CREST
CENTER FOR RESEARCH
IN ECONOMICS AND STATISTICS

WORKING PAPER SERIES

LATENCY TRADEOFFS IN BLOCKCHAIN CAPACITY MANAGEMENT

Michele Fabi

N°10 /October 2024

CREST
Center for Research in Economics and Statistics
UMR 9194

5 Avenue Henry Le Chatelier
TSA 96642
91764 Palaiseau Cedex
FRANCE

Phone: +33 (0)1 70 26 67 00
Email: communication.crest@ensae.fr
https://crest.science/

Latency Tradeoffs in Blockchain Capacity Management*

Michele Fabi†

September 2024

Abstract

We analyze the effect of block propagation latency on the performance and design of Nakamoto-style blockchains. Miners strategically choose block capacity, balancing the risk of invalidation from forking with transaction fee income. The model identifies a unique and symmetric Nash equilibrium block capacity, which increases with the ratio of block production time to transmission delay and decreases with the ratio of coinbase reward to transaction fee rate. We endogenize blockchain growth and derive the Fokker-Planck equation for pending mempool data. The results reveal a trade-off between efficiency (low transaction load) and security (high miner participation). Reducing the coinbase reward while raising transaction fees improves efficiency but may weaken security. We also discuss testable implications and extend the model to include uncle block rewards and discrete latency.

JEL Codes: C62, C63, G10, G14.

Keywords: blockchain design, Nakamoto consensus, forks, coinbase, stochastic storage.

*We thank the academic chair Blockchain@X for financial support. We also thank the audience at Blockchain@X-OMI Workshop on Blockchain and Decentralized Finance (September 2023), International Fintech Research Conference (November 2023), Torino Decentralized Finance Conference (April 2024), Finance and Business Analytics Conference (June 2024), as well as the audience at University of California Santa Barbara UCSB-ECON DeFi Seminar (Online, September 2023), CREST Microeconomics Seminar (October 2023), European University Institute (Online, October 2023), CESA Business School (Online, November 2023), University of Padova, Department of Management Seminars (January 2024), University of Bologna, Department of Economics Seminars (January 2024), University of Bergen, Department of Economics Seminars (February 2024). This work benefited from valuable discussions by Anantha Divakaruni, Julien Prat, Julien Combe, Matthijs Breugem.

†Telecom Paris, CREST, IP Paris (michele.fabi@ensae.fr)

1 Introduction

Despite the recent shift in blockchain research toward application-level issues—such as Miner Extractable Value (MEV) and Loss-versus-Rebalancing (LvR)—and scaling solutions—like rollups and danksharding—core questions about blockchain consensus mechanisms have yet to be thoroughly addressed. This paper aims to contribute to this discussion by analyzing the impact of block propagation latency on the performance and design of blockchains operating under Nakamoto consensus. Specifically, it examines how transmission delays affect miners' strategic decisions on block capacity (i.e., block size in units of data, such as kilobytes), the equilibrium speed of blockchain growth, and the volume of pending data awaiting confirmation. Furthermore, the analysis highlights a tension between optimizing rapid data processing and ensuring enough miner participation to protect the network from potential attacks.

This paper is particularly relevant for blockchains in fast block regimes, where transmission latency becomes significant. As blockchains evolve to support applications requiring near-instant finality, like financial services integration, managing latency becomes crucial. While Bitcoin produces blocks every 10 minutes, blockchains like Cardano and Ethereum do so in 10 to 20 seconds, but competitive use cases demand production speeds in fractions of a second. Blockchains such as Solana and Sui have introduced novel mechanisms like Proof of History (PoH) to achieve faster finality. However, the challenges of applying Nakamoto consensus in these fast-block environments remain understudied, which this paper aims to address. Our findings are also relevant for Layer 2 scaling solutions, where Layer 1 efficiency is essential for overall performance. For instance, Ethereum's current proto-danksharding introduces blobs, a new data structure designed to store rollup data and more. When attached to blocks, these blobs can burden the Layer 1 consensus mechanism and impact scalability.

We introduce the baseline model in Section 2, representing the blockchain as a dynamical system and its consensus mechanism as a stochastic game. To recap briefly, blockchains are decentralized ledgers maintained by nodes known as miners (or validators), who collect pending data into a *mempool* and record them in blocks. The sequential arrangement of these blocks creates a block-chain that gives a chronological timeline to the events recorded in the blocks. Miners are rewarded for their job with tokens as rewards for adding new blocks, the *coinbase* reward, and by transaction fees paid by users for including their data.

Given its decentralized nature, the blockchain is distributed across numerous copies within the miner network. Each miner updates its copy with new blocks and communicates these updates to others, relying on a *consensus protocol* to ensure consistency. Given the absence of a central authority, miners' actions are guided by the protocol's rules and incentives.

The widely used *Nakamoto consensus*, including Proof-of-Work (PoW) and Proof-of-Stake (PoS) variants, ensures blocks are produced at regular intervals and assigns block proposal rights through a lottery, based on computational power or token stakes.

In an ideal scenario of instantaneous data transmission, each block would seamlessly attach to its predecessor, allowing all miners to coordinate continuously on a unique blockchain. However, in reality, data transmission over the internet is subject to delays, leading at times multiple miners to propose blocks simultaneously. When these conflicts occur, they generate *forks*, which are diverging chains of blocks that reflect disagreement among miners over the blockchain state—the registry of recorded transactions.[1] In practice, miners typically resolve forks by considering the first block they receive as part of the main chain and discarding the others into forks. Adding more data to a block therefore increases its transmission time and the consequent risk that the block gets discarded. However, while the coinbase reward is fixed regardless of the block content, a larger block also provides a miner with higher revenues from transaction fee income, which scales with the amount of data recorded. Thus, when proposing a new block, miners face a trade-off between adding more data to earn higher transaction fees and the increased risk of block invalidation due to forking. We model this strategic choice as a *block proposal game*, where miners' strategies depend on the number of pending transactions, which act as a state variable.

Sections 2.2 and 2.3 present the first major result of this paper, which identifies the equilibrium block capacity based on blockchain configuration. Assuming transmission delays increase linearly with block capacity, the analysis finds a unique symmetric Nash equilibrium in the block proposal game. In this equilibrium, block capacity is positively related to the ratio of block production time to transmission time per unit of data, and negatively related to the ratio of the coinbase reward to the transaction fee income per unit of data. The reasoning behind this result is that, when the per-data transmission delay is short compared to block production time, miners are less concerned about increased forking risk from adding more transactions, leading to increased block capacity. Conversely, since the coinbase reward is fixed and unrelated to block capacity, a higher coinbase reward reduces miners' motivation to include more transactions due to the increased risk of invalidation.

In the atomistic limit where the blockchain is populated by a large number of small miners, the Nash equilibrium block capacity is miner-optimal. In other words, it maximizes miners' aggregate revenues, optimally trading off risk for revenues of each block. Thus, given that blockchain security increases in the number of participating miners, the equilib-

[1]The forks we refer to are spontaneous forks. However, there exist other types of forks, such as hard forks, that are created purposely in order to propose a structural modification of the consensus protocol. In this work we abstract from this second type.

rium block capacity is also optimal for blockchain security. Nevertheless, the block capacity resulting from miners' equilibrium is not necessarily user-optimal, meaning it does not necessarily minimize the time it takes miners to record a pending transaction. This occurs because, due to the presence of the coinbase, miners produce blocks that are inefficiently too small. As a result, the incentives of users and miners are aligned only if the coinbase reward is null.

In Section 3, we derive the user-optimal block capacity by studying the dynamical system that describes the growth of the blockchain and the evolution of the mempool. In doing so, we provide a neat characterization of its steady-state using a Fokker-Planck equation, and show that the equilibrium distribution of pending data follows a negative exponential density fully parameterized by the *load*: the ratio of the rate at which transaction are submitted to the miners per unit of time to the rate at which transactions are recorded by miners per unit of time. Through the dynamic analysis of mempool dynamics, we conclude that, given a fixed demand for data storage on the blockchain, the user-optimal block capacity is the one that minimizes the load. The logic of this result is that minimizing the load corresponds to also minimizing the waiting time a user bears before having a transaction recorded, as well as several other metrics of efficiency from the users' standpoint.

In Section 4, we then explore the design implications of the tradeoff between security and efficiency for users, focusing on maximizing efficiency (that is, minimizing the load) while ensuring security and individual rationality for users. Security requires miner participation to be above an exogenous threshold. Such threshold represents the power of a potential attacker willing to disrupt the blockchain. Individual rationality requires transaction fees to be below an exogenous threshold, to be interpreted as a reservation utility. The analysis suggests that, to achieve efficiency, that is, the minimum load, the blockchain would need to satisfy the security budget only with transaction fees. Doing so requires a fee rate such that the fee income raised over data transmitted in a unit of time is equal to the marginal cost of attacking the blockchain amplified by a factor e (2.718...). If such fee-only design is not feasible, a second-best implementation, leading to a higher load, can still be achieved by using the coinbase. Conversely, if the constraints on the optimization problem are too tight, it becomes impossible to jointly satisfy security and individual-rationality constraints.

The paper also includes two extensions of the main model, presented in Section 5. First, we consider a variant of the baseline model where miners are rewarded with a coinbase revenue also on forked blocks. In this setting, we find again a symmetric equilibrium with an analogous structure to the one of the main model. However, the coinbase reward on forked blocks can raise miners' incentive to include data, and if set equal to the regular coinbase reward, it allows achieving the efficient load without relying on transaction fees. Second, we

consider a variant where the transmission delay is discrete. In this case, while this version could be empirically relevant in some cases, it would lead to a multiplicity of equilibria. Despite such multiplicity, the minimum block capacity follows again the same comparative static features that are consistent with the baseline model. The paper wraps up with a discussion of main findings and testable implications in Section 6.

Literature Review

This paper contributes to the operational research and financial economics literature on blockchain design.

Blockchain Economics

Several market microstructure studies analyze the optimal configuration of Bitcoin transaction fees using queuing theory (Hinzen et al., 2019; Huberman et al., 2019; Easley et al., 2019). A parallel line of research studies Ethereum's fee structure (Roughgarden, 2021, 2020; Liu et al., 2022). More broadly, Chung and Shi (2022) employ an axiomatic approach to characterize optimal transaction fee mechanisms in Nakamoto-style blockchains. This work extends the above literature by considering block capacity as a miner-determined function of the transaction fee rate and latency, introducing a consensus-layer tradeoff. Taking a different perspective, Lehar and Parlour (2020) shows that Bitcoin miners may collude to keep blocks small, increasing their earnings through higher fees from priority auctions.

While this paper presents a latency tradeoff in block capacity choice, recent work highlights a broader set of latency-related issues faced by miners. For example, Schwarz-Schilling et al. (2023) discuss mining games where block proposal latency originates endogenously, as miners postpone blocks to capture as much value as possible without exceeding a critical time threshold. Latency is a salient issue also in MEV auctions, where it can have important implications for the optimal bidding strategy (Daian et al., 2019; Wu et al., 2024; Öz et al., 2024). In particular, latency reduces the effectiveness of adaptive bidding.

Beyond latency challenges, John et al. (2024) offers an overview of the economic design agenda of the Ethereum ecosystem.

Queuing Theory

The model presented in this work offers a continuous alternative to the discrete queuing theory models in the literature, which consider block capacity as a discrete variable. Specifically, our jump-diffusion mempool model provides a different approach to the limiting procedure used by Huberman et al. (2019) for their bulk-service $M/M^{[K]}/1$ queue. In our model,

the Markov-chain representation of the state-transition equations resembles a $D/M/1$ queue (Jansson, 1966). However, unlike in the $D/M/1$ queue, where each period starts with a new arrival, our model allows for periods to begin without a pending full block because miners remain active as long as the mempool contains any data.

Several papers adopt a binary block setting (with zero or one transaction per block) described by an $M/M/1$ queue (Hinzen et al., 2019; Easley et al., 2019). Under this assumption, miners earn the coinbase reward at the rate of blockchain growth and collect fees at the transaction arrival rate. However, this decomposition requires the steady-state probability of the idle state (i.e., the mempool being empty) to be linear in the load, a property that does not hold in general.

Our model also connects to stochastic inventory theory (Porteus, 2002). The key difference is that, while classical models feature a stock filled in batches and discharged continuously, our model involves continuous charging and batch discharging. Similar dynamics are used in neuroscience to model integrate-and-fire neurons (Dumont and Gabriel, 2020).

Algorithmic Trading

Beyond its relevance for blockchain scaling, latency creates important tradeoffs in algorithmic trading and market making. Budish et al. (2015) examines how latency in high-frequency trading leads to a costly speed race among traders, proposing frequent batch auctions to mitigate these inefficiencies by discretizing time and reducing the value of speed advantages. Moallemi and Sağlam (2013) analyzes the effects of latency on market-making strategies within a principal-agent framework. Gao and Wang (2020) studies the impact of latency on trading strategies under uncertain order flow conditions. Cartea et al. (2021) and Cartea and Sánchez-Betancourt (2023) explore the relationship between latency and liquidity risk, as well as optimal execution strategies in the presence of stochastic delays.

Computer Science

This paper is also closely related to the computer science literature on blockchain consensus. Among these works, Houy (2014) characterizes the equilibrium set of the block proposal game for the case of two miners with heterogeneous costs of investing in mining capacity. This work enhances its analysis by rigorously deriving the formula for the unique symmetric block capacity equilibrium and proving its uniqueness for a generic number of miners.

Other studies examine blockchain growth and security. Pass and Shi (2017) was the first to compute the growth rate of the blockchain using assumptions similar to ours regarding

transmission delays,[2] and their work laid the foundation for subsequent research. Building on this, Ren (2019a) and Dembo et al. (2020) developed continuous-time models of blockchain security assuming a Poisson process of block arrivals as in our study. Notably, these studies assume that all blocks face the same latency regardless their capacity. Addressing this limitation, Doger et al. (2024) extends the literature by incorporating the effect of block capacity on latency. In a different vein, Amoussou-Guenou et al. (2023) explores incentives under Byzantine Fault Tolerance (BFT) consensus—the leading alternative to Nakamoto consensus—which essentially eliminates forks by employing validator committees for block proposal and validation.

2 Nakamoto Consensus as a Stochastic Game

In the classical version of Nakamoto consensus, the blockchain nodes, known as *miners*, are responsible for reaching consensus on the blockchain's state. We let $m \in \{1, 2, \ldots, M\}$ denote a generic miner and $M \in \mathbb{N}$ the number of active miners. M will be determined by free entry in Section 3.2.

Miners construct the blockchain by concatenating blocks, which are structures containing records of user data or "transactions." Each block is indexed by a counter $B \in \mathbb{N}_0$[3], indicating the number of predecessor blocks in the chain up to the genesis block 0. The length of the blockchain is determined by the index of the furthest block from the genesis.

Miners also maintain a queue of pending transactions in a *mempool* (memory pool), denoted by $Q \in \mathbb{R}_+$. Transactions enter the mempool when submitted by users and exit when recorded on the blockchain by miners.

In principle, each miner stores a separate copy of the mempool and the blockchain. However, miners rely on a peer-to-peer network to synchronize their local mempools and follow *Nakamoto consensus* to synchronize their local blockchains. Therefore, we consider a single blockchain and mempool for analysis.

