The impact of block mining time distribution on the probability of forks

Thomas Lelièvre, Quentin Bramas

The impact of block mining time distribution on the probability of forks

1st Thomas LELIÈVRE
ICUBE, University of Strasbourg, CNRS
Strasbourg, France
thomas.lelievre@etu.unistra.fr

2nd Quentin BRAMAS
ICUBE, University of Strasbourg, CNRS
Strasbourg, France
bramas@unistra.fr

This is the accepted version of the manuscript published in the BRAINS 2024 conference proceedings.

***Abstract*—In blockchain networks, the probability of forks, where the chain splits due to simultaneous block discoveries, is a critical factor affecting both performance and security. This paper investigates how the probability distribution of block mining times influences the likelihood of fork occurrences. We develop a formal model to examine the impact of the number of miners, the distribution of their hashing power, and the probabilistic characteristics of block mining times on the fork probability. Our analysis reveals that the block mining time (BMT) distribution, which follows a geometric distribution in Bitcoin, can significantly affect fork rates. We propose an optimization approach to identify BMT distributions that minimize the probability of forks in various network configurations. Our findings suggest that tailored BMT distributions could enhance blockchain protocol design by reducing fork frequency, thus improving consensus reliability and overall network performance. This theoretical study provides insights into the potential customization of block mining processes to achieve more robust and efficient blockchain systems.**

***Index Terms*—blockchain, forks, mining distribution, distributed ledger technologies**

I. Introduction

Blockchain technology underpins the functionality of decentralized cryptocurrencies, providing a secure and transparent ledger through distributed consensus mechanisms. In blockchain systems like Bitcoin, the current state of the blockchain is represented by a single, linear chain of blocks, where each block contains a record of transactions. The integrity and reliability of this chain are crucial, as they ensure the immutability and consistency of the transactional history.

A critical aspect of blockchain network performance and security is the occurrence of forks, where the blockchain temporarily splits into separate chains. Forks primarily result from the near-simultaneous discovery of blocks by different miners, leading to a state of temporary inconsistency until one chain becomes the longest and is accepted as the canonical chain by the network.

This work was partially funded by the ANR project BASE-BLOC, ref. ANR-22-CE25-0012

While numerous studies have explored the multifaceted factors contributing to fork events—including network propagation delays, node connectivity, block size, and more—this paper aims to narrow the focus to a more specific inquiry. We investigate the probability of fork occurrences that are the results of the near-simultaneous discovery of blocks by different miners. In particular, we study the impact of the number of miners in the network, the hashing power distribution among them, and, more surprisingly, the probability distribution of the time it takes for a miner to find a block. This theoretical study may impact future blockchain design, especially when this distribution could be customized in the protocol to minimize the probability of forks. This distribution cannot be changed in PoW blockchain like Bitcoin, but may be changed in specific protocols such as Proof-of-Interactions [1].

a) Related work: Blockchain technology, introduced through Bitcoin by Nakamoto [6], has seen extensive research into its underlying mechanisms and various applications. A significant concern in blockchain networks is the occurrence of forks, which can disrupt the consensus process and lead to inefficiencies and security vulnerabilities.

Previous research has widely examined the relationship between network characteristics and fork probabilities. Croman et al. [2] analyzed Bitcoin's scalability and identified that network propagation delays are a critical factor in increasing fork rates. Similarly, Decker and Wattenhofer [4] provided an empirical analysis showing that faster block propagation decreases the likelihood of forks, emphasizing the importance of network infrastructure in maintaining blockchain integrity.

Research has also explored how the distribution of hashing power among miners impacts fork occurrences. Eyal and Sirer [5] demonstrated that the presence of mining pools with significant hashing power can lead to strategic behaviors like selfish mining, which exacerbates fork occurrences by intentionally delaying block propagation. On a similar topic, Zhang and Preneel [9] analyzed the impact of adjusting the branch selection algorithm to reduce the risk of selfish-mining. These papers differ from our work as we study the impact of the mining time distribution on legitimate forks, even when all nodes are honest.

A significant theoretical framework for analyzing fork probability in blockchain networks was presented by Yahya et al. [8]. They developed a mathematical model to predict the likelihood of forks by considering the interplay between block propagation delays and miner competition. Their model treats block arrivals as a Poisson process and uses stochastic methods to estimate the fork probability based on network latency and the rate of block discovery. However, this work does not study how variations in block mining time distributions affect the probability of concurrent block discoveries.

