Time Is Money: Strategic Timing Games in Proof-Of-Stake Protocols

Caspar Schwarz-Schilling
Ethereum Foundation, Berlin, Germany

Fahad Saleh
Wake Forest University, Winston Salem, NC, USA

Thomas Thiery
Ethereum Foundation, Lyon, France

Jennifer Pan
Jump Crypto, Chicago, IL, USA

Nihar Shah
Jump Crypto, Chicago, IL, USA

Barnabé Monnot
Ethereum Foundation, Berlin, Germany

Abstract

We propose a model suggesting that rational consensus participants may play *timing games*, and strategically delay their block proposal to optimize MEV capture, while still ensuring the proposal's inclusion in the canonical chain. In this context, ensuring economic fairness among consensus participants is critical to preserving decentralization. We contend that a model grounded in rational consensus participation provides a more accurate portrayal of behavior in economically incentivized systems such as blockchain protocols. We empirically investigate timing games on the Ethereum network and demonstrate that while timing games are worth playing, they are not currently being exploited by consensus participants. By quantifying the marginal value of time, we uncover strong evidence pointing towards their future potential, despite the limited exploitation of MEV capture observed at present.

2012 ACM Subject Classification Theory of computation → Algorithmic game theory and mechanism design

Keywords and phrases blockchain, proof-of-stake, game theory, maximal extractable value

Digital Object Identifier 10.4230/LIPIcs.AFT.2023.30

Acknowledgements The authors acknowledge helpful discussions and comments from Francesco d'Amato and Anders Elowsson. We also appreciate the significant contributions of Mike Neuder in obtaining the necessary data for this study.

1 Introduction

Consensus protocols are typically evaluated based on their ability to maintain liveness and safety [11], referring to the regular addition of new transactions to the output ledger in a timely manner, and to the security of confirmed transactions remaining in their positions within the ledger. However, beyond liveness and safety, blockchain protocols require fairness of economic outcomes amongst consensus participants to preserve decentralization. More specifically, a protocol should be designed to maximize profitability of honest participation, defined as adherence to the prescribed rules. Otherwise, a deviating participant may outcompete their honest peers, leading to centralization of the validation set over time and security implications for consensus itself.

However, the advent of Maximal Extractable Value (MEV) frustrates such fairness goals. MEV is defined as the value that consensus participants, in their duties as block proposers, accrue by selectively including, excluding and ordering user transactions [14, 6]. This concept has significant implications for the security of consensus protocols. For systems predominantly based on transaction fee rewards, increased variance in miner rewards may result in consensus instability [12]. Similarly, it was argued that a rational actor issuing a *whale transaction* with an abnormally large transaction fee can convince peers to fork the current chain, further destabilizing consensus [22]. As such, understanding and mitigating the impact of MEV on the security and fairness of blockchain networks has become a central concern of protocol designers [30].

Potential MEV accrues over time as users submit transactions and the value of the set of pending transactions increases for the block proposer. As a consequence, time is valuable to consensus participants, a feature obviated by the assumption of honest behavior in previous models of consensus. However, we argue that protocols who wish to preserve properties such as economic fairness amongst consensus participants must assume some share of rational consensus participation. In particular, the effects of MEV on the consensus participants' incentives must be better understood.

In this paper, we investigate the possibility for block proposers to delay their block proposal as long as possible while ensuring they become part of the canonical chain, aiming to maximize MEV extraction. The reader may note that in Proof-of-Work (PoW)-based leader selection protocols, delaying a proposal bears the risk of losing to a competing block proposer. PoW protocols exhibit an inherent racing condition that prevents these types of strategic delay deviations, or at least make them unprofitable in expectation. Thus, we investigate the implications of MEV on the incentives of consensus participants, particularly block proposers, in a Proof-of-Stake (PoS) context. More specifically, we consider propose-vote type of PoS protocols, where in each consensus round, one leader proposes a block, and a committee of consensus participants is selected in-protocol to vote on the acceptance of that block. This effectively grants block proposers a short-lived monopoly as the only valid proposer for some given round. During this time interval they can attempt to strategically deviate from their assigned block proposal time and delay the release of their block as long as possible in order to extract more MEV, while still ensuring that a sufficient share of attesters see the block in time to vote it into the canonical chain. This behavior leads to an environment in which honest validators earn less than their deviating counterparts, resulting in stake centralization and second-order effects for consensus stability.

