# Characterizing the Impact of Network Delay on Bitcoin Mining

Tong Cao*, Jérémie Decouchant†, Jiangshan Yu‡, Paulo Esteves-Verissimo§
*SnT - University of Luxembourg, †Delft University of Technology, ‡Monash University, §RC3 - KAUST

***Abstract*—While previous works have discussed the network delay upper bound that guarantees the consistency of Nakamoto consensus, measuring the actual network latencies and evaluating their impact on miners/pools in Bitcoin remain open questions. This paper fills this gap by: (1) defining metrics that quantify the impact of network latency on the mining network; (2) developing a tool, named miner entanglement (ME), to experimentally evaluate these metrics with a focus on the network latency of the top mining pools; and (3) quantifying the impact of the current network delays on Bitcoin's mining network. For example, we evaluated that Poolin, a Bitcoin mining pool, was able to gain between 0.5% and 1.9% of blocks in addition (i.e., from 36.27 BTC to 137.83 BTC) per week thanks to its low network latency. Moreover, as pools are rational in Bitcoin, we model the strategy a pool would follow to improve its network latency (e.g., by leveraging our ME tool) as a two party game. We show that a Bitcoin mining pool could improve its effective hash rate by up to 4.5%. For a multi-party game, we use a state-of-the-art Bitcoin mining simulator to study the situation where all pools attempt to improve their network latency and show that the largest mining pools would improve their revenue and reach a Nash equilibrium while the smaller mining pools would suffer from a decreased access to the network, and therefore a decreased revenue. These conclusions further incentivize the centralisation of the mining network in Bitcoin, and provide an empirical explanation for the observed tendency of pools to design and rely on low latency private networks.**

## I. Introduction

Bitcoin miners are incentivized to compete with each other to solve a cryptographic challenge by generating hashes. Upon solving the challenge, a miner generates a new block of transactions, propagates it to all miners using the peer-to-peer (P2P) network, and collects a reward. Under these settings, the longest proof of work chain is eventually accepted by all participants despite the network's asynchrony, so that Nakamoto Consensus (NC) can be achieved [1]. The original Bitcoin paper evaluated that an adversary would require more than 50% of the global computing power to attack the system, i.e., to double spend a coin. Since then, several works have shown that Bitcoin's security bounds decrease when the adversary has a relatively small network delay [2–6]. Specifically, the maximum network delay of honest participants is used as the main parameter to evaluate the attack bounds [7, 8], chain quality [5, 9] and consistency [5, 6] in Bitcoin. However, measuring and evaluating the actual delays of miners/pools and their impact on the mining process remained an open challenge [3].

The Bitcoin mining success rate depends not only on the mining power of a miner but also on its network latency. In this paper, we measure the network latencies of major mining pools and evaluate their impact on the mining process, and in particular, on the success rate of mining. Intuitively, a miner that is connected to other peers with a high latency has less time to try to solve the cryptographic challenge than others, and therefore earns less reward than it could expect.

In this paper, we investigate and quantify the impact of network latency on Bitcoin mining. First, we define new metrics to quantify the impact of network latency on mining. In particular, we define the effectiveness ratio to capture the impact of network latency on mining efficiency for individual miners/pools, by considering the length of time to receive a block and to generate a block. We define the effective hash rate share to quantify the competitiveness of miners/pools on the mining race.

Second, to quantify the impact on the current Bitcoin mining network, we develop a tool, named miner entanglement (ME), to measure the block reception latencies of major mining pools in Bitcoin. We infer the revenue changes of pools based on our metrics through Monte Carlo simulation. We evaluated that Poolin had the smallest network delay among major mining pools and was able to obtain between 0.5% and 1.9% of blocks in addition (which represent between 36.27 BTC and 137.83 BTC) per week depending on the performance of other pools. On the opposite, the anonymous pools who applied the default P2P protocol suffered from a degraded network access, and collected less than their fair share of the revenue. As a side observation, we show that pools can leverage ME to reduce their block reception latency. By applying our method, miners can increase their effective hash rate and revenue. We show that an individual Bitcoin miner can improve its effective hash rate from 95.2% to 99.7%.

Last, as miners are rational, it might happen that they attempt to compete with each other to optimize their network latency to gain more mining revenue. We show using a state-of-the-art mining network simulator [8] that this would lead to a Nash equilibrium among the most powerful miners, while the whole network's fairness would decrease. In other words, a small set of large mining pools would increase their revenue, while other miners would have a decreased revenue.

To summarize, this work makes the following contributions.

1) We establish several metrics to characterize the impact of heterogeneous network latencies on the mining process.
2) We design ME, a tool that leverages Bitcoin mining pools' API to estimate their block reception latency. We use ME to monitor the Bitcoin network for an entire

week, and quantify the impact of network latency on the revenue distribution using our metrics.

3) We discuss the situation where miners would decide to deviate from the official peer-to-peer protocol and try to optimize their connections and their revenue by connecting directly to other mining pools. We show that it would lead to a Nash equilibrium among the largest miners, while other miners will have a decreased revenue.

The rest of this paper is structured as follows. Section II discusses related work. Section III provides some background knowledge on PoW cryptocurrencies. Section IV defines metrics to quantify the impact of network latencies on the mining process. Section V describes how we monitored the Bitcoin network during one week using the ME tool, and evaluates the impact of network latencies. Section VI discusses the impact of network latency on a network that would consist of pools that would aim at optimizing their block reception latencies. Finally, Section VII concludes this paper.

## II. Related Work

**Network measurement.** Several tools have been designed to measure the block propagation, transaction propagation, and network coverage in the Bitcoin network [2, 10, 11]. Previous works measured the block propagation latency as the maximum block reception latency between miners [4–6], or the time required for 50% or 90% of the nodes to receive the blocks [12, 13]. Some of these works [2, 3, 11] evaluated that a block needed in average 8.7 seconds to reach 90% of the nodes before 2014. According to the Bitcoin network monitor [13], this delay has since been improved to less than 5 seconds. However, no mechanism has been implemented to make sure that all nodes have a fair access to new blocks. Pool Detective monitors 32 pools across 17 cryptocurrencies [14]. However, it does not quantify the impact of network delay on the mining process as we do in this paper.

Close to our work, Alzayat et al. build a model to evaluate the impact of network delay on miners' efficiency in Ethereum [15]. The efficiency of a mining pool is calculated based on the number of uncle blocks that it generated. The reported efficiencies of the top 5 Ethereum mining pools range from 88.68% to 93.58%. While we consider the Bitcoin network, we believe that the methods we present in this work could be useful to analyze the Ethereum network.

**Mining fairness and chain quality.** Mining fairness was cited as an important property of PoW cryptocurrencies [3, 16]. In a perfectly fair network, a miner having $x\%$ of the global mining power would earn $x\%$ of the blocks. However, existing mechanisms aiming to provide this property do not consider the impact of the P2P network. The chain quality was defined as the fraction of the main chain blocks that is mined by honest participants [4, 9]. In consequence, when the adversary controls a proportion $\beta$ of the global computing power, the system is expected to have an ideal chain quality equal to $1-\beta$. Previous works [17, 18] showed that the temporary block withholding strategies could harm chain quality. Pass et al. analyzed the impact of network delay on chain quality [5], and proved that the upper bound of chain quality is usually below the ideal chain quality in asynchronous networks.

