On Blockchain Commit Times: An analysis of how miners choose Bitcoin transactions

Johnnatan Messias
johnme@mpi-sws.org
Max Planck Institute for Software Systems (MPI-SWS)
Germany

Mohamed Alzayat
alzayat@mpi-sws.org
Max Planck Institute for Software Systems (MPI-SWS)
Germany

Balakrishnan Chandrasekaran
balac@mpi-inf.mpg.de
Max Planck Institute for Informatics (MPI-INF)
Germany

Krishna P. Gummadi
gummadi@mpi-sws.org
Max Planck Institute for Software Systems (MPI-SWS)
Germany

ABSTRACT

Blockchains suffer from a well-known, non-trivial scalability problem: The low throughput (i.e., transactions committed per unit time) of blockchains when paired with the increasingly high volume of issued transactions leads to significant delays in transaction commit times. In a month-long investigation of Bitcoin, we reveal that congestion (i.e., when there exist more transactions than can be included in a block) is typical and that commit times exhibit a significant variance during periods of congestion. Although the fee-per-byte dequeuing policy is widely considered the "norm" for prioritizing transactions—and explaining how and when transactions are committed—we show that miners *somehow* delay a significant fraction of transactions. Such deviations undermine the utility of blockchains for ensuring a "fair" ordering that might be required for some applications.

KEYWORDS

Blockchain, transaction commit times, transaction ordering, fairness, Bitcoin

1 INTRODUCTION

Blockchain is a decentralized ledger for recording transactions between two or more clients participating in the blockchain's peer-to-peer network. Transactions are recorded in chunks referred to as blocks, and the chain of blocks, with each pointing to an ancestor, constitutes the blockchain. Participants who validate a transaction, include it in a block, and extend the blockchain are referred to as miners. In return for extending the blockchain, the miners get a block reward. Moreover, clients (i.e., users issuing transactions) voluntarily attach a fee—to further subsidize the miners—with each transaction. Miners reap this transaction fee upon inclusion of the concerned transaction in a block that successfully extends the chain. In the context of cryptocurrencies, transactions entail the exchange of currency between clients, and the blockchain reflects the consensus (to a first level of approximation) among the miners on what sequence of blocks constitute the chain. The consensus on the blocks precipitates in the confirmation of transactions contained in them. In 2008, Nakamoto laid the ground rules for maintaining this distributed, decentralized ledger based on a proof-of-work scheme [23], and particularly absent from this specification was the requirement of any notion of trust in the miners.

The design of a distributed ledger, particularly by weaning it from a reliance on any notion of trusted entities, is remarkable; we refer to this ledger (and its variants) as simply blockchain. Blockchain has earned quite an ardent following: Blockchain-based solutions enjoy support across a wide range of domains including education [25], insurance [21], healthcare [11], supply-chain management [26, 27], and government [9, 14, 20]. The widespread use of blockchains, particularly of the *permissionless* variety—where *any* user can join and participate in the network, as in the case of Bicoin [23]—leads us to ask a simple question: Do the miners adhere to the "norms" prescribed by the blockchain? While our investigation concerns the behavior of miners in any blockchain, we focus, in this paper, on the blockchain underpinning Bitcoin [23].

Bitcoin is the largest cryptocurrency in the world, with a market capitalization of over 173.6 bn dollars (USD) as of June 2020 [7]. The increasing volume of transactions issued in the Bitcoin network introduces *congestion* among transactions for confirmation: At any point in time, there may be more transactions than can be immediately confirmed in a block. Unconfirmed transactions, hence, must wait for their "turn" to be confirmed, thereby introducing *delays*. This issue of congestion and of increasing delays in confirmation or commit[1] times is well-known and quite general to blockchains (refer to Figs. 1 & 2 in [17]). The rich body of prior work on addressing the delays and scaling the blockchain, however, have skirted around the subject of trust in miners.

In the context of Bitcoin, the conventional wisdom or "norm" is that miners follow a transaction ordering strategy similar to that in the *GetBlockTemplate (GBT)* [3] mining protocol. GBT uses transaction fees normalized by the size of the corresponding transactions—transaction fee-per-byte, in short—for determining a rank order of transactions for inclusion in the blockchain. We show that this conventional wisdom is *not* necessarily true, and we summarize our contributions as follows.

- ⋆ In two separate studies of Bitcoin, one spanning three weeks and the other covering four weeks, we found the network to be congested most of the time, i.e., approximately 75% of the three-week and 92% of the four-week periods.
- ⋆ Transaction commit times significantly vary due to congestion: While 60% of the transactions are immediately included in a block, 20% wait for at least 2 blocks (or 20 minutes on average).

[1] We use the terms 'confirmation' and 'commit' interchangeably to refer to the inclusion of a transaction in a block.

★ The "norm" that transactions are selected for inclusion based on fee-per-byte does *not* completely explain the commit delays: In 50% of blocks mined in Bitcoin, over three weeks, 22% of the transactions included in a block (on average) do *not* follow the fee-per-byte priority ordering.

Why should you care? These deviations from the "norm," consistently observed in Bitcoin across the top miners (i.e., top in the rank ordering of miners based on the fraction of blocks they have mined) has serious implications for the users. How can users, for instance, accurately estimate the fee that they should include in their transactions to minimize delays? Transaction-fee predictions from any predictor that assumes that miners follow the "norm," will be misleading, particularly since predictors are also bundled with some client-side software.[2]

Further, how can users or any third party ensure that the miners are adhering to *fee-per-byte* or any other norm? Today, the reward for mining a block is at least two orders of magnitude larger than the aggregate reward gained by the miner through transaction fees. The aberrant behavior of the miners, manifested by the deviations from the "norm," hence, will only become more prevalent when mining rewards decrease and transaction fees start representing a significant share[3] of the miners' payouts.

The rest of this paper is organized as follows. We review relevant literature in §2 and provide a brief background on blockchains and Bitcoin in §3. We discuss whether congestion in Bitcoin is a typical phenomenon and its implication for the transaction delays in §4. In §5, we analyze how miners prioritize transactions for inclusion and discuss the deviations from the "norm" in §6. We discuss the miners' aberrant behavior in §7 and its implications for the Bitcoin ecosystem. In §8, we summarize the results and present our conclusions.

2 RELATED WORK

It is well known that the decentralized nature of blockchains poses a non-trivial scalability problem: Bitcoin Relay Network [22], FIBRE [13], Falcon [2], and, more recently, BDN [17] address the problem by improving the underlying network. These solutions will alleviate congestion in the network, but it is also likely that a faster network (built using these efforts) will attract more users, invalidating the network's benefits (and reintroducing congestion). In its proposal to use CDNs to accelerate the announcements of blocks (and transactions), BDN [17], interestingly, defines a norm—the need for the network to be neutral to all peers—and proposes an approach to test the network's compliance to this norm. Investigation of miners' compliance to the norm, in a similar manner, unfortunately, has mostly remained unexplored.