Related Work

To the best of our knowledge, *timing games* have not been formally analyzed in previous literature on Proof-of-Stake. Selfish mining [19], studied in the context of Proof-of-Work, relies on appropriately timing the release of a block, in order to waste computation of honest miners and earn an outsize share of the rewards. Our timing games are also concerned with strategic behavior to capture a larger share of the total available rewards to consensus participants, yet do not feature the same dynamics as selfish mining in Proof-of-Work, since participants in many PoS-based consensus mechanisms are given a fixed time interval in which to perform their duties.

The security of PoS-based mechanisms has been discussed in terms of chain growth [18] or focusing on the safety and liveness properties of hybrid protocols such as Gasper [10, 24]. The economics literature has also examined Proof-of-Stake security with respect to particular attacks such as the double-spending attack [26] and 51% attacks [20]. Separately, incentive considerations in the presence of MEV led to the discovery of severe attacks on the Gasper consensus [28, 25] and protocol changes to address such attacks [15, 16, 17].

Our Contributions

Our work models the value of time to consensus participants and explores the potential emergence of timing games in Proof-of-Stake protocols. By understanding the strategic behavior of consensus participants within this model, we gain insights into how these dynamics affect the robustness of consensus protocols to exogenous incentives, and ultimately fairness.

- Despite initial pessimism regarding the existence of equilibria in timing games [23], we formally show how to sustain equilibrium behavior, where it is individually irrational for proposers to deviate from a schedule enforced by attesters, and reward-sharing is fair among participants (Sections 2 and 3).
- We then investigate whether such timing games might occur in real-world systems (namely, the Ethereum network), using a large, granular data set recording the MEV offered to block proposers over time. We show incidental deviations from the honest protocol specification, highlighting the feasibility of timing games, yet we do not conclude on the existence of intentional deviations from honest behavior (Section 4).

2 Model

We model an infinite horizon game among *block proposers* and *attesters*. Time is partitioned into slots $n \in \mathbb{N}$, each of time length $\Delta > 0$. Each slot n has a block proposer n and a unit measure of attesters $A_n = \{A_{(i,n)}\}_{i\in[0,1]}$ where $A_{(i,n)}$ refers to the ith attester within slot n.[1]

The game evolves as follows:

- At the beginning of slot n, proposer n acts by deciding whether to build on top of the block of proposer $n-1$ and also when to release their own block. More formally, proposer n selects $\phi_n \in \{0,1\}$ and $t_n \geq n \cdot \Delta$ where $\phi_n = 1$ ($\phi_n = 0$) refers to proposer n (not) building on top of the block of proposer $n-1$ and t_n denotes the time at which proposer n releases their own block. Note that we specify that proposer n cannot release their block before the start of slot n (i.e., $t_n \geq n \cdot \Delta$) but that they release their block after the end of the slot.
- After proposer n acts, all slot n attesters act simultaneously. In particular, attester $A_{(i,n)}$ decides whether to attest to the block of proposer n and also the time to release their attestation. More formally, attester $A_{(i,n)}$ select $\nu_{(i,n)} \in \{0,1\}$ and $\tau_{(i,n)} \geq n \cdot \Delta$ where $\nu_{(i,n)} = 1$ ($\nu_{(i,n)} = 0$) refers to attester $A_{(i,n)}$ (not) attesting to proposer n's block and $\tau_{(i,n)}$ refers to the time that they release their attestation. Notably, attester $A_{(i,n)}$ can attest to the block of proposer n only if they receive the block before releasing their attestation. We let $\delta_{n,(i,n)} \sim exp(\theta^{-1})$ refer to the random time required for the block of proposer n to reach attester $A_{(i,n)}$ where $\theta > 0$ denotes the average communication time across the network and we assume that the slot length is at least double the average communication time across the network (i.e., $\Delta \geq 2\theta$). In turn, the action of attester $A_{(i,n)}$ is constrained by $\nu_{(i,n)} = 1 \implies \tau_{(i,n)} \geq t_n + \delta_{n,(i,n)}$.