**Fast relay network.** The purpose of the fast relay network [19] is to maintain full Bitcoin nodes that miners and pools can connect to. These nodes use unsolicited relay patterns to broadcast a new block. In this way, miners are able to speed up the blocks propagation in the network. However, this method leads to a potential issue of centralization. So far, not all mining pools have adopted this technology due to a concern of having their network access controlled by the fast relay network.[1] Our work justifies the appearance of private networks by pointing out limitations of the default mining network.

**Game theory.** Game theory has been used to evaluate cryptocurrencies' network performance and property. For example, it has been shown that a Nash equilibrium can be achieved when two pools, or any number of pools use the block withholding attack [20–22]. Previous works [23, 24] provide a game-theoretic model to analyze the impacts of network attacks (such as DDoS) in mining pools. Another example concerns the network creation game, which has been introduced in the lightning payment channel [25]. It was shown that the Bitcoin payment network can be more stable and efficient in a centralized network structure. In our paper, we are interested in how nodes select their neighbors to optimize their effective hash rate, and consequently their expected mining revenue.

## III. Background

In this section, we recall the basic concepts of Bitcoin mining, which explain why the generation times of blocks fluctuates. We then introduce the core concepts behind PoW cryptocurrency networks, which disseminate blocks to all miners with variable delays.

### A. Mining Process

Mining is a trial-and-error process. From a candidate set of transactions, miners assemble a block and use their processors to identify a hash that is smaller than a target value[2]. Upon finding such a hash, a miner has successfully created a block.

**Mining as a Poisson process.** In practice, the discovery rate of blocks is not constant. Many works have captured this variation by modeling the PoW mining process as a Poisson process [1, 2, 26, 27], where the success rate $\lambda$ corresponds to a block being generated every 10 minutes in average. The probability density function of a Poisson process is $P(X = x) = \frac{\lambda^x e^{-\lambda}}{x!}$.

**Block interval distribution.** By modeling Bitcoin mining as a Poisson process, the block generation interval follows an exponential distribution [1, 2], whose probability density function is $G(t) = \lambda e^{-\lambda t}$, where $t$ denotes the block interval between two adjacent blocks and $\lambda$ is the average success rate. We present in Section V the empirical distribution of the Block receptions we observed during our experiments.

[1]https://bitcoinfibre.org/public-network.html

[2]https://en.bitcoin.it/wiki/Block_hashing_algorithm

## B. PoW Cryptocurrency Network

A PoW cryptocurrency network consists of a P2P overlay that interconnects solo miners and mining pools. This network disseminates the newly found blocks to all miners. To clarify the difference between the P2P overlay and the intra network of mining pools, we use mining pools and miners to denote the nodes of the P2P overlay, and sub-miners to denote the miners located within the mining pools.

**P2P overlay network.** The P2P overlay network mainly consists of full nodes[3] and client nodes. Each node maintains a peer list and periodically exchanges information with other peers to keep it up-to-date and to disseminate blocks and transactions. The solo miners could also be full nodes. The mining pools normally maintain some full nodes in the P2P overlay as the front end to exchange messages with other pools/miners. The main properties of cryptocurrency P2P networks, such as their network size, node degree distribution, and connectivity, have been widely studied [2, 10, 11, 28, 29]. Existing measurements (as we have introduced in Section II) rely on the P2P overlay network to evaluate the block propagation delay, which cannot reflect the delays of pools, mainly because of the unknown IP addresses of pools' front end nodes and network churn.

**Intra network of mining pools.** As the mining difficulty increases, the revenue of miners that have a small hash power becomes more irregular. To compensate this effect, miners can join mining pools. The block rewards that are collectively earned by the members of a mining pool are shared among them according to their participation. The internal networks of mining pool usually work as client-server infrastructures. A mining pool uses some dedicated servers to manage its sub-miners. The sub-miners are only connected with those servers who assign them jobs and inform them of the new blocks. When a sub-miner has finished a job, it sends its result to the servers. The mining pools also maintain front-end nodes in the P2P network to exchange messages with other miners and pools. The most popular protocol used by mining pools is Stratum [30].

## IV. Quantifying the Impact of Network Delay on Mining

This section defines metrics to evaluate the impact of network latency on mining.

### A. Mining and latencies

The mining process of a miner can be decomposed in three successive phases, among which the first and the last are affected by the miner's access to the network. We illustrate those phases in Fig. 1.

1. **New block reception.** Once a block has been discovered by a miner in the network, it is transmitted to the whole network. The delay between the discovery of a block and its reception by a miner is the block reception latency of this miner on this block. In Fig. 1, $d_{BR}(3,1)$ is the block reception latency of miner 3 on block 0, whose reception allows it to work on block 1. The sooner a miner receives a block the sooner it can start attempting to solve the cryptographic puzzle.
2. **Cryptographic puzzle solving.** Based on the new block, miners compute hashes to try to discover the next block, by attempting to solve the mining cryptographic puzzle. During this phase, which lasts until the next block is found in the network, miners are doing effective work. The number of hashes they can generate is the product of their computing power, expressed in hashes per second, and the time they dedicate to mining the current block. In Fig. 1, $d_{BI}(1) - d_{BR}(3,1)$ is the time of effective work of miner 3 on block 1, and $d_{BI}(1) - d_{BR}(5,1)$ is the time of effective work of miner 5 on block 1. In this example $d_{BR}(3,1) < d_{BR}(5,1)$, which means that miner 3 has worked more effectively than miner 5 on block 1.
3. **Next block dissemination.** After a new block has been discovered, it has to be disseminated to the nodes who collectively account for more than 50% of the global hash power to be validated. The position of a node in the network influences the probability with which a new block is validated when several blocks are simultaneously discovered.

[3]https://bitnodes.io/

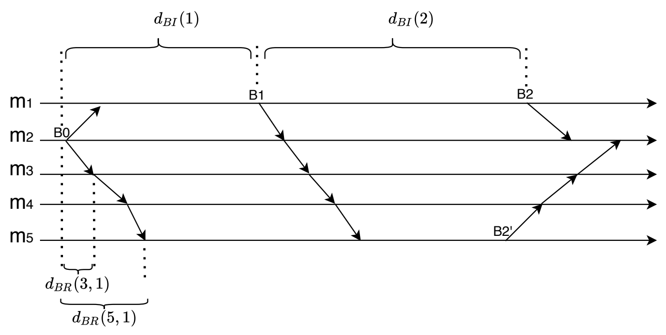

Figure 1: Network influence on the mining process in PoW-based cryptocurrencies. $d_{BR}$ represents the block reception latency, and $d_{BI}$ denotes the block interval.

### B. Block reception latency and effective hash rate

Miners might receive a newly found block at different times, which might lead those that receive a block after the others to waste hashing power on an already solved cryptographic puzzle. Because of this effect, two miners with equal hash power might consistently earn different revenue. The effect of the network latency on mining is particularly acute when a block is discovered in a short time, which happens in practice as captured by the Poisson distribution of mining process.

Obtaining the hash rate distribution in the network is difficult, as miners and pools do not reveal their real-time hash power and because the network is under constant evolution. It is therefore challenging to evaluate the impact of the mining network on miners based on the hash rate distribution and on statistics about block discoveries. We instead define the effectiveness ratio metric to evaluate the effectiveness of an individual miner independently of the global hash rate distribution as follows.

