To Broadcast or Not to Broadcast: Decision-Making Strategies for Mining Empty Blocks

Chon Kit Lao*, Rui Jiang*, Luyao Zhang†, Fan Zhang‡, and Ye Wang*§

*University of Macau, Macau, China
†Duke Kunshan University, Kunshan, China
‡Yale University, New Haven, United States
Email: yc37494@umac.mo, dc02764@umac.mo, luyao.zhang@dukekunshan.edu.cn,
f.zhang@yale.edu, wangye@um.edu.mo
§Corresponding author

***Abstract*—Resource efficiency in blockchain systems remains a pivotal concern in their design. While Ethereum often experiences network congestion, leading to rewarding opportunities for miners through transaction inclusions, a significant amount of block space remains underutilized. Remarkably, instances of entirely unutilized blocks contribute to resource wastage within the Ethereum ecosystem. This study delves into the incentives driving miners to produce empty blocks. We ascertain that the immediate rewards of mining empty blocks often lead miners to forego potential benefits from transaction inclusions. Moreover, our investigation reveals a marked reduction in empty blocks after the Ethereum's *Merge*, highlighting that the Proof-of-Stake (PoS) consensus mechanism enhances block space efficiency in the blockchain sphere.**

Keywords: Ethereum; Blockchain; Resource Waste; Empty Block

1. Introduction

Blockchain operates as a decentralized computing system, harnessing the collective power of distributed users through a peer-to-peer network [1]. In this system, all participants share a common set of computing and storage resources. Critical to its operation, each system execution is encapsulated as transactions within blocks. The efficiency with which this block space is utilized directly influences the computational efficacy of the entire blockchain network [2]. Consequently, any resource misallocation or wastage can profoundly impact the system's overall performance.

In the current blockchain systems, miners are incentived by a transaction fee mechanism to enhance block space usage [3]. According to this approach, miners strive to maximize their earnings by including numerous transactions within a single block. Nevertheless, Ethereum, being one of the common blockchain systems, reveals a discrepancy between this mechanism's anticipated outcomes and actual results through data analysis on Etherscan[1]. Despite the presence of numerous unconfirmed transactions on the Ethereum network, certain blocks exhibit low levels of block usage, and there are even instances of entirely empty blocks. This production of empty blocks not only squanders valuable block space resources but also undermines system efficiency.

The occurrence of empty blocks in a blockchain system is inconsistent with an ideal blockchain design. In this paper, we aim to understand the root causes behind the generation of empty blocks. By examining the gap between the design of the blockchain incentive system and miner behaviors, we aspire to gain insights that will help in designing a more efficient blockchain system.

Our study combines theoretical modeling with empirical analysis. We begin by modeling the decision-making of miners after successfully mining a block to understand their motives in block mining. Given Ethereum's prevalent use, we then chose it as our primary study object, collecting actual data from its network to examine patterns in mining empty blocks. We also consider the implications of Ethereum's *the Merge* update, especially the transition in its consensus mechanism [4], to evaluate its impact on mining behavior and overall blockchain performance.

From our study, we derive three primary insights: Our theoretical model suggests that, under heightened competition or when block rewards significantly surpass transaction fees, miners have the propensity to favor immediate block rewards over potential transaction fees during the mining process. This inclination results in the creation of empty blocks. Empirically, about 91.8% of empty blocks are processed within a mere 3 seconds, which stands in stark contrast to

1. https://etherscan.io/

the 14 seconds average for blocks containing transactions. This pronounced variance in processing times supports our model's assertions concerning miners' behaviors. Following *the Merge*, when block rewards were reduced to zero and mining competition was eliminated, the occurrence of empty blocks ceased. This observation bolsters our theoretical conclusions and emphasizes the role of consensus mechanisms in enhancing blockchain efficiency.

2. Background and Related Work

Blockchain functions as a widely accepted peer-to-peer (P2P) distributed ledger system where data updates are logged as transactions [1]. Some participants assume the responsibility of gathering these transactions, bundling them into a block and appending it to the blockchain structure. These participants are incentivized with rewards, including block rewards and transaction fees when they successfully mine a block [5]. Blockchain systems dictate which participant generates the new block by consensus algorithm. The two primary consensus algorithms in blockchain systems are Proof-of-Work (PoW) and Proof-of-Stake (PoS) [6]. In PoW, participants compete to compute the block's hash value rapidly, with the first to find the correct hash earning a reward [7]. In contrast, PoS entails randomly selecting participants to create and add a block to the blockchain network [8].