2.1 Nakamoto Consensus

Nakamoto consensus defines the rules for competition among miners to update the blockchain. Each time a block B is added to the blockchain, miners engage in a tournament to select the next block $B+1$. Only one proposal is accepted, and the discarded blocks form forks in

[2] While their work defines a parameter as an upper bound on transmission delays, our paper treats transmission delays as an exact value.

[3] The block index is commonly referred to as "block height."

the blockchain. In the baseline model, miners ignore forked blocks. However, in Section 5.1, we will consider a variant of the model where forked blocks affect miner revenues.

In each of these tournaments, block capacity cannot exceed the amount of data available in the mempool.[4] Therefore, the mempool size determines miners' action set. Miner interactions can thus be modeled as a stochastic game with Q as the state variable and these tournaments as stage games, indexed by the block number B of the tip of the blockchain. We refer to a tournament for proposing the next block as a *block proposal game*.

Miners participate in block proposal games by choosing block capacity, denoted k_m for $m \in \{1, 2, \ldots, M\}$. We assume miner strategies are Markovian, meaning the strategy of miner m is a function $k_m(Q) : \mathbb{R}_+ \to [0, Q]$, inducing the sequence $\{k_m(Q_B)\}_{B=0}^{\infty}$.[5] The resulting Markov Perfect Equilibrium (MPE) of the stochastic game determines the growth rate of the blockchain, $\lambda \in \mathbb{R}_+$, which is the frequency of the times $\{t_B\}_{B=0}^{\infty}$ at which the blockchain grows by one block ($t_0 = 0$ by convention). The equilibrium also induces a stationary distribution of the mempool size (when it exists), which we will characterize in Section 3. Miners take blockchain growth and the mempool distribution as given when choosing block capacity.

Block Proposal Mechanism

Each block proposal game spans the time it takes for a new block to be added to the blockchain, denoted by the interval $[t_B, t_{B+1})$, and thus has a duration $T \equiv t_{B+1} - t_B \in \mathbb{R}_+$. During this time, the blockchain protocol sets a block production rate of $\mu \in \mathbb{R}_+$ blocks per unit of time.

The block production rate is *fixed* and independent of the number of participating miners. As miner participation increases, the rate at which individual miners propose blocks decreases proportionally, ensuring that the aggregate rate remains constant. If M homogeneous miners participate in the block proposal game, each miner produces blocks at a Poisson rate of $\mu_m \equiv \mu/M$. We refer to μ^{-1} as the *block production time* or *block time*.

Nakamoto consensus has several variations, but all determine block proposers through a lottery. The most well-known implementation is Bitcoin's Proof-of-Work (PoW) consensus, where a miner's selection probability is proportional to its share of the network's computing power. In contrast, Nakamoto-style Proof-of-Stake (PoS) protocols require miners to make a token deposit, or "stake," and miners are selected to propose new blocks based on their

[4] Most protocols also impose an upper bound on block capacity to avoid spamming, but we neglect this aspect as it does not significantly impact the analysis.

[5] Relaxing this restriction would require studying strategies that depend not just on Q_B but also on the time elapsed since the last observed block. With that information, miners could infer the block capacity choices of others.

share of the total stake. Appendix B provides a brief description of popular Nakamoto-style consensus protocols.

Consensus Blocks and Fork Resolution

According to the classical Nakamoto consensus, whenever a miner observes multiple chains, it should build its next block on the longest chain (the one with the largest number of blocks) and consider the other chains as forks.[6] Therefore, unless miners act maliciously, they should all build their blocks on top of the longest chain.

More relevant to the analysis that follows is the case in which miners act in accordance with the protocol but two or more blocks extend the longest chain simultaneously. In this case, miners consider the next valid block to be the first one they observe. This criterion is stated by Nakamoto (2008) and is a practical choice for breaking ties among conflicting blocks in the vast majority of Nakamoto-style blockchain protocols.[7]

This mechanism introduces the main friction of our model: In an ideal world with immediate block transmission, the blockchain would grow without forks since conflicting blocks are precluded. However, introducing *transmission delays* that increase with the size of proposed blocks gives rise to forks and strategic tradeoffs. A miner updates the blockchain if its block is the first to be proposed *and transmitted* to the other miners. Therefore, larger blocks reduce a miner's probability of winning. Nevertheless, miners may still be willing to increase the capacity of their blocks if doing so allows them to collect higher revenues.

To model these elements, we assume that a miner who produces a block of capacity $k \in [0, Q]$ faces a linear transmission delay of Δk before propagating the block to the entire miner community, where $\Delta > 0$ is the delay per unit of data. The linearity assumption makes the model tractable and closely resembles observed propagation times (Shahsavari et al., 2022; Mighan et al., 2022).[8]

[6] To be precise, the consensus chain is the one with the highest cumulative mining power, which is typically the longest.

[7] Nakamoto (2008) states that "Nodes always consider the longest chain to be the correct one and will keep working on extending it. If two nodes broadcast different versions of the next block simultaneously, some nodes may receive one or the other first. In that case, they *work on the first one they received*, but save the other branch in case it becomes longer. The tie will be broken when the next proof-of-work is found and one branch becomes longer; the nodes that were working on the other branch will then switch to the longer one."

[8] Given that communication over the internet occurs in data packets, a more realistic approach would be to assume piecewise linearity or discrete block capacity. However, doing so would lead to a multiplicity of equilibria that the continuity assumption removes.

2.2 Equilibrium Block Capacity

In this section, we determine the Nash equilibrium block capacity choice in a given block proposal game.

Miners choose block capacity by balancing the increased revenues from larger blocks against the higher risk of losing the block proposal game due to their blocks being discarded in forks. Each block that successfully updates the blockchain provides its creator with a coinbase revenue, denoted by $\pi \in \mathbb{R}_+$, from the creation of new tokens. The miner also receives transaction fees amounting to τk, where $\tau \in \mathbb{R}_+$ is the transaction fee rate.

For simplicity, we take τ as given. Nevertheless, the main insights of the analysis hold when τ results from an auction or mechanisms such as Ethereum's EIP-1559. The analysis applies directly to blockchains such as Cardano, where a proportional transaction fee rate is imposed by the protocol.

The expected revenue a miner receives for proposing a block of capacity k is given by

$$R(k, \mathbf{k}_{-m}) = P(k, \mathbf{k}_{-m}) (\pi + \tau k), \tag{1}$$

where $\mathbf{k}_{-m} \in \mathbb{R}_+^{M-1}$ represents the block capacity choices of the other miners, and $P(k, \mathbf{k}_{-m})$ is the probability that the proposed block becomes the new head of the blockchain.

Win Probabilities

A miner m wins the block proposal game if it is the first to propose and transmit a new block. The time it takes the miner to do so is given by $T_m + \Delta k$, where T_m is an exponential random variable (ERV) with rate $\mu_m \equiv \mu/M$. Thus, the win probability of a miner proposing a block of capacity k is expressed as

$$P(k, \mathbf{k}_{-m}) = \mathbb{P}\left(T_m + \Delta k < \min\left\{T_{m'} + \Delta k_{m'}\right\}_{m' \in \{1,\dots,M\}\setminus m}\right). \tag{2}$$

The generic expression for $P(k, \mathbf{k}_{-m})$ is rather intricate and not very informative. Nevertheless, for the purpose of the analysis to come, it will be sufficient to evaluate $P(k, \mathbf{k}_{-m})$ only at specific values of $(k, \mathbf{k}_{-m})$ that make the expression tractable. For example, as stated below, miners have equal probability of establishing the next block if they all choose the same capacity. Furthermore, some general properties of the win probabilities can be immediately established by their definition:

Lemma 1. *The win probability $P(k, \mathbf{k}_{-m})$ satisfies the following properties:*

1. ***Monotonicity:*** *$P(k_m, \mathbf{k}_{-m})$ is increasing in $\mathbf{k}_{-m}$, decreasing in k_m, and decreasing in $M = |\mathbf{k}_{-m}| + 1$.*

2. ***Proportionality:*** $P(k, (k, k, \ldots, k)) = \dfrac{1}{M}$.

Monotonicity is obvious from Eq. (2), while proportionality is a standard property of the minimum of IID random variables.

Equally important and more nuanced are the properties of the likelihood ratio, defined as:

$$L_{k,k'}(\mathbf{k}_{-m}) = \frac{P(k', \mathbf{k}_{-m})}{P(k, \mathbf{k}_{-m})}.$$

The likelihood ratio reflects the *relative* change in win probability when block capacity is increased or decreased. Specifically, for $k' > k$ ($k' < k$), a likelihood ratio that decreases (increases) with the block capacity chosen by other miners indicates strategic complementarity in miners' choice of k_m: the reduction in win probability from increasing block capacity is smaller when others also propose larger blocks. Thus, a miner is more incentivized to increase capacity and collect higher fees when also the other miners choose higher capacities.

To analyze the Nash equilibrium of the block proposal game, it is enough to analyze the monotonicity of $L_{k,k'}(\mathbf{k}_{-m})$ for specific values of $\mathbf{k}_{-m}$. Thus, we prove a weaker form of monotonicity, *local monotonicity*, where $\mathbf{k}_{-m}$ is fixed. The next lemma establishes what we call local symmetric monotonicity.

Lemma 2. *The likelihood ratio $L_{k,k'}(\mathbf{k}_{-m})$ exhibits local symmetric monotonicity. Namely,*

$$L_{k,k'}(\mathbf{k}) > L_{k,k'}(\mathbf{k} + \epsilon\mathbf{i}) \quad \textit{for all} \quad k' > k \quad \textit{and} \quad \epsilon > 0,$$

where $\mathbf{k}$ is a vector of $M - 1$ repetitions of $k > 0$, and $\mathbf{i}$ is the basis vector for the i-th dimension.[9]

Proof *in Appendix A.*

In Lemma 2, the term 'symmetric' is used because the monotonicity condition is defined under the assumption that competing miners choose the same block capacity. It is also possible to prove local asymmetric monotonicity, where elements of $\mathbf{k}_{-m}$ differ, but doing so involves more complex computations. However, in Appendix A, we prove it for a specifically chosen value of $\mathbf{k}_{-m}$. This ensures that the resulting monotonicity of $L_{k,k'}(\mathbf{k}_{-m})$ implies that uniqueness of the Nash equilibrium of the game.

[9] $\mathbf{i} = (0, \ldots, 1, \ldots, 0)$.

Nash Equilibrium

A Nash equilibrium in the block proposal game with state Q is a block capacity vector $\mathbf{k} = (k_1, k_2, \ldots, k_M)$ such that

$$k_m \in \arg\max_{k \in [0,Q]} R(k, \mathbf{k}_{-m}), \quad \text{for all } m.$$

The next proposition shows that the block proposal game admits a unique and symmetric Nash equilibrium.

Proposition 1. *The block proposal game has a unique and symmetric Nash equilibrium block capacity, given by* $k(Q) = \min\left\{Q, k_m(\pi, \tau, \mu)^+\right\}$, *where*

$$k_m(\pi, \tau, \mu) = \frac{(\mu_{-m})^{-1}}{\Delta} - \frac{\pi}{\tau}, \tag{3}$$

and $\mu_{-m} \equiv \mu \frac{M-1}{M}$ *is the block production rate of the other miners.*

Proof *in Appendix A.*

Proposition 1 shows that the equilibrium block capacity $k(\pi, \tau, \mu)$, when positive and not constrained by the mempool size, is a function of two ratios: on the one hand, it increases linearly with the ratio of the block production time of the other miners, μ_{-m}^{-1}, to the transmission delay per unit data, Δ. On the other hand, it decreases linearly with the ratio of the coinbase reward π to the transaction fee rate τ.

These effects are intuitive: if the transmission delay is short relative to the block production time, then the increase in forking risk caused by recording more transactions becomes less of a concern, thus miners increase block capacity. Conversely, the coinbase reward is independent of block capacity, and hence the higher it is, the lower miners' incentive to record transactions and increase the invalidation risk.

Notice also that block capacity depends on the mining power of the other miners, represented by $\mu_{-m} = \mu(M-1)/M$. As the number of miners M decreases (i.e., mining power is concentrated among fewer miners), μ_{-m} decreases, leading to a higher $k_m(\pi, \tau, \mu)$. This implies that a miner with significant mining power has an incentive to propose larger blocks because the risk of its block being invalidated by others is lower. Conversely, when there are many small miners (large M), $\mu_{-m} \approx \mu$, and the individual miner's incentive to propose larger blocks diminishes due to increased competition and higher invalidation risks. In the limit as M approaches infinity, we have

$$k_m(\pi, \tau, \mu) \approx k(\pi, \tau, \mu) \equiv \frac{\mu^{-1}}{\Delta} - \frac{\pi}{\tau}. \tag{4}$$

Interestingly, when $k(Q)$ is interior, miner revenues on a *single* block are independent of the coinbase reward, as they are given by

$$\pi + \tau k_m(\pi, \tau, \mu) = \tau \times \frac{\mu_{-m}^{-1}}{\Delta}.$$

This happens because increasing π leads miners to produce smaller blocks, causing an exact compensatory reduction in transaction fee income. However, π will enter the calculation of the *present value* of revenues as it affects the fork rate and hence the frequency at which miners collect their rewards. We will explore this aspect next.

2.3 Equilibrium Blockchain Growth and Block Revenues

In equilibrium, each block will record an amount of data given by $k(Q)$ from Proposition 1. Nevertheless, not all blocks produced by miners will be included in the blockchain: unless $k(Q) = 0$ for every Q, the blockchain grows at a rate $\lambda < \mu$ because some of the produced blocks may become orphaned due to forks and thus discarded.

As illustrated in Fig. 1, for $k(Q) \equiv k$, a block B contributes to the blockchain growth only if $t_B - t_{B-1} \geq \Delta k$; that is, with probability $e^{-\mu \Delta k}$. For analytical tractability, we approximate the process of blocks by a thinned Poisson process with rate[10]

$$\lambda = \mu e^{-\mu \Delta k}. \tag{5}$$

This approximation keeps the process Markovian and is standard in the computer science literature—see, for example, Ren (2019a), Dembo et al. (2020), and Doger et al. (2024).

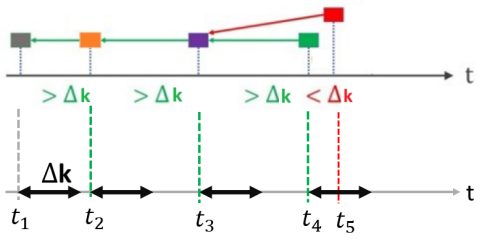


Figure 1: Fork probability

When evaluated at $k(\pi, \tau, \mu)$ from Eq. (4), the blockchain growth rate becomes $\lambda =$

[10] Fig. 1 is adapted from Ling Ren's blog post on Nakamoto Consensus (Ren, 2019b).

$\mu e^{-1+\mu\Delta\pi/\tau}$, and the resulting rate of miner revenues becomes[11]

$$\lambda\left(\pi+\tau k(\pi,\tau,\mu)\right)=\begin{cases}\dfrac{\tau}{\Delta}e^{-1+\mu\Delta\pi/\tau} & \text{if} \quad \dfrac{\mu^{-1}}{\Delta}>\dfrac{\pi}{\tau},\\ \mu\pi & \text{if} \quad \dfrac{\mu^{-1}}{\Delta}\leq\dfrac{\pi}{\tau}\end{cases}. \tag{6}$$

Note that, when $\mu^{-1}/\Delta > \pi/\tau$, block capacity is strictly positive ($k(Q) > 0$). In this case, π and τ can act as complements in raising miner revenues. Specifically, setting a positive π can lead to higher overall revenues because it incentivizes miners to produce smaller blocks, thereby reducing the probability of forks and increasing the effective block inclusion rate. On the other hand, when $\mu^{-1}/\Delta \leq \pi/\tau$, block capacity is zero, and revenues are solely from the coinbase reward, earned at the rate of block production μ.