We consider a (snapshot of a) mining network of $N$ miners. Let $H_i$ be the hash rate of miner $i \in [1, N]$, $d_{BI}(j)$ be the block generation interval between the $(j-1)^{th}$ block and the $j^{th}$ block (i.e., the length of time between the generation of the $(j-1)^{th}$ and $j^{th}$ block), and $d_{BR}(i,j)$ be the block latency of receiving the $(j-1)^{th}$ block at miner $i$.

**Effectiveness ratio.** The effectiveness ratio $f_{i,j}$ of miner $i$ on the $j^{th}$ block is $\left(1 - \frac{d_{BR}(i,j)}{d_{BI}(j)}\right) \in [0, 1]$.

The effectiveness ratio of a miner on a block is the proportion of time it can compute hashes during the associated block interval. An effectiveness ratio close to 1 indicates that miner $i$ is well positioned in the network. This formula captures the fact that a mining pool that quickly receives a block and starts early to mine the next block has an advantage over the other pools, which receive the same block later. Moreover, since $d_{BI}$ follows an exponential distribution (the probability density function $G_0(t) = \lambda \times e^{-\lambda \times t}$, where $\lambda \approx \frac{1}{600}$ in Bitcoin) [1, 2], the effectiveness ratio of a miner differs depending on the blocks. For instance, even though the expected $d_{BI}$ in Bitcoin is 600 seconds, 1.65% ($G_0(10)$) of the blocks are such that $d_{BI} < 10$ seconds, 8.0% ($G_0(50)$) are such that $d_{BI} < 50$ seconds, and 15.3% ($G_0(100)$) are such that $d_{BI} < 100$ seconds. A few seconds delay in the reception of a block can significantly affect the effective hash rate of miners in some block rounds, and drift their revenue share.

We capture the impact of heterogeneous delays on the whole network over the $j^{th}$ block using a 2-tuple $(G_j, D_j)$, where $G_j \in [0, 1]$ is the Gini coefficient $gini(f_{1,j}, f_{2,j}, ..., f_{n,j})$ [31] and $D_j \in [0, 1]$ is the difference between the highest and the lowest effectiveness ratio observed among miners on block $j$. A small $G_j$ indicates that the miners have in average similar effectiveness ratios over the $j^{th}$ block, while a larger $G_j$ shows unfairness. $D$ indicates the amplitude of the distribution of effectiveness ratios. Note that the Gini coefficient alone describes the degree of inequality of a distribution, and that we use $D$ to provide additional information.

We define the effective hash rate (EHR) using the effectiveness ratio, as follows.

**EHR.** The effective hash rate $EHR(i,j)$ of miner $i$ on the $j^{th}$ block is equal to $H_i \left(1 - \frac{d_{BR}(i,j)}{d_{BI}(j)}\right) \in [0, H_i]$.

The effective hash rate $EHR(i,j)$ of a miner $i$ corresponds to the number of hashes it is able to compute after having received block $j-1$ and before block $j$ is discovered.

### C. Impact of heterogeneous network delays on revenue

The competitiveness of a miner or a pool in the mining race not only depends on its hash rate, but also on its network delay. In the following, we detail how the block revenue of a miner is influenced by the heterogeneous delays.

**HR Share.** The hash rate share $HR\ Share_i$ of miner $i$ is $\frac{H_i}{\sum_{k=1}^{N} H_k} \in [0, 1]$.

The HR Share of miner depends on its real hash rate and on the global hash rate. Without considering the impact of network delay, the mining success rate of a miner is equal to its HR Share. However, we take the impact of network delay into account to evaluate a miner's success rate and define the EHR Share as follows.

**EHR Share.** The effective hash rate share $EHR\ Share_i$ of miner $i$ is $\frac{H_i \times f_i}{\sum_{k=1}^{N}(H_k \times f_k)} \in [0, 1]$.

The revenue of a miner is determined by its EHR share, which in turn depends on its hash rate and on the network delays. We use this metric to further evaluate the impact of network delays on the revenue distribution.

## V. Measurement and Evaluation

To evaluate the impact of network latency on mining, we develop a tool called Miner Entanglement (ME) to monitor the block reception latency of mining pools. ME leverages the mining pools' API to measure the time it takes for the mining pools to learn about newly discovered blocks. We deploy Bitcoin nodes running ME and quantify the impact of network delays with the metrics that we defined in Section IV.

### A. Leveraging the API of mining pools

Miner Entanglement (ME) uses BFGminer [32] to build direct TCP connections between a local machine and mining pools. More specifically, we deploy ME in our local machine, which is registered as sub-miners in several mining pools. Since the pools recognize a ME-empowered node as a sub-miner, they directly inform the ME-empowered node with the information about new blocks. This enables us to estimate the block reception delays of the mining pools. We connected to 10 Bitcoin mining pools, which we list in Appendix A. Collectively, these pools own approximately 69.88% of the global computing power. We could not establish connections with the pools that own the remaining 30.12% of the global computing power because they either do not accept non-ASIC devices, or because they could not be identified.

We illustrate the design of ME in Figure 2 where Pool D represents our local machine, and where we deploy sub-miners in pools A, B, and C to receive block discovery information directly via the ME channels (which are established directly between our machine and the pools). We also run a full Bitcoin node on our machine using the default P2P protocol to evaluate the latency of a normal Bitcoin full node via the P2P channel (which is randomly built between peers) as a comparison. We use an Ubuntu 16.04 desktop with an Intel Core i7-7700 processor.

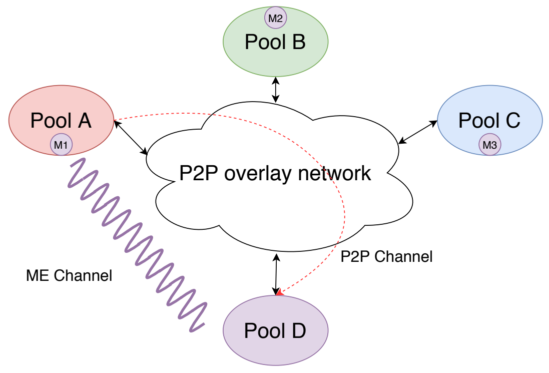

Figure 2: Illustration of the miner entanglement design. Pool D registers three sub-miners $M_1$, $M_2$, and $M_3$ in Pool A, Pool B, and Pool C respectively. This allows Pool D to receive the information from other pools directly, thus, avoiding the delay of the P2P overlay.

We run ME for an entire week and have collected discovery notifications related to 1,116 blocks (from block 641,767 to 642,882), which represented 68.1 MB of data overall. From this dataset, we report raw data such as the block intervals and the block reception delays. We then evaluate the impact of network delays on Bitcoin mining by using our metrics.

## B. Discussion

For each pool we managed to use ME with, we use half of the round trip time (as which has been used in other works [11, 33]) between our machine and the mining pool, i.e., $\frac{RTT}{2}$, to estimate the time needed for a mining pool to receive a block and start sending it to its sub-miners. We obtain the round trip time with the ping command, which uses ICMP packets. We calculate the block sending time of the target pool using the block reception time of our sub-miner minus the block propagation delay between the pool and the sub-miner. We continuously monitored the RTTs between 10 pools and our sub-miners. These RTTs were stable during a week as indicated in Appendix A. Moreover, we assume that the pools relay new blocks to their sub-miners as soon as they learn about them. This assumption is reasonable if mining pools are seen as rational entities aiming to increase their revenue.