[1] Note that we have a continuum of attesters, rather than a discrete set. In Ethereum, over 19,000 attesters emit a vote per slot (as of 2023-06-16).

2.1 Block proposers

The pay-off for proposer n is given as follows:

$$U^P(t_n, \phi_n) = \begin{cases} \alpha + \mu \cdot (t_n - t_{n_-})^+ \text{ if } \chi_n = 1 \\ 0 \text{ otherwise} \end{cases} \tag{1}$$

where $\alpha, \mu > 0$ are exogenous constants while t_{n_-} corresponds to the time of the most recent canonical block before slot n and $\chi_n \in \{0, 1\}$ corresponds to whether the block in slot n is canonical on the blockchain. We introduce the conditions for a block to become canonical in our model in the following, and delay until Section 3.2 its interpretation with respect to established consensus models.

Note that we assume that the reward of proposer n increases linearly with time relative to the most recent canonical block so long as block n eventually becomes canonical. This assumption reflects that proposer n accrues incremental MEV over time by delaying the release of their block but that they risk being skipped if they delay release for too long. The time of the most recent canonical block, t_{n_-}, is endogenous where slot n_- refers to the most recent canonical slot and is thus given explicitly as follows:

$$n_- = \max\{k \in \mathbb{N} : \chi_k = 1, k \leq n - 1\} \tag{2}$$

For a block to be canonical, we require both that it receives sufficiently many successful attestations and that the subsequent block producer builds on top of it. More formally, letting $\tilde{A}_n$ denote the successful attestations for block n, χ_n is given explicitly as follows:

$$\chi_n = \begin{cases} 1 & \text{if } \phi_{n+1} = 1, \tilde{A}_n \geq \gamma \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where the number of successful attestations for block n is given as the measure of attesters in slot n voting for block n:

$$\tilde{A}_n = |\{i \in [0, 1] : \nu_{(i,n)} = 1\}| \tag{4}$$

2.2 Attesters

Attester (i, n) receives a pay-off if and only if two conditions are met:

- **Correctness:** A vote by attester (i, n) is correct if their vote is consistent with the canonical blockchain. Recall that the vote of attester (i, n) is given by $\nu_{(i,n)}$ and the eventual canonical status of the block is given by χ_n; thus, this condition is equivalent to $\nu_{(i,n)} = \chi_n$.
- **Freshness:** A vote by attester (i, n) is fresh if it was received by proposer $n+1$ soon enough that it could be included in the block in slot $n+1$ and the block in slot $n+1$ is eventually made canonical. We let $\delta_{(i,n),n+1} \sim exp(\theta^{-1})$ denote the random communication time between attester (i, n) and proposer $n + 1$, implying that the first part of this condition equates with $\tau_{(i,n)} + \delta_{(i,n),n+1} \leq t_{n+1}$. Moreover, the second part of this condition equates with $\chi_{n+1} = 1$.