Throughput, also measured in transactions per second (TPS), is a widely recognized metric for measuring the efficiency of blockchain systems [9]. It depends on two main factors: the block processing time and the number of transactions included in each block. The block processing time is affected by the chosen consensus algorithm. In PoW consensus, the block processing time varies but the algorithm maintains a constant expected block processing time through its difficulty parameter [7]. In contrast, PoS consensus sets a fixed block time by design [8]. With a consistent expected block processing time in both consensus methods, the quantity of transactions within each block becomes a crucial factor in evaluating blockchain efficiency.

Previous research has highlighted the limited TPS of blockchain systems [10]. Consequently, significant research efforts have focused on improving efficiency, primarily through scaling methods aimed at accommodating more resources within the system. These scaling strategies can be broadly categorized into two approaches. The first approach involves making modifications to the core code to enhance efficiency [11], such as altering block data [12], proposing alternative consensus mechanisms [13] and implementing sharding [1]. The second approach aims to expand blockchain capabilities while retaining the core consensus mechanism [11], achieved by introducing novel transaction handling platforms like channels [14] and side chains [15]. However, it's worth noting that most scaling research assumes an ideal block space utilization and doesn't delve into the potential impact of varying block space usage on efficiency. Addressing this research gap may be a priority for future exploration in this field.

The behavior of miners plays a pivotal role in system utilization. Existing literature has delved into various facets of this behavior, ranging from transaction ordering strategies to specific block mining decisions. Within the scope of transaction ordering, research efforts have predominantly focused on two streams: one focused on reward optimization like Maximal Extractable Value (MEV) [16], [17], analyzed maximize reward within blocks; and the other, emphasizing transaction fairness as a critical determinant for blockchain system security [18], [19]. When it comes to block-level decisions, previous works have mainly scrutinized selfish mining strategies, where miners may find it beneficial to withhold their discovered blocks rather than broadcast them immediately [20]. Despite these extensive studies, there remains a conspicuous gap: the nuanced interplay between miner behavior and block space utilization has yet to be exhaustively explored.

Additionally, the choice of consensus algorithms exerts a substantial influence on the overall performance metrics of a blockchain system. This impact is most vividly illustrated by Ethereum shift from Proof-of-Work (PoW) to Proof-of-Stake (PoS) [4], which has led to a lot of new research that compares the two types of consensus algorithms. Key metrics scrutinized in these comparative analyses include, but are not limited to, energy efficiency and economic competitiveness, as highlighted by Sunny et al. [21]. Further, unique vulnerabilities associated with PoS systems [22] are absent in their PoW analogs. Importantly, the choice of a consensus algorithm also shapes block creation and transaction inclusion patterns, thereby influencing the volume of transactions within each block. Consequently, a holistic understanding of the far-reaching implications of consensus algorithms on blockchain system efficiency stands as an intriguing frontier for future research.

3. Theoretical Model Analysis

To gain insights into the decision-making processes of miners during block mining, we develop theoretical models. Our analysis begins within the context of the PoW consensus mechanism.

In this model, miners take actions at discrete time intervals. The process initiates at time $t = 0$ with the miner starting with an empty block. Subsequent actions occur at each discrete time step, denoted as $\{0, 1, \ldots\}$. At each time step t, two sequential actions are undertaken. In the first step, the miner calculates the hash of the current block, which may already contain some transactions. In the second step, the miner decides whether to enrich the block with additional transactions or to broadcast it, should a valid hash be found.

The process continues until one of two scenarios unfolds: either the miner successfully mines and broadcasts the block, or abandons mining that specific block, typically because another miner accomplishes the mining first.

Regarding the reward, the total reward R for successfully mining and broadcasting a block consists of two elements: R_B, which represents the system-generated reward for block mining, and $\Sigma_{i=1}^{n} R_{T_i}$, the summation of transaction fees from the included transactions T_i. Therefore, R can be articulated as:

$$R = R_B + \Sigma_{i=1}^{n} R_{T_i} \tag{1}$$

Aside from assessing the value of the total reward R, it's crucial to account for the probability p of a miner actually securing it. In PoW ecosystems, this probability p is a function of the miner's computational power C_m relative to the network's total computational power C_{total}, encompassing the computational capabilities of all participating miners. Thus, the probability p is formulated as:

$$p = \frac{C_m}{C_{total}} \tag{2}$$

In the context of our theoretical model, it's worth noting that while selfish mining has been identified as a profitable strategy for miners with more than a third of the network's mining power [20], we limit our analysis to miners with less significant mining capabilities. Consequently, selfish mining is not a strategy these miners will employ within the scope of our study.