Our question concerning the behavior of miners, at its core, casts doubts upon the *incentive compatibility* of blockchain's design. Eyal et al. [12] were the first to broach this subject and show that the design is indeed not incentive compatible. Our question is, however, orthogonal to this prior work: We focus on how the participants select transactions for recording in the blockchain, and whether this *dequeuing policy* of candidates, from a pool of available transactions, follows the established *norm*. The behavior of the participants after the transactions have been selected for inclusion in the blockchain, explored in [12], is irrelevant to our question. In a similar way, we consider the rich literature on the security of Bitcoin or, in general, blockchains (e.g., [15, 16, 28]) to be orthogonal to our work.

Lavi et al. [18] and Basu et al. [1] highlight the inefficiencies in the existing (transaction) fee setting mechanisms, and propose alternatives. Except for the agreement on the topic of a lack of trust in miners' adhering to a norm, these efforts are orthogonal to our work; they highlight that miners might not be trustworthy, but do not substantiate that claim with empirical observations. Lastly, we show that miners somehow deviate from the fee-per-byte dequeuing policy, and this behavior will only become worse in the future.

3 PRELIMINARIES

Bitcoin was introduced in 2008 by Satoshi Nakamoto [23]. A bitcoin user or client issues transactions that move currency from one or more addresses or *wallets* owned by the client to another. Clients are connected to one another via a peer-to-peer network, and the transactions issued are broadcast over this network via a gossip protocol. A subset of the users, referred to as *miners* validate the transactions, and bundle them in a *block*. A block constitutes a set of zero (i.e., empty block)[4] or more transactions in addition to the *coinbase* transaction, which moves the block reward to the miner's wallet.

Transactions pending inclusion in a block are deemed *unconfirmed*. Miners create a block by including these unconfirmed transactions, and solving a cryptographic puzzle that includes, among other things, a hash of the most recent block mined in the network. The chain of cryptographic hashes linking each block to an ancestor all the way to the initial (or *genesis*) block [5] constitutes the blockchain. Miners are rewarded for their work in two ways. First, a miner reaps a block reward upon mining a block. Second, miners also collect the fees, if any, from each transaction; fees are included by users for incentivizing the miners to commit their transaction. To maximize their chances of solving the cryptographic challenge and minimize the variance in their revenues, miners often work together in groups, called as *mining pools*: Members of the pool share their computational power over a network; upon mining a block, the reward is split between the members according to the fraction of power each contributed for the mining.

The difficulty of the cryptographic puzzle is adjusted periodically, once for every 2016 blocks mined (i.e., two weeks, on average), to ensure that the blocks are mined at a steady rate.

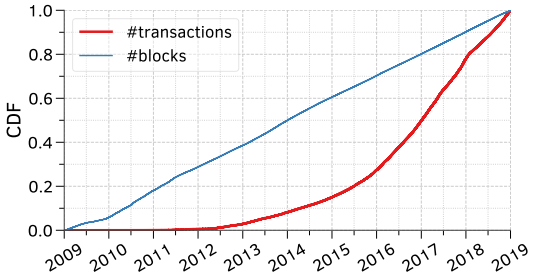
Nodes[5] queue the unconfirmed transactions received via broadcasts in an in-memory buffer, called the *Mempool*, from where they are dequeued for inclusion in a block. Bitcoin Core [4], the most widely used software [6], uses a dequeuing policy (referred to as the *GetBlockTemplate* mining protocol [3]) based on the fee-per-byte (i.e., transaction fees normalized by the transaction's size) metric. Throughput the paper, the term size refers to *virtual size*, each unit

[2] Coinbase is one of the top cryptocurrency exchange that hard-codes transaction fees (https://help.coinbase.com/en/coinbase/trading-and-funding/pricing-and-fees/what-are-miner-fees-and-does-coinbase-pay-them.html).

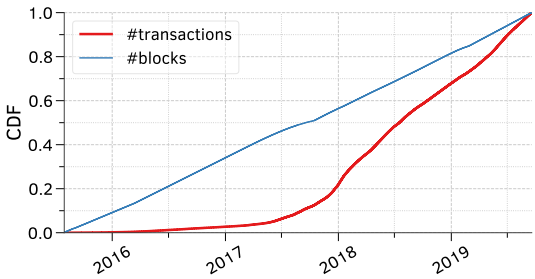
[3] Fig. 1 in Easley et al. [10] shows that yearly revenue from transaction fees in 2016 was five times that from the previous three years combined.

[4] As miners can also mine a block without including any transaction on it, we refer to those blocks as *empty blocks*.

[5] The machines running the software that allows users to participate—i.e., either actively mining or passively observing—in the Bitcoin network are referred to as *nodes*.



(a) ***Bitcoin***



(b) ***Ethereum***

Figure 1: ***The transaction-issue rate easily outpaces the block-mining rate: (a)*** 60% ***of all Bitcoin transactions were added from mid-2016; and similarly, (b) on Ethereum*** 80% ***of all transactions were added from the end 2018.***

of which corresponds to four *weight units* as defined in bitcoin improvement proposal BIP-141 [19]. Conventional wisdom holds that the miners following GBT's dequeuing policy (i.e., dequeuing transactions in the order defined by the fee-per-byte metric) is the expected "norm."

4 CHARACTERIZING TRANSACTION COMMITS

A congestion in the Mempool leads to a contention among transactions for inclusion in a block. The congestion inevitably results in delaying the transaction-commit times. In this section, we discuss whether congestion of the Mempool is a typical phenomenon and its implications for transaction-commit delays. We follow up this discussion with an analyses on if, and how, users adjust transaction fees to cope with congestion, and the impact of these fee adjustments on commit delays.