The Miner-Optimal Block Capacity

The block capacity $k(\pi, \tau, \mu)$, besides being the Nash equilibrium of the game (provided a sufficiently large mempool), is also the one that maximizes miner revenues. In fact, with one line of algebra, one can immediately verify that

$$\arg\max_{k} \quad \mu e^{-\mu\Delta k}(\pi+\tau k) = k(\pi,\tau,\mu).$$

In conclusion:

Proposition 2. *The Nash equilibrium block capacity in Eq. (3) maximizes miner revenues.*

This proposition closes the analysis of miner strategic behavior and resulting payoffs. The next two sections will characterize the dynamics implied by Nakamoto consensus and discuss blockchain design implications of the theory.

3 Blockchain and Mempool Dynamics

In this section, we analyze the dynamics of the mempool and the blockchain growth, focusing on how they affect user experience. To model this, we assume that data flow continuously into the mempool at rate $\alpha > 0$ per unit of time. Thus, the mempool evolves according to a stochastic inventory model, with continuous entry and discrete release of k units of data in each block. This stochastic inventory representation extends queuing theory models from the literature, such as the batch-service queue proposed by Huberman et al. (2019),

[11] Notice that the right side of Eq. (6) is a continuous function of μ^{-1}/Δ.

to a continuous setting. As a complement to the core analysis, Appendix C provides an characterization of mempool dynamics using a random-walk representation. This last will lead to identical dynamics as its continuous-time counterpart.

Continuous-Time Process Representation

We model the dynamics of the mempool size Q_t as a jump-diffusion process constrained by the lower barrier $Q_t \geq 0$. Specifically, this process is described by the following stochastic differential equation (SDE):

$$\mathrm{d}Q_t = \alpha\,\mathrm{d}t - \min(k, Q_t)\,\mathrm{d}B_t, \tag{7}$$

where B_t is a Poisson counting process of blocks arriving at rate λ.

To better understand the behavior of Q_t, Fig. 2 illustrates its dynamics. The mempool size increases continuously due to the inflow of data at rate α. When a block is successfully added to the blockchain (e.g., at times t_1, t_2, t_4), up to k units of data are removed from the mempool. However, not all blocks produced by miners are included in the blockchain due to possible forks; for example, the block generated at t_3 is invalidated because $t_3 - t_2 < \Delta k$. Additionally, if the mempool contains less than k units of data when a new block is created (e.g., at time t_5), the block will be only partially utilized.

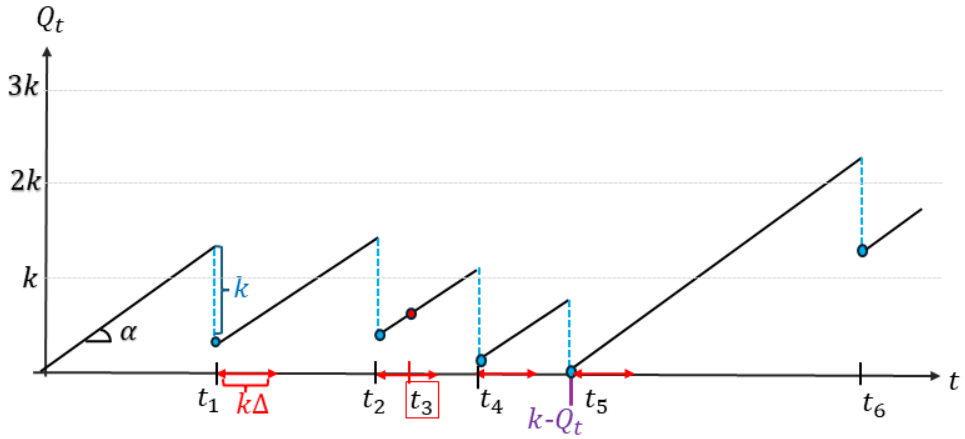


Figure 2: Dynamics of Q_t

To determine under what conditions a stationary distribution for Q_t exists, we analyze the long-run behavior of the dynamical system. Interestingly, this behavior depends on a single parameter: the *load*, defined as

$$\rho \equiv \frac{\alpha}{\lambda k}. \tag{8}$$

The load ρ represents the ratio of the data inflow rate to the data processing rate of the blockchain. The restriction $\rho < 1$ guarantees the existence of a stationary distribution, as it ensures that the mempool does not grow indefinitely. Intuitively, when $\rho < 1$, the blockchain can process transactions faster than they arrive on average, preventing unbounded growth of the mempool.

For convenience, we can re-parameterize the stochastic process in Eq. (7) so that ρ becomes the sole parameter of the associated stochastic process. By expressing time and mempool size in block units, we consider the following change of variables:

$$b = \frac{Q}{k}, \quad x = \lambda t.$$

Dividing both sides of Eq. (7) by k and noting that $\mathrm{d}t = \frac{\mathrm{d}x}{\lambda}$, we obtain the normalized SDE:

$$\mathrm{d}b_x = \rho \, \mathrm{d}x - \min(1, b_x) \, \mathrm{d}B_x, \tag{9}$$

where B_x is a Poisson process with unit rate.

In Appendix A, we show that the Fokker-Planck equation corresponding to the SDE in Eq. (9) is given by:

$$\partial_x f_x(b) = -\rho \, \partial_b f_x(b) + \left[f_x(b+1) - f_x(b) \right], \quad \text{for } b \in (0, \infty). \tag{10}$$

By setting $f_x(b) = f(b)$ for all x and imposing the condition $\partial_x f(b) = 0$, we derive the steady-state distribution. As stated in the following proposition, the equilibrium stationary distribution $f(b)$ is exponential.

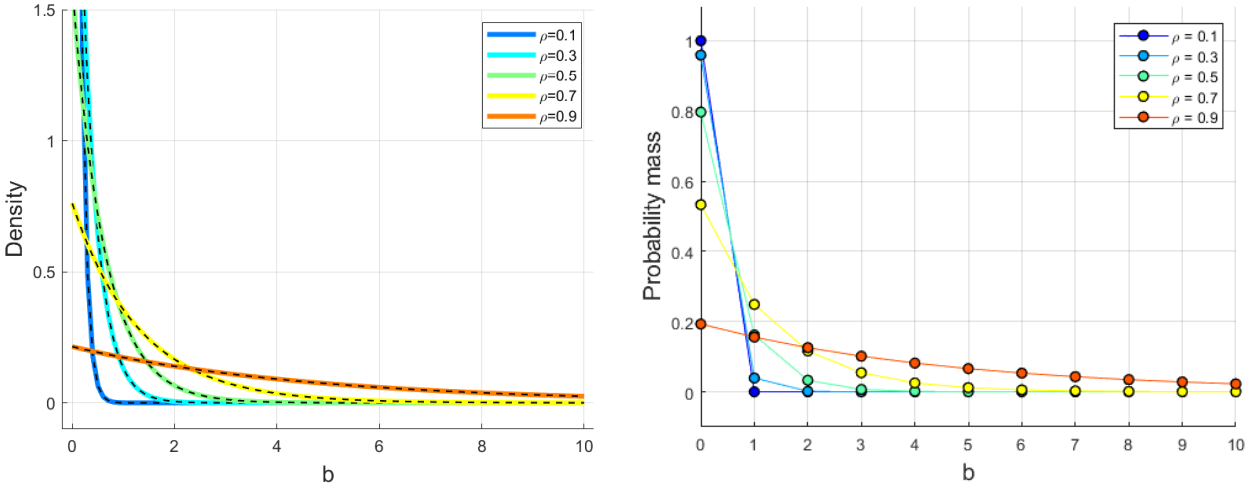
Proposition 3. *The mempool Q_t admits a stationary distribution for $\rho \in [0, 1)$. Furthermore, the mempool process expressed in block units, as in Eq. (9), admits a unique stationary exponential distribution,*

$$f(b) = \kappa(\rho) e^{-\kappa(\rho) b}, \qquad \kappa(\rho) \equiv \frac{1}{\rho} + \mathcal{W}\left(-\frac{1}{\rho} e^{-1/\rho} \right), \tag{11}$$

where $\mathcal{W}$ is the Lambert W function.

Proof *in Appendix A.*

Fig. 3 illustrates the stationary distribution $f(b)$ for various values of ρ. The left panel depicts the density function, while the right panel shows the probability mass function for observing b pending blocks of data in the mempool, given by $P_b(\rho) = e^{-\kappa(\rho) b} - e^{-\kappa(\rho)(b+1)}$.



(a) Stationary distribution density $f(b)$ (b) Probability of observing b pending blocks

Figure 3: Stationary distribution of the mempool

3.1 User Performance Metrics

In this subsection, we develop metrics to evaluate the blockchain's performance from the users' perspective. Since users experience disutility from waiting, minimizing transaction confirmation times is crucial for user satisfaction. Therefore, we introduce two key metrics related to transaction inclusion times: the partial-utilization probability and the block-inclusion probability.

The first important metric is the *partial-utilization probability*, $P_0(\rho)$, which measures the probability that a transaction arriving at the mempool at a given point in time will be included *for sure* in the next produced block. This occurs when the next block has enough capacity to accommodate all pending transactions, meaning the system is underutilized. Mathematically, this probability is given by

$$P_0(\rho) \equiv \Pr(b \leq 1) = 1 - e^{-\kappa(\rho)}. \tag{12}$$

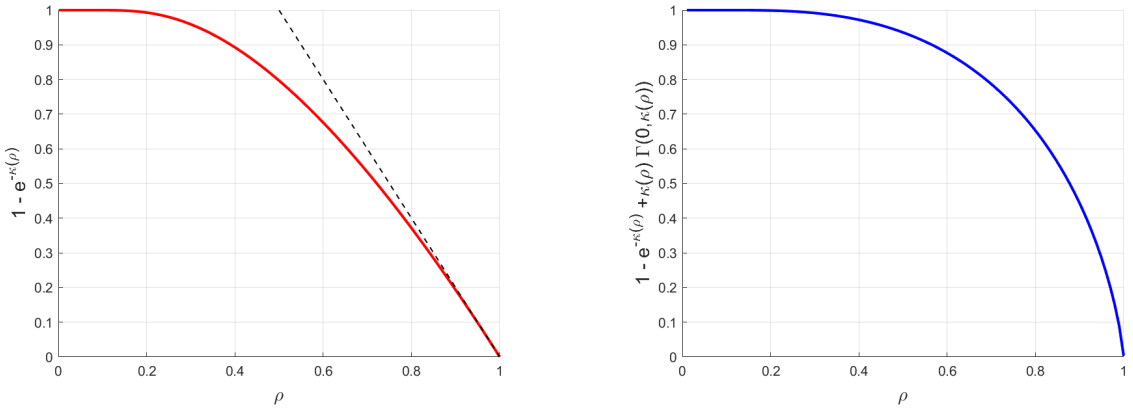
As shown in Fig. 4a, $P_0(\rho)$ decreases as the load ρ increases, indicating that blocks are more likely to be fully utilized and unable to include all pending transactions when the system is heavily loaded. Specifically, $P_0(0) = 1$ (every transaction is included immediately when there is no load), and $\lim_{\rho \to 1} P_0(\rho) = 0$ (transactions are rarely included immediately at high loads).

The second performance metric is the *block-inclusion probability*, $\nu(\rho)$, which quantifies the likelihood that a user's transaction is included in the next block, even when the next block cannot process all pending transactions simultaneously. This metric is particularly

important under heavy load. Assuming a Random-Order-of-Service (ROS), where each unit of data is equally likely to be included, the probability that a random transaction is included in the next block is:

$$\nu(\rho) = \int_0^\infty \frac{1}{\max(1,b)} f(b)\,\mathrm{d}b = 1 - e^{-\kappa(\rho)} + \kappa(\rho)\,\Gamma(0,\kappa(\rho)), \tag{13}$$

where $\Gamma(\cdot,\cdot)$ is the incomplete Gamma function. Fig. 4b illustrates that $\nu(\rho)$ is decreasing and concave in ρ.



(a) Partial-utilization probability $P_0(\rho)$ (b) Block-inclusion probability $\nu(\rho)$

Figure 4: User performance metrics and load

These performance metrics highlight that the system's load ρ is a fundamental determinant of user experience. A lower load leads to higher probabilities of immediate transaction inclusion, reducing users' disutility from waiting. In contrast, a higher load results in more congestion and delays, increasing the waiting time for transactions to be recorded. Therefore, we can directly use the load ρ as a measure of performance from the users' perspective, aiming to keep ρ low to enhance user satisfaction.

3.2 Miner Value Function and Participation

With the mempool dynamics and user performance metrics established, we turn to the miners' perspective to determine their profits and participation levels. Miners obtain revenues from block rewards and transaction fees, but they also incur costs from energy consumption and potential delays due to forks. Therefore, a miner's profit per unit of time is given by

$$V_m = \frac{\lambda}{M}\left(\pi + \tau k(\pi,\tau,\mu)\right) - c - O\,(1-\rho)\,. \tag{14}$$

Here, miners pay an energy cost c per unit of time. Blocks are added to the blockchain at rate λ, and each miner has a $1/M$ chance of being selected to produce a block. The term $O(1-\rho)$ accounts for the reduction in transaction fee revenues when the blockchain is in a partial-utilization state ($Q < k$). As Fig. 4a indicates, this adjustment becomes negligible when ρ is close to 1, which is often the case in practice. For tractability, we assume this adjustment can be safely ignored.

Miner Entry

To determine the equilibrium level of miner participation, we consider the case of atomistic miners, treating their participation as a continuum. We fix the total mining power to $M \in \mathbb{R}_+$ and assume each miner has infinitesimal power $\mathrm{d}m$, faces a marginal energy cost $c\,\mathrm{d}m$, and produces blocks at rate $\mu\,\mathrm{d}m/M$.

Under free entry, miners will participate until their expected profit is zero. Therefore, M is determined by the condition:

$$\lim_{\mathrm{d}m \to 0} \frac{\left(\lambda\,\mathrm{d}m/M\right)\left[\pi + \tau k_m(\pi,\tau,\mu)\right]}{c\,\mathrm{d}m} = 1.$$

Proposition 4. *Miner participation under free entry of atomistic miners is given by*

$$M = \frac{\lambda\left(\pi + \tau k(\pi,\tau,\mu)\right)}{c}, \tag{15}$$

where $k(\pi,\tau,\mu)$ is defined by Eq. (4) and λ is defined by Eq. (5).

As discussed in Section 2.2, in the atomistic limit, equilibrium block capacity depends on the total block rate, so miners have no incentive to produce blocks that are suboptimally large.

4 Blockchain Design

In this section, we analyze the implications of the above theory for designing miner compensation and block production rates. We take the objectives of the design exercise to be the *efficiency* and *security* of the blockchain.[12]

By *efficiency*, we refer to how quickly pending data are processed by miners. A natural metric for measuring efficiency is the load ρ from Eq. (8). For a *fixed* demand for data

[12] These concepts closely resemble the notions of *liveness* and *safety* in the distributed algorithms literature.

storage, a lower load reduces the time it takes for pending data to be recorded. Therefore, in this context, maximizing efficiency corresponds to minimizing the load ρ.