Building upon this model, we examine the decision-making process miners undergo after successfully mining a block. Our analysis reveals that miners are inclined to broadcast blocks immediately, rather than appending additional transactions, only if the transactions not yet included offer sufficient profitability. This behavioral tendency can be formalized through the following lemma:

Lemma 1. A miner will opt to include additional transactions in a newly mined block rather than immediately broadcasting it if the following inequality is satisfied:

$$\Delta R > \frac{(C_{total} - C_m)(R_B + \Sigma_{i=1}^{n} R_{T_i})}{C_m} \tag{3}$$

Here, ΔR represents the transaction fee obtained from adding an extra transaction to the block.

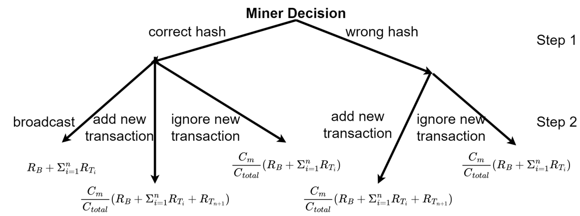

Figure 1. This Figure shows the decision tree of a miner when mining blocks and the expected reward for each action.

Proof. At time t, a miner commences by calculating the block's hash value in Step 1, incorporating all transactions collected up to that point. This hash can either be valid, satisfying the requirements of blockchain PoW mechanism, or invalid. We consider the miner's options in each scenario (cf. Figure 1).

Scenario: Invalid Block Hash Computed

In this situation, the miner moves on to Step 2 with an invalid block hash. The miner has two choices: either to add new transactions or to refrain from adding any.

Adding a Transaction: If the miner chooses to include an additional transaction with a fee of $\Delta R = T_{i+1}$, the expected reward becomes:

$$\begin{aligned} E(R)_{add} &= p \times (R_B + \Sigma_{i=1}^{n} R_{T_i} + \Delta R) \\ &= \frac{C_m}{C_{total}}(R_B + \Sigma_{i=1}^{n} R_{T_i} + \Delta R) \\ &= \frac{C_m}{C_{total}}(R_B + \Sigma_{i=1}^{n+1} R_{T_i}) \end{aligned} \tag{4}$$

Not Adding a Transaction: If the miner chooses not to include any new transactions, the expected reward remains:

$$\begin{aligned} E(R)_{not} &= p \times (R_B + \Sigma_{i=1}^{n} R_{T_i}) \\ &= \frac{C_m}{C_{total}}(R_B + \Sigma_{i=1}^{n} R_{T_i}) \end{aligned} \tag{5}$$

From this, it's apparent that adding new transactions ($E(R)_{add}$) strictly dominates doing nothing ($E(R)_{not}$) when the miner has not computed a valid hash. This is because $E(R)_{add} > E(R)_{not}$ whenever $\Delta R > 0$.

Scenario: Valid Block Hash Computed

In this scenario, the miner in Step 2 has a valid block hash and faces three options: broadcasting the mined block, adding new transactions, or withholding the block for the next iteration. As shown in the first scenario, not adding any new transactions is dominated by adding new transactions, so our discussion here will focus on the trade-off between broadcasting the block and adding new transactions.

Adding a Transaction: If the miner opts to include an additional transaction with a fee of $\Delta R = T_{i+1}$, the expected reward becomes:

$$E(R)_{\text{add}} = \frac{C_m}{C_{\text{total}}} \left(R_B + \sum_{i=1}^{n+1} R_{T_i} \right)$$

Broadcasting the Block: If the miner decides to broadcast the block, the reward becomes confirmed and is given by:

$$E(R)_{\text{broadcast}} = p \times R = R_B + \sum_{i=1}^{n} R_{T_i}$$

To identify the optimal strategy, we compare $E(R)_{\text{add}}$ and $E(R)_{\text{broadcast}}$. The following inequality reflects the system's preference for miners to add transactions until the block is full:

$$\begin{aligned} E(R)_{\text{add}} &> E(R)_{\text{broadcast}} \\ \frac{C_m}{C_{\text{total}}} \left(R_B + \sum_{i=1}^{n} R_{T_i} + \Delta R \right) &> R_B + \sum_{i=1}^{n} R_{T_i} \\ \Delta R &> \frac{(C_{\text{total}} - C_m)\left(R_B + \sum_{i=1}^{n} R_{T_i}\right)}{C_m} \end{aligned}$$

□

Next, we turn our attention to a unique scenario—when a miner successfully computes a valid block hash while the block itself is empty. Drawing upon insights from Lemma 1, we formulate the following lemma:

Lemma 2. In the context of a PoW mechanism, if a miner successfully mines an empty block, the optimal strategy is to broadcast the block immediately rather than adding a new transaction.