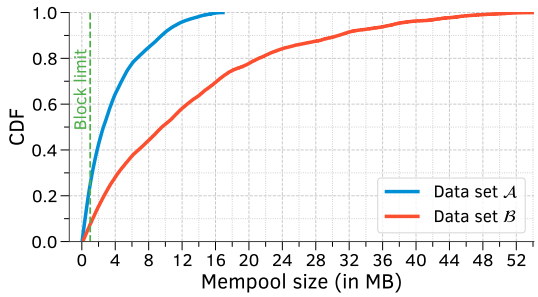
4.1 Congestion and delays

Bitcoin's design—specifically, the adjustment of hashing difficulty to keep mining rate constant—virtually ensures that there is a steady flow of currency generation in the network. Fig. 1a confirms that the aggregate number of blocks mined in Bitcoin, obtained from a *full node*[6], increases linearly over time. Transactions introduced in the network, in contrast, are subject to no such constraints and have been increasing more aggressively since mid-2016: 60% of all

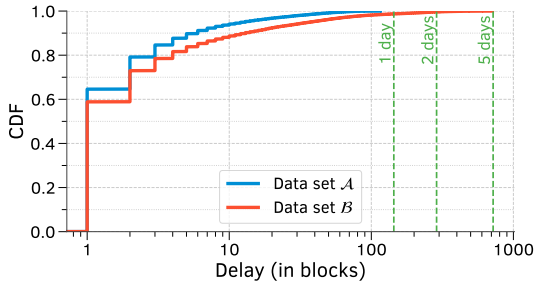
[6] A *full node* maintains a copy of the entire ledger (i.e., the blockchain history), receives broadcasts of blocks and transactions, and re-broadcasts these announcements to others. It also fully validates transactions and blocks, helping to keep the blockchain tamper-evident.

Table 1: ***Overview of the Bitcoin data sets.***

Attributes	*Data set $\mathcal{A}$*	*Data set $\mathcal{B}$*
Time span	Feb. 20^{th} – Mar. 13^{th}, 2019	Jun. 1^{st} – 30^{th}, 2019
#Tx. committed	6,816,375	10,487,966
#blocks	3119	4522
#empty-blocks	38	18



(a)



(b)

Figure 2: ***(a) Mempool is, typically, congested, i.e., for nearly*** 75% ***of the time period of data set $\mathcal{A}$ and for*** 92% ***of that of $\mathcal{B}$; (b) While*** 60% ***(***65%***) of transactions in $\mathcal{A}$ ($\mathcal{B}$) get included in the next block,*** 15% ***(***20%***) of them wait for at least*** 3 ***blocks (i.e.,*** 30 ***minutes on average).***

transactions ever introduced were added in only in the last two and a half years of the decade-long life of the cryptocurrency. Per Fig. 1b, a similar trend among blocks and transactions also manifests in Ethereum,[7] but we restrict our analyses in this paper to only Bitcoin. If this trend in transaction growth continues, users submitting the transactions will have to contend with one another for the limited space (of 1 MB) in a block to have their transaction(s) committed. This claim is more than simply an eventuality: Our measurements of the Bitcoin network indicates that congestion among transactions is already common.

We measured the number of transactions added to Bitcoin over time by running a full node. We recorded the size of the node's Mempool, once every minute, across two study periods. The first study ran for three weeks from February 20^{th} through March 13^{th} of 2019, while the second ran throughout the month of June 2019. The state of the Mempool in each one-minute interval, including the transactions and blocks received during this interval, constitutes

[7] Based on data obtained from https://etherscan.io

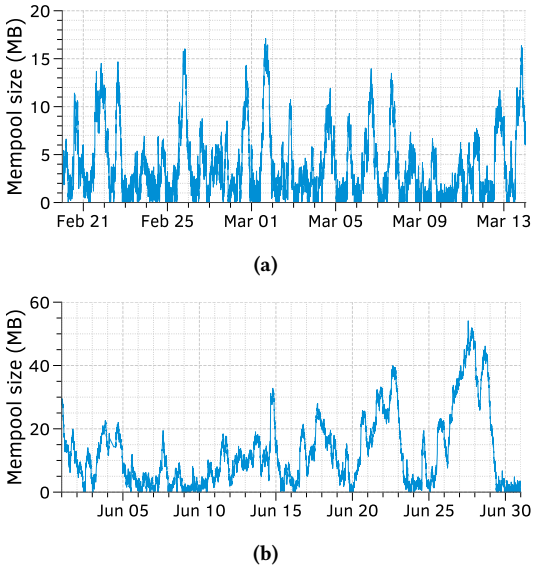


Figure 3: ***Mempool size as a function of time in both data sets (a) $\mathcal{A}$ and (b) $\mathcal{B}$. Data set $\mathcal{B}$ shows much higher levels of congestion compared to $\mathcal{A}$.***

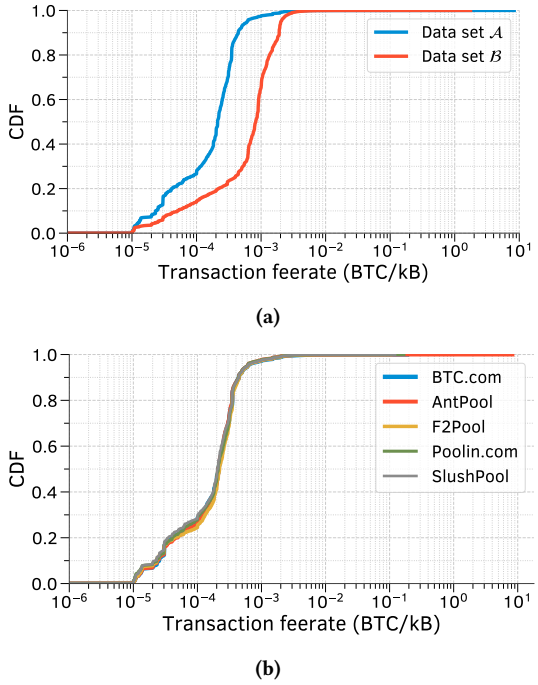


Figure 4: ***(a) Feerates show a large variance, with the majority being one to two orders of magnitude higher the recommended minimum. (b) Feerates of transactions committed by different mining pool operators mostly looks similar.***

a *snapshot* of the Mempool. We refer to these studies, together with the data set obtained from each, collectively as $\mathcal{A}$ and $\mathcal{B}$, respectively; Tab. 1 presents an overview of the data sets. Across both the time periods, the aggregate size (in MB) of all unconfirmed transactions in the Mempool of our full node shows a huge variance, as shown in Fig. 2a. The green line marks the maximum size of a block (i.e., 1 MB), and indicates that the Mempool is, *typically*, congested: During the three-week period of $\mathcal{A}$, the Mempool size was above the 1 MB threshold for nearly 75% of the time, and during the four-week period of $\mathcal{B}$ the Mempool was congested for nearly 92% of the time period.

Figs. 3a and 3b provide a complementary view of the Mempool congestion. They show the timelines of the Mempool sizes, measured every minute, in both the data sets, $\mathcal{A}$ and $\mathcal{B}$. Measurements from data set $\mathcal{B}$ exhibit much higher levels of Mempool congestion compared to that of $\mathcal{A}$: Mempool size fluctuations in $\mathcal{B}$ are approximately three times higher than that in $\mathcal{A}$. Consequently, at times, Mempool in $\mathcal{B}$ takes much longer durations than in $\mathcal{A}$ to be drained of all transactions, e.g., around June 22$^{\text{th}}$ or June 25$^{\text{th}}$, when there was a surge in Bitcoin price following by the announcements of Facebook Libra[8] and the US dollar depreciation [24].

