# BitcoinF: Achieving Fairness for Bitcoin in Transaction-Fee-Only Model

by

Shoeb Siddiqui, Ganesh Vanahalli, Sujit Prakash Gujar

in

*International Conference on Autonomous Agents and Multi-agent Systems, AAMAS*

Report No: IIIT/TR/2020/-1

Centre for Visual Information Technology
International Institute of Information Technology
Hyderabad - 500 032, INDIA
March 2020

# BitcoinF: Achieving Fairness for Bitcoin in Transaction-Fee-Only Model

Shoeb Siddiqui, Ganesh Vanahalli, Sujit Gujar
*International Institute of Information Technology*
Hyderabad, India
shoeb.siddiqui@research.iiit.ac.in, ganesh.vanahalli@students.iiit.ac.in, sujit.gujar@iiit.ac.in

***Abstract*—A blockchain, such as Bitcoin, is an append-only, secure, transparent, distributed ledger. A fair blockchain is expected to have healthy metrics; high honest mining power, low *processing latency*, i.e., low wait times for transactions and stable *price of consumption*, i.e., the minimum transaction fee required to have a transaction processed. As Bitcoin matures, the influx of transactions increases and the block rewards become insignificant. We show that under these conditions, it becomes hard to maintain the *health of the blockchain*. In Bitcoin, under these *mature operating conditions* (MOC), the miners would find it challenging to cover their mining costs as there would be no more revenue from merely mining a block. It may cause miners not to continue mining, threatening the blockchain's security. Further, as we show in this paper using simulations, the cost of acting in favor of the health of the blockchain, under MOC, is very high in Bitcoin, causing all miners to process transactions greedily. It leads to *stranded transactions*, i.e., transactions offering low transaction fees, experiencing unreasonably high processing latency. To make matters worse, a compounding effect of these stranded transactions is the rising price of consumption. Such phenomena not only induce unfairness as experienced by the miners and the users but also deteriorate the health of the blockchain.**

**We propose BitcoinF transaction processing protocol, a simple, yet highly effective modification to the existing Bitcoin protocol to fix these issues of unfairness. BitcoinF resolves these issues of unfairness while preserving the ability of the users to express urgency and have their transactions prioritized.**

## I. Introduction

Blockchain, introduced in Bitcoin [1] by Nakamoto, is an append-only, secure, transparent, distributed ledger, storing data in blocks connected through immutable cryptographic links, with each block extending exactly one previous block. In blockchain technology, the *miners* validate transactions that, the *users publish* (create and broadcast). Miners add valid transactions into the next block(s). Different miners attempt to publish (create and broadcast) the next block. In *Proof of Work* (PoW) blockchains, such as Bitcoin, the miner who solves a cryptographic puzzle first is whose published block is accepted as the extension. Each miner has a different puzzle, yet of the same level of difficulty, which needs computations to solve.

In Bitcoin, there are two types of rewards offered to the miners: *block rewards* and *transaction fees*. Block rewards are incentives that the miners are allowed to pay to themselves, minting currency in every block mined, regardless of the contents of the block. Transaction fees, on the other hand, are incentives offered by the users to the miners to prioritize their transactions. In Bitcoin and similar PoW blockchains, the miners invest resources, such as electricity and hardware, in such computations in anticipation of these rewards.

It is due to the investment on the part of the miners that they benefit from the *health of the blockchain*. In the context of this paper, we characterize the *health of a blockchain* by (i) the fraction of mining power held by honest nodes; higher, the better (ii) *processing latency*: one of the performance parameters of the blockchain; lower the better, and (iii) the *price of consumption*; lower the variance, the better. All of these three metrics are linked to the perceived value of the blockchain and its currency. If the health of the blockchain is good, it is prudent to say that the underlying crypto-currency possesses a good value.

One of the key differences between traditional currency and Bitcoin is inflation control. To control inflation, block rewards are halved every four years in Bitcoin. Over time, a scenario develops, in which block rewards are negligible, and the only incentive for the miners is the transaction fee, i.e., the *transaction-fee-only model* (TFOM).

When the block rewards are high in value, we say the Bitcoin protocol satisfies *individual fairness for the miners* as it is believed that the current block rewards at least cover the marginal costs of mining blocks. Since the miners are not hard-pressed for revenue, they can include transactions for free in the order that they arrive. This not only ensures that no transaction is stranded but also ensures that the price of consumption does not rise. A blockchain ecosystem is *fair for the users* if it has (i) low processing latency and (ii) stable price of consumption. Currently, the Bitcoin ecosystem is fair to the miners and users. The vital question we study is, do these three notions of fairness carry forward to TFOM?

The authors in [2] showed that in TFOM under low *influx* (incoming volume of transactions), the rational miners will *undercut* instead of following default strategy. While this analysis considers the impact of rational miners in TFOM w.r.t. *forking*, it does not consider the processing latency and the price of consumption.

In this paper, our goal is to quantify fairness to the miners and the users and study the impact of TFOM under *standard influx*. Standard influx refers to the case when influx on an average is equal to the maximum outflux (processing capacity) of the blockchain. The two conditions, TFOM and standard influx, inherently go hand-in-hand as the Bitcoin matures [3]. Thus, making it very important to study and contemplate such