Proof. Consider the unique case where the mined block is empty. In such a scenario, the miner cannot accrue transaction fees, and thus the only potential reward comes from the block reward R_B.

We start by revisiting Inequality 3, adapted for an empty block:

$$\Delta R > \frac{(C_{\text{total}} - C_m)R_B}{C_m}$$

Here, ΔR represents the transaction fee that could be earned by including an additional transaction in the block, while $\frac{C_m}{C_{\text{total}}}$ denotes the proportion of computational power controlled by the miner relative to the total computational power in the network.

To explore the circumstances under which adding a transaction would be advantageous, we rearrange the inequality:

$$\frac{C_m}{C_{\text{total}}} > \frac{R_B}{\Delta R + R_B}$$

We illustrate the relationship between ΔR and $\frac{C_m}{C_{\text{total}}}$ in Figure 2. In systems governed by the PoW consensus mechanism, it is generally true that the transaction fee for a single transaction (ΔR) is less than the block reward (R_B), i.e., $\Delta R < R_B$.

Given this inequality, the miner would need to control a majority (i.e., at least 50%) of the network's computational resources to make adding a transaction more profitable than broadcasting the block. This is widely considered impractical or infeasible in a decentralized system. Hence, the rational strategy for the miner is to broadcast the empty block as-is, without including additional transactions.

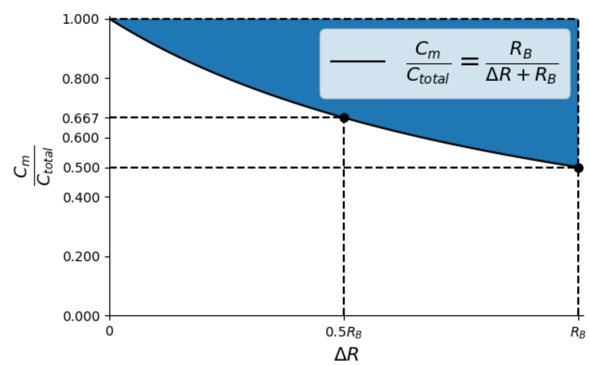

Figure 2. The relation between ΔR and $\frac{C_m}{C_{\text{total}}}$. The blue area indicates the scenarios where miners would prefer adding transactions to broadcasting the block.

□

Building on the previous analysis of miners' strategic choices under the PoW mechanism, we extend our theoretical framework to the PoS consensus mechanism. Unlike PoW, where the probability of mining the next block is proportional to computational power, PoS employs a deterministic algorithm to select the next block producer, irrespective of computational resources. Consequently, a miner selected under PoS has the luxury of iteratively adding transactions to their block until they opt to broadcast it. Given this dynamic, there's no apparent benefit for miners to broadcast an empty block, as articulated in the following lemma:

Lemma 3. In a Proof-of-Stake (PoS) consensus mechanism, when a miner is in possession of an empty block, they will opt to include a new transaction in that block rather than broadcasting the block as empty.

4. Data Collection

To investigate the behavior of miners in blockchain systems, we carried out an in-depth analysis of Ethereum blocks. Specifically, our dataset spans blocks mined between July 30, 2015, and July 31, 2023, capturing block numbers from 0 to 17,816,433.

We configured an Ethereum full node on an Ubuntu 22.04.1 Linux server using the 'eth-docker' framework[2]. For the roles of execution and consensus clients, we opted for Geth and Prysm, respectively. Upon achieving full synchronization, we extracted the block data employing the 'ethereumetl' tool[3].

The collected block data includes the following attributes: Block Number (B), Gas Used (Gas_B), Gas Limit ($Limit_B$), Base Fee per Gas ($BaseFee_B$), Timestamp (T_B), and Transaction Count ($Count_B$). It is important to note that the parameter $BaseFee_B$ is applicable only for blocks subsequent to number 12,965,000, owing to the implementation of the EIP-1559 protocol.