The Mempool congestion, which in turns leads to the contention among transactions for inclusion in a block, has one serious implication for users: delays in transaction commit times. While 65% (60%) of all transactions in data set $\mathcal{A}$ ($\mathcal{B}$) get committed in the next block (i.e., in the block immediately following their arrival in the Mempool), Fig. 2b shows that nearly 15% (20%) of them wait for at least 3 blocks (i.e., 30 minutes on average). Moreover, 5% (10%) of the transactions wait for 10 or more blocks, or 100 minutes on average, in data set $\mathcal{A}$ ($\mathcal{B}$). While no transaction waited for more than a day in data set $\mathcal{A}$, a small percentage of transactions waited for up to five days (because of the high levels of congestion during June 2019) in data set $\mathcal{B}$.

Takeaways. Mempool is typically congested, and transactions must, hence, contend with one another for inclusion in the block. Mempool congestion has a non-trivial impact on the commit times of transactions.

4.2 Combating delays via transaction fees

To combat the delays and ensure that a transaction is committed "on time" (i.e., selected for inclusion in the earliest block), users may incentivize the miner by including a transaction fee. While the block reward today is 6.25 BTC[9], by the time we conduct our study it was 12.5 BTC, then the aggregate fees accrued per block remains a measly fraction (≈1%) of this reward.[10] Revenue from transaction fees are, nevertheless, increasing [10]. With the volume of transactions aggressively growing (as shown in Fig. 1a) over time and the block rewards, in Bitcoin, halving every four years, it is inevitable that transactions fees will be an important, if not the only, criterion for considering a transaction for inclusion. Whether users resort to incentivizing miners, and whether such incentives even work, today, are the focus of this section.

The transaction fee rate (also spelled "feerate") of committed transactions exhibits a wide range (Fig. 4a), from 10^{-6} to beyond 1 BTC/kB. A few transactions (0.001% in $\mathcal{A}$ and 0.07% in $\mathcal{B}$) were committed, despite offering feerates less than the recommended

[8] On June 18th, Facebook announced its cryptocurrency known as Libra: https://libra.org/

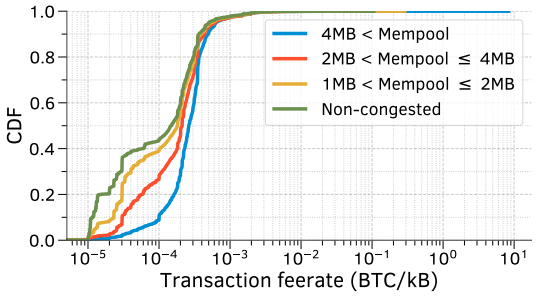
[9] As of May 11$^{\text{th}}$, 2020

[10] Based on analysis of ≈3000 blocks over a three-week period.

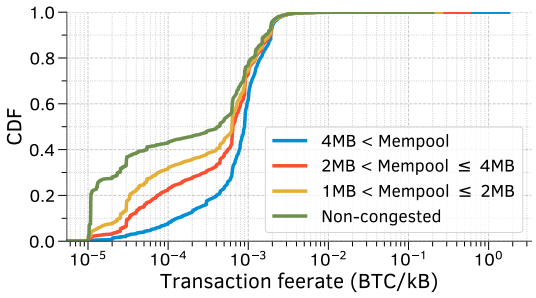
minimum of 10^{-5} BTC/kB. A non-trivial percentage of transactions offered feerates that are two orders of magnitude higher than the recommended value; particularly, in data set $\mathcal{B}$, perhaps due to the comparatively high levels of congestion (refer Fig. 3), 34.7% of transactions offered feerates higher than 10^{-3} BTC/kB. Comparison of the feerates of transactions in $\mathcal{A}$ committed by the top five mining pool operators (in a rank ordering of mining pool operators based on the number of blocks mined), in Fig. 4b, shows no major differences. Approximately 70% (51.3%) of the transactions in $\mathcal{A}$ ($\mathcal{B}$) offer feerates between 10^{-4} and 10^{-3} BTC/kB, i.e., between one and two orders of magnitude more than the recommended minimum.

Perhaps the high feerates observed are proportional to the level of congestion. Stated differently, our hypothesis is that users increase the feerates to overcome the delays introduced by congestion. To test the hypothesis, we marked a Mempool snapshot (refer §4.1) as congested if its Mempool size was larger than the block limit (i.e., 1 MB). Additionally, we marked snapshots with Mempool sizes being two-times (or four-times) the block limit as denoting periods of high (or higher) congestion levels. Fig. 5 plots the feerates of the transactions observed during periods of different congestion levels, and validates our hypothesis: The fraction of transactions with feerates in the range from 10^{-4} and 10^{-3} BTC/kB significantly increase as a function of the level of congestion.

Users' strategy of increasing feerates to combat congestion works well in practice according to Fig. 6. The figure compares and contrasts the CDF of commit delays of transactions with low (i.e., less than 10^{-4} BTC/kB), high (i.e., between 10^{-4} and 10^{-3} BTC/kB), and exhorbitant (i.e., more than 10^{-3}) feerates, in both data sets $\mathcal{A}$ and $\mathcal{B}$. Per Fig. 6, an increase in the transaction feerates is consistently rewarded (by miners) with a decrease in the commit delays. The observations, perhaps, suggest that miners follow the "norm" and prioritize transactions for inclusion based on the fee-per-byte ordering.