Further expanding on practical implications, Nourmohammadi and Zhang [7] explored the specific impacts of Ethereum Improvement Proposal (EIP) 1559 on fork occurrences within the Ethereum network. Their study focused on how the introduction of EIP-1559, which revised the fee structure and block size management, influenced the dynamics of block propagation and miner incentives.

b) Contributions: This paper makes three key contributions: (1) We present a formal model to analyze how block mining time (BMT) distributions affect fork probabilities. (2) We develop an optimization approach to identify BMT distributions that minimize fork occurrences. (3) We provide insights that could inform the design of blockchain protocols to enhance consensus reliability and network performance

II. Model

a) The block mining time distribution: We consider a set of n fully connected nodes, $u_1, u_2, \ldots, u_n$, simultaneously searching for the next block of a blockchain. The time it takes for node u_i to find a block is a positive random variable X_i that follows a distribution D_i. All the random variables are independent. The distribution D_i depends on the blockchain protocol. For instance, in Bitcoin, the distribution D_i is a geometric distribution whose parameter depends on the hashing power of node u_i. In a Proof-of-Stack blockchain, D_i may depend on the stake of node u_i.

In the remaining, unless explicitly stated, we assume that the distributions are all identical, called the Block Mining Time (BMT) distribution and denoted D. This may represent the base distribution of a simple node running the protocol and our analysis can be generalized by removing this assumption as discussed in Section V.

b) The ordering of BMT and the inter-block time: In many blockchains, such as Bitcoin, the protocol is such that the expected time between two blocks is fixed (10 minutes in Bitcoin). This means that the expectation of the minimum BMT among the nodes is a given value $\mathfrak{m}$. Formally, let $X_{(i)}$ be the time of creation of the i-th block among the n nodes, in particular $X_{(1)} = \min_{i\in[1,n]} X_i$. Then, by assumption, $X_{(1)} = \mathfrak{m}$.

c) The probability of fork: In this paper, we focus on the forks that occur when two (or more) nodes find a block in a small interval of time, that is, the probability that a fork occurs is inversely proportional to the time difference between the creation of the first and the second block. Since we want to minimize the probability of fork, we are looking for a distribution D that maximizes the mean time between the first and the second mined block, called the *first range*.

d) formalization: We have the following property [3].

Proposition 1 (Distribution of an order Statistic)**.** *Let $X_1, \ldots, X_n$ a sequence of independent and identically distributed random variables on the $(\Omega, \mathcal{A}, \mathbb{P})$ probability space and let $X_{(1)} \leq \ldots \leq X_{(n)}$ be an order statistic of $(X_i)_{i\in\{1,\ldots,n\}}$.*

$$\forall r \in \{1, \ldots, n\}, F_{X_{(r)}} = \sum_{k=r}^{n} \binom{n}{k} F_X^k (1 - F_X)^{n-k}$$

Definition 1. *The* first range *is the difference $X_{(2)} - X_{(1)}$ and is denoted Y in the remaining of this paper.*

Our goal is to find the probability distribution D that maximizes the expectation of Y.

We restrain ourselves to the cases where X is a discrete random variable such that $X(\Omega) \subseteq \{0, \ldots, T\}$ and $\forall i \in X(\Omega),\ \mathbb{P}(X = i) = p_i$.

As X is a positive random variable, so is $\left(X_{(i)}\right)_{i\in\{1,n\}}$. And using the formula of the expectancy for a positive random variable we have

$$\begin{aligned}
E(Y) &= E(X_{(2)}) - E(X_{(1)}) \\
&= \sum_{k=0}^{T-1} \left(\sum_{i=1}^{n} \binom{n}{i} F_X^i(k)(1 - F_X(k))^{n-i} \right. \\
&\qquad \left. - \sum_{i=2}^{n} \binom{n}{i} F_X^i(k)(1 - F_X(k))^{n-i} \right) \\
&= \sum_{k=0}^{T-1} n F_X(k)(1 - F_X(k))^{n-1} \\
&= \sum_{k=0}^{T-1} n \left(\sum_{i=0}^{k} p_i \right) \left(1 - \sum_{i=0}^{k} p_i \right)^{n-1}
\end{aligned}$$

We should remark that if, for a given $k \in X(\Omega)$, $X = k$ almost surely, then $E(Y) = 0$, which minimizes $E(Y)$. Hence, we consider $p = (p_0, \ldots, p_T)$ to be in $[0, 1[^{T+1}$.