Security, on the other hand, refers to the resilience of the blockchain protocol against malicious attacks. We consider security to be achieved when miner participation M is sufficiently high relative to the mining power of a potential attacker. Thus, a security requirement for the blockchain can be stated as[13]

$$M \geq A, \tag{16}$$

where M is miner participation determined by Eq. (15), and A is a safety threshold. This approach is analogous to that used in the computer science literature to discuss the resilience of blockchain protocols against Byzantine (arbitrarily malicious) attackers.

4.1 The Efficient Load

We focus on tuning the parameters of the model to minimize the load (maximize efficiency) subject to the security constraint in Eq. (16). Before considering the incentive-constrained problem, it is useful to define the *efficient load* as a benchmark:

$$\rho^* = \min_{(\mu,k)\in\mathbb{R}^2_+} \quad \frac{\alpha}{\mu k e^{-\mu\Delta k}}, \tag{17}$$

It is evident that the block capacity and block rate that solve Eq. (17) are given by pairs (μ^*, k^*) such that

$$\mu^* = \frac{1}{\Delta k^*}, \quad k^* \in (0, \infty). \tag{18}$$

Here, efficiency requires the block production time μ^{-1} to equal the transmission delay for a block of equilibrium capacity Δk, optimally trading off data inclusion and forking risk. This result confirms an intuition already stated in the economics and computer science literature (John et al., 2020; Pass and Shi, 2017) and demonstrates its generality for the case of arbitrary block capacity. Furthermore, since ρ depends only on the product of μ and k, the two variables are perfect substitutes in the optimization problem and can be used interchangeably to achieve the efficient load. Notice that the efficient load

$$\rho^* = e \times \frac{\Delta}{\alpha^{-1}}, \tag{19}$$

is the ratio of the transmission delay to the arrival time of a unit of data, scaled by the Euler

[13]Technically, the requirement is typically stated with a strict inequality, but for a well-defined solution of the design problem, we take it as weak.

number e.

Efficiency-Security Tradeoff

Comparing k^* from Eq. (18) with $k(\pi, \tau, \mu)$ from Eq. (3), we observe a tension between efficiency and security, as

$$k^* = k(\pi, \tau, \mu) + \frac{\pi}{\tau}.$$

Since $\frac{\pi}{\tau} \geq 0$, it follows that $k(\pi, \tau, \mu) \leq k^*$, with equality only if $\pi = 0$. So, unless the coinbase reward π is null, miners have an incentive to choose a lower block capacity than the efficient one. Lowering capacity allows miners to collect the coinbase reward more frequently, thereby increasing their profits.

Proposition 5. *To minimize the load, the block production time should be inversely proportional to the block transmission delay. It follows that the efficient block capacity is higher than the Nash equilibrium block capacity unless $\pi = 0$.*

Remark. On the contrary, the blockchain growth rate, λ, is maximized when $\tau = 0$. A null fee rate results in empty blocks and a blockchain that grows without forks.

4.2 The Incentive-Compatible and Security-Constrained Load

We now consider the case where $k = k(\pi, \tau, \mu)$. The design problem becomes:

$$\rho^{**} = \min_{(\pi,\tau,\mu)\in\mathbb{R}^3_+} \frac{\alpha}{\mu k(\pi, \tau, \mu) e^{-\mu\Delta k(\pi,\tau,\mu)}}, \tag{20}$$

subject to the security constraint in Eq. (16) and exogenous upper bound on the fee rate: $\tau \in [0, \overline{\tau}]$. This constraint account for the fact that an excessively high fee rate could discourage blockchain usage.

For convenience, we restate the security constraint as

$$M = \frac{(\tau/\Delta) \times \exp\left(-1 + \mu\Delta\pi/\tau\right)}{c} \geq A. \tag{21}$$

Constrained Solution

The efficient load can be achieved as long as setting $\pi = 0$ is feasible and the security constraint can be satisfied using only τ. This requires $\overline{\tau} \geq \hat{\tau}^*$, where

$$\frac{\hat{\tau}^*}{\Delta} = e \times Ac. \tag{22}$$

Essentially, the marginal fee on transmitted data must outweigh the marginal cost of attacking the blockchain by a factor of e (approximately 2.718).[14]

In the alternative case, satisfying the security constraint requires using both the coinbase reward and transaction fees. By setting $\tau = \overline{\tau}$, we observe that

$$\lambda k(\pi, \overline{\tau}, \mu) = \left(\frac{1}{\Delta} - \frac{\mu\pi}{\overline{\tau}}\right) e^{-1+\mu\Delta\pi/\overline{\tau}}, \tag{23}$$

is a decreasing function of $\mu\pi$. Therefore, the second-best load can be achieved by setting $\mu\pi$ as low as possible. The optimal μ^{**} and π^{**} satisfy

$$\frac{\pi^{**}}{\overline{\tau}} = \frac{1}{\Delta\mu^{**}}\left[1 - \ln\left(\frac{\overline{\tau}/\Delta}{Ac}\right)\right], \qquad k(\pi^{**}, \overline{\tau}, \mu^{**}) = \frac{1}{\Delta\mu^{**}} \ln\left(\frac{\overline{\tau}/\Delta}{Ac}\right).$$

In this scenario, block capacity is suboptimally low. Moreover, transaction fees must be sufficiently high for blocks to have positive capacity, which requires $\overline{\tau} \geq \hat{\tau}^{**} = Ac\Delta$. If this condition is not met, the design problem has no solution.

Proposition 6. *The solution to the incentive- and security-constrained design problem depends on the parameter $\overline{\tau}$:*

1. *For $\dfrac{\overline{\tau}}{\Delta} > e \times Ac$, the efficient load is implementable: $\rho^{**} = \rho^*$.*
2. *For $Ac < \dfrac{\overline{\tau}}{\Delta} \leq e \times Ac$, a second-best load is implementable with $\rho^{**} > \rho^*$ and $k^{**} < k^*$.*
3. *For $\dfrac{\overline{\tau}}{\Delta} \leq Ac$, the design problem has no solution.*

5 Extensions

5.1 Uncle Block Rewards

Up to now, we have seen that the risk of block invalidation due to forking induces a trade-off between the speed at which the blockchain can process data (efficiency) and the profitability for miners to join the consensus protocol (security). However, modern blockchains mitigate the inefficiencies created by forking risks by rewarding miners even for producing forked blocks. We now present a variant of the baseline model incorporating this feature, showing that it allows the blockchain designer to achieve maximum efficiency over a wider range of parameters.

[14] Multiplying by $1/\Delta$ expresses the fee rate in units of time rather than units of data.

Consider again the block proposal game of Section 2, but this time suppose that miners are awarded new tokens even if their blocks get forked. Let ϕ denote the coinbase reward for each forked block. The expected payoff for a miner is now given by

$$R(k, \mathbf{k}_{-m}) = \left[P_{\mathsf{Win}}(k, \mathbf{k}_{-m})\,(\pi + k\tau) + P_{\mathsf{Fork}}(k, \mathbf{k}_{-m})\phi\right]. \tag{24}$$

Here, P_{Fork} is the probability that the proposed block is forked by the block proposal tournament's winner:

$$P_{\mathsf{Fork}}(k, \mathbf{k}_{-m}) = \mathbb{P}\left(\min_{m' \neq m}\left\{T_{m'} + \Delta(k_{m'} - k_m)\right\} < T_m < \min_{m' \neq m}\left\{T_{m'} + \Delta k_{m'}\right\}\right). \tag{25}$$

As before, the expressions for $P_{\mathsf{Win}}(k, \mathbf{k}_{-m})$ and $P_{\mathsf{Fork}}(k, \mathbf{k}_{-m})$ are cumbersome for an arbitrary input vector $\mathbf{k} = (k_m, \mathbf{k}_{-m})$. However, for a symmetric action profile $k_{-m} = k$, it is possible to express them neatly. Moreover, even when extending the model with uncle block rewards, it can be shown that there exists a symmetric Nash equilibrium with the following properties:

Proposition 7. *In the block proposal game with uncle block rewards, there exists a symmetric Nash equilibrium given by* $k_m = \min(Q, k(\pi, \tau, \mu, \phi)^+)$*, where*

$$k(\pi, \tau, \mu, \phi) = \frac{(\mu_{-m})^{-1}}{\Delta} - \frac{\pi - \phi}{\tau}. \tag{26}$$

Design Implications

For M sufficiently large, the efficient block capacity $\frac{1}{\Delta\mu}$ can be achieved regardless of τ by setting $\pi = \phi$. Thus, uncle block rewards widen the parameter range over which the efficient block capacity is achieved.

More formally, for $\pi = \phi$, miner participation is given by $M = (\pi + \lambda\tau k)/c$. Therefore, in the presence of an upper bound on the coinbase $\overline{\pi}$, the efficient load can be achieved when $\overline{\tau} \geq A\,(e\Delta c - \overline{\pi})$, which is a wider range than that in Proposition 6, as long as $\overline{\pi} > 0$.

The above conclusion holds as long as we consider a security constraint where the security threshold A is not affected by miner rewards. This corresponds to the case where the designer seeks security against a Byzantine adversary that ignores incentive compatibility. However, if we consider a rational attacker, uncle block rewards can have perverse effects. For example, they can incentivize the attacker to fork the blockchain, as they provide income even if the attack fails.

5.2 Discrete Latency

Up to now, we have considered latency to be linear in k. In this section, we analyze a variant of the model where only blocks of capacity above a threshold $\underline{k}$ face a transmission delay of length Δ. Specifically, the transmission time of a block with capacity k, denoted Δ_k, is given by

$$\Delta_k = \begin{cases} 0, & \text{for } k < \underline{k}, \\ \Delta, & \text{for } k \geq \underline{k}. \end{cases} \tag{27}$$

This model variant may be of interest for aligning the model with empirical data, given that communication over the internet (more precisely, over the TCP/IP protocol) occurs in such a way that data are transmitted in packets rather than continuously. Thus, it is worth investigating a model where adding data contributes to latency only when the additional data form a new data packet. As we will see below, the transmission delay function in Eq. (27) leads to a multiplicity of equilibria.

Likelihood Ratios and Nash Equilibria

In this variant of the model, the decision of a miner m involves choosing $k_m \in \{\underline{k}, Q\}$ to optimize the revenues in Eq. (1), under the belief that $\hat{m}$ other miners are choosing $k_{m'} = \underline{k}$, and consequently, $M - \hat{m} - 1$ other miners are choosing $k_{m'} = Q$.

Letting $P(k_m; \hat{m})$ denote miner m's estimate of the probability of winning the block proposal game, miner m has an incentive to choose $k_m = Q$ if and only if

$$P(Q, \hat{m})\,(\pi + \tau Q) - P(\underline{k}, \hat{m})\,(\pi + \tau \underline{k}) \geq 0. \tag{28}$$

Using Eq. (28), we see that a miner has an incentive to propose $k_m = Q$ rather than $k_m = \underline{k}$ if

$$L_{\underline{k},Q}(\hat{m}) \equiv \frac{P(\underline{k}, \hat{m})}{P(Q, \hat{m})} \leq \frac{\pi/\tau + Q}{\pi/\tau + \underline{k}}. \tag{29}$$

In words, the likelihood ratio of win probabilities must be sufficiently low relative to the threshold on the right side. Notice that the threshold is decreasing in π/τ, making the condition harder to meet. Thus, as in previous iterations of the model, a higher ratio of coinbase to fees reduces the incentives to increase block capacity. The threshold is also decreasing in Q and increasing in $\underline{k}$.

As in Section 2.2, a monotonicity property of the likelihood ratio allows us to characterize the equilibrium set. For this discrete set, we can, in fact, prove monotonicity in $\hat{m}$ rather than the local monotonicity we proved in the baseline model:

Lemma 3. *The likelihood ratio*

$$L_{\underline{k},Q}(\hat{m}) = \frac{M}{\hat{m}+1}\left(e^{\mu\hat{m}\Delta/M} - e^{-\mu\Delta/M}\right) + e^{-\mu\Delta/M} \tag{30}$$

is increasing in $\hat{m}$. It follows that miner payoffs exhibit strategic complementarity in k.

Proof *in Appendix A.*

Here, strategic complementarity means that the more miners switch action from $k_{m'} = \underline{k}$ to $k_{m'} = Q$, the more a miner's payoff from choosing $k_m = Q$ increases relative to the payoff from choosing $k_m = \underline{k}$. Strategic complementarity has two important implications. First, it rules out mixed-strategy equilibria based on stability refinements (Echenique and Edlin, 2004). Second, as we will see soon, it restricts the profile of equilibrium payoffs to symmetric ones: $\mathbf{k} = (k, k, \ldots, k)$.

We can now determine miners' block capacity choices as a Nash equilibrium outcome. In this context, a Nash equilibrium is a pair $(\hat{m}_0, \hat{m}_1)$ representing the number of miners choosing actions $k_m = \underline{k}$ and $k_m = Q$, respectively. Equilibrium requires two incentive-compatibility (IC) constraints to hold: one to prevent a deviation from Q among the $\hat{m}_0$ miners choosing it, and the other to prevent the $M - \hat{m}_0$ miners from deviating from $\underline{k}$. We denote $\text{IC}^{k,k'}$ as the constraint for action k and deviation k'.

$$L_{\underline{k},Q}(\hat{m}_0) \leq \frac{\pi/\tau + Q}{\pi/\tau + \underline{k}}; \tag{IC$^{Q,\underline{k}}$}$$

$$L_{\underline{k},Q}(\hat{m}_0 - 1) \geq \frac{\pi/\tau + Q}{\pi/\tau + \underline{k}}. \tag{IC$^{\underline{k},Q}$}$$

By Lemma 3, the game features strategic complementarity in block capacity. Strategic complementarity rules out asymmetric action profiles since they require both constraints to hold simultaneously, which is impossible because $L_{\underline{k},Q}(\hat{m}_0 - 1) < L_{\underline{k},Q}(\hat{m}_0)$. Therefore, the only possible equilibrium profiles are $(\hat{m}_0, \hat{m}_1) \in \{(M, 0), (0, M)\}$.

Proposition 8. *Under the latency function in Eq. (27), the block proposal game exhibits multiple equilibria in which all miners choose the same block capacity:*

1. *For $\dfrac{\pi/\tau + Q}{\pi/\tau + \underline{k}} \leq L_{\underline{k},Q}(M-1)$, $(\hat{m}_0, \hat{m}_1) = (M, 0)$ is the unique Nash equilibrium.*

2. *For $L_{\underline{k},Q}(0) \leq \dfrac{\pi/\tau + Q}{\pi/\tau + \underline{k}} \leq L_{\underline{k},Q}(M-1)$, the game has two Nash equilibria:*

$$(\hat{m}_0, \hat{m}_1) \in \{(M, 0), (0, M)\}.$$

3. *For* $\dfrac{\pi/\tau + Q}{\pi/\tau + \underline{k}} \geq L_{\underline{k},Q}(0)$, $(\hat{m}_0, \hat{m}_1) = (0, M)$ *is the unique Nash equilibrium.*

Despite the multiplicity of equilibria described in Proposition 8, block capacity satisfies the same comparative statics properties as is Section 2.2. Namely, it increases μ^{-1}/Δ and decreases in π/τ.