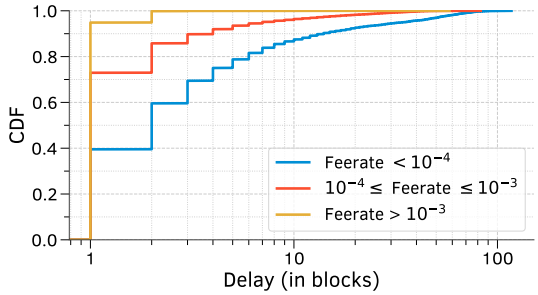


(a) ***Data set*** $\mathcal{A}$

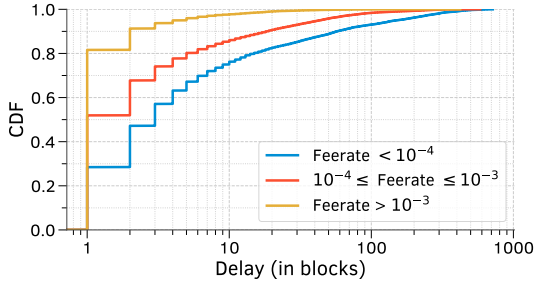


(b) ***Data set*** $\mathcal{B}$

Figure 5: ***Feerates observed in both study periods increase during periods of congestion. The fraction of transactions offering feerates between*** 10^{-4} ***and*** 10^{-3} ***BTC/kB increases proportional to the level of congestion.***



(a) ***Data set*** $\mathcal{A}$



(b) ***Data set*** $\mathcal{B}$

Figure 6: ***Increasing transaction feerates results in decreasing the commit delays.***

Takeaways. A significant fraction of transactions offering feerates that are well above the recommended minimum. Congestions typically explains the high feerates: Users increase feerates of transactions to decrease their commit delays.

5 ON TRANSACTION COMMIT PRIORITIZATION

In a decentralized system, it is vital that all participants follow a "norm," to avoid compromising the stability and fairness of the system. Bitcoin Core [4], the widely used software to access the Bitcoin blockchain history, uses the fee-per-byte metric (as part of the GBT mining protocol [3]), for prioritizing transactions for inclusion. Conventional wisdom, therefore, suggests that miners would be greedy and follow the same strategy as GBT—the "norm." In this section, we check the adherence of miners to this norm.

5.1 On deviations from the norm

If miners do not follow the norm, feerate prediction has serious economic implications for the users. It could become even worse in the following years. To justify our concerns of the miners' behaviors, we checked for transaction pairs that unequivocally show that miners are not completely following the norm. To this end we sampled, uniformly at random, 30 Mempool snapshots (see §4.1) from the set of all available snapshots. Suppose that, in each snapshot, we

denote, for any transaction i, the time at which it was received in the Mempool by t_i, its feerate by f_i, and the block in which it was committed by b_i. We then selected, from each snapshot, all pairs of transactions (i, j) such that $t_i < t_j$ and $f_i > f_j$, but $b_i > b_j$. Each pair essentially has one transaction that appeared earlier and has a higher feerate than the other, but still was committed later than the other; such pairs clearly constitute a violation of the "norm".

Bitcoin supports notions of dependent transactions, involving a parent and child, where the child pays a high fee to incentivize miners to also confirm the parent from which it draws its inputs. This mechanism enables users to "accelerate" a transaction that has been "stuck" because of low fee [8]. As the existence of such *child-pays-for-parent (CPFP)* transactions would introduce false positives in our analysis we decided to discard them. Fig. 7 shows a CDF of the percentage of the number of such transaction pairs (line labelled "$*$") violating the norm across all sampled snapshots. Even if relax the time constraint as $t_i + \epsilon < t_j$ and use an ϵ of either 10 seconds or 10 minutes, there exist (in Fig. 7) a non-trivial number of violations.

5.2 Establishing a baseline

To systematically evaluate how well the fee-per-byte metric explains the dequeuing behavior of miners, we establish a *baseline* as follows. We run a full node and stamp each transaction added to the Mempool with the *chain length*. Chain length represents the number of blocks already present in the blockchain when a given transaction was received in the node's Mempool. For every block B_i mined (in reality, in Bitcoin), we estimate the *candidate* set of transactions that were available to the miner. More concretely, the candidate set of B_i comprises all transactions that were observed in the Mempool before block B_i but have not been confirmed yet. We order the transactions within a candidate set using the fee-per-byte metric (the same adopted on the GBT mining protocol and well believed to be the norm) and create a *baseline* block $\hat{B}_i$ of the same size as that of B_i, i.e., $|B_i| = |\hat{B}_i|$, from the candidate set. To simplify the analyses, we removed child-pays-for-parent transactions prior to creating the baseline block. The number of such transactions dropped out from both the baselines and actual blocks, represents (in the median) 29.6% of the size of the candidate sets.

5.3 Deviations from the baseline

We examined the blocks and transactions in data set $\mathcal{A}$ and estimated the baselines for 3079 actual blocks, observed during this period. The ratio of the size of the intersection between each actual block (B_i) and its corresponding baseline ($\hat{B}_i$), i.e., $|B_i \cap \hat{B}_i|$, to the size of the corresponding B_i (or $\hat{B}_i$) quantifies the extent to which miners adhere to the fee-per-byte dequeuing policy; Fig. 8a plots the CDF of the ratios across all 3079 blocks. In the median, there is a 78% overlap between actual and baseline blocks: The fee-per-byte metric seems, on average, to explain the dequeuing of transactions from the Mempool.

The magnitude of the intersection between the baselines and actual blocks is, however, not 100%! 22% of the transactions in the baselines do not appear in the corresponding actual blocks, i.e., $B_i \setminus \hat{B}_i$; by symmetry, 22% of the actual blocks do not intersect with the corresponding baselines, i.e., $\hat{B}_i \setminus B_i$. Succinctly, 22% of the composition of a block on average deviates from the "norm." There exist, hence, a significant number of transactions whose inclusion (or the lack thereof) in corresponding actual blocks cannot be explained by the GBT like strategy where miners rank transactions based on a fee-per-byte metric.

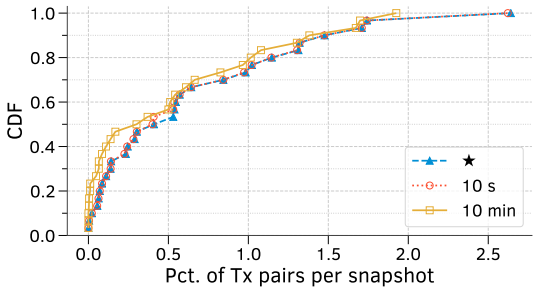


Figure 7: *There exists a non-trivial fraction of Non-CPFP transaction pairs across all snapshots, clearly indicating that miners do not adhere to the norm.*

Could the deviating behavior be attributed to a small subset of miners? Performing the same analysis (of quantifying the overlap between baseline and actual blocks), but only for the blocks mined by the top five[11] mining pools (Fig. 9a) indicates that the pools exhibit almost identical behavior. The CDFs in Fig. 8b are similar to those for the bottom five mining pools as well. The discrepancies between actual and baseline blocks is consistent across all miners, regardless of size: Deviations from the "norm" are consistent across all mining pools (or miners).