Previous approaches [2, 3, 11, 13] deployed multiple geo-graphically distributed collectors in the P2P overlay to infer block propagation delays, partly because the network latencies between the observer and mining pools were unknown. We use ME to build direct connections with mining pools, which does not suffer from transmission delays possibly incurred by the P2P overlay. In our case, one collector is sufficient to measure the block reception latencies between mining pools. Deploying multiple collectors would help to further reduce the noise that comes from RTTs variation, which is nonetheless already negligible. The data we collected might have been different if the experiment had been conducted over a longer period, e.g., because of network churn. However, our conclusions on the network impact on mining would remain valid.

The effectiveness ratio of the pools would be affected when concurrent blocks are discovered simultaneously. In this case, the pools that had worked on the stale block wasted their hash power, and decreased the effectiveness ratio. The probability for this to happen in Bitcoin is low (0.41% in 2016) [8]. During our one week measurement in August 2020, we did not find any stale block. Therefore, we do not consider the impact of stale blocks in this paper.

## C. Block reception latency and block interval

At the top of Fig. 3, we report the block intervals, and the median and maximum block reception latencies that we observed from 10 mining pools. In a week of measurement, the 1 second median block reception latency (indicated by $Med$) that we measured through ME is similar to the block propagation delay to reach 50% of the nodes in Bitcoin that was reported by KIT [13]. However, some pools waited for more than 10 seconds in some mining rounds (as indicated by $Max$). We observed a maximum block reception latency between 10 pools of 13.76 seconds, which is larger than the block propagation delay of 90% reachable nodes (less than 5 seconds, reported by KIT). This difference of block reception latency between pools leads to different effective hash rates, and its impact on the effective hash rate is amplified in the mining rounds with short block intervals. For instance, in some cases, some mining pools completely missed the chance to win the mining race as indicated by the crossing points between $d_{BI}$ and $Max$. That is, before they received the next block, other pools had already produced the next block. The bottom of Fig. 3 reports the distribution of block intervals during one week, which seems coherent with the expected exponential distribution.

## D. Effectiveness ratio

We calculate the effectiveness ratio of each pool on each of the blocks to analyze the impact of heterogeneous block reception delays in detail. First, Fig. 4 shows the CDF of the effectiveness ratios for each of the 10 pools. Each distribution shows how the effectiveness ratio of the pools varies during a week. For instance, the effectiveness ratio of “BTC.com” was smaller than 90% on 34 blocks during a week, which decreased its competitiveness in the mining competition. As a comparison, the default Bitcoin full node had the worst effectiveness ratio (95.2% in average), and our node had the highest effectiveness ratio (99.7% in average) thanks to the ME channels. The average effectiveness ratio of the pools is distributed between 96.2% to 99.2%, Pools might use some routing strategies [19, 34] to speed up their block propagation. However, independently of the strategy mining pools might use, ME would perform better (as indicated by ME channel in Fig. 4).

Second, for each block, we plot the Gini coefficient $G$ and the maximum difference $D$ between the pools' effectiveness ratios to illustrate the difference in the effective hash rate in the middle figure of Fig. 3. Notice that for better readability we plot $1 - G$ instead of $G$. The average Gini coefficient was

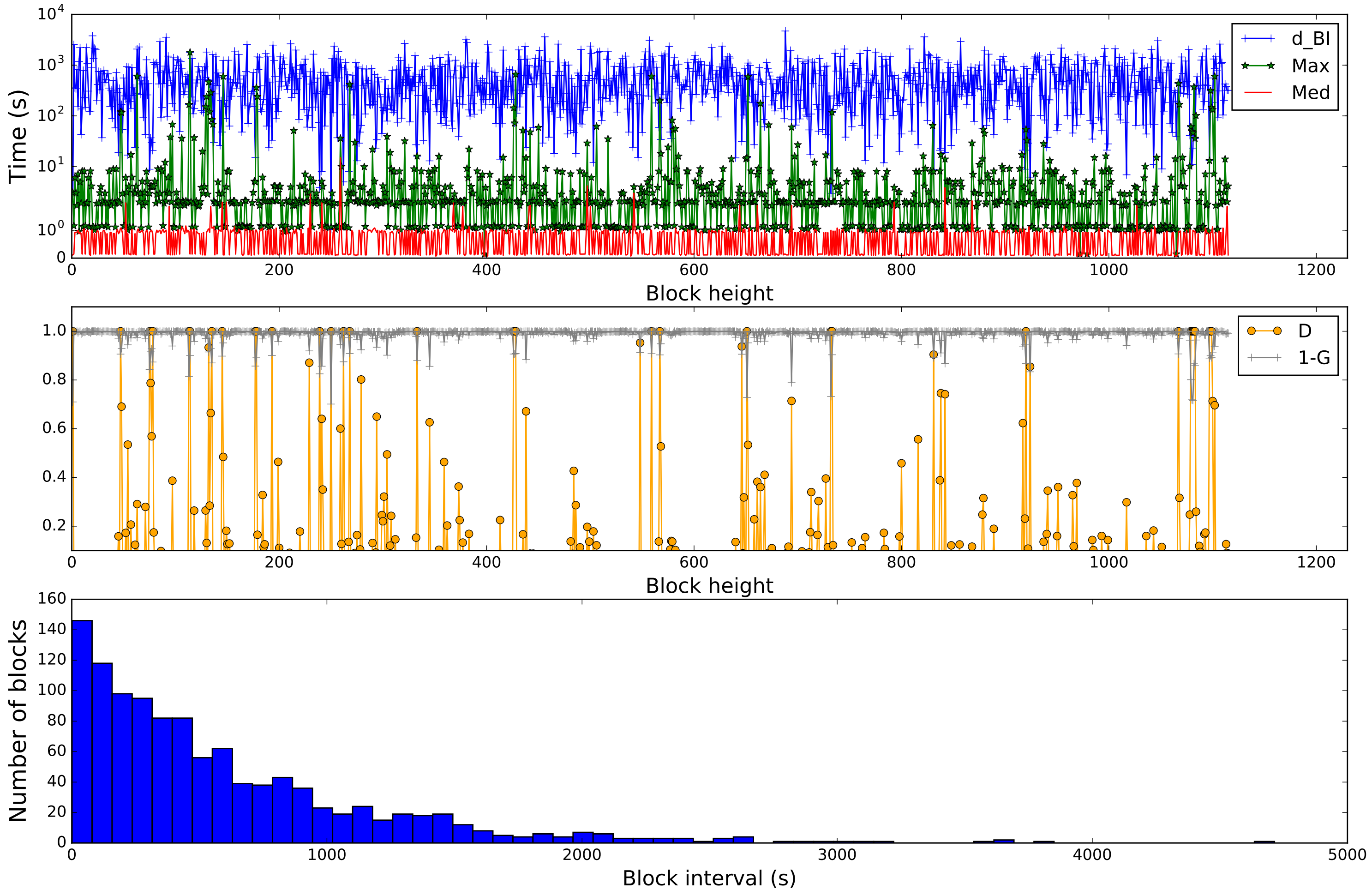

Figure 3: Results of measurement in Bitcoin. The top figure shows the block interval $d_{BI}$, and the maximum $Max$, median $Med$ block reception latency among 10 pools from block 641,767 to 642,882. The middle figure indicates the changes of $(G, D)$ between 10 pools during a week. The bottom figure shows that the block interval follows the exponential distribution during a week.

approximately equal to 0.011, which indicates a small average deviation between the 10 pools. However, we also observed that $D > 0.2$ for 9.3% of blocks, which means that some pools could work more effectively than others on the corresponding blocks, which in turn influenced their revenue.