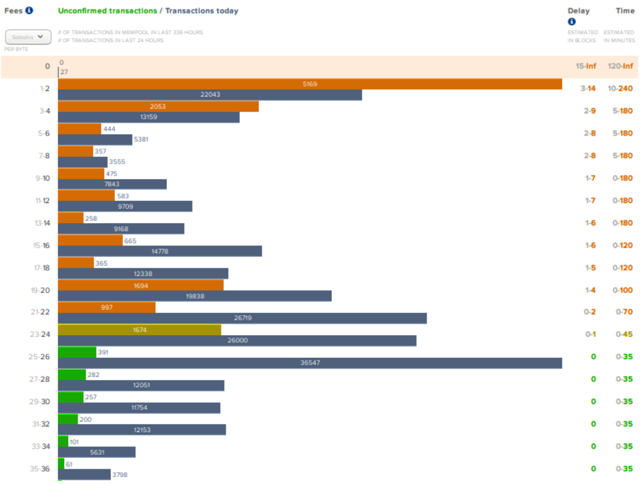


Fig. 1: Transaction Fees vs Processing Latency [4]

scenarios. In this paper, we analyze Bitcoin in TFOM and under standard influx, which we term as *mature operating conditions* (MOC), and show that it is unfair for both miners and users. This unfairness, in a nutshell, is exemplified in Fig. 1. We observe that those paying lesser transaction fees are expected to wait upto 9 blocks, and those who pay an insignificant amount of fees are expected to experience a processing latency of 14 blocks.

In TFOM, even when transaction volumes are sufficiently high enough to fill the processing capacity of the blockchain, it is not assured that the miners earn sufficient revenue. Insufficient transaction fees can be a major issue, as this puts the blockchain at a security risk due to the possibility of reduced honest mining power, which in turn will deteriorate the health of the blockchain, reducing its value, and hence further dropping of the honest miners.

Miners following *First-In-First-Out* (FIFO) processing ensures low and reasonable processing latency, avoiding stranding transactions entirely. Since no transactions are stranded, the price of consumption does not increase. Hence, miners following FIFO help sustain the blockchain's health through good performance and maintaining a stable price of consumption. In Bitcoin, under MOC, we show that miners following FIFO processing take heavy losses as compared to the ones mining greedily. This is an issue, as miners depend entirely on transaction fees to sustain mining and cannot take considerable losses in order to follow FIFO processing. This results in all miners processing transactions greedily. As we show in our analysis, this causes transactions to experience unreasonably high processing latency; such transactions are referred to as *stranded transactions*. This leaves the users with uncertainty about whether or not their transactions will be processed. A compounding effect of stranded transactions is the rising price of consumption. These issues culminate in unfairness for both the miners and the users. Thus, we say that Bitcoin, in its current form, is unfair under MOC (Proposition 1).

We solve these issues of unfairness by proposing a novel protocol, BitcoinF, for processing transactions. BitcoinF enforces a minimum transaction fee and uses two queues, instead of one to process transactions. BitcoinF allows the users to express urgency and have their transactions prioritized, just as they can do in Bitcoin. Our game-theoretic analysis proves that the proposed modification to Bitcoin, BitcoinF, ensures good health of the blockchain. Thus, we believe BitcoinF will lead to a stable ecosystem and hence will be fair to the miners and the users (Proposition 2). While there have been many published works analyzing TFOM, using collected data or using game-theoretic models, to the best of our knowledge, this is the first formal attempt at solving the pressing issues that are bound to arise in TFOM.

## II. Related work

There are many papers in the literature studying transaction fees offered in Bitcoin. The authors in [5] study the behavior of transaction fees over a period of time, whereas Li et al. [6] conduct a theoretical analysis of a queuing game to study the transaction fees, and both conclude that the users paying lesser fee faced higher processing latency. The authors in [7], point out that if transaction fees were to be determined by the free market alone without a block size limit, it would be detrimental to Bitcoin as the fees would eventually become zero and miners will no longer have an incentive to mine. Easley et al. [8] develop a game-theoretic model, based on observational data, to study transaction fees and explain the behavior of miners and users in equilibrium. The authors in [9] conclude briefly that transaction fees would not play any major role unless the underlying rules of Bitcoin are changed. However, [10] suggests otherwise, clearly stating the importance of transaction fees and suggest increasing block rate to check congestion and increase miners' revenue. The authors in [11] study the effects of transactions paying a small fee and block size limit on the transaction confirmation time using queuing theory. The authors in [12] study how the users tend to offer high fees for their transactions to get included in the block when the demand exceeds block capacity, they model the confirmation time as a particular stochastic fluid queuing process.

## III. Preliminaries

In this paper, we focus on Bitcoin when operating under *mature operating conditions* (MOC), i.e., when blocks rewards are negligible (Transaction-fee-only model (TFOM)), and there is standard demand (influx on an average is equal maximum outflux). First, we define all the important terms. Then, we present our model and describe our assumptions. Next, we explain how we simulate miners' behavior and users' behavior.

### A. Important Definition

*Definition 1 (Honest miner):* We say a miner is *honest* or *non-adversarial* if it does not willingly attempting to disrupt the Bitcoin ecosystem by adding *invalid transactions* to blocks, attempting to *double spend* or by extending other than the *longest chain*.

*Definition 2 (Rational Miner):* We say a miner is *rational* if it; continues mining when *individual fairness for the miners* is guaranteed, acts in favor of the health of the blockchain if the cost of doing so is marginal.

The act of *processing transactions* simply involves selecting the transactions from the set of received but yet unprocessed transactions, adding them to the block and then publishing the block. This is also known as mining, and it is performed by miners. The system requires a *honest majority* of miners to maintain the security of the blockchain, vis-a-vis *persistence*, against adversarial miners. Persistence is a property that must be ensured by blockchains; It ensures that the *confirmation* (different from processing transactions) of a transaction by an honest node is never disputed by any other honest node. In this paper, we consider that all miners are honest but *rational*. Miners are incentivized to participate in honest mining by rewards. If the miners are not compensated appropriately for their mining efforts, the rational, though honest miners may choose not to mine. Thus reducing the honest power in the network, weakening the blockchain against adversarial attacks, adversely affecting the health of the blockchain. When a miner chooses to stop mining, they are essentially giving up on the value of the blockchain and hence giving up on the significantly high investment they have in it, either in the form of the mining equipment or in the form of the blockchain currency token they hold.