For exposition, we normalize the pay-off for attester (i, n) to unity, implying that their pay-off function is given explicitly as follows:

$$U^A(\nu_{(i,n)}, \tau_{(i,n)}) = \begin{cases} 1 & \text{if } \nu_{(i,n)} = \chi_n, \tau_{(i,n)} + \delta_{(i,n),n+1} \leq t_{n+1}, \chi_{n+1} = 1 \\ \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

3 Analysis

3.1 Equilibrium analysis

There exists a multiplicity of Nash equilibria. In particular, attesters can coordinate to implement proposers acting at any particular time $\Delta^\star \in [0, \Delta]$ within the slot. Formally, we have the following result:

▶ **Proposition 1** (Multiple Equilibria). *For any $\Delta^\star \in [0, \Delta]$, there exists an equilibrium as follows:*

Proposer n selects t_n as follows:

$$t_n = n \cdot \Delta + \Delta^\star \tag{6}$$

and selects ϕ_n as follows:

$$\phi_n = \begin{cases} 1 & \text{if } t_{n-1} \leq (n-1) \cdot \Delta + \Delta^\star \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

Attester (i, n) selects $\nu_{(i,n)}$ as follows:

$$\nu_{(i,n)} = \begin{cases} 1 & \text{if (6) and (7) hold} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

and selects $\tau_{(i,n)}$ as follows:

$$\tau_{(i,n)} = \begin{cases} t_n + \delta_{n,(i,n)} & \text{if (6) and (7) hold} \\ n \cdot \Delta & \text{otherwise} \end{cases} \tag{9}$$

Proposition 1 arises because a proposer receives a zero pay-off unless their block earns sufficiently many attestations. In turn, if attesters coordinate on voting for a proposer's block only if the proposer releases their block at a particular time, then the proposer earns a strictly positive pay-off only if she releases their block at that particular time. Thus, since a proposer prefers a strictly positive pay-off to a zero pay-off, each proposer optimally releases their block at the release time on which attesters coordinate.

As an aside, we emphasize that the referenced coordination by attesters is equilibrium behavior. In particular, an attester receives a strictly positive pay-off only if their attestation is correct, and their attestation is correct only if it agrees with the majority of attesters in their slot. As a consequence, when all other attesters vote in one direction, each attester optimally votes in that same direction to avoid a zero pay-off.

Proof. We begin by establishing that (8) - (9) are optimal responses for any attester (i, n). Formally, we take as given that all attesters other than (i, n) follow the equilibrium actions (8) - (9) and also that all proposers follow the equilibrium actions (6) - (7); in that context, we demonstrate that (8) - (9) maximize (5) and thus these are equilibrium actions for each attester (i, n).

If (6) and (7) hold, then $\phi_n = 1$ follows directly for all $n \in \mathbb{N}$. Moreover, if all attesters other than (i, n) follow (8), then (6) and (7) imply $\nu_{(-i,n)} = 1$ which implies $\tilde{A}_n = 1$ for all $n \in \mathbb{N}$. Then, since (6) and (7) imply $\phi_n = 1$ for all $n \in \mathbb{N}$ and also $\tilde{A}_n = 1 \geq \gamma$, (3) therefore implies $\chi_n = 1$ for all $n \in \mathbb{N}$. In turn, since $\nu_{(i,n)} \neq \chi_n$ implies the lowest possible pay-off in (5), we have that $\nu_{(i,n)} = \chi_n = 1$ whenever (6) and (7) holds. If (6) and (7) do not hold, then (8) implies $\nu_{(-i,n)} = 0$ which implies $\tilde{A}_n = 0$ for all $n \in \mathbb{N}$. Moreover, (3) implies $\chi_n = 0$ for all $n \in \mathbb{N}$. In turn, since $\nu_{(i,n)} \neq \chi_n$ implies the lowest possible pay-off, we have that $\nu_{(i,n)} = \chi_n = 0$ whenever the conjunction of (6) and (7) do not hold. Thus, $\nu_{(i,n)} = 1$ is an optimal response if (6) and (7) and $\nu_{(i,n)} = 0$ is an optimal response otherwise, thereby establishing (8) as the equilibrium action for any attester (i, n).