### *E. Impact of block intervals on effectiveness ratios*

As we have shown in Fig. 3, the block reception latency has an impact over the duration of a block interval and modifies the effectiveness ratio. Here, we study the pools' effectiveness ratio on blocks with different interval delays. We calculate the average effectiveness ratio of pools for different ranges of block interval, and illustrate the result in Fig. 5. The main observation one can make is that pools have smaller effectiveness ratio on the blocks with short generation times. For instance, the average effectiveness ratio of "BTC.com" is 97.39%, 96.12%, and 92.61% on the blocks with generation time $\leq$ 4717 seconds (the maximum block interval), $<$ 600 seconds, and $<$ 200 seconds respectively. We then use the actual blocks to evaluate the effect of effectiveness ratio. As shown in Tab. I, we collected 1,116 blocks during one week. The revenue distribution was the following: "Poolin" earned 14.53% of blocks, "F2Pool" earned 14.26%, "BTC.com" earned 12.20%, "Huobi" earned 8.97%, "1THash" earned 7.00%, "OKpool" earned 6.64%, "ViaBTC" earned 5.47%, and "Novapool" earned 0.81%. "bitcoin.com" and "kanopool" did not get any block during one week. The pool's revenue share was affected by the block interval. When $d_{BI} < 200$, the actual revenue share of "BTC.com" was 10.8%, which represents a 13.11% decrease from 12.20%. This decrease is explained by the observations in Fig 4 and Fig 5, where "BTC.com" had the worst average effectiveness ratio during one week, in particular on the blocks with block intervals lower than 200 seconds. To evaluate the revenue losses or gains associated to the effectiveness ratio changes, we use the EHR Share metric to infer the bounds on revenue share of pools in Section V-F.

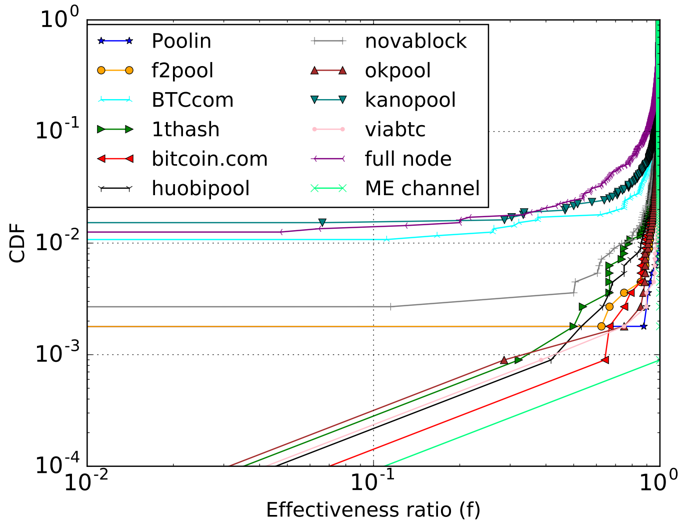

Figure 4: CDF of the observed effectiveness ratio of several Bitcoin mining pools during one week. The legend denotes the name of the mining pools, except for "full node", which represents the Bitcoin full node we maintained, and "ME channel" represents the results obtained with ME.

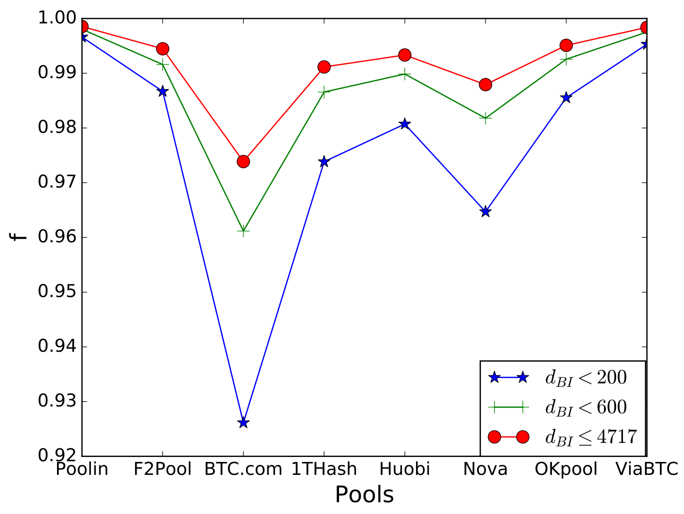

Figure 5: Effectiveness ratio of mining pools with three ranges of block intervals.

### F. Inferring the revenue bounds

To exactly evaluate the impact of network delays on the revenue distribution, one would need to know the real hash rate distribution, and evaluate the EHR Share. It is however challenging to obtain the actual hash rate distribution. Our approach to evaluate the impact of network delays on miners revenue follows the one developed in previous works [7, 8]: (1) we calculate the hash rate distribution based on the discovered blocks[4]; (2) we consider the impact of network delay on the mining process; (3) we use a Monte Carlo method to simulate 10K blocks to evaluate the revenue distribution. We calculate the revenue drift rate as $\frac{EHR\ Share - HR\ Share}{HR\ Share}$. A positive value indicates extra earnings, while a negative value shows a loss. Figure 6 shows the revenue drift rate per mining pool depending on the assumptions one could make regarding the proportion of the global hash rate we could not connect establish ME channels with.

[4]We estimated the hash rate share of a miner/pool following the best practices of the literature [7, 8] and block explorers, e.g., blockchain.com and BTC.com

Table I: Distribution of blocks during one week.

| | Poolin | F2Pool | BTCcom | Huobi | 1THash |
|---|---|---|---|---|---|
| $d_{BI}$<200 | 47 | 45 | 35 | 39 | 18 |
| $d_{BI}$<600 | 111 | 107 | 77 | 63 | 42 |
| In total | 163 | 159 | 136 | 101 | 78 |
| | OKpool | ViaBTC | Nova | | |
| $d_{BI}$<200 | 20 | 17 | 1 | | |
| $d_{BI}$<600 | 43 | 33 | 8 | | |
| In total | 74 | 61 | 9 | | |

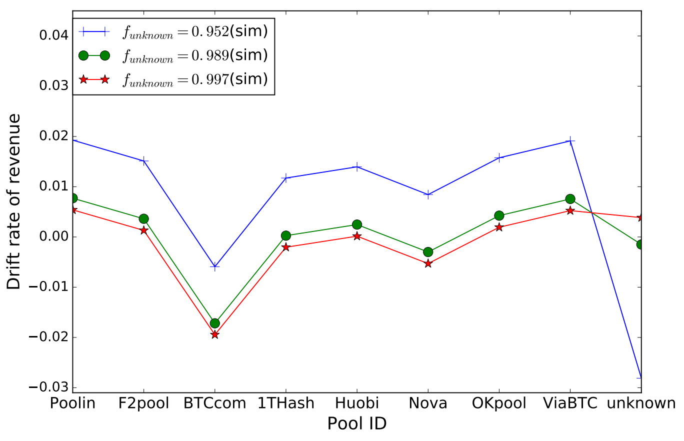

Figure 6: The bounds of mining fairness in Bitcoin. The drift rate of revenue is calculated by using EHR Share minus HR Share, and validated through the revenue share. The x-axis represents different mining pools, and "unknown" represents the remaining mining power.