We define the function $J : \mathbb{R}^{T+1} \to \mathbb{R}$,

$$J(p) = n \sum_{k=0}^{T-1} S_k(p)(1 - S_k(p))^{n-1}$$

with $S_k(p) : \mathbb{R}^{T+1} \to R,\ k = 0, \ldots, T,\ S_k(p) = \sum_{i=0}^{k} p_i$.

III. Optimization Problem

Formally, the problem $(\mathcal{P})$ can be expressed as follows

$$\text{argmax}_{p\in[0,1]^{T+1}}\ J(p) \tag{1}$$

$$\text{s.t} \quad \sum_{i=0}^{T} p_i = 1 \tag{2}$$

$$E(X_{(1)}) = \sum_{k=0}^{T-1} \left(1 - \sum_{i=0}^{k} p_i \right)^{n} = \mathfrak{m} \tag{3}$$

Recall that (3) represents the requirement on the expected time between two created blocks (*i.e.*, the time it takes in average for the first node to find a block).

Let the set of constraints $\mathcal{S}$ be

$$\mathcal{S} = \left\{p \in [0,1]^{T+1} \mid h_1(p) = 0 \text{ and } h_2(p) = 0\right\}$$
$$\text{with} \quad h_1(p) = S_T(p) - 1$$
$$h_2(p) = \sum_{k=0}^{T-1} (1 - S_k(p))^n - \mathfrak{m}$$

Then $(\mathcal{P})$ is $\text{argmax}_{p\in\mathcal{S}} J(p)$.

a) Existence: The functions S_k, $k = 0, \ldots, T$ are polynomial functions, then J, h_1 and h_2 are also polynomial functions and $S_0, \ldots, S_T, J, h_1, h_2$ are, therefore, smooth on $\mathbb{R}^{T+1}$.

Because, $[0,1]^{T+1}$, h_1^{-1}, h_2^{-1} are closed, so is $S = [0,1]^{T+1} \cap h_1^{-1} \cap h_2^{-1}$. Moreover, $S \subset [0,1]^{T+1}$ is bounded. Then, since J is continuous on the compact set $\mathcal{S}$, there exists a least one solution to the problem $(\mathcal{P})$.

b) A simple solution to $(\mathcal{P})$: For $k \in \{0, \ldots, T\}$ and $p \in \mathcal{S}$,

$$0 \leq S_k(p) \leq 1.$$

Also, observe that $f : [0,1] \to R$, $f(x) = x(1-x)^{n-1}$, has on unique maximum in $[0,1]$, obtained when $x = \frac{1}{n}$. So each term of the sum in $J(p)$ is smaller than $\frac{1}{n}\left(1 - \frac{1}{n}\right)^{n-1}$ so that

$$\forall p \in \mathcal{S}, J(p) \leq T\left(1 - \frac{1}{n}\right)^{n-1}$$

By taking $\bar{p} = \left(\frac{1}{n}, 0, \ldots, 0, 1 - \frac{1}{n}\right)$ we obtain $S_k(\bar{p}) = \frac{1}{n}$, $\forall k \in \{0, T-1\}$, and $J(\bar{p}) = T\left(1 - \frac{1}{n}\right)^{n-1}$ is maximum.

It remains to see when $\bar{p}$ in $\mathcal{S}$. Clearly $c_1(\bar{p}) = 0$ and

$$\begin{aligned} c_2(\bar{p}) = 0 \quad &\Leftrightarrow \quad \sum_{k=0}^{T-1} (1 - S_k(\bar{p}))^n = \mathfrak{m} \\ &\Leftrightarrow \quad T = \left(1 - \frac{1}{n}\right)^{-n} \mathfrak{m} \end{aligned} \tag{4}$$

Problem $(\mathcal{P})$ has a unique solution $\bar{p}$ when $T = \left(1 - \frac{1}{n}\right)^{-n} m$.

IV. Optimal Scaled Bernoulli Distribution

In this section, we only consider a scaled Bernoulli distribution. Thanks to the previous section, we know such a distribution is a global optimal when it verifies equation (4), and we now show that it has interesting properties in general.

Consider that each X_i follows a Bernoulli distribution scaled by T, that is $P(X_i = 0) = 1 - p$ and $P(X_i = T) = p$, with T a positive integer representing the maximum duration for finding a block. In this case, the goal is to find the value p such that the first constraint is satisfied: $E(X_{(1)}) = \mathfrak{m}$.