6 ON DIFFERENTIAL OBSERVABILITY

In estimating the baseline blocks, we relied *only* on the transactions in our full-node's Mempool. This "view" of the Mempool, however, could be substantially different from that of another node or miner simply because of where that miner is geographically located. Information on transactions introduced or blocks mined in the network disseminates through the peer-to-peer network via a gossip protocol at different speeds depending, at least, on the latency between the nodes in the network. The discrepancies observed, $B_i \setminus \hat{B}_i$ and $\hat{B}_i \setminus B_i$, could, perhaps, be explained by such network delays. The *ignored* transactions[12] in $\hat{B}_i \setminus B_i$, for instance, could have been committed in block B_{i+1} or later instead of B_i, perhaps because the miner received them later than when we observed it. They could also have been committed earlier in block B_{i-1} or earlier instead of B_i, perhaps because our full node observed these transactions later than when the miner observed them. We explore, hence, different scenarios that could explain the discrepancies and ascertain whether to give miners the benefit of doubt—rather than flag their behavior as a violation of the norm—and to what extent.

6.1 Where "our view" is at fault

Let us suppose that the discrepancies between the baseline ($\hat{B}_i$) and actual blocks (B_i) are due to our full node "missing" some of

[11] Based on the number of blocks mined by each pool over the three-week study period.
[12] Ignored by the miner and, hence, missing in the actual block, but included in the baseline.

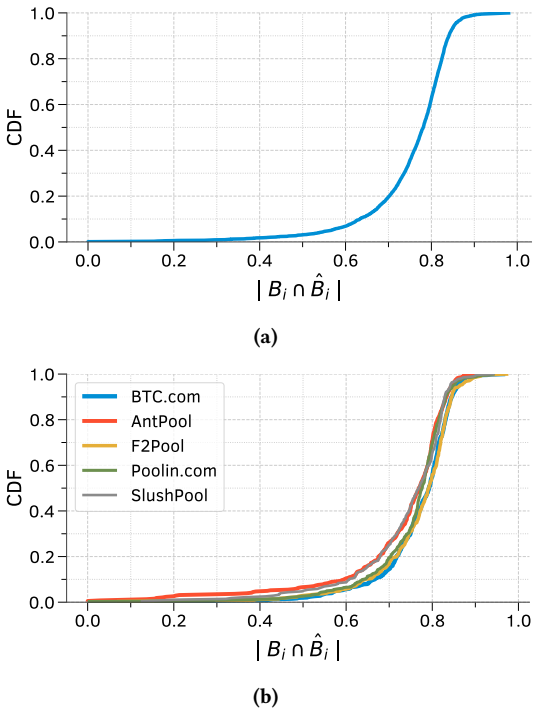


Figure 8: ***(a) In the median,*** 78% ***of the transactions in baselines appear in the corresponding actual blocks, and (b) the observations are consistent across the top-5 mining-pool operators.***

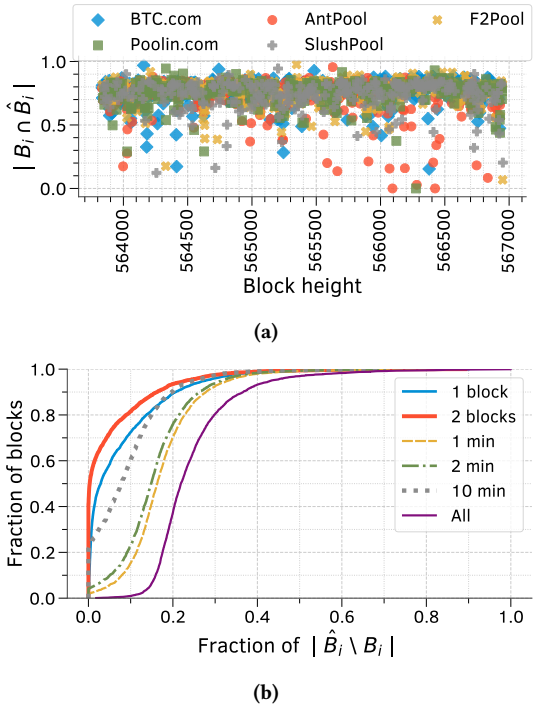


Figure 9: ***(a) All mining pools occasionally deviate from the norm. (b) Even with a 2-block cutoff period, miners ignore some transactions half the time***

the transactions observed by the miners. We now examine to what extent this premise holds true.

Were miners privy to certain transactions? Of the set of transactions in $B_i \setminus \hat{B}_i$, we measured the fraction that our full node *never* observed in its Mempool. In nearly 80% of the 3079 blocks, the full node does not miss even one transaction; stated differently, transactions in $B_i \setminus \hat{B}_i$ were observed, in most cases, at some point in time in our Mempool (and were included in some baseline block, but not $\hat{B}_i$). Even in the 99-th percentile, the full node fails to observe fewer than 10% of the transactions in $B_i \setminus \hat{B}_i$. These small number of cases could be explained by network delays, resulting in some transactions being received after "their" blocks: Transactions received after the block, in which they were included, causes the full node to drop the transaction (silently) even before adding them to the Mempool. *Even if miners were being privy to certain transactions, the small fraction of transactions our full node "misses" to observe cannot explain the large discrepancies.*

Did we miss transactions with high fees? Recall that in computing the baselines we used the fee-per-byte metric to prioritize transactions. If our full node missed some transactions with higher fee-per-byte values than the minimum across all transactions in a given baseline, these missed transactions will explain some of the discrepancies. Across the 3079 blocks in our data set, we observe, however, only 1% of the transactions in $\hat{B}_i \setminus B_i$ (i.e., observed in the baseline but not in the actual) to have higher fee-per-byte values than the minimum across all transactions in B_i. *Therefore, of the* 22% *discrepancy (i.e.,* $\frac{|\hat{B}_i \setminus B_i|}{|B_i|}$*) we observe, only* 1% *are perhaps because of our full node missing some transactions with high fee rates.*

In summary, the transactions that our full node either never observes or fails to observe "on time" explain, at best, a percent of those ignored (or absent) in $B_i \cap \hat{B}_i$.

6.2 Where "their view" is at fault

We are now left with only one premise to verify: Perhaps it is the miners who do not observe some of the transactions "on time".

The "All" line in Fig. 9b shows, for each block, the number of (ignored) transactions in baseline but not in the actual block (i.e., $\hat{B}_i \setminus B_i$) as a fraction of that in the actual block. For this line to be true, every node in the peer-to-peer network should have observed the transactions at the same time (i.e., with zero delays), which is infeasible. To account for delays, we remove from this set of ignored transactions, those we received within some *cutoff period*, e.g., one minute, before a given block is mined. More concretely, if a transaction t_i belongs to $\hat{B}_i \setminus B_i$, but the time at which we received this transaction was within the cutoff period before we received the actual block B_i, we drop it from the ignored set. Per Fig. 9b, the "1 min." line, corresponding to a 1-minute cutoff period, significantly reduces the fraction of ignored transactions: Fraction of ignored transactions drops (from 22% in "All") to 12% or less; 2% of the blocks (compared with *none* in "All") also have no ignored transactions. Increasing the cutoff period to 2 minutes further reduces the discrepancies, in favor of the miners.