Most of the top mining pools could earn more than 1% of extra blocks if we assume that the remaining 30.12% computing power only use a Bitcoin full node ($f_{unknown} = 0.952$), except for the 3rd pool (BTCcom) whose revenue drift rate is smaller than 0 due to the deviation of $f$ as we have indicated in Fig. 4. It is interesting to note that when the remaining 30.12% computing power have a connectivity that is equal to the mean of the pools we connected to ($f_{unknown} = 0.989$), the 3rd pool finds 1.7% less blocks, and the 6th pool finds 0.3% less blocks, while the other pools' drift rate is still positive (between 0 and 1%). When the unknown 30.12% computing power uses ME ($f_{unknown} = 0.997$), the revenue of the 8 connected pools decreases, as one could assume. However, even in that scenario some mining pools have an increased revenue, which confirms that some mining pools have exceptionally good network connectivity. The fact that some mining pools use different methods to optimize their connectivity [10, 35] could explain this observation.

## VI. The impact of large scale deviations

We have shown in the previous section that the miners that do not optimize their connections have a disadvantage in the mining network. In this context, a rational miner would try to improve its network access. This section discusses the situation where miners would massively decide to deviate from the official P2P protocol to try to improve their revenue. Assuming such a scenario, we first show that the top miners would reach a Nash equilibrium as they would connect to each other, while the rest of the network would not be able to do so. We conduct a simulation using 1,000 miners on a state-of-the-art mining simulator [8] to evaluate the resulting metrics.

### A. Nash equilibrium among the biggest mining pools

We consider a network with $N$ nodes $V = \{0, \cdots, N-1\}$ that includes mining nodes (a mining node represents a mining pool that could include many sub-miners) and non-mining nodes. If mining node $i$ uses a strategy $s_i$ to select a subset of $V$ as its neighbors, then its strategy space, i.e., the universe of possible neighbors sets, can be denoted as $S_i = 2^{V \setminus \{i\}}$. For $i \in V$, it may use different strategies $s_0, \cdots, s_x, \cdots, s_{n-1}$ to select its neighbors. Therefore, we would obtain graph $G[s] = (V, \bigcup_{i=0}^{n-1} (\{i\} \times s_i))$, where $(s_0, s_1, ..., s_x, ..., s_{n-1}) \in S_0 \times S_1 \times ... \times S_{(n-1)}$.

When mining node $i$ uses strategy $s_i$ to select $p$ neighbors, we can use the set $N_i = \{N_{i,1}, N_{i,2}, ..., N_{i,p}\}$ to denote the set of neighbors it selects, where $N_i \in V$ and $p \le n$. The set of edges between mining node $i$ and its neighbors can be denoted as $E_i = \{\{i, N_{i,1}\}, \{i, N_{i,2}\}, ..., \{i, N_{i,p}\}\}$.

According to the cost function that was described in [36], we have $c_i(s) = \alpha \times |E_i| + \sum_{i \neq j} stretch_{G[s]}(i,j)$, where $stretch_G(i,j)$ defines the distance cost between mining node $i$ and $j$, which is equal to the shortest distance between i and j using links of the graph divided by the direct distance. For instance, in a complete graph, $stretch_G(i,j) = 1$. $\alpha$ defines the relative importance of a pool's connectivity versus the distance cost (for more details on the definition of $\alpha$, see [36]). Assuming there is a set of mining nodes $M \in V$, for any mining node $i$, the cost function can be expressed as $c_i(s) = \alpha \times |E_i| + \sum_{i \neq j} stretch_{G[s]}(i,j),\ where\ i,j \in M$.

By using ME, the mining nodes can maintain direct connections with other mining nodes, which can optimize the $stretch$ cost. Therefore, if the number of mining nodes is $k$, the cost of mining node $i$ that uses ME strategy $s_{me}$ is $c_i(s_{me}) = \alpha \times |E_i| + (k-1)$. We define that a Nash equilibrium can be achieved by using ME as follows.

**Lemma 1.** *In a cryptocurrency network with $k$ mining nodes and $n-k$ non-mining nodes ($k \le n$), if each node can maintain $l$ connections ($l > k$), then ME is a strategy that allows a Nash equilibrium to be reached.*

*Proof.* For any mining node $i \in V$, if it uses the ME strategy $s_{me}$ to connect with other mining nodes, it would be able to build $k-1$ direct connections with other mining nodes and to reach 100% of the global hash power within one hop, where $|E_i| = k-1$, and $\sum_{i \neq j} stretch_{G[s_{me}]}(i,j) = k-1$. Thus, we have $c_i(s_{me}) = \alpha \times (k-1) + k - 1$. If mining node $i$ does not use ME, then its strategy space can be denoted by $S_i = 2^{V \setminus \{i\}}$. By using $l$ connections, for any strategy $s \in S_i$, we have a connectivity cost $\alpha \times |E_i| = \alpha \times l > \alpha \times (k-1)$, and $\sum_{i \neq j} stretch_{G[s]}(i,j) \ge k-1$. The cost of any strategy $s$ can be calculated as $c_i(s) = \alpha \times l + \sum_{i \neq j} stretch_{G[s]}(i,j)$, which implies that $c_i(s) > c_i(s_{me})$. Thus, ME is optimal for any mining node. When all mining nodes use ME, a complete graph would be achieved. □

In practice, $l$ is significantly greater than $k$. For instance, the Bitcoin network contains around 20 to 30 mining pools, and the default connectivity of a node is 125. Moreover, the connectivity of a node can easily be extended to 1,000. Therefore, a Nash equilibrium would be obtained if all Bitcoin mining pools were to adopt ME.

### B. Decreased revenue of the smaller mining pools

Table II: Rationality against fairness.

| | non-selfish ($m_b$) | selfish ($m_b$) |
|---|---|---|
| non-selfish ($m_a$) | $f_a = A$, $f_b = B$ | $f_a = A$, $f_b > B$ |
| selfish ($m_a$) | $f_a > A$, $f_b = B$ | $f_a > A$, $f_b < B$ |

We now discuss the case of a network with massive mining nodes, and $l < k$, which would imply that all mining nodes could not connect to each other and create a complete graph. Here, we point out that rationality would push the mining nodes to select their neighbors selfishly, and thereby causes unfairness. Let us assume a network of $n$ mining nodes $\{m_1, \cdots, m_n\}$ with hash rates $\{H_1 > H_2 > \cdots > H_n\}$. We consider two mining nodes $m_a$ and $m_b$ such that $a, b \in n, a < b$. Each miner can maintain $l$ connections, and $l < a - b$, which mean that $m_a$ would not build direct connection with $m_b$ when they both are selfish. We prove the game of rationality against fairness as follows, and summarize our findings in Table II.