Besides persistence, another property that must be ensured by blockchains is *liveness*. Liveness ensures that a transaction will eventually be processed; however, it is not sufficient as it does not guarantee that transactions will not get *stranded* for a long time, let alone be processed in a reasonable amount of time. As the blockchain technology matures, to maintain competitive performance, a blockchain must ensure that transactions are processed within a reasonable amount of time, i.e., low *processing latency*. Furthermore, another reason to avoid *stranded transactions* is that it leads to increasing *price of consumption*, further deteriorating the *health of the blockchain*. While stranded transactions can be avoided by simply restricting the amount of transaction fee offered to a single value, this trivial solution is unacceptable as the users' ability to offer a range of fees is required to express urgency and importance in a setting where there is varying demand.

*Definition 3 (Processing Latency): Processing latency* refers to the duration, which we measure in terms of blocks, between a user publishing a transaction and a miner processing it (publishing a block containing it).

*Definition 4 (Stranded Transactions): Stranded transactions* are those transactions, that experience unreasonably high processing latency ( $> 100$ blocks).

*Definition 5 (Price of Consumption):* The price of consumption is the minimum transaction fee, as perceived by the users, that must be paid for the transaction to be processed.

*Definition 6 (Health of a Blockchain): Health of a blockchain* is characterized by (i) the fraction of mining power held by honest nodes (ii) *processing latency* and (iii) the *price of consumption*.

For the security of the blockchain, mainly Bitcoin, more than 50% of miners should be honest; higher, the better. This is an essential aspect for the system to be fair to both the honest miners and the users, as the decentralized nature of the system depends on it. Processing latency, one of the performance parameters of the blockchain, is the time taken to process a transaction. It is crucial to the blockchain's usability and adopt-ability. It also ensures that transactions are not *stranded* in the system and are added to the blockchain within a reasonable time. The price of consumption is the minimum transaction fee that must be paid for the transaction to be processed; for stability; lower the variance, the better.

The health of the blockchain would be better if more miners are honest. Given the miner's stake in the ecosystem they participate in, it is fair to say that they would act in favor of the health of the blockchain as long as the cost of doing so is marginal.

*Definition 7 (Individual Fairness for The Miners):* We say the given blockchain protocol satisfies *individual fairness for the miners* if the rewards from a block are at-least the cost of mining it.

*Definition 8 (Fairness of The Users):* We say the given blockchain protocol satisfies *fairness for the users* if the users experience
(i) reasonable processing latency,
(ii) stable price of consumption (i.e., $f_{min}$), and
(iii) decreasing average processing latency with increasing $\eta$.

*Definition 9 (Fair Blockchain):* We say that the blockchain ecosystem is fair if it satisfies individual fairness for the miners and fairness for the users.

Since acting in the blockchain's favor yields optimal results required for the blockchain to perform competitively, it is imperative that miners do not deviate from it. Deviations from FIFO processing can not be clearly detected and hence can not be actively discouraged; the only solution is to create an environment that ensures that the miner cannot benefit from such deviations. We quantify this as a game-theoretic equilibrium. Let $\mathcal{M} = \{m_1, m_2, \ldots, m_k\}$ be the set of miners and $S$ be the set of strategies for the miners in the blockchain to act upon.

*Definition 10 ($\epsilon$-*Expected Dominant Strategy Equilibrium*):* We say $s = (s^*_{m_1}, s^*_{m_2}, \ldots, s^*_{m_k}), s^*_{m_i} \in S$ is $\epsilon$-Expected Dominant Strategy Equilibrium for the miners if for all the miners,

$$\mathbb{E}f_{txn}(s'_{m_i}, s_{-m_i}) < (1+\epsilon)\mathbb{E}f_{txn}(s^*_{m_i}, s_{-m_i}) \qquad \forall s_{-m_i} \in S_{-m_i}, \forall m_i \in \mathcal{M}$$

where $s'_{m_i} \neq s^*_{m_i}$. $s_{-m_i}$ and $S_{-m_i}$, indicate the strategy profile and set of strategy profiles, followed by the miners except $m_i$. The expectation is w.r.t. randomness in influx of the transactions and the variance in the transaction fees.

The above definition may be too strong and difficult to achieve. Hence, we also work with the following, a weaker equilibrium concept from game theory.

*Definition 11 ($\epsilon$-Expected Nash Equilibrium):* We say $s^* = (s^*_{m_1}, s^*_{m_2}, \ldots, s^*_{m_k})$, $s^*_{m_i} \in S$, is $\epsilon$-Expected Nash Equilibrium for the miners, if for each miner, the expected revenue per block by following any strategy is not more than $(1+\epsilon)$ times what it would have obtained by following $s^*_{m_i}$; provided the other miners are following $s^*_{-m_i}$. I.e.,

$$\mathbb{E}f_{txn}(s'_{m_i}, s^*_{-m_i}) \leq (1+\epsilon)\mathbb{E}f_{txn}((s^*_{m_i}, s^*_{-m_i})) \; \forall s'_{m_i} \in S, \; \forall m_i \in \mathcal{M}$$

where, $s^*_{-m_i}$ indicates the strategy profile in $s^*$ by the miners except $m_i$. The expectation is w.r.t. randomness in influx of the transactions and the variance in the transaction fees.