The cutoff period accounts for the scenario that perhaps the miners received the transactions "later" than our full node. It is unlikely, however, that the mining pools would experience a delay, as high as, one minute: It is in the best interests (economically speaking) of the mining-pools to equip their infrastructure with low-latency network connections, after having spent millions in hardware [29]; the fixed infrastructure costs alone should reduce the cost of providing low-latency Internet connectivity (e.g., [13] and [2]) to their nodes. The economic argument notwithstanding, even with a 2-minute cutoff period, 50% of the blocks ignore nearly 10% of the transactions.

Rather than use absolute time spans for the cutoff period we also used block-based cutoff periods. A one-block cutoff period implies that we drop any transaction in the ignored set if it was received anytime before the current block B_i (where it is being flagged as ignored) but after we received the prior block B_{i-1}. Recall that block generation times, in Bitcoin, vary with an average of about 10 minutes. Even with a two-block cutoff period (i.e., 20 minutes on average), we observe, in Fig. 9b, 50% of the blocks to have at least some ignored transactions!

The analyses using (absolute) time-based as well as block-based cutoff periods indicate that a significant fraction of transactions is being ignored from immediate inclusion, for whatever reason, by miners.

7 DISCUSSION

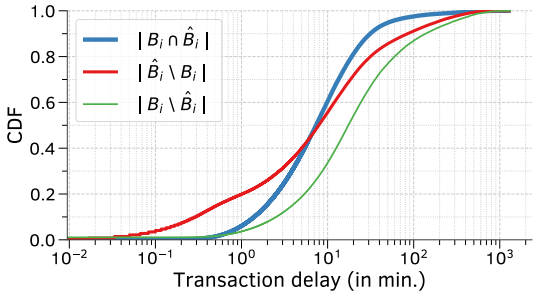
We now present a few conjectures, backed by empirical observations, that could explain the miners' behavior and its implications for the Bitcoin—or more generally the blockchain—ecosystem.

7.1 Musings on miners' behavior

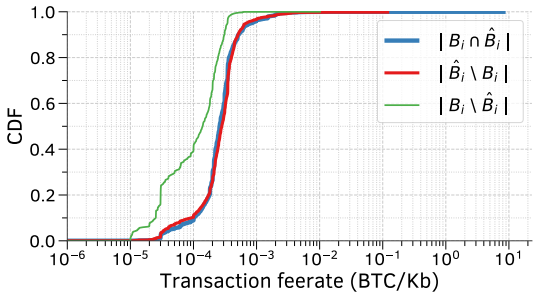
For each transaction, regardless of whether it was present in both the baseline and actual blocks ($B_i \cap \hat{B}_i$) or only in the actual ($B_i \setminus \hat{B}_i$) or just in the baseline ($\hat{B}_i \setminus B_i$), we compute the transaction delay as the difference between when the transaction was received in Mempool and when it was included in the baseline or actual, depending on to which of the three aforementioned categories the transaction belongs. Fig. 10a shows the CDF of the delays for all transactions, separately for each of the three categories.

Per Fig. 10a, some of the transactions that the miners included (in the actual block) that are not in the baseline (i.e., the category $B_i \setminus \hat{B}_i$) arrived several minutes later than those that appear in both the actual and baseline blocks (i.e., the category $B_i \cap \hat{B}_i$). Some of the transactions in the baseline but not in the actual block (i.e., category $\hat{B}_i \setminus B_i$), arrived much earlier compared to those in the $B_i \setminus \hat{B}_i$. *Perhaps the miners are using a different protocol, which takes other parameters, in addition to fee-per-byte, into consideration.*

Perhaps the miners are being "altruistic". Said differently, miners might be committing transactions that have been waiting in the Mempool for a "long" time, despite those transactions having comparatively lower fees—the CDF of $|B_i \setminus \hat{B}_i|$ in Fig. 10a lends some credence to this line of reasoning. Fig. 10b, which is similar to Fig. 10a except that the x-axis is transaction fee rate instead of delays, also shows that the transactions prioritized by miners have comparatively lower fee rates. The observation that 60% or more of the transactions that have been waiting for 10 minutes or longer



(a)



(b)

Figure 10: ***(a) It is unlikely that the miners are ignoring transactions because these arrived too close to when the block was mined: Some, if not all, of the transactions in $B_i \setminus \hat{B}_i$ arrived minutes later than those in $B_i \cap \hat{B}_i$. (b) Transactions in $B_i \setminus \hat{B}_i$ have significantly lower fee-per-byte compared to those in $B_i \cap \hat{B}_i$, suggesting that fee-per-byte does not completely explain miners' dequeuing policies.***

(in the CDF of $|\hat{B}_i \setminus B_i|$ in Fig. 10a) were ignored by the miners, however, clearly refutes the claim of an "altruistic" behavior.

Are miners adopting other strategies? We hypothesize that mining pool operators might be sending a different set of transactions to each miner (or perhaps changing this set every a fixed time interval or event) to reduce the network overhead, avoiding miners to request a new task often. It is possible due to the stratum protocol[13]. Stratum is a pooled mining protocol that focuses on reducing network communication between the mining pool and its miners by allowing miners to change some bytes on the coinbase transaction and consequently changing the Merkle root. Another reason is that some clients might be using services like transaction accelerators to speed up the commit time of a particular transaction. They pay the mining pool to use this service off-chain (i.e., with another cryptocurrency or via credit-cards) to, hopefully, increase the probability of their transactions get included in the next block. One example of this service is the *BTC.com transaction accelerator*[14].

Implications. Regardless of whether the miners are altruistic, Fig. 10b strongly suggests that the dequeuing policy is not simply a function of the fee-per-byte metric. The transactions in $B_i \setminus \hat{B}_i$ have, for instance, significantly smaller fees than those available in

[13] Stratum is a pooled mining protocol available at https://stratumprotocol.org
[14] *BTC.com* is one of the biggest Bitcoin mining pools operators currently available. Its transaction accelerator service is available at: https://pushtx.btc.com/

$B_i \cap \hat{B}_i$. Further, the $B_i \cap \hat{B}_i$ and $B_i \setminus \hat{B}_i$ lines in Fig. 10b suggest that even if users pay a fee significantly higher—one or two orders of magnitude higher—than the lowest fee (10^{-5} BTC/kB), there is virtually no guarantee that their transaction will be included in the next block. Only beyond an exorbitant fee rate (10^{-1} BTC/kB) there is, unsurprisingly, a guarantee that the concerned transaction will be immediately committed. Today, virtually all of the fee predictors, however, falsely assume that miners follow the fee-per-byte metric for prioritizing transactions for inclusion.