- If $m_a$ and $m_b$ follow the default P2P protocol, their effectiveness ratio would be respectively equal to $f_a = A$ and $f_b = B$.
- A miner can improve its effectiveness ratio if it selects its neighbors selfishly. For instance, miner $m_b$ could establish connections with the top miners to reduce its average block reception latency $d_{BR}$ and improve its effectiveness ratio, which would then be greater than $B$.
- The effectiveness ratio of a miner is greatly influenced by its hash power when all miners are selfish. In this case, the miner with high hash power will connect with the miners whose hash power are closest to hers, eventually, leading to a chain-like network. The miners with higher hash powers then have relatively small block reception latency, thus, $f_a$ increases, and $f_b$ decreases. We use simulation to validate this evolution.

A miner could use the selfish neighbor selection strategy to improve its effectiveness ratio $f$. If all miners were to select their neighbors selfishly, a miner with a larger hash power

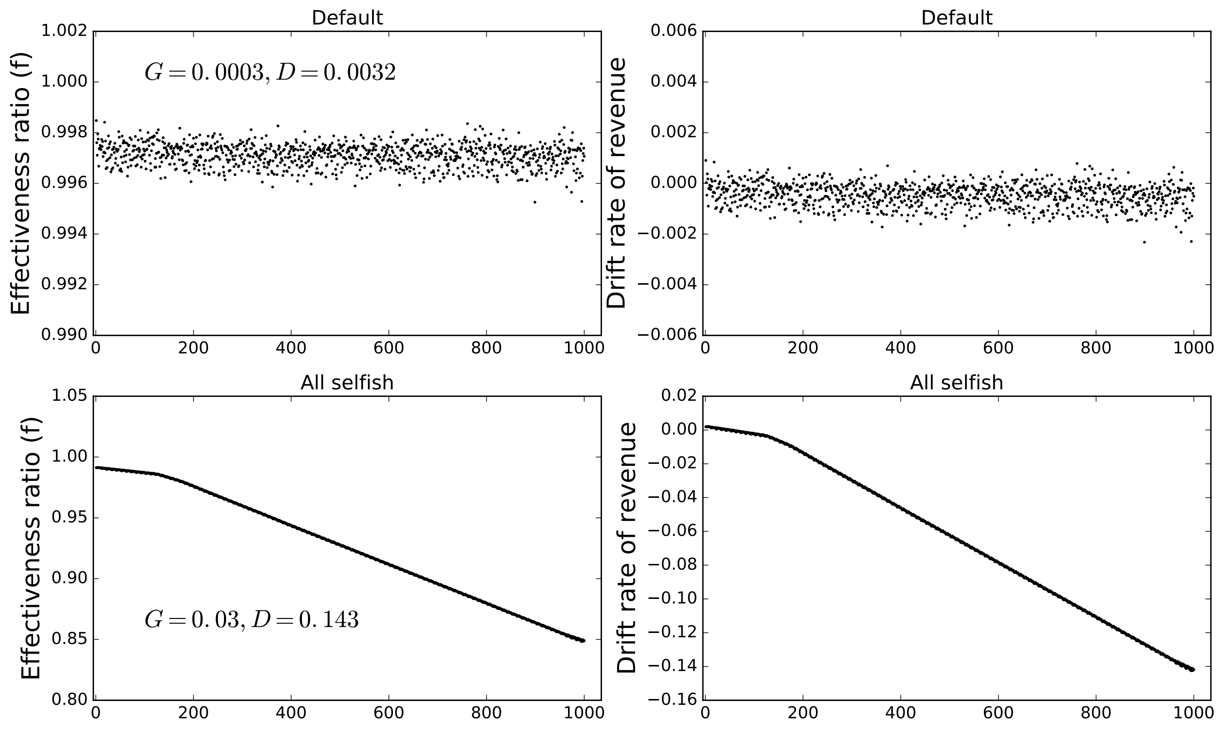

Figure 7: Impact of selfish behaviors on miners' effectiveness ratio and fairness. The x-axis represents miner's ID

would gain a larger advantage, and the network's fairness would decrease.

### C. Simulation

We modified Gervais et al.'s mining simulator [8] and simulated a network of 1,000 miners. The hash power distribution partially follows the one of Bitcoin: we assign the hash rates of the top 15 miners based on publicly accessible data [5], and the remaining miners equally share the remaining hash rate.

**Modifications of the simulator.** We i) enable 1,000 miners instead of the original 16 miners used by default; ii) remove the accumulating delay; iii) create an interface to connect with the network topology generator. This interface allows us to examine a dynamic network and observe how the miners' network latencies would evolve depending on different strategies.

**Network topology generator.** Our network topology generator mimics the existing cryptocurrency P2P protocol, it includes the following main functions:

*Peer list.* A table to maintain a node's peers information.

*Peer information exchanging.* To keep the network connectivity, each node exchanges a partial information of its peer list with others.

*Neighbor selection.* By default, each node has to periodically select a random peer from its peer list to try to establish the new connection. However, a rational node can select its neighbors selfishly. We simulate both a random neighbor selection and a selfish neighbor selection.

[5]https://btc.com/; https://www.blockchain.com/explorer

**Selfish behaviors.** We assume that miners know each other's hash power (the hash rate distribution could be estimated based on the discovered blocks as we have shown in Section V-F). A selfish miner tries to connect to the peer with the largest hash power in its peer list. We compare two scenarios: *i) default:* every miner selects its neighbors randomly according to the default protocol; and *ii) all selfish:* all miners select the neighbors selfishly.

We show in Fig. 7 the results of these simulations, where the mining power of a miner decreases when its ID increases. Our goal is to simulate a large scale mining network.

As shown at the top of Fig. 7, with the default P2P protocol, all miners select their neighbors randomly, which results in a fair network. By using the metrics we defined in Section IV, we have $(G, D) = (0.0003, 0.0032)$. Under this network topology, the revenue rate of miners is modified from -0.2% to 0.1%, independently of the computing power. However, the default setting does not show rational miners, and in practice miners are able to select their neighbors. At the bottom of Fig. 7, we assume that every miner is rational, and wants to have powerful neighbors. By using our network topology generator, we show that after 1,000 rounds of selfish neighbor selections, a miner is connected with other miners with similar hash power (i.e., network assortativity [37]). The miners' effectiveness ratio is then correlated with their ID: the more powerful miners have a shorter block reception latency and a higher effectiveness ratio. Thus, the top miners increase their revenue, and the smaller miners are losing revenue. The network fairness is evaluated to (0.03, 0.143), which indicates that the revenue distribution is unfair.

## VII. Conclusion

In this paper, we explored the effects of heterogeneous block reception delays of participants on the mining process. We first defined metrics to quantify the network impact on mining, in particular, on the mining success rate. We developed a tool to experimentally evaluate these metrics in Bitcoin. We indicated that the hash rate effectiveness of 10 Bitcoin mining pools varied during one week depending on their network latency and on the block intervals. Indeed, we evaluated that the impact of the network on the average effectiveness ratio of pools was more severe on the blocks with short generation time, which reinforced the effects of the network latency on the revenue. We then use a Monte Carlo method to simulate 10K blocks, and bound the revenue of mining pools by considering different scenarios. In a large scale mining network, we proved that a Nash equilibrium would be achieved among the larger miners (i.e., mining pools). However, this would also lead to an increased unfairness of the network, and in particular for smaller miners. It is an open problem to design fair and efficient P2P networks for PoW Blockchains, which we plan to tackle in future work. Furthermore, it is feasible to use our approach to monitor other Bitcoin-like cryptocurrencies.