6 Concluding Remarks

In this paper, we analyzed incentives in Nakamoto-style consensus protocols, focusing on how network latency impacts the tradeoffs that determine miners' choice of block capacity. Our theoretical model demonstrates that latency introduces a tension between efficiency (processing transactions quickly) and security (ensuring sufficient miner participation). We showed that under certain conditions, miners may choose a suboptimally small block capacity to maximize their individual profits, potentially reducing the overall efficiency of the blockchain.

By understanding how latency affects miners' decisions on block capacity, protocol designers can better balance efficiency and security to optimize the performance of blockchain networks. Future research could extend our model to consider additional factors such as variable network conditions, miner heterogeneity, or the impact of mining pools on block capacity decisions.

To illustrate the practical implications of our findings, we conclude by discussing how these tradeoffs manifest in popular blockchains such as Bitcoin, Ethereum, and Cardano, and present testable implications for future research.

Bitcoin

In Bitcoin, the block production time is considerably longer than the block propagation time. While the block production time is set at 10 minutes, the propagation time for a 1 MB block is on the order of 2 seconds at most. Under these parameters, the ratio μ^{-1}/Δ is approximately 300MB. This suggests that the theoretical optimal block capacity is much larger than the current 2MB limit imposed by the protocol.[15] Given that the coinbase reward is still substantial relative to transaction fees for most Bitcoin blocks, miners likely perceive the risk of accidental forking as minimal. This implies that Bitcoin's block capacity is shaped more by protocol-imposed limits and other factors rather than by latency-induced

[15]The theoretical upper bound on block capacity in Bitcoin is 4MB, but for all practical purposes it is 2MB.

incentives. For example, Lehar and Parlour (2020) suggest that block capacity choices may be influenced by miner collusion.

Ethereum Pre-Merge and EIP-1559

In Ethereum, the faster block production time (with block times around 12 seconds) makes latency more consequential than in Bitcoin. The impact of larger block capacity on forking has been acknowledged in previous studies and technical discussions (Liu et al., 2022; Buterin and Griffith, 2017). Moreover, the presence of frequent forks due to shorter block times led to the adoption of the GHOST protocol, a variant of Nakamoto consensus where miners receive rewards for uncle blocks (stale blocks).

The tradeoffs discussed in this paper highlight potential limitations of Ethereum's EIP-1559 transaction fee mechanism in managing block capacity. While EIP-1559 aims to maintain an equilibrium block capacity by dynamically adjusting a base fee, this fee is burned, offering no direct incentives to miners. As noted by Roughgarden (2021), burning the base fee reduces the risk of off-chain collusion and ensures strategy-proofness in low-congestion scenarios. However, in the context of latency tradeoffs, the absence of direct miner compensation could undermine the mechanism's ability to regulate block capacity, particularly during demand-side shocks. Previous studies by Leonardos et al. (2021) and Reijsbergen et al. (2021) have identified the instability of EIP-1559 during periods of fluctuating demand.

Ethereum Proof of Stake

The shift to an epoch-based Proof of Stake (PoS) system reduces the significance of latency on the block proposal race, as block proposers are known ahead of time. In this context, abstracting from MEV, block capacity decisions can be modeled as an individual optimization problem. Due to the absence of competition over propagation speed, the block capacity choice primarily depends on the length of the time slot allocated for validators to propose blocks. This suggests that involuntary forks (also known as "reorgs" in this context) will be less common than in the previous Ethereum version.

Cardano

Cardano utilizes an epoch-based PoS system integrated with Verifiable Random Functions (VRFs) (Micali et al., 1999). Unlike in PoS Ethereum, where the identity of the next block proposer is public, in Cardano, this information is known only to the block proposer due to the use of VRFs. This feature introduces competition at the block proposal phase, making the incentive issues discussed in this paper more salient than in epoch-based Ethereum. From

the perspective of validator rewards, Cardano employs a mechanism where transaction fees are proportional to the size of the data stored in blocks, aligning its design with the incentive compatibility conditions emphasized in this paper.

References

Amoussou-Guenou, Y., Biais, B., Potop-Butucaru, M., and Tucci-Piergiovanni, S. (2023). Committee-based blockchains as games between opportunistic players and adversaries. *The Review of Financial Studies*, page hhad051.

Budish, E., Cramton, P., and Shim, J. (2015). The high-frequency trading arms race: Frequent batch auctions as a market design response. *The Quarterly Journal of Economics*, 130(4):1547–1621.

Buterin, V. and Griffith, V. (2017). Casper the friendly finality gadget. *arXiv preprint arXiv:1710.09437*.

Cartea, Á., Jaimungal, S., and Sánchez-Betancourt, L. (2021). Latency and liquidity risk. *International Journal of Theoretical and Applied Finance*, 24(06n07):2150035.

Cartea, Á. and Sánchez-Betancourt, L. (2023). Optimal execution with stochastic delay. *Finance and Stochastics*, 27(1):1–47.

Chung, H. and Shi, E. (2022). Foundations of transaction fee mechanism design.

Daian, P., Goldfeder, S., Kell, T., Li, Y., Zhao, X., Bentov, I., Breidenbach, L., and Juels, A. (2019). Flash boys 2.0: Frontrunning, transaction reordering, and consensus instability in decentralized exchanges.

Dembo, A., Kannan, S., Tas, E. N., Tse, D., Viswanath, P., Wang, X., and Zeitouni, O. (2020). Everything is a race and nakamoto always wins. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 859–878.

Doger, M., Ulukus, S., and Akar, N. (2024). Pow security-latency under random delays and the effect of transaction fees. *arXiv preprint arXiv:2405.04526*.

Dumont, G. and Gabriel, P. (2020). The mean-field equation of a leaky integrate-and-fire neural network: measure solutions and steady states. *Nonlinearity*, 33(12):6381–6420.

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

Echenique, F. and Edlin, A. S. (2004). Mixed equilibria are unstable in games of strategic complements. *Journal of Economic Theory*, 118(1):61–79.

Gao, X. and Wang, Y. (2020). Optimal market making in the presence of latency. *Quantitative Finance*, 20(9):1495–1512.

Hinzen, F. J., John, K., and Saleh, F. (2019). Proof-of-work's limited adoption problem. *NYU Stern School of Business*.

Houy, N. (2014). The economics of bitcoin transaction fees. *GATE WP*, 1407.

Huberman, G., Leshno, J., and Moallemi, C. C. (2019). An economic analysis of the bitcoin payment system. *Columbia Business School Research Paper*, (17-92).

Jansson, B. (1966). Choosing a good appointment system-a study of queues of the type (d, m, 1). *Operations Research*, 14(2):292–312.

John, K., Monnot, B., Mueller, P., Saleh, F., and Schwarz-Schilling, C. (2024). Economics of ethereum. *Available at SSRN 4783695*.

John, K., Rivera, T. J., and Saleh, F. (2020). Economic implications of scaling blockchains: Why the consensus protocol matters. *Available at SSRN*.

Lehar, A. and Parlour, C. A. (2020). Miner collusion and the bitcoin protocol. *Available at SSRN 3559894*.

Leonardos, S., Monnot, B., Reijsbergen, D., Skoulakis, E., and Piliouras, G. (2021). Dynamical analysis of the eip-1559 ethereum fee market. In *Proceedings of the 3rd ACM Conference on Advances in Financial Technologies*, pages 114–126.

Liu, Y., Lu, Y., Nayak, K., Zhang, F., Zhang, L., and Zhao, Y. (2022). Empirical analysis of eip-1559: Transaction fees, waiting time, and consensus security. *arXiv preprint arXiv:2201.05574*.

Meyn, S. P. and Tweedie, R. L. (1993). Stability of markovian processes iii: Foster–lyapunov criteria for continuous-time processes. *Advances in Applied Probability*, 25(3):518–548.

Micali, S., Rabin, M. O., and Vadhan, S. P. (1999). Verifiable random functions. *40th Annual Symposium on Foundations of Computer Science (Cat. No.99CB37039)*, pages 120–130.

Mighan, S. N., Mišić, J., and Mišić, V. B. (2022). On block delivery time in ethereum network. In *GLOBECOM 2022-2022 IEEE Global Communications Conference*, pages 2867–2872. IEEE.

Moallemi, C. C. and Sağlam, M. (2013). The cost of latency in high-frequency trading. *Operations Research*, 61(5):1070–1086.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. *Cryptography Mailing list at https://metzdowd.com*.

Pass, R. and Shi, E. (2017). Rethinking large-scale consensus. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pages 115–129. IEEE.

Porteus, E. L. (2002). *Foundations of stochastic inventory theory*. Stanford University Press.

Reijsbergen, D., Sridhar, S., Monnot, B., Leonardos, S., Skoulakis, S., and Piliouras, G. (2021). Transaction fees on a honeymoon: Ethereum's eip-1559 one month later. In *2021 IEEE International Conference on Blockchain (Blockchain)*, pages 196–204. IEEE.

Ren, L. (2019a). Analysis of nakamoto consensus. *Cryptology ePrint Archive*.

Ren, L. (2019b). Security proof for nakamoto consensus. Decentralized Thoughts: https://decentralizedthoughts.github.io/2019-11-29-Analysis-Nakamoto/. Accessed: 2023-10-18.

Roughgarden, T. (2020). Transaction fee mechanism design for the ethereum blockchain: An economic analysis of eip-1559.

Roughgarden, T. (2021). Transaction fee mechanism design.

Schwarz-Schilling, C., Saleh, F., Thiery, T., Pan, J., Shah, N., and Monnot, B. (2023). Time is money: Strategic timing games in proof-of-stake protocols.

Shahsavari, Y., Zhang, K., and Talhi, C. (2022). A theoretical model for block propagation analysis in bitcoin network. *IEEE Transactions on Engineering Management*, 69(4):1459–1476.

Wu, F., Thiery, T., Leonardos, S., and Ventre, C. (2024). Strategic bidding wars in on-chain auctions.

Öz, B., Sui, D., Thiery, T., and Matthes, F. (2024). Who wins ethereum block building auctions and why?

Appendix A Proofs

Proof of Lemma 2. For convenience, let us define $k_i \equiv k + \epsilon$, $\boldsymbol{k}_{-m} = (k, \ldots, k)$ and $\boldsymbol{k}'_{-m} = \boldsymbol{k}_{-m} + \boldsymbol{i}(k_i - k)$. To compare the likelihood ratios $L_{k,k'}(\mathbf{k}_{-m})$ and $L_{k,k'}(\mathbf{k}'_{-m})$, we first determine the following probabilities:

$$P(k, \boldsymbol{k}_{-m}) = \int_0^\infty \prod_{m' \in \{1,2\ldots,M\}\setminus\{m\}} \mathbb{P}\left(T_{m'} \geq T_m\right) \mathrm{d}f_m(t) = \int_0^\infty (e^{-(M-1)\mu_m t})(\mu_m e^{-\mu_m t})\, \mathrm{d}t = \frac{1}{M}$$

$$P(k', \boldsymbol{k}_{-m}) = \int_0^\infty \prod_{m' \in \{1,2\ldots,M\}\setminus\{m\}} \mathbb{P}\left(T_{m'} \geq T_m + \Delta(k'-k)\right) \mathrm{d}f_m(t) = \frac{1}{M} e^{-(M-1)\mu_m \Delta(k'-k)}$$

$$\begin{aligned} P(k, \boldsymbol{k}'_{-m}) &= \int_0^\infty \prod_{m' \in \{1,2\ldots,M\}\setminus\{m,i\}} \mathbb{P}\left(T_{m'} \geq T_m\right) \mathbb{P}\left(T_i \geq T_m - \Delta(k_i - k)\right) \mathrm{d}f_m(t) \\ &= \mu_m \int_0^\infty e^{-(M-2)\mu_m t} e^{-\mu_m (t - \Delta(k_i-k))^+} e^{-\mu_m t}\, \mathrm{d}t \\ &= \mu_m \int_0^{\Delta(k_i-k)} e^{-(M-1)\mu_m t}\, \mathrm{d}t + \mu_m e^{\mu_m \Delta(k_i-k)} \int_{\Delta(k_i-k)}^\infty e^{-M\mu_m t}\, \mathrm{d}t \\ &= \frac{1}{M-1}\left[1 - e^{-(M-1)\mu_m \Delta(k_i-k)}\right] + \frac{1}{M} e^{-(M-1)\mu_m \Delta(k_i-k)} \end{aligned}$$

$$\begin{aligned} P(k', \boldsymbol{k}'_{-m}) &= \int_0^\infty \prod_{m' \in \{1,2\ldots,M\}\setminus\{m,i\}} \mathbb{P}\left(T_{m'} \geq T_m + \Delta(k'-k)\right) \mathbb{P}\left(T_i \geq T_m - \Delta(k'-k_i)\right) \mathrm{d}f_m(t) \\ &= \mu_m e^{-(M-2)\mu_m \Delta(k'-k)} \int_0^\infty e^{-(M-1)\mu_m t} e^{-\mu_m (t - \Delta(k'-k_i))^+}\, \mathrm{d}t, \end{aligned}$$

Thus, for $k' > k_i$,

$$\begin{aligned} \int_0^\infty e^{-(M-1)\mu_m t} e^{-\mu_m (t - \Delta(k'-k_i))^+}\, \mathrm{d}t, &= \int_0^{\Delta(k'-k)} e^{-(M-1)\mu_m t}\, \mathrm{d}t + e^{-\mu_m \Delta(k'-k_i)} \int_{\Delta(k'-k)}^\infty e^{-M\mu_m t}\, \mathrm{d}t \\ &= \frac{1}{\mu_m}\left[\frac{1}{M-1}\left(1 - e^{-(M-1)\mu_m \Delta(k'-k_i)}\right) + \frac{1}{M} e^{-(M-1)\mu_m \Delta(k'-k_i)}\right], \end{aligned}$$

giving

$$P(k', \boldsymbol{k}'_{-m}) = e^{-(M-2)\mu_m \Delta(k'-k)}\left[\frac{1}{M-1}\left(1 - e^{-(M-1)\mu_m \Delta(k'-k_i)}\right) + \frac{1}{M} e^{-(M-1)\mu_m \Delta(k'-k_i)}\right].$$

Conversely, for $k' \leq k_i$,

$$\int_0^\infty e^{-(M-1)\mu_m t} e^{-\mu_m (t-\Delta(k'-k_i))^+} \, \mathrm{d}t, = \int_0^\infty e^{-(M-1)\mu_m t} e^{-\mu_m (t+\Delta(k_i-k'))} \, \mathrm{d}t = \frac{1}{\mu_m} \frac{1}{M} e^{\mu_m \Delta(k_i-k')}$$

so that

$$P(k', \boldsymbol{k}'_{-m}) = e^{-(M-2)\mu_m \Delta(k'-k)} \frac{1}{M} e^{\mu_m \Delta(k_i-k')}.$$

Comparison of likelihood ratios

Now let us prove that $L_{k,k'}(\mathbf{k}_{-m}) > L_{k,k'}(\mathbf{k}'_{-m})$, that is,

$$\frac{P(k', \boldsymbol{k}_{-m})}{P(k, \boldsymbol{k}_{-m})} > \frac{P(k', \boldsymbol{k}'_{-m})}{P(k, \boldsymbol{k}'_{-m})}. \tag{31}$$