### B. Model

For a tractable analysis, we simulate the execution of the Bitcoin protocol in steps of 10 mins, i.e., each block is published (created and broadcast) every 10 mins. Typically, the time duration between two consecutive blocks is random, with the expected value of 10 mins. We assume that each block can contain a maximum $bs_{max}$ number of transactions. It is important to note that the maximum size of a block in Bitcoin is an inherent limitation of the Bitcoin protocol [13], [14]. In our analysis, we assume, in accordance with the standard influx condition under MOC, the number of transactions that arrive in each step is *not* constant and follow a Poisson distribution with mean $bs_{max}$. A Poisson distribution is prudent here as the influx is a discrete number of events (w.r.t. arriving transactions) occurring in step, which is a fixed interval of time, with a known constant rate of $bs_{max}$. Hence, the average influx (in terms of the number of transactions) is equal to the maximum outflux (i.e., maximum block size). Here, *outflux* is the transactions that are processed by the miners.

We split the transaction fee offered by the user as, $f_{txn} = f_{min} + f_{extra}$. $f_{min}$ is the minimum amount of fee, as perceived by the user, that must be included for the miner to process the transaction; it reflects the price of consumption. $f^0_{min} \leq f_{min}$, is the minimum transaction fee set by the protocol, and hence it is the initial value of $f_{min}$ that the users start with. As there is no restriction on the minimum $f_{txn}$ in Bitcoin, $f^0_{min}$ is 0 as per the protocol. The $f_{extra}$ is the extra fee the user would like to give to the miners to prioritize their transaction over others.

We model the aggression of a user towards $f_{extra}$ through a parameter $0 \leq \eta < \infty$; higher the $\eta$, higher the $f_{extra}$. Each user, when publishing the transaction, calculates the amount of extra fee they would like to pay as a monotonically increasing function, $f_{extra} = \phi(\eta)$. Let $\psi_\lambda(\eta)$, a *pdf* characterized by a static parameter $\lambda$, be the distribution that captures the fraction of users having aggression level towards $f_{extra}$ as $\eta$.

During each step, transactions are collected by the miners. At the end of each step, the miners form a block out of the currently unprocessed transactions followed by "instantly" publishing it. This merely a simplification of the process of continually updating the block while attempting to solve the cryptographic puzzle along with the assumption that a block would be mined by the end of the step.

We consider two modes that the miners may operate in; i) *greedy* processing, where miners greedily include transactions, i.e., they include the highest *or* lowest valued transactions, whichever may be more profitable ii) FIFO processing, where miners process transactions on a *First-In-First-Out* basis. These two modes of miner operation form the strategy space of the miners. Since the influx, as well as outflux, is in terms of the number of transactions, it is understood that all transactions are considered to be of the same size in terms of the space taken on the block. Hence *greedy* processing only considers the transaction fee offered and not the size of the transaction in bytes.

*Assumptions on Miners' Behaviour:* We assume that all miners are honest but rational. The rationality of the miners implies the following:

- Miners would continue to mine and sustain the blockchain, as long as mining costs are covered.
- Since the health of the blockchain is crucial to the value miners obtain from mining, and that all miners inherently understand this, miners act in favor of sustaining the health of the blockchain, i.e., follow FIFO processing, if the cost of doing so is marginal.

*Assumptions on Users' Behaviour:* We assume that the user would like to have his transaction processed within a reasonable time and that their inclination to have their transaction prioritized is characterized by their aggression level, $\eta$ towards paying a higher $f_{extra}$. This assumption implies the following:

- The user chooses $f_{txn}$ by first considering the $f_{min}$ and then deciding $f_{extra} = \phi(\eta)$ where $\eta$ captures its aggression parameter. Note that, the system need not know the $\eta$ for each individual user, but for analysis, we use the distribution of users against $\eta$ (i.e., $\psi_\lambda(\eta)$ is known).
- The user observes transactions, below a certain threshold of fees, being stranded, they concede to making the presumption that this threshold of fees is the new $f_{min}$, in the sense that transaction below this fees will not be processed in a reasonable time. This is because a user will not attempt to publish a transaction if they do not expect it to be processed.

### C. Simulation Setup

Since the theoretical analysis is intractable in such a complex scenario. We use simulations to support our arguments. For a fair and consistent analysis, we use the same simulation setup and parameters to simulate both Bitcoin as well as our protocol. All simulation results were averaged over 10 runs.

As mentioned in Section III-B, the execution of the protocol in consideration, proceeds in steps. Each step represents the time between blocks. At the end of each step, a block is added to the chain. Each block has a maximum capacity, $bs_{max} = 1000$, in terms of transactions. All the simulations are run in the setting where the average influx is Poisson distributed with mean equal to $bs_{max}$.

We take, $\psi_\lambda(\eta) = \lambda \cdot e^{-\lambda \cdot \eta}$, an *exponential distribution* characterized by $\lambda = 3$, where $\lambda$ is the rate parameter of the exponential distribution. $\phi$, the function used to calculate the $f_{extra}$ a user with aggression parameter $\eta$ gives, we take to be $\phi(\eta) = f_{extra} = e^{\eta} - 1$.

We take the granularity of the size of miners in terms of their mining power to be $\delta = 0.05$; and $0 \leq \beta \leq 1$ to be the fraction of miners that follow *greedy*, while the rest follow FIFO.

To observe the cost of mining as per FIFO as opposed to acting *greedy*, and further investigate the establishment of equilibrium, we simulate both, BitcoinF and Bitcoin with varying values of $\beta$. We then consider the resulting *average revenue per block* mined for both FIFO and *greedy* behavior, in both BitcoinF and Bitcoin. The fraction of mining power controlled is represented in the simulation by setting the same fraction as the probability of mining a block.

To emulate the user's characteristic of observing the change in $f_{min}$ based on the stranded transactions, we use epochs of observation. Each *observational epoch* is a series of subsequent steps at the end of which the users change their $f_{min}$ accordingly. At the end of each epoch, the users check the average processing latency of transactions that were published in this epoch. The highest transaction fee that, experiences an average processing latency high enough to be considered stranded, is considered to be the new $f_{min}$. The length of the observational epoch, i.e., the number of steps the users observe after which they change their presumption of $f_{min}$, we take to be $1000$ steps. We use $100$ to be limit to the processing latency in units of steps, after which we consider a transaction to be stranded.