## VIII. Acknowledgments.

This work is partially supported by the Fonds National de la Recherche Luxembourg (FNR) through PEARL grant FNR/P14/8149128.

## References

[1] S. Nakamoto. "Bitcoin: A peer-to-peer electronic cash system". In: (2008).

[2] C. Decker and R. Wattenhofer. "Information propagation in the Bitcoin network". In: *IEEE P2P*. 2013.

[3] K. Croman, C. Decker, I. Eyal, A. E. Gencer, A. Juels, A. E. Kosba, A. Miller, P. Saxena, E. Shi, E. G. Sirer, D. Song, and R. Wattenhofer. "On Scaling Decentralized Blockchains". In: *FC 2016*.

[4] J. Garay, A. Kiayias, and N. Leonardos. "The Bitcoin Backbone Protocol: Analysis and Applications". In: *Advances in Cryptology - EUROCRYPT 2015*.

[5] R. Pass, L. Seeman, and A. Shelat. "Analysis of the blockchain protocol in asynchronous networks". In: 2017.

[6] P. Gaži, A. Kiayias, and A. Russell. "Tight consistency bounds for bitcoin". In: *CCS*. 2020.

[7] K. Nayak, S. Kumar, A. Miller, and E. Shi. "Stubborn Mining: Generalizing Selfish Mining and Combining with an Eclipse Attack". In: *EuroS&P 2016*.

[8] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun. "On the Security and Performance of Proof of Work Blockchains". In: *CCS*. 2016.

[9] R. Zhang and B. Preneel. "Lay down the common metrics: Evaluating proof-of-work consensus protocols' security". In: *IEEE SP*. 2019.

[10] A. Miller, J. Litton, A. Pachulski, N. S. Gupta, D. Levin, N. Spring, and B. Bhattacharjee. "Discovering Bitcoin's Public Topology and Influential Nodes". In: *eprint*. 2015.

[11] T. Neudecker, P. Andelfinger, and H. Hartenstein. "Timing Analysis for Inferring the Topology of the Bitcoin Peer-to-Peer Network". In: *IEEE UIC*. 2016.

[12] I. Eyal, A. E. Gencer, E. G. Sirer, and R. Van Renesse. "Bitcoin-ng: A scalable blockchain protocol". In: *NSDI*. 2016.

[13] *Bitcoin Monitor*. https://dsn.tm.kit.edu/bitcoin/index.html. Accessed: 2020-09-24.

[14] *Pool Detective*. https://dci.mit.edu/research/tag/pool+detective. Accessed: 2020-09-24.

[15] M. Alzayat, J. Messias, B. Chandrasekaran, K. P. Gummadi, and P. Loiseau. "Modeling Coordinated vs. P2P Mining: An Analysis of Inefficiency and Inequality in Proof-of-Work Blockchains". In: *CoRR* (2021).

[16] J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A. Kroll, and E. W. Felten. "SoK: Research Perspectives and Challenges for Bitcoin and Cryptocurrencies". In: *SP*. 2015.

[17] I. Eyal and E. G. Sirer. "Majority Is Not Enough: Bitcoin Mining Is Vulnerable". In: *FC*. 2014.

[18] A. Sapirshtein, Y. Sompolinsky, and A. Zohar. "Optimal Selfish Mining Strategies in Bitcoin". In: *FC*. 2016.

[19] M. Corallo. *BIP 152: Compact block relay*. https://github.com/bitcoin/bips/blob/master/bip-0152.mediawiki. 2016.

[20] I. Eyal. "The Miner's Dilemma". In: *SP*. 2015.

[21] L. Yu, J. Yu, and Y. Zolotavkin. "Game Theoretic Analysis of Reputation Approach on Block Withholding Attack". In: *NSS*. 2020.

[22] C. Chen, X. Chen, J. Yu, W. Wu, and D. Wu. "Impact of Temporary Fork on the Evolution of Mining Pools in Blockchain Networks: An Evolutionary Game Analysis". In: *IEEE Transactions on Network Science and Engineering* (2020).

[23] B. Johnson, A. Laszka, J. Grossklags, M. Vasek, and T. Moore. "Game-Theoretic Analysis of DDoS Attacks Against Bitcoin Mining Pools". In: *FC 2014 Workshops*.

[24] A. Laszka, B. Johnson, and J. Grossklags. "When Bitcoin Mining Pools Run Dry - A Game-Theoretic Analysis of the Long-Term Impact of Attacks Between Mining Pools". In: *FC*. 2015.

[25] Z. Avarikioti, L. Heimbach, Y. Wang, and R. Wattenhofer. "Ride the Lightning: The Game Theory of Payment Channels". In: *FC 2020*.

[26] M. Rosenfeld. "Analysis of Bitcoin Pooled Mining Reward Systems". In: *CoRR* (2011).

[27] A. Miller and J. J. LaViola Jr. "Anonymous byzantine consensus from moderately-hard puzzles: A model for bitcoin". In: *University of Central Florida Tech. Report* (2014).

[28] T. Cao, J. Yu, J. Decouchant, X. Luo, and P. Veríssimo. “Exploring the Monero Peer-to-Peer Network”. In: *FC 2020*.

[29] P. Silva, D. Vavricka, J. Barreto, and M. Matos. “Impact of geo-distribution and mining pools on blockchains: a study of ethereum”. In: *DSN*. 2020.

[30] *Stratum mining protocol*. en . bitcoinwiki . org / wiki / Stratum_mining_protocol. Accessed: 2020-09-24.

[31] R. Dorfman. “A formula for the Gini coefficient”. In: *The review of economics and statistics* (1979).

[32] *BFGminer*. http://bfgminer.org/. Accessed: 2020-09-24.

[33] M. Grundmann, T. Neudecker, and H. Hartenstein. “Exploiting Transaction Accumulation and Double Spends for Topology Inference in Bitcoin”. In: *FC 2018 International Workshops*.

[34] *Falcon: A Fast Bitcoin Backbone*. ttps://www.falcon-net.org/.

[35] *Mining cartel attack*. https://bitcointalk.org/index.php?topic=2227. 2010.

[36] T. Moscibroda, S. Schmid, and R. Wattenhofer. “Topological Implications of Selfish Neighbor Selection in Unstructured Peer-to-Peer Networks”. In: *Algorithmica* (2011).

[37] M. E. Newman. “Assortative mixing in networks”. In: *Physical review letters* (2002).

## Appendix

We used a CPU miner to connect to mining pools. We report below the names of these pools, and the average RTTs that we observed between our machine and the mining pools during one week.

Poolin: stratum+tcp://btc.ss.poolin.com:1883, 22ms
f2pool: stratum+tcp://btc.f2pool.com:3333, 24ms
BTC.com: stratum+tcp://us.ss.btc.com:1800, 19ms
1thash pool: stratum.1thash.btc.top:8888, 329ms
bitcoin.com: connect.pool.bitcoin.com:3333, 171ms
Huobi pool: stratum+tcp://hk.huobipool.com:8888, 272ms
Nova pool: btc.s.novablock.com:443, 109ms
Okpool: stratum.okpool.me:3333, 267ms
Kano pool: stratum+tcp://sg.kano.is:3333, 263ms
viabtc pool: URL:btc.viabtc.com:3333, 22ms