For $k' \geq k_i$, Eq. (31) becomes

$$e^{-(M-1)\mu_m \Delta(k'-k)} > \frac{e^{-(M-2)\mu_m \Delta(k'-k)} \left[\frac{1}{M-1} \left(1 - e^{-(M-1)\mu_m \Delta(k'-k_i)} \right) + \frac{1}{M} e^{-(M-1)\mu_m \Delta(k'-k_i)} \right]}{\frac{1}{M-1} \left[1 - e^{-(M-1)\mu_m \Delta(k_i-k)} \right] + \frac{1}{M} e^{-(M-1)\mu_m \Delta(k_i-k)}},$$

which, after a few algebraic manipulations, is equivalent to have (left-hand side) $\mathsf{LHS} > \mathsf{RHS}$ (right-hand side), where

$$\mathsf{LHS} = e^{\mu_m \Delta(k'-k)} \left[\frac{1}{M-1} \left(1 - e^{-(M-1)\mu_m \Delta(k'-k_i)} \right) + \frac{1}{M} e^{-(M-1)\mu_m \Delta(k'-k_i)} \right],$$

$$\mathsf{RHS} = \frac{1}{M-1} \left(1 - e^{-(M-1)\mu_m \Delta(k_i-k)} \right) + \frac{1}{M} e^{-(M-1)\mu_m \Delta(k_i-k)}.$$

The inequality holds because LHS is a strictly increasing function of $\epsilon' \equiv k' - k_i$ and satisfies $\mathsf{LHS}|_{\epsilon'=0} > \mathsf{RHS}$. To verify these properties, notice that

$$\mathsf{LHS}|_{\epsilon'=0} = e^{\mu_m \Delta(k_i-k)} \left[\frac{1}{M-1} + \frac{1}{M} \right] > \mathsf{RHS}$$

since $e^{\mu_m \Delta(k_i-k)} > 1$ and RHS applies weights below one to $1/(M-1)$ and $1/M$. Second,

$$\frac{\partial \mathsf{LHS}}{\partial k_i} = \mu \Delta [\mathsf{LHS} + (1 - 1/M)] > 0.$$

For $k' \leq k_i$, recall that $\epsilon = k_i - k$ and let $z \equiv e^{-(M-1)\mu_m \Delta\epsilon}$. Then, Eq. (31) becomes

$$e^{(M-1)\mu_m \Delta(k'-k)} > \left[\frac{1-z}{M-1} + \frac{z}{M}\right] e^{-\mu_m \Delta\epsilon} e^{(M-1)\mu_m \Delta(k'-k)}.$$

$$\iff 1 > \left[\frac{1-z}{M-1} + \frac{z}{M}\right] e^{-\mu_m \Delta\epsilon} \iff 1 > \left[\frac{1-z}{M-1} + \frac{z}{M}\right] z^{1/(M-1)}.$$

Since $0 < z < 1$, the expression on the right is less than 1, confirming the monotonicity property. ■

Proof of Proposition 1.

Closed-form solution and uniqueness of the symmetric equilibrium

Suppose that $k_m = k$ for all m is a Nash equilibrium. Let $(k, \mathbf{k})$, where the vector $\mathbf{k}$ contains $M-1$ copies of the scalar k, denote the strategy profile. The payoff a miner m achieves at this profile is clearly

$$R(k, \mathbf{k}) = \frac{1}{M}(\pi + \tau k).$$

Now suppose that miner m deviates from the equilibrium by choosing $k + \epsilon$, with $\epsilon > 0$. Let $T_{-m} \equiv \min_{m'}\{T_{m'}\}$, an exponential random variable with rate $\mu_m(M-1)$. Since $\min_{m'}\{T_{m'} + \Delta k\} = \min_{m'}\{T_{m'}\} + \Delta k = T_{-m} + \Delta k$, miner m wins with probability

$$\mathbb{P}\left(T_m + \Delta(k+\epsilon) \leq T_{-m} + \Delta k\right) = \mathbb{P}\left(T_{-m} \geq T_m + \Delta\epsilon\right) = \int e^{-\mu_m(M-1)(t+\Delta\epsilon)} \mu_m e^{-\mu_m t}\, \mathrm{d}t$$

$$= e^{-\mu_m(M-1)\Delta\epsilon} \int \mu_m e^{-\mu_m M t}\, \mathrm{d}t = \frac{1}{M} e^{-\mu_m(M-1)\Delta\epsilon} = \frac{1}{M} e^{-\mu \frac{M-1}{M} \Delta\epsilon}. \tag{32}$$

The payoff of this deviation is

$$R(k+\epsilon, \mathbf{k}) = e^{-\mu \frac{M-1}{M} \Delta\epsilon} \left(\frac{\pi + \tau(k+\epsilon)}{M}\right). \tag{33}$$

Given that $R(k+\epsilon,\mathbf{k}) = R_k(\mathbf{k})$ at $\epsilon = 0$, the necessary condition to preclude the deviation is $\frac{dR(k+\epsilon,\mathbf{k})}{d\epsilon}|_{\epsilon=0} \leq 0$:

$$\frac{dR(k+\epsilon,\mathbf{k})}{d\epsilon} = e^{-\mu\frac{M-1}{M}\Delta\epsilon}\left[\frac{\tau}{M} - \mu\frac{M-1}{M}\Delta\left(\frac{\pi+\tau(k+\epsilon)}{M}\right)\right]$$

$$\left.\frac{dR(k+\epsilon,\mathbf{k})}{d\epsilon}\right|_{\epsilon=0} = \left[\frac{\tau}{M} - \mu\frac{M-1}{M}\Delta\left(\frac{\pi+\tau k}{M}\right)\right] \leq 0 \iff k \geq \frac{1}{\mu\Delta(M-1)/M} - \frac{\pi}{\tau}.$$

The sufficient condition to rule out the deviation is $\frac{d^2R(k+\epsilon,\mathbf{k})}{d\epsilon^2} < 0$ whenever $\frac{dR(k+\epsilon,\mathbf{k})}{d\epsilon} \geq 0$. We can see that such condition holds since

$$\frac{d^2R(k+\epsilon,\mathbf{k})}{d\epsilon^2} = -\left(\mu\frac{M-1}{M}\Delta\right)\left[\frac{dR(k+\epsilon,\mathbf{k})}{d\epsilon} + e^{-\mu\frac{M-1}{M}\Delta\epsilon}\frac{\tau}{M}\right] < 0 \text{ if } \frac{dR(k+\epsilon,\mathbf{k})}{d\epsilon} \geq 0.$$

Now consider a deviation to $k-\epsilon$. The probability of winning the tournament at the new strategy is

$$\begin{aligned}\mathbb{P}\left(T_m + \Delta(k-\epsilon) \leq T_{-m} + \Delta k\right) &= \mathbb{P}\left(T_m \leq T_{-m} + \Delta\epsilon\right) \\ &= 1 - \mathbb{P}\left(T_m \geq T_{-m} + \Delta\epsilon\right) = 1 - \frac{M-1}{M}e^{-\mu\Delta\epsilon/M},\end{aligned} \tag{34}$$

so that

$$R(k-\epsilon,\mathbf{k}) = \left(M - (M-1)e^{-\mu\Delta\epsilon/M}\right)\left(\frac{\pi+\tau(k-\epsilon)}{M}\right). \tag{35}$$

Equilibrium requires:[16]

$$\frac{dR(k-\epsilon,\mathbf{k})}{d\epsilon} = \left[\mu\frac{M-1}{M}\Delta e^{-\mu\Delta\epsilon/M}\left(\frac{\pi+\tau(k-\epsilon)}{M}\right) - \left(M-(M-1)e^{-\mu\Delta\epsilon/M}\right)\frac{\tau}{M}\right]$$

$$\left.\frac{dR(k-\epsilon,\mathbf{k})}{d\epsilon}\right|_{\epsilon=0} = \left[\mu\frac{M-1}{M}\Delta\left(\frac{\pi+\tau k}{M}\right) - \frac{\tau}{M}\right] \leq 0 \iff k \leq \frac{1}{\mu\Delta(M-1)/M} - \frac{\pi}{\tau}$$

$$\frac{d^2R_{k-\epsilon}(\mathbf{k})}{d\epsilon^2} = -\left[\frac{dR_{k-\epsilon}(\mathbf{k})}{d\epsilon} + \tau\left(1 + \frac{M-1}{M}e^{-\mu\Delta\epsilon/M}\right)\right] < 0 \text{ if } \frac{dR(k-\epsilon,\mathbf{k})}{d\epsilon} \geq 0.$$

We can see that the only way to satisfy $\frac{dR(k+\epsilon,\mathbf{k})}{d\epsilon}|_{\epsilon=0} \leq 0$ and $\frac{dR(k-\epsilon,\mathbf{k})}{d\epsilon}|_{\epsilon=0} \leq 0$ simultaneously is by having $k = (\mu\Delta(M-1)/M)^{-1} - \pi/\tau$. If $(\mu\Delta(M-1)/M)^{-1} > \pi/\tau$, then k is

[16] The sufficient condition holds strictly for $\tau > 0$.

the unique Nash equilibrium of the mining game. If instead $(\mu\Delta(M-1)/M)^{-1} < \pi/\tau$, then $\frac{dR(k-\epsilon,\mathbf{k})}{d\epsilon}|_{\epsilon=0} > 0$ for all $k \geq 0$, so the only Nash equilibrium is $k = 0$. ■

Impossibility of asymmetric equilibria

Impossibility of asymmetric equilibria follows from strategic complementarity. For the purpose of this proof, I will show that there is always (at least) one profitable deviation from an asymmetric equilibrium.

Let us proceed by contradiction. Any asymmetric strategy profile is a vector $(k_1, k_2, \ldots, k_M)$ where $k_m \neq k_{m'}$ for at least one pair (m, m'). Without loss of generality, let $k_{m-1} \geq k_m$ for all $m \leq M$ and $k_1 > k_2$. Let $\mathbf{k'} \equiv (k_3, \ldots, k_M)$. For $\mathbf{k}$ to be an equilibrium, miner 1 should not find profitable to choose k_2 and vice-versa. So the following inequalities have to hold simultaneously:

$$P_{k_1}(k_2, \mathbf{k'})\,(\pi + \tau k_1) \geq P_{k_2}(k_2, \mathbf{k'})\,(\pi + \tau k_2)\,, \qquad P_{k_2}(k_1, \mathbf{k'})\,(\pi + \tau k_2) \geq P_{k_1}(k_1, \mathbf{k'})\,(\pi + \tau k_1)\,.$$

Combining the previous inequalities gives:

$$L_{k_1,k_2}(k_2, \mathbf{k'}) \leq \frac{\pi + \tau k_2}{\pi + \tau k_1}, \quad L_{k_1,k_2}(k_1, \mathbf{k'}) \geq \frac{\pi + \tau k_2}{\pi + \tau k_1}, \qquad L_{k_1,k_2}(\mathbf{k}_{-m}) \equiv \frac{P_{k_2}(\mathbf{k}_{-m})}{P_{k_1}(\mathbf{k}_{-m})}$$

which is only possible if $L_{k_1,k_2}(k_1, \mathbf{k'}) \geq L_{k_1,k_2}(k_2, \mathbf{k'})$ for $k_1 > k_2$. However, I will now show that

$$L_{k_1,k_2}(k_1, \mathbf{k'}) < L_{k_1,k_2}(k_2, \mathbf{k'}) \tag{36}$$

establishing the needed contradiction. Notice that Eq. (36) requires the likelihood ratio $L_{k_1,k_2}(k, \mathbf{k'})$ to decrease in k. To prove Eq. (36), let us compute

$$P_{k_1}(k_1, \mathbf{k'}) = \int_0^\infty \mathbb{P}(T_2 > T \mid T = t) \prod_{i=3}^M \mathbb{P}(T_i > T + \Delta(k_1 - k_i) \mid T = t)\, \mathrm{d}F_T(t)$$

$$= \int_0^\infty e^{-\mu_m t} e^{-\mu_m [t(M-2) + \sum_{i=3}^M \Delta(k_1 - k_i)]} \mu_m e^{-\mu_m t}\, \mathrm{d}t = \mu_m e^{-\mu_m \sum_{i=3}^M \Delta(k_1 - k_i)} \int_0^\infty e^{-\mu_m t M}\, \mathrm{d}t$$

$$= \frac{1}{M} e^{-\mu_m \sum_{i=3}^M \Delta(k_1 - k_i)},$$

and

$$P_{k_2}(k_1, \mathbf{k'}) = \int_0^\infty \mathbb{P}(T_2 > T - \Delta(k_1 - k_2) \mid T = t) \prod_{i=3}^M \mathbb{P}(T_i > T + \Delta(k_2 - k_i) \mid T = t) \, \mathrm{d}F_T(t)$$

$$= \int_0^\infty \left(\mathbb{1}_{\{t < \Delta(k_1 - k_2)\}} + \mathbb{1}_{\{t > \Delta(k_1 - k_2)\}} e^{-\mu_m t} \right) e^{-\mu_m [t(M-2) + \sum_{i=3}^M \Delta(k_2 - k_i)]} \mu_m e^{-\mu_m t} \, \mathrm{d}t$$

$$= \mu_m e^{-\mu_m \sum_{i=3}^M \Delta(k_2 - k_i)} \int_0^\infty \left(\mathbb{1}_{\{t < \Delta(k_1 - k_2)\}} + \mathbb{1}_{\{t > \Delta(k_1 - k_2)\}} e^{-\mu_m t} \right) e^{-\mu_m t(M-1)}$$

$$= \mu_m e^{-\mu_m \sum_{i=3}^M \Delta(k_2 - k_i)} \left[\int_0^{\Delta(k_1 - k_2)} e^{-\mu_m t(M-1)} \, \mathrm{d}t + \int_{\Delta(k_1 - k_2)}^\infty e^{-\mu_m t M} \, \mathrm{d}t \right]$$

$$= e^{-\mu_m \sum_{i=3}^M \Delta(k_2 - k_i)} \left[\frac{1}{M-1} \left(1 - e^{-\mu_m (M-1) \Delta(k_1 - k_2)} \right) + \frac{1}{M} e^{-\mu_m M \Delta(k_1 - k_2)} \right],$$

so that

$$L_{k_1,k_2}(k_1, \mathbf{k'}) = e^{\mu_m \Delta(M-2)(k_1 - k_2)} \left[\frac{M}{M-1} \left(1 - e^{-\mu_m (M-1) \Delta(k_1 - k_2)} \right) + e^{-\mu_m M \Delta(k_1 - k_2)} \right] \quad (37)$$

Also, by the same logic as before,

$$P_{k_1}(k_2, \mathbf{k'}) = \int_0^\infty \prod_{i=2}^M \mathbb{P}(T_i > T + \Delta(k_1 - k_i) \mid T = t) \, \mathrm{d}F_T(t) = \frac{1}{M} e^{-\mu_m \sum_{i=2}^M \Delta(k_1 - k_i)},$$

$$P_{k_2}(k_2, \mathbf{k'}) = \int_0^\infty \mathbb{P}(T_2 > T \mid T = t) \prod_{i=3}^M \mathbb{P}(T_i > T + \Delta(k_2 - k_i) \mid T = t) \, \mathrm{d}F_T(t)$$

$$= e^{-\mu_m \sum_{i=3}^M \Delta(k_2 - k_i)} \frac{1}{M},$$

giving

$$L_{k_1,k_2}(k_2, \mathbf{k'}) = e^{\mu_m \Delta(M-1)(k_1 - k_2)}. \quad (38)$$

Now, define the constant $\kappa \equiv \mu_m \Delta(k_1 - k_2) > 0$ (as $k_1 > k_2$). We have that $L_{k_1,k_2}(k_1, \mathbf{k'}) < L_{k_1,k_2}(k_2, \mathbf{k'})$, or $L_{k_1,k_2}(k_1, \mathbf{k'}) / L_{k_1,k_2}(k_2, \mathbf{k'}) < 1$ holds if $e^{-\kappa} \left[\frac{M}{M-1} (1 - e^{-\kappa(M-1)}) + e^{-\kappa M} \right] < 1$. Since $e^{-\kappa} < 1$, the inequality is satisfied if

$$\left[\frac{M}{M-1} (1 - e^{-\kappa(M-1)}) + e^{-\kappa M} \right] < 1 \iff \frac{1 - e^{-\kappa(M-1)}}{M-1} < \frac{1 - e^{-\kappa M}}{M}.$$

The last condition holds since $(1 - e^{-\kappa M})/M$ is an increasing function of M, as it can be seen by differentiating with respect to M: The numerator of the derivative obtained using the quotient rule is $1 - e^{-\kappa M} - \kappa M e^{-\kappa M} > 0$ since $1 = e^{-\kappa M} + \kappa M e^{-\kappa M} + \sum_{x=2}^\infty (\kappa M)^x e^{-\kappa M} / x!$.