Now we study Bitcoin protocol in the next section.

## IV. Bitcoin Transaction Processing: Analysis under MOC

First, we describe Bitcoin protocol, and then explain what assumptions we make, highlight the specifics of Bitcoin simulation, and the inference from the simulations.

### A. Bitcoin Protocol

In Bitcoin, the market for "space on the block" is a completely free market (FM), there is no regulation on how the transactions must be processed or how much transaction fee must be given. In Bitcoin, a block contains only one section (we refer to this as the FM section) where there are no restrictions (except our assumption on the maximum number of transactions per block, i.e., block size). The users publish only one instance per transaction containing $f_{txn} = f_{min} + f_{extra}$. Initially, $f_{min} = 0$ as the Bitcoin protocol does not state any minimum transaction fee that must be included.

In TFOM, each miner must collect transaction fees to sustain their mining efforts. Even when assuming an influx of transactions that is on an average sufficient to fill the blocks, there is no guarantee that the incoming transactions will contain sufficient transaction fees to sustain the mining efforts. Hence, individual fairness for the miners can not be established.

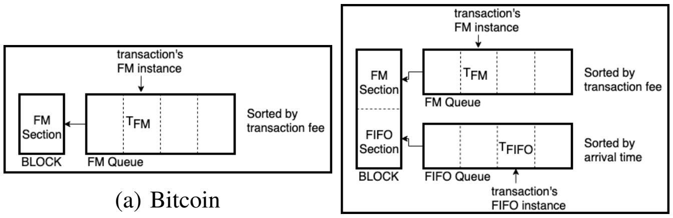
In Bitcoin, while the block rewards alone are sufficient to sustain mining efforts, the rational miners can be expected to process transactions in a FIFO manner, especially since the number of users offering a competitive fee is minimal. However, under MOC, we assume in our analysis that when processing transactions from the FM queue to be added to the FM section of the block, the miners pick the transactions offering highest transaction fees, as shown in Fig. 2a. The miners could choose to follow FIFO while processing transactions from the FM queue. They might want to do this to preserve the health of the blockchain. However, we expect our simulation-based game-theoretic investigation into the establishment of equilibrium will yield that the miners lose a large part of their revenue by following FIFO processing in the FM queue. Hence the miners cannot be expected to act in favor of the blockchain's health as the loss is considerable. Thus, we assume the miners will gather transactions greedily.

The users suffer as a consequence of the miners not being able to follow FIFO processing. Not wanting to follow FIFO would not be cause for concern to the users if the influx of transactions to be processed was low. If the influx of transactions is low, then no matter the transaction fees, all transactions would be included in the next block. In low influx scenarios, the competition is not high, and the users realize that there is no need to pay any more than a marginal fee, as their transaction would get included in the next block regardless. However, as we show in our analysis, issues arise in higher influx scenarios where the users *must* pay competitive fees to have their transactions processed or risk having them stranded. Further, since these stranded transactions are public knowledge, this, as we see in our analysis, causes the price of consumption to rise, adversely affecting the health of the blockchain.

In Bitcoin, under MOC, as we show using simulations, there are a large number of transactions that get stranded. This causes the users to be uncertain about when their transaction will be processed, or even if it ever will be. These stranded transactions are public knowledge. Thus, over a sufficiently long series of consecutive steps, if it is observed that there are transactions that are stranded, the users are likely to treat the highest transaction fees offered by the stranded transactions as the new $f_{min}$ to avoid their transactions being pushed further down in the queue. Given this new $f_{min}$, the phenomenon will repeat. Depending on the observations of stranded transactions, $f_{min}$ can increase, decrease (not below $f^0_{min}$), or remain constant. We study what is likely happen for $f_{min}$ Bitcoin via simulations.

### B. Simulation Specifics

Since the miners only include transactions greedily, to emulate miners in this scenario, we have only one section (FM section) of the block, which spans the entire capacity of the block, to which we keep adding the highest valued transaction till the block is full. As per the Bitcoin protocol, we keep the initial $f_{min} = 0$.



(a) Bitcoin

(b) BitcoinF

Fig. 2: Transaction processing in Bitcoin and BitcoinF

First, we conduct simulations to investigate the establishment of equilibrium, which we expect to state that to follow *greedy* processing is *$\epsilon$-Expected Dominant Strategy Equilibrium* validating the assumption that the miners follow *greedy* processing from the FM queue. We assume the same in our simulation estimating average processing latency and the change in $f_{min}$ with observation epochs.

### C. Simulation Results and Inference

First, we discuss the result of our investigation of equilibrium of miner behavior in the Bitcoin ecosystem. The strategy space of the miners here is $S = \{FIFO, \textit{greedy}\}$, where $s \in S$ is the mode of processing (as in Section III-B) of the miner in the FM queue.

Clearly, from Fig. 3a, we see that:

$$\mathbb{E}f_{txn}(s'_{m_i}, s_{-m_i}) < (1+\epsilon)\mathbb{E}f_{txn}(s^*_{m_i}, s_{-m_i}) \qquad \forall s_{-m_i} \in S_{-m_i}, \forall m_i \in \mathcal{M}$$

where $s'_{m_i} = FIFO$, $s^*_{m_i} = \textit{greedy}$ and $\epsilon = 0$.

Intuitively, miners make significantly higher revenue if they followed *greedy* processing as opposed to FIFO processing regardless of $\beta$. Thus we say that following *greedy* processing in the FM queue is *$\epsilon$-Expected Dominant Strategy Equilibrium* with $\epsilon = 0$.