We define the following terms and metrics for the further analysis:

Empty Blocks: An empty block is one that contains no transactions. To be precise, a block B is categorized as empty if $Count_B = 0$.

Block Processing Time: According to the Ethereum Yellow Paper [5], a block's timestamp reflects the Unix time at which the mining process was initiated. Therefore, the processing time P_B for block B can be computed using the equation:

$$P_B = T_{B+1} - T_B \tag{6}$$

Block Space Usage: Block space usage, denoted as U_B, is calculated as the ratio of the Gas Used (Gas_B) to the Gas Limit ($Limit_B$). This relationship is mathematically expressed as:

$$U_B = \frac{Gas_B}{Limit_B} \tag{7}$$

5. Empirical Analysis Results

In this section, we shift our focus from theoretical models to empirical analysis. Initially, we examine the frequency of empty blocks in Ethereum's blockchain, comparing it under both PoW and PoS consensus mechanisms to validate our theoretical predictions. Subsequently, we investigate the correlation between block processing time and block usage, considering both PoW and PoS mechanisms.

5.1. Empty Blocks in Ethereum

We first quantify both the daily ratio of empty blocks and the daily average block usage, as illustrated in Figure 3 and Figure 4, respectively. Our data highlights specific time points that reveal trends in empty block ratios and block usage. Initially, when there were fewer transactions in the pool, we observed a high and fluctuating ratio of empty blocks and low average block usage. However, as the number of transactions increased, the empty block ratio declined, and block usage became more stable. The introduction of the EIP-1559 protocol led to a decrease in average block usage to about 0.5 and caused the empty block ratio to rise due to reduced rewards from transactions. Lastly, the system's transition to a PoS mechanism eliminated block rewards, which drastically lowered the empty block ratio to 0.001.

We also analyzed the average rate at which empty blocks occurred during two distinct periods: between the implementation of EIP-1559 and The Merge, and

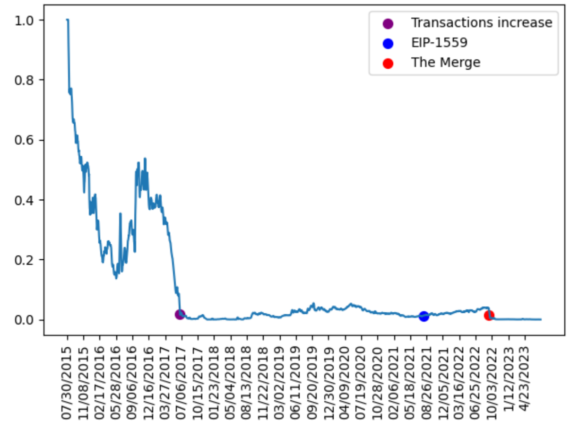

Figure 3. The percentage of daily empty blocks from July 31, 2015, to July 31, 2023. The purple point denotes the time when the total transactions massively increased. The blue point denotes the first time the system applied EIP-1559. The red point denotes the time of *The Merger*, when the system switched from PoW to PoS.

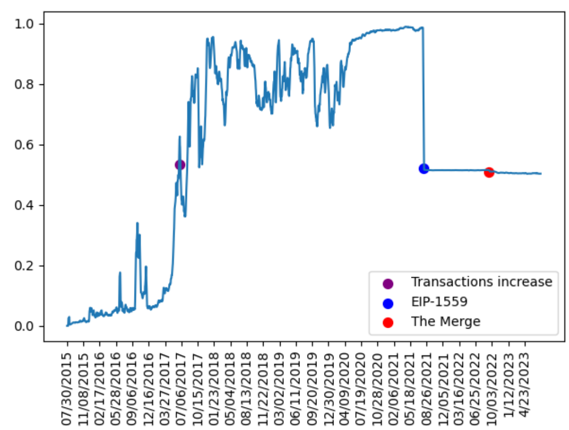

Figure 4. The daily average block usage from July 31, 2015, to July 31, 2023. The 3 points contain the same meanings as points in Figure 3.

2. https://eth-docker.net
3. https://ethereum-etl.readthedocs.io/en/latest/

the period following The Merge. During the first period, the average rate stood at 2.50%, which was considerably higher than the mere 0.11% observed after The Merge. These statistics underline the effectiveness of the consensus mechanism change from PoW to PoS in substantially reducing the frequency of empty blocks on the Ethereum network.