■

Proof of Proposition 3. To derive the Fokker-Planck equation, we notice that the infinitesimal generator $\mathcal{A}$ for a generic function $h(\cdot)$ of the process Q_t is given by

$$\begin{aligned}
\mathcal{A}h(Q_t) &\equiv \lim_{\epsilon \downarrow 0} \frac{\mathbb{E}[h(Q_{t+\epsilon})] - h(Q_t)}{\epsilon} \\
&= \lim_{\epsilon \downarrow 0} \frac{\Big[h(Q_t + \alpha\epsilon) + \lambda\epsilon\left(h(Q_t + \alpha\epsilon - \min(Q_t, k)) - h(Q_t + \alpha\epsilon)\right)\Big] - h(Q_t)}{\epsilon} + o(\epsilon) \\
&= \lim_{\epsilon \downarrow 0} \frac{h(Q_t + \alpha\epsilon) - h(Q_t)}{\epsilon} + \lim_{\epsilon \downarrow 0} \lambda\left[h(Q_t + \alpha\epsilon - \min(Q_t, k)) - h(Q_t + \alpha\epsilon)\right] \\
&= \alpha \frac{\partial h(Q_t)}{\partial Q_t} + \lambda\left[h(Q_t - \min(k, Q_t)) - h(Q_t)\right] \qquad (39)
\end{aligned}$$

The rate of change in the expected value of a function h over the probability density $f_t(Q)$ is given by

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}[h(Q_t)] \equiv \frac{\mathrm{d}}{\mathrm{d}t}\int_0^\infty h(Q) f_t(Q)\, dQ = \int_0^\infty h(Q)\left(\frac{\partial f_t(Q)}{\partial t}\right) dQ \qquad (40)$$

Moreover, by the continuity of $\mathbb{E}[h(Q_t)]$ with respect to t, the following identity holds for *any* function h:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}[h(Q_t)] = \mathbb{E}[\mathcal{A}h(Q_t)],$$

or analogously,

$$\int_0^\infty h(Q)\left(\frac{\mathrm{d}}{\mathrm{d}t} f_t(Q)\right) dQ = \int_0^\infty \left(\alpha\frac{\partial h(Q)}{\partial Q} + \lambda\left[h(Q - \min(k, Q)) - h(Q)\right]\right) f_t(Q)\, dQ. \quad (41)$$

We can now manipulate Eq. (41) to obtain the Fokker-Planck equation. To proceed, we split the right-hand side into three integrals:

$$\alpha\int_0^\infty \frac{\partial h(Q)}{\partial Q} f_t(Q) dQ + \lambda\int_0^\infty h(Q - \min(k, Q)) f_t(Q) dQ - \lambda\int_0^\infty h(Q) f_t(Q) dQ. \qquad (42)$$

PDE for the density of Q

Now, consider test functions h that are continuous, positive and satisfy $h(0) = \lim_{Q\to\infty} h(Q) = 0$. The first integral in Eq. (42) simplifies after integrating by parts, with $dv = \frac{\partial h(Q)}{\partial Q}\, dQ$ and

$u = f_t(Q)$:

$$\int_0^\infty u\, dv = \underbrace{h(Q) f_t(Q)\big|_{Q=0}^{Q\to\infty}}_{=0} - \int_0^\infty h(Q) \frac{\partial f_t(Q)}{\partial Q}\, dQ.$$

The first term is null because the test function vanishes at the endpoints of the domain. In this way we obtain:

$$\alpha \int_0^\infty \frac{\partial h(Q)}{\partial Q} f_t(Q) dQ = -\alpha \int_0^\infty h(Q) \frac{\partial f_t(Q)}{\partial Q}\, dQ. \tag{43}$$

To simplify the second integral in Eq. (42) we notice that

$$\int_0^\infty h(Q - \min(k, Q)) f_t(Q) dQ = \underbrace{h(0) \int_0^k f_t(Q) dQ}_{=0} + \int_k^\infty h(Q-k) f_t(Q) dQ.$$

The first term vanishes because $h(0) = 0$ by definition. Now shifting the low integration endpoint from Q to $Q - k$ we obtain

$$\int_k^\infty h(Q-k) f_t(Q) dQ = \int_0^\infty h(Q) f_t(Q+k) dQ \tag{44}$$

Finally plugging Eqs. (43) and (44) into Eq. (42) and using our initial identity Eq. (41),

$$\int_0^\infty h(Q) \left(\frac{\partial f_t(Q)}{\partial t} \right) dQ = \int_0^\infty h(Q) \left(-\alpha \frac{\partial f_t(Q)}{\partial Q} + \lambda (f_t(Q+k) - f_t(Q)) \right) dQ. \tag{45}$$

For Eq. (45) to hold for all test functions h, it must be the case that

$$\frac{\partial f_t(Q)}{\partial t} \equiv -\alpha \frac{\partial f_t(Q)}{\partial Q} + \lambda [f_t(Q+k) - f_t(Q)]. \qquad \text{for all } Q \in (0, +\infty)$$

Which expressed in normalized quantities reads

$$\frac{\partial f_x(b)}{\partial x} \equiv -\rho \frac{\partial f_b(b)}{\partial b} + f_x(b+1) - f_x(b). \qquad \text{for all } b \in (0, +\infty)$$

■

Stationary distribution

Stationarity requires $\partial f_t(Q)/\partial t = 0$ so that $f_t(b) = f(b)$. So $f(b)$ has to satisfy the following delay ODE:

$$\frac{\partial f_b(b)}{\partial b} = \rho^{-1} (f(b+1) - f(b))$$

We can verify that the (negative) exponential density solves such ODE by using the test function $f(b) = z_1 e^{-z_2 b}$. Plugging such test function into the ODE, we can see that the values z_2 compatible with the ODE are the roots of the following equation:

$$z_2 = \rho^{-1}(1 - e^{-z_2}) \tag{46}$$

Eq. (46) clearly always admits the trivial root $z_2 = 0$. Nevertheless, for $\rho \in (0, 1)$, there exists also a non-trivial root, whose value is given by

$$z_2 = \kappa(\rho) \equiv \rho^{-1} + \mathcal{W}\left(-\rho^{-1} e^{-\rho^{-1}}\right)$$

where $\mathcal{W}$ is the principal branch of the Lambert-W function (also known as ProductLog function).

The existence of the non-trivial solution follows by noticing that the two sides of Eq. (46) intersect above zero. First, at $z_2 = 0$, both sides are equal and evaluate at 0. Then, the slope of the left expression is 1, while the slope of the right expression is $\rho^{-1} e^{-z_2}$, which satisfies $\rho^{-1} e^{-z_2} = \rho^{-1} > 1$ at $z_2 = 0$ and $\lim_{z_2 \to \inf} \rho^{-1} e^{-z_2} = 0$. Thus the left hand side is increasing and concave, tending towards a straight line asymptotically. This ensure that it stays above the right hand side for small values of z_2 and below for large values.

The other normalizing constant z_1 follows immediately by requiring the density to integrate at unity:

$$z_1 \int_0^\infty e^{-z_2 b} = \frac{z_1}{z_2} \equiv 1. \implies z_1 = z_2.$$

It follows that $f(b)$ is exponentially distributed with rate parameter $\kappa(\rho)$. ■

Stability condition

To prove that the process described by the stochastic differential equation (SDE)

$$\mathrm{d}Q_t = \alpha \, \mathrm{d}t - \min(k, Q_t) \, \mathrm{d}B_t$$

is ergodic and admits a stationary distribution for $\rho = \alpha/(\lambda k) < 1$, we can use the Foster-Lyapunov condition. Such condition states that there exists a function $h(Q)$ such that for some constants $u_0 > 0$ and $u_1 < \infty$,

$$\mathcal{A}h(Q) \leq -u_0 \quad \text{for } Q \text{ outside a compact set}$$

$$\mathcal{A}h(Q) \leq u_1 \quad \text{for } Q \text{ inside a compact set}$$

In our case, we can choose $h(Q) = Q$ and the compact set to be $[0, k]$. Outside this set, i.e., for $Q > k$, $\mathcal{A}h(Q) = \alpha - \lambda k < 0$. Thus, there exists a constant $u_0 = \lambda k - \alpha > 0$ such that $\mathcal{A}h(Q) \leq -u_0$. For $Q \in [0, k]$, $\mathcal{A}h(Q) = \alpha - \lambda Q$, so for $u_1 = \alpha$, we have that $\mathcal{A}h(Q) \leq u_1$.

The stability of the stochastic process in Eq. (7) obviously implies stability of its normalized representation in Eq. (10). ■

Proof of Proposition 7. Suppose that every miner m chooses $k_m = k$. Let $\mathbf{k}$ de a vector with element k repeated $M - 1$ times. The probability that a proposed block gets forked can be computed as

$$P_{\mathsf{Fork}}(k, \mathbf{k}) = Pr[T_m \in (T_{-m}, T_{-m} + \Delta k)] = \int_0^\infty \left(e^{-\mu_m t} - e^{-\mu_m (t+\Delta k)}\right) \mu_{-m} e^{-\mu_{-m} t} \, \mathrm{d}t$$

$$= \mu_{-m} \left[\int_0^\infty e^{-\mu t} \, \mathrm{d}t - e^{-\mu_m \Delta k} \int_0^\infty e^{-\mu t} \, \mathrm{d}t\right] = \left(1 - e^{-\mu_m \Delta k}\right) \mu_{-m} \int_0^\infty e^{-\mu t} \, \mathrm{d}t = \frac{\mu_{-m}}{\mu} \left(1 - e^{-\mu_m \Delta k}\right)$$

so that

$$P_{\mathsf{Fork}}(k, \mathbf{k}) = \frac{M-1}{M} \left(1 - e^{-\mu \Delta k / M}\right)$$

$$R(k, \mathbf{k}_{-m}) = \left[\frac{1}{M} (\pi + \tau k) + \frac{M-1}{M} \, \phi \left(1 - e^{-\mu \Delta k / M}\right)\right]. \tag{47}$$

Now consider a deviation to $k + \epsilon$. The fork probability becomes

$$P_{\mathsf{Fork}}(k + \epsilon, \mathbf{k}) = \Pr[T_m \in (T_{-m} - \Delta\epsilon, T_{-m} + \Delta k)]$$

$$= \mu_{-m} \left[\int_0^{\Delta\epsilon} \left(1 - e^{-\mu_m (t+\Delta k)}\right) e^{-\mu_{-m} t} \, \mathrm{d}t + \int_{\Delta\epsilon}^\infty \left(e^{-\mu_m (t-\Delta\epsilon)} - e^{-\mu_m (t+\Delta k)}\right) e^{-\mu_{-m} t} \, \mathrm{d}t\right]$$

$$= \mu_{-m} \left[\int_0^{\Delta\epsilon} \left(e^{-\mu_{-m} t} - e^{-\mu_m \Delta k} e^{-\mu t}\right) \mathrm{d}t + \int_{\Delta\epsilon}^\infty \left(e^{\mu_m \Delta\epsilon} e^{-\mu t} - e^{-\mu_m \Delta k} e^{-\mu t}\right) \mathrm{d}t\right].$$

The first and second integral simplify as follows:

$$\mu_{-m} \int_0^{\Delta\epsilon} \left(e^{-\mu_{-m} t} - e^{-\mu_m \Delta k} e^{-\mu t}\right) \mathrm{d}t = \left(1 - e^{-\mu_{-m} \Delta\epsilon}\right) - \frac{\mu_{-m}}{\mu} e^{-\mu_m \Delta k} \left(1 - e^{-\mu \Delta\epsilon}\right),$$

$$\mu_{-m} \left(e^{\mu_m \Delta\epsilon} - e^{-\mu_m \Delta k}\right) \int_{\Delta\epsilon}^\infty e^{-\mu t} \, \mathrm{d}t = \frac{\mu_{-m}}{\mu} e^{-\mu \Delta\epsilon} \left(e^{\mu_m \Delta\epsilon} - e^{-\mu_m \Delta k}\right) = \frac{\mu_{-m}}{\mu} \left(e^{-\mu_{-m} \Delta\epsilon} - e^{-\mu_m \Delta k} e^{-\mu \Delta\epsilon}\right).$$

Combining both, we obtain the following fork probability:

$$P_{\mathsf{Fork}}(k+\epsilon, \mathbf{k}_{-m}) = \left(1 - e^{-\mu\Delta\epsilon(M-1)/M}\right) + \frac{M-1}{M}\left(e^{-\mu\Delta\epsilon(M-1)/M} - e^{-\mu\Delta k/M}\right).$$

The win probability is determined by Eq. (32):

$$P_{\mathsf{Win}}(k+\epsilon, \mathbf{k}) = \frac{1}{M} e^{-\mu\Delta\epsilon(M-1)/M}(1 - e^{-\mu/M}).$$

Thus the miner payoff is given by

$$R(k+\epsilon, \mathbf{k}_{-m}) = \left[\pi + \tau(k+\epsilon)\right] P_{\mathsf{Win}}(k+\epsilon, \mathbf{k}) + \phi P_{\mathsf{Fork}}(k+\epsilon, \mathbf{k})$$

Now, precluding a deviation to $k+\epsilon$ requires

$$\left.\frac{R(k+\epsilon, \mathbf{k}_{-m})}{\partial\epsilon}\right|_{\epsilon=0} \leq 0. \tag{48}$$