To establish (9) as an optimal response for attester (i, n), note that (5) pointwise decreases in $\tau_{i,n}$ and thus it is optimal to set $\tau_{(i,n)}$ as low as possible subject to feasibility. In general, $\tau_{(i,n)} \geq n \cdot \Delta$ but $\nu_{(i,n)} = 1 \implies \tau_{(i,n)} \geq t_n + \delta_{n,(i,n)} = n \cdot \Delta + \Delta^\star + \delta_{n,(i,n)} > n \cdot \Delta$. As such, whenever $\nu_{(i,n)} = 0$, then $\tau_{(i,n)} = n \cdot \Delta$, whereas whenever $\nu_{(i,n)} = 1$, then $\tau_{(i,n)} = t_n + \delta_{n,(i,n)}$. Then, as per our proof of (8), (6) and (7) imply $\nu_{(i,n)} = 1$ which implies $\tau_{(i,n)} = t_n + \delta_{n,(i,n)}$, whereas if either (6) or (7) does not hold, then $\nu_{(i,n)} = 0$ which implies $\tau_{(i,n)} = n \cdot \Delta$, which thereby establishes (9).

We conclude by demonstrating that (6) - (7) are an optimal response for any proposer n. More formally, we take as given that all attesters follow the equilibrium actions (8) - (9) and also that all proposers other than proposer n follow the equilibrium actions (6) - (7); in this context, we establish that (6) - (7) maximize (5) and thus these are equilibrium actions for each proposer n.

Due to (8), any deviation in (6) or (7) implies $\nu_{(i,n)} = 0$ for all (i, n) which further implies $\tilde{A}_n = 0$. Then, under such a deviation, (3) implies $\chi_n = 0$ which implies a zero pay-off as per (1). Finally, since pay-offs are bounded below by zero, not deviating from (6) and (7) necessarily produces a higher pay-off than any such deviation and thus (6) and (7) are equilibrium actions. ◀

3.2 Model justification

The model presented in Section 2 is an idealized description of a blockchain consensus mechanism. A sequence of proposers is selected, each of which is given the right to produce a block for the slot they are assigned to in the sequence. Once the block is released, a set of attesters assigned for the current slot gets to vote for the presence or absence of the block.

When the proposer chooses to build on the previous block, they affirm its place in the canonical chain. There is no block tree: either the current proposer recognizes the block produced by the proposer before them as part of the canonical chain ($\phi = 1$), or they recognize that the previous proposer failed to produce a block which is part of the canonical chain ($\phi = 0$). With the assumption of a continuum of attesters, at equilibrium, sufficiently many votes reach the following proposer, allowing them to make the call on whether or not the previous proposer's block is canonical.

This model resembles the Streamlet protocol [13]. A proposer submits a block for consideration to the rest of the network. If $\gamma = 2/3$ share of attesters vote the block in, the block is notarized. If attesters do not, e.g., because the block is unavailable, the chain height is not increased, but the next slot starts, giving the opportunity to the next block producer to submit a block for consideration. Leaders extend the longest chain of notarized blocks they have seen.

The model also bears resemblance with the proposed (block, slot) fork choice rule of the Ethereum Gasper protocol [1], specifically the dynamically available chain produced by the protocol, when $\gamma = 1/2$. In this model of the fork choice, attesters submit a vote attesting to the presence or absence of a block at some given slot. The canonicity of a block is however complicated by the LMD-GHOST rule for block weight accumulation. Obtaining more than half of the attesters' vote may then neither be a sufficient nor a necessary condition to be part of the canonical chain.

Generally, we formulate the hypothesis that most Proof-of-Stake-based leader selection protocols will be exposed to timing games. As long as duties are assigned according to an absolute (wall-clock) time schedule, there exists no pressure to complete duties in a timely manner comparable to the random arrival process of leaders in Proof-of-Work. For instance, PBFT-based finalization protocols such as Tendermint [21] or HotStuff [31] do not perform a view change until some timeout is reached, which a leader may use to time their release appropriately. While a sufficiently decentralized committee of validators is an existing feature of these protocols, our model further highlights its role in enforcing timeliness at equilibrium, as described in Section 3.1.