8 CONCLUSION

In selecting transactions for inclusion in a block, miners somehow deviate from the conventional wisdom or the norm, which dictates that transactions are prioritized for inclusion based on the fee-per-byte metric. This deviation is consistent across all miners (or mining pools), regardless of size, and becomes more pronounced during periods of congestion.

While our inferences are only with reference to the Bitcoin network, we believe the incentives for a miner to deviate from the "norm", especially during periods of congestion, most likely exist in other proof-of-work-based blockchains. Especially given the lack of any notion of fairness of transaction ordering in such systems, miners at least seem to lack strong incentives to follow the norm. We plan to investigate other cryptocurrencies and blockchain-based implementations as part of future work. In the meantime, we hope that this paper serves as an incentive for researchers to investigate mechanisms that will allow any observer to validate if the miners are adhering to the norm.

ACKNOWLEDGMENTS

This research was supported in part by a European Research Council (ERC) Advanced Grant for the project "Foundations for Fair Social Computing", funded under the European Union's Horizon 2020 Framework Programme (grant agreement no. 789373).

REFERENCES

[1] Soumya Basu, David Easley, Maureen O'Hara, and Emin Gün Sirer. 2019. Towards a Functional Fee Market for Cryptocurrencies. *CoRR* abs/1901.06830 (2019).
[2] Soumya Basu, Ittay Eyal, and Emin Gün Sirer. 2016. Falcon Network: A High-Performance, Wide Area Interconnect. https://www.falcon-net.org/papers/falcon-retreat-2016-05-17.pdf.
[3] Bitcoin Wiki. 2019. getblocktemplate. https://en.bitcoin.it/wiki/Getblocktemplate.
[4] bitcoin.org. 2019. Bitcoin Core. https://bitcoin.org/en/bitcoin-core.
[5] blockchain.com. 2019. Bitcoin Block #0 (Genesis block). https://tinyurl.com/genesis-block.
[6] Coin Dance. 2019. Bitcoin Nodes Summary. https://coin.dance/nodes.
[7] CoinMarketCap. 2020. Cryptocurrency Market Capitalizations: Top 100 Cryptocurrencies. https://coinmarketcap.com/.
[8] CoinStaker. 2018. Bitcoin CPFP Experience—Bitcoin Child Pays for Parent. https://www.coinstaker.com/bitcoin-cpfp/.
[9] DemocracyEarth. 2019. On decentralized digital democracy. https://github.com/DemocracyEarth/paper.
[10] David Easley, Maureen O'Hara, and Soumya Basu. 2017. From Mining to Markets: The Evolution of Bitcoin Transaction Fees. *SSRN* (2017).
[11] Ariel Ekblaw and Asaf Azaria. 2017. MedRec: Medical Data Management on the Blockchain. *Viral Communications* (12 2017).
[12] Ittay Eyal and Emin Gün Sirer. 2018. Majority is Not Enough: Bitcoin Mining is Vulnerable. *Commun. ACM* 61, 7 (June 2018), 95–102.
[13] FIBRE. 2019. Fast Internet Bitcoin Relay Engine (FIBRE). http://bitcoinfibre.org.
[14] Follow My Vote. 2016. Blockchain Voting: The End To End Process. https://followmyvote.com/blockchain-voting-the-end-to-end-process/.
[15] Adem Efe Gencer, Soumya Basu, Ittay Eyal, Robbert van Renesse, and Emin Gün Sirer. 2018. Decentralization in Bitcoin and Ethereum Networks. In *Financial Cryptography and Data Security 2018.*
[16] Ghassan Karame. 2016. On the Security and Scalability of Bitcoin's Blockchain. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16).* ACM, New York, NY, USA, 1861–1862.
[17] Aleksander Kuzmanovic. 2019. Net Neutrality: Unexpected Solution to Blockchain Scaling. *Queue* 17, 1 (Feb. 2019), 20:20–20:78.
[18] Ron Lavi, Or Sattath, and Aviv Zohar. 2019. Redesigning Bitcoin's Fee Market. In *The World Wide Web Conference (WWW '19).* ACM, New York, NY, USA, 2950–2956.
[19] Eric Lombrozo, Johnson Lau, and Pieter Wuille. 2015. BIP-141: Segregated Witness (Consensus layer). https://github.com/bitcoin/bips/blob/master/bip-0141.mediawiki.
[20] Martin Ruubel. 2016. Blockchain-Enabled Cloud: Estonian Government selects Ericsson, Apcera and Guardtime. https://guardtime.com/blog/blockchain-enabled-cloud-estonian-government-selects-ericsson-apcera-and-guardtime.
[21] Martin Ruubel. 2018. World's First Blockchain Platform for Marine Insurance Now in Commercial Use. https://guardtime.com/blog/world-s-first-blockchain-platform-for-marine-insurance-now-in-commercial-use.
[22] Matt Corallo. 2017. Bitcoin Relay Network. http://bitcoinrelaynetwork.org.
[23] Satoshi Nakamoto. 2008. Bitcoin: A Peer-to-Peer Electronic Cash System. https://bitcoin.org/bitcoin.pdf.
[24] Paul R. La Monica. 2019. Bitcoin's march to $10,000 propelled by Facebook and the Fed. https://edition.cnn.com/2019/06/21/investing/bitcoin-price-increase/.
[25] Philipp Schmidt. 2015. Certificates, Reputation, and the Blockchain. https://medium.com/mit-media-lab/certificates-reputation-and-the-blockchain-aee03622426f.
[26] Provenance. 2015. Blockchain: the solution for transparency in product supply chains. https://www.provenance.org/whitepaper.
[27] Robert Hackett. 2017. Walmart and 9 Food Giants Team Up on IBM Blockchain Plans. http://fortune.com/2017/08/22/walmart-blockchain-ibm-food-nestle-unilever-tyson-dole/.
[28] Marie Vasek, Micah Thornton, and Tyler Moore. 2014. Empirical Analysis of Denial-of-Service Attacks in the Bitcoin Ecosystem. In *Financial Cryptography and Data Security 2018 (Lecture Notes in Computer Science, Vol. 8438).*
[29] Wolfie Zhao. 2019. Bitmain Set to Deploy $80 Million Worth of Bitcoin Miners, Sources Say. https://www.coindesk.com/bitmain-bitcoin-mining-farms-antminer.