Confirming our assumption about miner behavior, we now discuss the results of our simulations regarding the fairness of the blockchain. As we can see from Fig. 3b, the $f_{min}$ rises with observational epochs. This causes instability in the price of consumption. The figure also implies that when the influx has not been *standard* for long enough, Bitcoin cannot ensure that mining costs will be covered, i.e., individual fairness for miners is not guaranteed. In Bitcoin, Fig. 3c shows that, the users that have very little aggression towards paying $f_{extra}$, experience unreasonably high processing latency, i.e., stranded transactions. These stranded transactions, in turn, cause the price of consumption to rise, as seen in Fig. 3b.

These observations are summarized as Proposition 1.

*Proposition 1:* If the miners strategy space is $S = \{FIFO, \textit{greedy}\}$, it is *$\epsilon$-Expected Dominant Strategy Equilibrium* for the miners to follow *greedy* with $\epsilon = 0$. As a consequence, the Bitcoin ecosystem is not a fair under MOC.

## V. Achieving Fairness under MOC

As discussed in the previous section, under MOC, Bitcoin faces challenges. To resolve this, we propose a simple modification to Bitcoin, which we call BitcoinF.

### A. BitcoinF

To solve the issues of unfairness for both the miners and the users, we propose a protocol to process transactions. Our approach is two-fold: Firstly, we enforce a minimum fee of $f^0_{min} (> 0)$ that is to be included in every transaction. Secondly, we introduce a section in the block that only accepts transaction instances with $f_{txn} = f^0_{min}$, called the FIFO section of the block. So now, there are two sections in the block: FM and FIFO. The FIFO section has a size of $\alpha \cdot bs_{max}$, whereas FM has the remaining. This is illustrated in Fig. 2b. In our simulation of BitcoinF, we set $\alpha = 0.2$.

Formally, we propose BitcoinF as a *block validation rule*. This rule will have two parameters, $\alpha$, and $f^0_{min}$. Miners shall only accept and extend blocks that follow the rule. We expect that when this protocol is implemented, it will be enforced by the honest miners, as is commonplace in blockchain ecosystems.

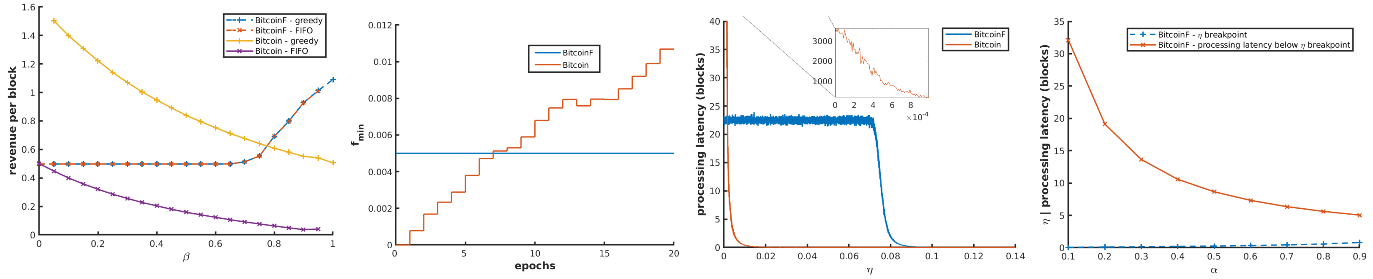
*Definition 12 (BitcoinF: Block Validation Rule):* Each block must contain $\alpha \cdot bs_{max}$ transactions offering $f_{txn} = f^0_{min}$.

A typical execution is described as follows:

- Users when they want to add a transaction to the blockchain broadcast two instances of the same transaction; one instance that has $f_{extra} = 0$, and the other instance where $f_{extra}$ is as chosen by the user. Both instances must include at least $f^0_{min}$ as required.
- The miners collect these instances of every transaction and add the instance with $f_{extra} = 0$ to the FIFO queue and the other instance, the one with $f_{extra}$ to the FM queue.
- When an instance of a transaction is processed, the other instance is invalidated.
- The block can have transactions in the FM section *only* if the FIFO section of the block is completely filled. The honest but rational miners naturally would first add as many transactions as possible to the FM section from the FM queue (they may do this however they please, but naturally they choose the ones with highest transaction fees), while keeping aside sufficient transactions to fill the FIFO section of the block. Then, the miners fill the FIFO section of the block with transactions selected from the FIFO queue in a FIFO manner.

When processing transactions from the FM queue to be added to the FM section of the block, the miners naturally pick the transactions offering the highest fees. Further, we assume that while processing transactions from the FIFO queue, the miners follow FIFO.

The miners could choose to follow *greedy* processing, i.e., add the minimum valued transactions instead of following FIFO processing to fill the FIFO section. They might want to do this, as FIFO processing might process some slightly higher valued transactions through the FIFO section of the block, voiding the $f_{extra}$ it offers; thus, by following *greedy* processing, the miners process the least valued transactions through the FIFO section, while keeping the slightly higher valued transactions (which would otherwise get processed



(a) Difference in revenue of FIFO and *greedy*

(b) $f_{min}$ vs steps

(c) Processing latency vs $\eta$

(d) Trend w.r.t. alpha

Fig. 3: Simulation Results

through FIFO) for later, to be processed through the FM section of the block. However, we expect our simulation-based game-theoretic investigation into the establishment of equilibrium will yield that the miners gain a negligible profit by following *greedy* processing as opposed to FIFO processing in the FIFO queue. Hence the miners can be expected to act in favor of the blockchain's health as the loss is insignificant. Thus, we assume the miners will follow FIFO processing in the FIFO queue.

