invalid. While the above encoding has the desired properties of a soft fork, there are two main limitations to this approach. The first limitation is that each miner does not get paid for a block immediately and their payouts are held until more blocks are mined. In Bitcoin, this holding period is 100 blocks. To overcome this limitation, miners must use their own funds to correct for overpayments in the short term, to be repaid in 100 blocks. In practice, such a repayment mechanism, within a day, should not constitute a major barrier for miners because the capital required for mining dwarfs the capital required for the fee advance. Additionally, some overpayments may be very small, to the point where spending that amount will actually cost more money in fees than the amount itself. Such amounts are called *dust*, and overpayments in that range cannot be paid out. Since dust payments are very small, not paying users such small amounts will still confer most of StableFees' benefits.

Of course, any cryptocurrency can always choose to do a hard fork and explicitly alter the payoff structure from each transaction to only pay the miner after applying StableFees. This will require all stakeholders (users, miners, exchanges, etc) to update their code, otherwise they will be left behind. However, a hard fork implementation will be able to fully leverage the benefits to our mechanism and is a lot more elegant than a soft fork design.

C Proofs

Proof of Proposition 1: Consider user i with value V_i and suppose that all other users bid truthfully. Let V_{K-1} and V_K be the $K-1$st highest bid of others and the Kth highest bid of others. If $V_i > V_{K-1}$ and $V_{K-1} > V_K$ then a bid by i of b_i such that $V_{K-1} > b_i > V_K$ gives i a slot on the block at price b_i while a truthful bid gives i a slot on the block at price $V_{K-1} > b_i$. So truthful bidding is not a dominant strategy.

If a user's value is below V_K there is no possible gain from bidding strategically as the price will be greater than the user's value for any bid. If $V_i \geq V_K$ then user i would gain from a slot on the block. The price of that slot will be V_{K-1} if $b_i \geq V_{K-1}$, as V_{K-1} will be the lowest successful bid, and it will be b_i if $V_{K-1} > b_i > V_K$, as in this case b_i will be the lowest successful bid. So the gain that user i can earn from strategic bidding (a non-truthful bid) is bounded by $(V_{K-1} - V_K)$, and this maximal gain can be earned only if $V_{K-1} \geq V_i \geq V_K$. Alternatively, in the event $V_i \notin [V_K, V_{K-1}]$ user i's maximal gain is 0. Thus, user i's expected gain from strategic bidding is bounded by $E[(V_{K-1} - V_K)|V_i] = E[(V_{K-1} - V_K)]$ which converges to 0 in the number of users.

Proof of Proposition 2: In the text preceding Proposition 2.

Proof of Proposition 3: Follows immediately from an application of Swinkels Swinkels (2001) and Jackson and Kremer Jackson and Kremer (2006), proof of Theorem 1.

Proof of Proposition 4: By the Remark in the text this proposition is equivalent to showing that a discriminatory, multi-unit, first price auction has a symmetric Bayes-Nash equilibrium in which bids are increasing in values. This is a standard result, see Weber Weber (1983) and Milgrom Milgrom (1985).

Proof of Proposition 5: The modification yields a multi-unit, second price auction. That this auction has weakly dominant strategies and an efficient equilibrium is a standard result, see Weber Weber (1983) and Milgrom Milgrom (1985).

We provide a simple, direct proof of this claim as we use the logic elsewhere in the paper. Consider bidder i with value V_i. We need to show that bidding more than V_i or less than V_i cannot increase the profit of bidder i.

Consider a bid $b_i > V_i$. Bidder i's bid only affects whether he wins or loses the auction; it does not affect the price he pays conditional on winning. So this high bid only changes the payoff to i if bidder i would not win with a bid of V_i and would win with a bid of b_i. That is, only if $b_i > V^K > V_i$, where V^K is the Kth lowest bid of the other bidders. In this case i wins with a bid of b_i, but pays $V^K > V_i$ as V^K is now the K+1 st highest bid. So high bidding reduces i's payoff.

Alternatively suppose that i bids $b_i < V_i$. This only affects i's payoff if i would have won with a bid of V_i and does not win with a bid of b_i. That is, only if $V_i > V^K > b_i$. In this case i would have won with a bid of V_i and paid $V^K < V_i$ and does not win with a bid of b_i. So a low bid also reduces i's payoff.