One can easily see that $P(X_{(1)} = T) = p^n$ so that

$$E(X_{(1)}) = Tp^n$$

which gives $p = \left(\frac{\mathfrak{m}}{T}\right)^{1/n}$. In this case, we can easily compute the first range by observing that the range is null except if there is exactly a single node i such that $X_i = 0$. Hence

$$E(Y) = T\,\mathbb{P}\left(\bigcup_{i=1}^{n}\left([X_i = 0] \bigcap_{\substack{j\in\{1,\ldots,n\}\\ j\neq i}} [X_j = 1]\right)\right) = Tn(1-p)p^{n-1}$$

Interestingly, when n tends to infinity, $E(Y)$ tends to $\mathfrak{m}\log(T/\mathfrak{m})$. So it is possible to obtain an arbitrarily large expected first range by choosing T large enough.

V. Non-identical Distributions

We now consider a more general scenario where the distribution of the participants may be different. In this context, we consider that each participant has an associated weight in the system, and this weight remains constant even if the participant creates multiple virtual identities, provided that the sum of their weights remains the same. This principle is evident in Bitcoin, where distributing computational power across several identities neither increases nor decreases the expected time to discover the next block. Specifically, it does not alter the time distribution for finding the next block. We contend that maintaining this property is crucial for preventing Sybil attacks.

We consider that node i has a weight $W_i \in \mathbb{N}\backslash\{0\}$. One can see that the previous property can be obtained by considering that the node i simulates W_i trials following a base distribution D (with corresponds to the distribution a node with weight 1 would follow), and takes the minimum among the obtained values. More formally, the BMT distribution of node i is

$$X_i = \min\left((X^j)_{j\in[1,W_i]}\right) \tag{5}$$

where $(X^j)_{j\in[1,W_i]}$ are W_i i.i.d. random variables that follow distribution D. One can easily see that by splitting its weight into several new participants. The chances of seeing one of the new participants obtaining the minimal BMT are exactly the same as when a single participant has the combined weight. In this context, the distribution of each participant differs from the base distribution D. For instance, when D is our Bernoulli distribution scaled by T with parameter p, the distribution of node i is also a Bernoulli distribution scaled by T but with parameter p^{W_i}. This implies that we have the same property as in the previous simple case: by choosing T large enough, we can obtain an expected first range arbitrarily large.

VI. Comparison with Bitcoin

In this section, we compare the expected first range obtained with our scaled Bernoulli distribution and with the geometric distribution used by Bitcoin. We consider $\mathfrak{m} = 10$ (so 1 unit of time represents 1 minute).

a) The case of Bitcoin: The BMT distribution for a participant follows a geometric distribution, parameterized by the difficulty and scaled by the node's weight W_i, where the weight represents the hashing power. The speedup is linearly proportional to the hashing power. This scenario can be equivalently described as the BMT being the minimum

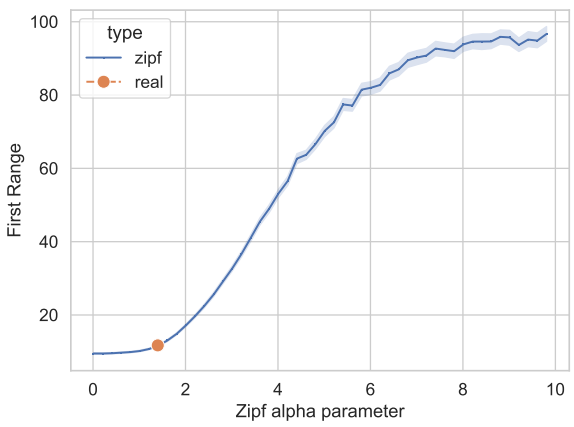

Fig. 1. First range average (Y) depending on the parameter of the Zipf distribution describing how the hashing power is distributed among the network of 100 nodes. The greater the parameter, the greater the centralization (almost all the weight is given to a single node). The point corresponds to the value obtained using the real hashing power distribution.

of W_i independent geometric distributions, each with a given parameter. Indeed, it is equivalent to consider that the participant performs each hash at the same speed but just performs W_i hashes in parallel. In this case, the difficulty is defined so that the minimum among all the $W = \sum_{i=0}^{n-1} W_i$ random variables, each following the base geometric distribution, is exactly a geometric distribution with parameter $\frac{1}{\mathfrak{m}}$.

So in the case of Bitcoin, the base distribution D is geometric with parameter p:

$$p = 1 - \left(\frac{\mathfrak{m}-1}{\mathfrak{m}}\right)^{1/W}.$$

By a simple calculation, the expected value of W i.i.d. variables following D is exactly $\mathfrak{m}$.