### B. Simulation Specifics

The size of the FIFO section is set to be $\alpha \cdot bs_{max} = 200$ transactions, to which transactions are added in a FIFO manner. The size of the FM section is set to be $(1-\alpha) \cdot bs_{max} = 800$ transactions, to which transactions are added greedily. The FM queue is processed before the FIFO queue, as is the expected behavior of the miners. To emulate the random order of the transactions' arrival during a step, when processing from the FIFO queue, random transactions are picked from the set of transactions with the highest processing latency. As per the protocol, initially, the value of $f^0_{min}$ is set appropriately to compensate miners, we, in our simulation, set it to be $0.005$.

First, we conduct a simulation to investigate *$\epsilon$-Expected Nash Equilibrium* which we expect to state that to follow the FIFO is *$\epsilon$-Expected Nash Equilibrium* validating the assumption that the miners follow FIFO processing from the FIFO queue. We assume the same in our simulation estimating average processing latency and the change in $f_{min}$ with observation epochs.

### C. Simulation Results and Inference

First, we discuss the result of our investigation of equilibrium of miner behavior in the BitcoinF ecosystem. The strategy space of the miners here is $S = \{FIFO, greedy\}$, where $s \in S$ is the mode of processing (as in Section III-B) of the miner in the FIFO queue.

Clearly, from Fig. 3a, we see that:

$$\mathbb{E}f_{txn}(s'_{m_i}, s^*_{-m_i}) \leq (1+\epsilon)\mathbb{E}f_{txn}((s^*_{m_i}, s^*_{-m_i}))\ \forall s'_{m_i} \in S,\ \forall m_i \in \mathcal{M}$$

where $s^* = \{FIFO, FIFO, \ldots, FIFO\}$ and $\epsilon = 0.00037$.

Intuitively, miners gain negligible profit if they followed *greedy* processing as opposed to FIFO processing in the FIFO queue when $\beta = 0$. Thus we say that all miners following FIFO processing in the FIFO queue is *$\epsilon$-Expected Nash Equilibrium* with $\epsilon = 0.00037$.

Here, *$\epsilon$-Expected Nash Equilibrium* is clearly a much weaker property than the one established by BitcoinF. Motivated by the analysis in [15], where the authors consider a fraction of agents (our case miners) following honest strategy and remaining agents follow greedy strategy, it is easy to see that our protocol exhibits *$\epsilon$-Expected Dominant Strategy Equilibrium* with $\epsilon = 0.00037$ when $\beta \leq 0.55$. The protocol does not establish *$\epsilon$-Expected Dominant Strategy Equilibrium* for all $\beta$, as after a point, the fraction of miners following FIFO processing in the FIFO queue drops low enough that transactions start to get stranded, raising the price of consumption and hence raising the average revenue of all miners.

Confirming our assumption about miner behavior, we can now discuss the results of our simulations regarding the fairness of the blockchain. As seen in Fig. 3b, $f_{min}$ remains constant with time. This implies a stable price of consumption. Since the $f^0_{min}$ can be set as per requirements to cover mining costs, BitcoinF guarantees individual fairness for miners under MOC. Fig. 3c shows that in BitcoinF, no matter the users' aggression towards paying $f_{extra}$, the users experience reasonable processing latency. Since there are no stranded transactions, the price of consumption does not rise, as seen in Fig. 3b. In fact, under MOC, the users would know what processing latency to expect as soon as they publish the transaction. If their $\eta$ is higher than a certain $\eta$ *break-point*, then their transaction will get processed almost immediately; if not, they know the upper bound on the processing latency they will experience. This trade-off between the $\eta$ break-point and processing latency below the $\eta$ break-point as a function of $\alpha$ is visualized in Fig. 3d. These simulation based observations can be summarized as Proposition 2.

*Proposition 2:* If the miners strategy space is $S = \{FIFO, greedy\}$, it is *$\epsilon$-Expected Nash Equilibrium* for the miners to follow FIFO with $\epsilon = 37 * 10^{-5}$. As a consequence, BitcoinF ecosystem is a fair under MOC.

### D. Security Analysis

A strategic and intelligent miner could attempt to use our protocol to leverage an unfair advantage. In this section, we

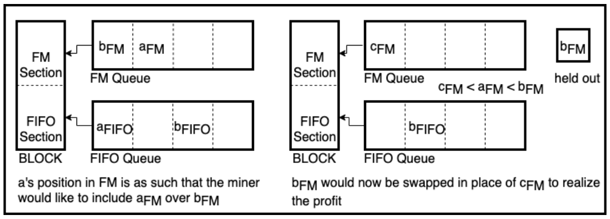


Fig. 4: Depiction of swapping

show that such attempts are not quite effective and hence, do not threaten our protocol.

*a) Ignoring the FIFO Section of the block:* The miners would ideally like to ignore all the $f^0_{min}$ instances of the transactions. The miners cannot do this as any valid block requires that the FIFO section of the block must be filled entirely before any transactions are added to the FM section of the block. Also, all transaction instances in the FIFO section are only supposed to offer $f^0_{min}$ fee.

*b) Swapping in transactions that are about to be processed through FIFO queue:* The miner can always process a transaction of lower value, say transaction $a$, than it should from the FM queue, by swapping out a transaction of higher value, say transaction $b$. Please refer to Fig. 4. The miner might be inclined to do this if $a$ is about to be processed via the FIFO queue, whereas $b$ is not. The miner might want to do this if he notices that $a$ is, in fact, of higher value than what is typically included in the FM section. In this sense, the miners keep $b$ held out, to be swapped in later (finally) for another transaction, say transaction $c$, that is lower in value as compared to $b$ and $a$, thus realizing a profit. The miner may swap several times (swapping out transactions $b$, $b'$, $b''$ ...) before finally swapping out transaction $c$.

Note that during this attempt, it is easy to see that the miner is risking losing a profit as he is betting on finding the transaction $c$. He may not find such a transaction if the lowest value of the transactions included in the FM section never drops sufficiently, or the miner does not maintain a mining monopoly, and some other miner mines the next block gaining the risked amount.