These findings underscore the relationship between system mechanisms and miner behavior. The data also accentuates the pivotal role of consensus algorithms and protocol upgrades in affecting both the frequency of empty blocks and the overall efficiency of block space utilization. In the subsequent sections, we delve deeper into miners' decision-making processes when encountering empty blocks. Specifically, we validate our theoretical analysis by examining the duration for which empty blocks are generated and the correlations between the block utilization and processing time under both PoW and PoS consensus mechanisms.

5.2. Block Processing Time in Ethereum

To further substantiate the notion that the generation of empty blocks is a strategic choice by miners to maximize immediate returns, we analyze the relationship between block processing time and block usage rates in the Ethereum blockchain. Given the dynamic adjustments to block usage and gas fees introduced by EIP-1559, our analysis is partitioned into two time frames: one extending from the onset of Ethereum's active ecosystem to the implementation of EIP-1559, and another from the implementation of EIP-1559 to the transition to The Merge.

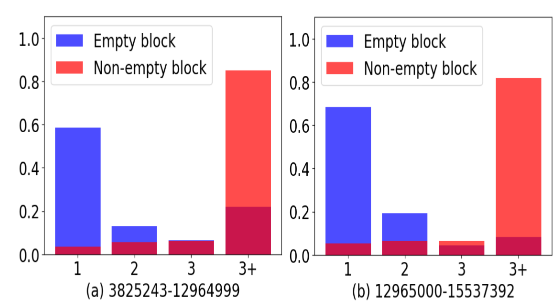

Figure 5. The distribution of the proportion of Ethereum empty blocks based on their processing time (second). (a) shows the blocks after the transaction increase and before the EIP-1559. (b) shows the blocks after the EIP-1559 and before The Merge

In our analysis, we examine the processing times of empty and non-empty blocks across two distinct timeframes: pre- and post-EIP-1559 implementation. Figure 5 shows the distribution of processing times for both types of blocks during these periods.

The data highlights a marked difference in processing times. Specifically, the overwhelming majority of empty blocks were processed within a narrow window of just 3 seconds—comprising 78.2% and 91.8% of empty blocks in the pre- and post-EIP-1559 eras, respectively. In contrast, a mere 15% and 18.2% of blocks containing transactions were processed within that same 3-second window for the two periods, respectively. Furthermore, the average processing time for blocks containing transactions was substantially longer, clocking in at 14.6 seconds as opposed to 4 seconds in the pre-EIP-1559 era, and 13.9 seconds as opposed to 2.1 seconds in the post-EIP-1559 era.

These data points support the hypothesis that the prevalence of empty blocks is largely attributable to their quick processing times. Importantly, our findings also indicate that the implementation of EIP-1559 did not meaningfully alter these processing time distributions.

We proceeded with correlation analyses to examine the link between block processing time and block usage. As mentioned earlier, we segmented the dataset into two distinct time frames: before and after the EIP-1559 activation, up until The Merge. The coefficients for these periods were 4.24 and 23.58, respectively, and the extremely low p-values (virtually zero) confirm the statistical significance of these results (t-values are 280.41 and 1256.10 respectively). This strong positive correlation between block processing time and block usage within Ethereum's PoW architecture lends credibility to our initial theoretical assertions.

These empirical findings illuminate the key dynamics affecting block processing times and usage rates. They confirm a strong relationship between these variables and lend empirical support to our theoretical insights, suggesting that miners are inclined to broadcast blocks as soon as they are successfully mined under PoW.

Further extending our investigation to the PoS phase, the derived coefficient was 0.01, supported by a p-value of 0.01 and a t-value of 3.40. This demonstrates an absence of a pronounced positive correlation between block processing time and block usage in Ethereum's PoS regime. While seemingly trivial, this regression finding aligns with our theoretical predictions.

6. Conclusion

This study delved into the dynamics of block space utilization and miner behavior in the Ethereum blockchain, transitioning from theoretical to empirical analysis. Our key findings indicate significant changes in empty block rates and block utilization with the switch from PoW to PoS. Statistical analysis confirmed a strong correlation between block processing time and block usage under PoW. Overall, our work sheds light on how system upgrades and consensus mechanisms influence Ethereum's efficiency, providing useful insights for future optimizations.

References

[1] G. Yu, X. Wang, K. Yu, W. Ni, J. A. Zhang, and R. P. Liu, "Survey: Sharding in blockchains," *IEEE Access*, vol. 8, pp. 14 155–14 181, 2020.