Then, thanks to our definition in equation (5), the random variable X_i of node i having hashing power W_i follows a geometric distribution of parameter $1-(1-p)^{W_i}$

The way the hashing power is distributed among the nodes in the network has no impact on the base distribution we use. However, it has a big impact on the *first range* and the probability of fork. Clearly, the more important the spread, the greater the probability of forks.

We can see this in Fig. 1 where we consider that the hashing power is distributed among the network following a Zipf distribution. We run 10000 simulations in a network of 100 nodes, where the hashing power is distributed among the nodes following a Zipf distribution with a parameter ranging from 0.01 to 10. The figure shows the average first range and the 95% confidence interval. When the Zipf parameter is very low, the hashing power is uniformly spread among the network, which is the worst case, as any node has the same BMT distribution. In this case, the first range is less than 10 on average. However, when a minority of the nodes holds almost all the hashing power (the Zipf parameter is high) the first range reaches a value close to 100.

We also consider the real-world scenario for the Bitcoin network using the real weight distribution of the 13 most important Bitcoin pools[1] (representing almost 95% of the estimated hashing power) and considering the unknown miners as the 14th one (with 5% of the hashing power). In this real-world scenario, a numerical computation of the expected value of the first range gives $E(Y) \approx 11.7954$. This corresponds to the theoretical value with a Zipf parameter of 1.4.

b) Using a scaled Bernoulli distribution: When using the scaled Bernoulli distribution defined in Section IV, we can easily tune the maximum value T to obtain the desired expected first range. As in the case with geometric distributions, the constraint on the expected value of the minimum $E(X_{(1)}) = \mathfrak{m}$ gives the value p of the probability of obtaining T in the base distribution D.

We require

$$E(X_{(1)}) = Tp^W = \mathfrak{m} \quad \Rightarrow \quad p = \left(\frac{\mathfrak{m}}{T}\right)^{1/W}$$

As in the previous case, the expected first range depends on the weight distribution in the network. However, using our custom distribution, even in the worst weight distribution, *i.e.*, when there are W nodes with weight 1, we can choose T so that the expected value of the first range is fixed.

VII. Conclusion and Discussion

In this paper, we examined how the block mining time (BMT) distribution impacts the probability of forks in blockchain networks. Our analysis, using a formal model, revealed that tailored BMT distributions can significantly minimize fork occurrences by increasing the time difference between the first and second block discoveries. This suggests that optimizing BMT distributions could enhance blockchain consensus mechanisms, leading to improved network performance and security. Future research could explore practical implementations of these findings as BMT usually cannot be customized, except in very specific protocols, such as Proof-of-Interactions [1].

References

[1] Jean-Philippe Abegg, Quentin Bramas, and Thomas Noël. Blockchain using proof-of-interaction. In *Networked Systems: 9th International Conference, NETYS 2021, Virtual Event, May 19–21, 2021, Proceedings*, pages 129–143. Springer, 2021.

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

[3] Herbert Aron David and Haikady Navada Nagaraja. *Order statistics*. John Wiley, 3rd ed edition, 2003.

[4] Christian Decker and Roger Wattenhofer. Information propagation in the bitcoin network. In *IEEE P2P 2013 Proceedings*, pages 1–10. IEEE, 2013.

[5] Ittay Eyal and Emin Gün Sirer. Majority is not enough: Bitcoin mining is vulnerable. *Communications of the ACM*, 61(7):95–102, 2018.

[6] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.

[1] From https://www.blockchain.com/explorer/charts/pools (June, 12th 2024)

[7] Reza Nourmohammadi and Kaiwen Zhang. Modeling the fork probability of blockchains: Did eip-1559 improve ethereum? In *2022 Fourth International Conference on Blockchain Computing and Applications (BCCA)*, pages 33–40, 2022.

[8] Yahya Shahsavari, Kaiwen Zhang, and Chamseddine Talhi. A theoretical model for fork analysis in the bitcoin network. In *2019 IEEE International Conference on Blockchain (Blockchain)*, pages 237–244, 2019.

[9] Ren Zhang and Bart Preneel. Publish or perish: A backward-compatible defense against selfish mining in bitcoin. In *Topics in Cryptology–CT-RSA 2017: The Cryptographers' Track at the RSA Conference 2017, San Francisco, CA, USA, February 14–17, 2017, Proceedings*, pages 277–292. Springer, 2017.