The theoretical analysis of this attack strategy is not tractable, and nor is the simulating the attack feasible due to the complexity of the actions available and state-space. Thus, to show that this attack is ineffective, we give the adversary generous and impractical advantages and show that even in the best case of executing this attack in the backdrop of our simulation, the adversary gains as little as less than $1\%$ of the total rewards gained.

Since the transaction $a$ must ultimately be swapped with a transaction $c$, we simply consider the number of potentially ultimately successful swaps, as the minimum of; the number of transactions processed by the FM queue, that are valued below the maximum value of the transactions processed by the FIFO queue; and the number of transactions processed by the FIFO queue, that are valued above the minimum value of the transactions processed by the FM queue. We multiply the value obtained by the difference between; the maximum value of the transactions processed by the FIFO queue and the minimum value of the transactions processed by the FM queue. In our simulations, the resultant value turns out to be less than $0.59 \pm 0.29\%$ of the total value of transactions processed.

Now, this is clearly an over-valuation for the following reasons; (i) The risk of the attempt is completely ignored here. (ii) The number of ultimately successful swaps considered is the result of the best (perhaps better than the practical best) possible exploitation of the one-to-one correspondence of the swapped transactions by the adversary. (iii) The value of profit gained with each successful swap is just taken to be the maximum profit gained from the best possible swap. (iv) The attacking miner is assumed to have a monopoly over mining, i.e., the attacking miner is the only one mining and hence can carry out this attack unhindered, i.e., without the possibility of another miner publishing a block impeding the attacking efforts.

### E. Discussion

The parameters of BitcoinF, $\alpha$ and $f^0_{min}$ can be chosen by consensus and should be agreeable by both the miners and the users; we suggest $\alpha = 0.2$ and $f^0_{min} = \frac{\text{average cost of mining a block}}{bs_{max}}$. If $\alpha = 0$, then BitcoinF reduces to Bitcoin, whereas $\alpha = 1$ would be strictly FIFO. Bitcoin, as we have seen, is not fair. Strictly FIFO processing disables the ability of the users to express urgency, and the average processing latency will not decrease with increasing $\eta$, which is a requirement for fairness for the users. An $f^0_{min}$ too low will discourage mining, an $f^0_{min}$ too high will discourage usage of the blockchain.

While we have chosen specific functions and parameters for $\phi$ and $\psi$, we believe that any monotonically increasing function for $\phi$ and any monotonically decreasing function for $\psi$ would yield similar yet scaled results.

## VI. Conclusion

In this paper, we studied Bitcoin under *mature operating conditions* (MOC), i.e., in TFOM and standard influx. To study a given blockchain, we introduced notions of fairness (i) for the miners and the users, and (ii) the health of a blockchain. Under reasonable assumptions, we showed using simulations that miners act greedily in Bitcoin, as it is $\epsilon$-*Expected Dominant Strategy Equilibrium* to do so, and as a consequence, Bitcoin ecosystem is not fair. To achieve fairness in Bitcoin, we propose BitcoinF, a simple yet powerful modification to Bitcoin. In BitcoinF; each transaction must include a minimum amount, to ensure that transaction fees cover marginal mining costs under MOC; and must have a minimum number of transactions offering the minimum specified amount. We showed using simulation analysis that in BitcoinF, miners act in favor of the health of the blockchain as it is $\epsilon$-*Expected Nash Equilibrium*, and as a consequence, BitcoinF ecosystem is fair.

# References

[1] S. Nakamoto *et al.*, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[2] M. Carlsten, H. Kalodner, S. M. Weinberg, and A. Narayanan, "On the instability of bitcoin without the block reward," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 154–167.

[3] blockchain.com, "Mempool transaction count," https://www.blockchain.com/charts/, 2019, accessed: 15-November-2019.

[4] earn.com, "Bitcoin fees for transactions," https://bitcoinfees.earn.com/, 2019, accessed: 15-November-2019.

[5] M. Möser and R. Böhme, "Trends, tips, tolls: A longitudinal study of bitcoin transaction fees," in *International Conference on Financial Cryptography and Data Security*. Springer, 2015, pp. 19–33.

[6] J. Li, Y. Yuan, S. Wang, and F.-Y. Wang, "Transaction queuing game in bitcoin blockchain," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 114–119.

[7] N. Houy, "The economics of bitcoin transaction fees," *GATE WP*, vol. 1407, 2014.

[8] D. Easley, M. O'Hara, and S. Basu, "From mining to markets: The evolution of bitcoin transaction fees," *Journal of Financial Economics*, 2019.

[9] J. A. Kroll, I. C. Davey, and E. W. Felten, "The economics of bitcoin mining, or bitcoin in the presence of adversaries," in *Proceedings of WEIS*, vol. 2013, 2013, p. 11.

[10] G. Huberman, J. Leshno, and C. C. Moallemi, "An economic analysis of the bitcoin payment system," *Columbia Business School Research Paper*, no. 17-92, 2019.

[11] S. Kasahara and J. Kawahara, "Effect of bitcoin fee on transaction-confirmation process," *arXiv preprint arXiv:1604.00103*, 2016.

[12] D. Koops, "Predicting the confirmation time of bitcoin transactions," *arXiv preprint arXiv:1809.10596*, 2018.

[13] P. R. Rizun, "A transaction fee market exists without a block size limit," *Block Size Limit Debate Working Paper*, 2015.

[14] B. Magazine, "What is the bitcoin block size limit?" https://bitcoinmagazine.com/guides/what-is-the-bitcoin-block-size-limit, 2019, accessed: 15-November-2019.

[15] J. Zou, S. Gujar, and D. Parkes, "Tolerable manipulability in dynamic assignment without money," in *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.