[2] M. V. Ferreira, D. J. Moroz, D. C. Parkes, and M. Stern, "Dynamic posted-price mechanisms for the blockchain transaction-fee market," in *Proceedings of the 3rd ACM Conference on Advances in Financial Technologies*, 2021, pp. 86–99.

[3] J. Li, Y. Yuan, and F.-Y. Wang, "A novel gsp auction mechanism for ranking bitcoin transactions in blockchain mining," *Decision Support Systems*, vol. 124, p. 113094, 2019.

[4] U. Pavloff, Y. Amoussou-Guenou, and S. Tucci-Piergiovanni, "Ethereum proof-of-stake under scrutiny," in *Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing*, 2023, pp. 212–221.

[5] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," https://ethereum.github.io/yellowpaper/paper.pdf, 2014.

[6] D. Mingxiao, M. Xiaofeng, Z. Zhe, W. Xiangwei, and C. Qijun, "A review on consensus algorithm of blockchain," in *2017 IEEE international conference on systems, man, and cybernetics (SMC)*. IEEE, 2017, pp. 2567–2572.

[7] J. Yun, Y. Goh, and J.-M. Chung, "Analysis of mining performance based on mathmatical approach of pow," in *2019 International conference on electronics, information, and communication (ICEIC)*. IEEE, 2019, pp. 1–2.

[8] M. V. Ferreira and S. M. Weinberg, "Proof-of-stake mining games with perfect randomness," in *Proceedings of the 22nd ACM Conference on Economics and Computation*, 2021, pp. 433–453.

[9] 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 *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 3–16.

[10] M. Vukolić, "The quest for scalable blockchain fabric: Proof-of-work vs. bft replication," in *Open Problems in Network Security: IFIP WG 11.4 International Workshop, iNetSec 2015, Zurich, Switzerland, October 29, 2015, Revised Selected Papers*. Springer, 2016, pp. 112–125.

[11] A. Gangwal, H. R. Gangavalli, and A. Thirupathi, "A survey of layer-two blockchain protocols," *Journal of Network and Computer Applications*, vol. 209, p. 103539, 2023.

[12] J. Levine, "Scalability controversy: understanding past cryptocurrency returns through segregated witness," 2019.

[13] I. Eyal, A. E. Gencer, E. G. Sirer, and R. Van Renesse, "{Bitcoin-NG}: A scalable blockchain protocol," in *13th USENIX symposium on networked systems design and implementation (NSDI 16)*, 2016, pp. 45–59.

[14] A. Miller, I. Bentov, S. Bakshi, R. Kumaresan, and P. McCorry, "Sprites and state channels: Payment networks that go faster than lightning," in *International conference on financial cryptography and data security*. Springer, 2019, pp. 508–526.

[15] A. Singh, K. Click, R. M. Parizi, Q. Zhang, A. Dehghantanha, and K.-K. R. Choo, "Sidechain technologies in blockchain networks: An examination and state-of-the-art review," *Journal of Network and Computer Applications*, vol. 149, p. 102471, 2020.

[16] K. Qin, L. Zhou, and A. Gervais, "Quantifying blockchain extractable value: How dark is the forest?" in *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2022, pp. 198–214.

[17] A. Capponi, R. Jia, and Y. Wang, "The evolution of blockchain: from lit to dark," *arXiv preprint arXiv:2202.05779*, 2022.

[18] K. Kursawe, "Wendy, the good little fairness widget: Achieving order fairness for blockchains," in *Proceedings of the 2nd ACM Conference on Advances in Financial Technologies*, 2020, pp. 25–36.

[19] S. Siddiqui, G. Vanahalli, and S. Gujar, "Bitcoinf: Achieving fairness for bitcoin in transaction-fee-only model," *arXiv preprint arXiv:2003.00801*, 2020.

[20] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," *Communications of the ACM*, vol. 61, no. 7, pp. 95–102, 2018.

[21] S. King and S. Nadal, "Ppcoin: Peer-to-peer crypto-currency with proof-of-stake," *self-published paper, August*, vol. 19, no. 1, 2012.

[22] S. Azouvi and M. Vukolić, "Pikachu: Securing pos blockchains from long-range attacks by checkpointing into bitcoin pow using taproot," in *Proceedings of the 2022 ACM Workshop on Developments in Consensus*, 2022, pp. 53–65.