Given that

$$\left.\frac{P_{\mathsf{Fork}}(k+\epsilon, \mathbf{k}_{-m})}{\partial\epsilon}\right|_{\epsilon=0} = \mu\Delta\frac{M-1}{M}\left(1 - \frac{M-1}{M^2}\right),$$

$$\left.\frac{R(k+\epsilon, \mathbf{k}_{-m})}{\partial\epsilon}\right|_{\epsilon=0} = \left[\frac{\tau}{M} - \mu\Delta\frac{M-1}{M}\left(\frac{\pi+\tau k}{M}\right)\right] + \phi\mu\Delta\frac{M-1}{M^2}$$

$$= \frac{\tau}{M} + \mu\Delta\frac{M-1}{M^2}\left[(\phi-\pi) - \tau k\right],$$

the deviation is precluded if and only if

$$\frac{\tau}{M} + \mu\Delta\frac{M-1}{M^2}\left[(\phi-\pi) - \tau k\right] \leq 0 \iff \tau + \mu\Delta\frac{M-1}{M}\left[(\phi-\pi) - \tau k\right] \leq 0$$

$$\iff k \geq \frac{\mu^{-1}}{\Delta}\frac{M-1}{M} - \frac{\phi-\pi}{\tau}. \tag{49}$$

Now consider the deviation to $k-\epsilon$. The fork probability is given by

$$P_{\mathsf{Fork}}(k-\epsilon, \mathbf{k}_{-m}) = \Pr[T_m \in (T_{-m}+\Delta\epsilon, T_{-m}+\Delta k)]$$

$$= \mu_{-m}\int_0^\infty \left(e^{-\mu_m(t+\Delta\epsilon)} - e^{-\mu_m(t+\Delta k)}\right) e^{-\mu_{-m}t}dt = \mu_{-m}\int_0^\infty \left(e^{-(\mu t+\mu_m\Delta\epsilon)} - e^{-(\mu t+\mu_m\Delta k)}\right) dt$$

$$= \frac{\mu_{-m}}{\mu}\left(e^{-\mu_m\Delta\epsilon} - e^{-\mu_m\Delta k}\right) = \frac{M-1}{M}\left(e^{-\mu\Delta\epsilon/M} - e^{-\mu\Delta k/M}\right).$$

The win probability is given by Eq. (34),

$$P_{\mathsf{Win}}(k+\epsilon, \mathbf{k}_{-m}) = 1 - \frac{M-1}{M}e^{-\mu\Delta\epsilon/M}$$

The miner payoff is given by

$$R(k-\epsilon, \mathbf{k}_{-m}) = \left[\pi + \tau(k-\epsilon)\right] P_{\mathsf{Win}}(k-\epsilon, \mathbf{k}_{-m}) + \phi P_{\mathsf{Fork}}(k-\epsilon, \mathbf{k}_{-m}).$$

Differentiating with respect to ϵ and evaluating at $\epsilon = 0$ we get

$$\left.\frac{R(k+\epsilon, \mathbf{k}_{-m})}{\partial\epsilon}\right|_{\epsilon=0} = (\pi+\tau k)\frac{\partial P_{\mathsf{Win}}}{\partial\epsilon}(k) - \tau P_{\mathsf{Win}}(k) + \phi\frac{\partial P_{\mathsf{Win}}}{\partial\epsilon}(k) \tag{50}$$

where

$$\frac{\partial P_{\mathsf{Win}}}{\partial\epsilon}(k) = \Delta\mu\frac{M-1}{M^2}, \quad P_{\mathsf{Win}}(k) = \frac{1}{M}, \quad \frac{\partial P_{\mathsf{Fork}}}{\partial\epsilon}(k) = -\Delta\mu\frac{M-1}{M^2}. \tag{51}$$

By plugging the expressions in Eq. (51) back into Eq. (50) we can see that

$$\left.\frac{\partial P_{\mathsf{Fork}}(k-\epsilon, \mathbf{k}_{-m})}{\partial\epsilon}\right|_{\epsilon=0} = -\left.\frac{\partial P_{\mathsf{Fork}}(k+\epsilon, \mathbf{k}_{-m})}{\partial\epsilon}\right|_{\epsilon=0}$$

Therefore, the deviation to $k-\epsilon$ is precluded for

$$k \le \frac{\mu^{-1}}{\Delta}\frac{M-1}{M} - \frac{\phi-\pi}{\tau}.$$

The only value of k that satisfies both condition is the symmetric equilibrium value shown in Eq. (26). ■

Proof of Lemma 3. First of all, let us compute the win probabilities of the miners choosing $k_m = \underline{k}$ and $k_m = Q$.

Let us denote by T_0 the fastest block produced by the *other* miners choosing capacity

$\overline{k}$. Similarly, let us denote T_1 the fastest block produced by the other miners with capacity Q. We denote the block production rate of these two groups of miners by μ_0 and μ_1. As in the previous proofs, we denote the production rate of a given miner by μ_m and the arrival time of its first block produced by T_m.

To proceed with the computation, we notice that $\mathbb{P}(T_0 - \Delta > t) = \mathbb{P}(T_0 > t + \Delta) = e^{-\mu_0(t+\Delta)}$ and $\mathbb{P}(T_1 > t) = e^{-\mu_1 t}$. So we have that

$$\begin{aligned} P(Q, \hat{m}) &= \mathbb{P}\left(T_m < \min\{T_0 - \Delta, T_1\}\right) \\ &= \int_0^\infty e^{-(\mu_0+\mu_1+\mu_m)t-\mu_0\Delta}\mu_m \,\mathrm{d}t = \frac{\mu_m}{\mu_0+\mu_1+\mu_m}e^{-\mu_0\Delta}. \end{aligned} \tag{52}$$

Similarly, since $\mathbb{P}(T_0 > t) = e^{-\mu_0 t}$ and $\mathbb{P}(T_1 + \Delta > t) = \mathbb{P}(T_1 > t - \Delta) = \mathbb{1}_{\{t<\Delta\}} + \mathbb{1}_{\{t>\Delta\}}e^{-\mu_1(t-\Delta)}$, integration over the density of T_m gives

$$\begin{aligned} P(\underline{k}, \hat{m}) &= \mathbb{P}(T_m < \min\{T_0, T_1 + \Delta\}) \\ &= \int_0^\infty \left[\mathbb{1}_{\{t<\Delta\}}e^{-\mu_0 t} + \mathbb{1}_{\{t>\Delta\}}e^{-(\mu_1+\mu_0)t+\mu_1\Delta}\right]\mu_m e^{-\mu_m t}\,\mathrm{d}t \\ &= \mu_m\int_0^\Delta e^{-(\mu_0+\mu_m)t}\,\mathrm{d}t + \mu_m e^{\mu_1\Delta}\int_\Delta^\infty e^{-(\mu_0+\mu_1+\mu_m)t}\,\mathrm{d}t \\ &= \frac{\mu_m}{\mu_0+\mu_m}\left(1 - e^{-(\mu_0+\mu_m)\Delta}\right) + \frac{\mu_m}{\mu_0+\mu_1+\mu_m}e^{-(\mu_0+\mu_m)\Delta}. \end{aligned} \tag{53}$$

$L_{\underline{k},Q}(\hat{m})$ in Eq. (30) follows from setting $\mu_m = 1/M, \mu_0 = \hat{m}/M$ and $\mu_1 = m_1/M$ in Eqs. (52) and (53). Now monotonicity of $L_{\underline{k},Q}(\hat{m})$ with respect to $\hat{m}$ can be shown as follows:

$$L_{\underline{k},Q}(\hat{m}) - L_{\underline{k},Q}(\hat{m}-1) > 0 \iff \frac{\hat{m}}{\hat{m}+1} > \frac{e^{\mu\hat{m}\Delta/M} - 1}{e^{\mu(\hat{m}+1)\Delta/M} - 1} \iff 1 < \frac{\mu\hat{m}\Delta}{M}e^{\mu\hat{m}\Delta/M}.$$

The last inequality is true by the properties of the exponential function. ■

Appendix B Block Proposer Election

In every Nakamoto-style consensus protocol, the selection of block proposers is done through a lottery mechanism implemented using a cryptographic hash function H (e.g., SHA256). In the classical Nakamoto consensus (used in Bitcoin), a miner can propose the next block B

if it finds a nonce (i.e., a 'number used once') such that

$$\mathsf{hash}_B = \mathsf{H}(\mathsf{hash}_{B-1}, \mathsf{payload}_B, \mathsf{nonce}) < \mathsf{difficulty}.$$

Here, hash_B is the hash of the proposed block B, hash_{B-1} is the hash of the previous block, $\mathsf{payload}_B$ represents the block's data (transactions), and nonce is a random value that miners adjust to find a valid hash.

The $\mathsf{difficulty}$ parameter determines the hardness of the random trial: the result of the hash function evaluation must be less than this target to be valid. The higher the difficulty (i.e., the more leading zeros required), the lower the probability that a randomly chosen nonce will satisfy the inequality.

In Proof-of-Stake (PoS) consensus protocols with "perfect randomness," the selection mechanism mimics that of Bitcoin but replaces the nonce with the validator's public key PK and adjusts the difficulty threshold based on the validator's stake. The hash inequality becomes

$$\mathsf{hash}_B = \mathsf{H}(\mathsf{hash}_{B-1}, \mathsf{time}, \mathsf{PK}) < \mathsf{difficulty} \times \mathsf{stake}.$$

Here, time denotes the current time slot or timestamp, and $\mathsf{difficulty}$ is a base difficulty target set by the protocol. By adjusting the threshold with the validator's stake, validators with more stake have a higher probability of satisfying the hash inequality.

A slight departure from the previous style of inequalities is found in epoch-based Nakamoto-consensus protocols. In these variants, the random seed that determines the result of the new hash inequality is not updated every block but every *epoch*, which are simply collections of time slots. The hash inequality for an epoch-based PoS protocol looks like

$$\mathsf{hash}_B = \mathsf{H}(\mathsf{hash}_{B(\mathsf{epoch})}, \mathsf{time}, \mathsf{PK}) < \mathsf{difficulty} \times \mathsf{stake},$$

where $\mathsf{hash}_{B(\mathsf{epoch})}$ is the hash of the block setting the seed for the current epoch. In Ethereum PoS, time slots last 12 seconds, and an epoch contains 32 time slots, lasting roughly 6 minutes. The block setting the seed for the current epoch $\mathsf{hash}_{B(\mathsf{epoch})}$ for $\mathsf{epoch} = e$ is, most of the time, the first block added in epoch $e - 2$.[17]

Proof-of-Stake with Verifiable Random Functions (VRF)

In Proof-of-Stake systems that utilize Verifiable Random Functions (VRFs), the selection of the block proposer is based on a cryptographic procedure that ensures unpredictability and

[17] To be precise, it is the checkpoint block of epoch $e-2$, which is usually the first block produced in that epoch.

fairness. Each validator possesses a key pair: a secret (private) key SK and a public key PK.

The VRF-based block proposer election works as follows:

1. **VRF Hash Computation:** Validators compute a VRF hash h using their secret key SK and the current seed, derived from the previous block's hash $\mathsf{hash}_{B(\mathsf{epoch})}$ and the current time slot time:

$$h = \mathsf{VRF}_{\mathsf{hash}}(\mathsf{hash}_{B(\mathsf{epoch})}, \mathsf{time}, \mathsf{SK}).$$

2. **Eligibility Check:** Validators check whether the VRF hash falls below a threshold adjusted by their stake:

$$h < \mathsf{difficulty} \times \mathsf{stake}.$$

   If the inequality holds, the validator is eligible to propose the next block.

3. **VRF Proof Generation:** To prove eligibility, the validator generates a VRF proof:

$$\pi = \mathsf{VRF}_{\mathsf{proof}}(\mathsf{hash}_{B(\mathsf{epoch})}, \mathsf{time}, \mathsf{SK}).$$

4. **Broadcast:** The validator broadcasts $(\pi, h, \mathsf{hash}_{B(\mathsf{epoch})}, \mathsf{time}, \mathsf{PK}, \mathsf{stake})$ to the validator network.

5. **Verification by Other Nodes:** Other nodes verify the proof using the VRF verification function:

$$\mathsf{VRF}_{\mathsf{verify}}(\mathsf{hash}_{B(\mathsf{epoch})}, \mathsf{time}, \pi, h, \mathsf{PK}) \rightarrow \mathsf{True},$$

   and confirm the eligibility condition $h < \mathsf{difficulty} \times \mathsf{stake}$. This enables them to verify that the validator rightfully proposed a block based on their stake.

This process ensures that the block proposer is selected fairly and that the proof of proposer selection is verifiable by all nodes in the network. Since the VRF output is pseudo-random and can only be computed by the holder of the secret key SK, but is verifiable by others using the corresponding public key PK, it provides a trustless and transparent mechanism for leader election in Proof-of-Stake protocols.

In epoch-based PoS systems employing VRFs, the randomness seed is updated every epoch, and validators use this seed along with their secret key to compute their VRF outputs for each slot within the epoch. This method is used in protocols like Algorand and Cardano, where the VRF provides a secret random selection of block proposers and committee members.

Appendix C Random walk representation of mempool dynamics

To derive the statistics of interest, I provide a random-walk representation of the dynamics in Eq. (10). We can form a discrete partition of the timeline $\{t_n\}_{n=0}^{\infty}$ such that the length of a period is $t_n - t_{n-1} = k/\alpha$. This is the time it takes a block to be filled at the equilibrium capacity. By the Poisson assumption for block arrival, the dynamics of b evolve according to the following difference equation:

$$b_n = (b_{n-1} + 1 - \mathsf{Poiss}_n(\rho))^+,$$

where $\mathsf{Poiss}_n(\rho)$ is a Poisson random variable with parameter ρ. The resulting process is a Markov chain described by the following transition probabilities:

$$\mathbb{P}(b_n = j \mid b_{n-1} = i) = \begin{cases} 0 & \text{if } j > i+1, \\ P_{i+1-j} & \text{if } 0 < j \leq i+1, \\ 1 - F_{i+1} & \text{if } j = 0, \end{cases} \quad P_k = \frac{\rho^k e^{-\rho}}{k!}, \quad F_k = \sum_{l=0}^{k} P_l.$$

Here, P_k and F_k are the Poisson mass function and cumulative distribution function with parameter ρ evaluated at k.

The resulting (infinite) state-transition matrix has the following form:

$$\begin{bmatrix} 1-F_1 & P_0 & \mathbf{0} & \cdots & \cdots & \cdots & \mathbf{0} \\ 1-F_2 & P_1 & P_0 & \mathbf{0} & \cdots & \cdots & \mathbf{0} \\ 1-F_3 & P_2 & P_1 & P_0 & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

To compute the steady-state probabilities, we can evaluate the transition matrix for a large limit value of b, for example $b = 10^6$, and take the left eigenvector of the transition matrix with unit eigenvalue.

CREST
CENTER FOR RESEARCH
IN ECONOMICS AND STATISTICS

CREST
Center for Research in Economics and Statistics
UMR 9194

5 Avenue Henry Le Chatelier
TSA 96642
91764 Palaiseau Cedex
FRANCE

Phone: +33 (0)1 70 26 67 00
Email: info@crest.science
https://crest.science/

The Center for Research in Economics and Statistics (CREST) is a leading French scientific institution for advanced research on quantitative methods applied to the social sciences.

CREST is a joint interdisciplinary unit of research and faculty members of CNRS, ENSAE Paris, ENSAI and the Economics Department of Ecole Polytechnique. Its activities are located physically in the ENSAE Paris building on the Palaiseau campus of Institut Polytechnique de Paris and secondarily on the Ker-Lann campus of ENSAI Rennes.

cnrs

GENES Groupe des écoles nationales d'économie et statistique