4 An empirical case study: Ethereum

Following a formal analysis of the coordination game between proposers and attesters, we now investigate the occurrence of such strategic timing games in real-world systems. To this end, we examine Ethereum, an ideal candidate for the empirical analysis of potential timing games, owing to its mature MEV market structure and the availability of accessible, informative data points.

We show that timing games are indeed worth playing. However, we find that proposers do not delay their block release with the intention to capture more MEV. Instead, we find that delays are mostly due to latency in their signing processes. Thus, we can conclude that timing games are rational to engage in, but do not yet occur to their full possible extent.

4.1 Consensus mechanism

The Ethereum consensus mechanism is a composite of two protocols: variants of LMD GHOST [29] and Casper FFG [9], often referred to together as Gasper [10]. In this paper, we focus exclusively on Ethereum's *available chain* that is built roughly following LMD GHOST. This is because timing games only occur on the available chain. Within this protocol, time progresses in 12-second slots [3]. For each slot, one consensus participant, referred to as a validator, is selected as the *block proposer*. According to the honest validator specifications [4], which define the rules for honest protocol participation, a block should be released at the beginning of the slot (0 seconds into the slot). Furthermore, the protocol selects a committee of *attesters* from the validator set who vote on what they consider to be the latest canonical block as soon as they hear a valid block for their assigned slot, or 4 seconds into the slot, whichever comes first [4]. We refer to this 4-second mark as the *attestation deadline*. This dynamic, in which block proposers must release their block early enough for attesters to receive it via the peer-to-peer network before the attestation deadline, results from the attestation deadline serving as a coordination Schelling point [27]. It is worth noting that the honest validator specification prescribes block proposers to release their block at the beginning of the slot, while attesters only attest 4 seconds into the slot (unless a valid block is heard prior to the attestation deadline). This opens up room for block proposers to release their block strategically – i.e., as late as possible while ensuring they accumulate a sufficient share of attestations.

4.2 Block production process

To assess the potential benefits of timing games for block proposers, it is important to comprehend the value of time and the process by which MEV opportunities are captured in the block proposing process. In Ethereum, the MEV market structure evolved and matured significantly over time, turning the block production process into an intricate interplay between specialized actors [30]. This division of labor enables validators to profit from MEV without engaging in the complex process of identifying MEV opportunities themselves. Instead, validators can outsource the task of building a maximally profitable block to an out-out-protocol block auction process known as MEV-Boost [5].

Searchers look for MEV opportunities (e.g., arbitrages), and submit bundles of transactions alongside bids to express their order preference to *block builders*. Block builders, in turn, specialize in packing maximally profitable blocks using searcher bundles and other available user transactions before submitting their blocks with bids to *relays*. Relays act as trust facilitators between block proposers and block builders, validating blocks received by block builders and forwarding only valid headers to validators. This ensures validators cannot steal the content of a block builder's block, but can still commit to proposing this block by signing the respective block header. In the long run, Ethereum's plans include enshrining this currently out-of-protocol mechanism into the protocol [7, 8] to eliminate relay trust assumptions. It is worth noting that MEV-Boost is an opt-in protocol, and validators can always choose to revert to local block building. Finally, when a validator is selected to propose a block in a given slot, they request the highest-bidding block header from the relay, sign it, and return the signed block header to the relay, which then releases the block to the peer-to-peer network.

In summary, searchers find MEV opportunities and express their transaction-ordering preferences within a block via bids. Block builders aim to build maximally profitable blocks using searcher bundles and user transactions, then submit their block content and bids to relays. Validators ultimately request the highest-paying block header, sign it and return it to relays, which release the signed block to the peer-to-peer network. Due to competition at all levels in this block production process (except for the block proposing monopoly), the block proposer is able to capture most of the MEV via this block auction.

MEV-Boost block auction

Here, we granularly outline the sequence of events that take place during the block construction of MEV-Boost block auctions on the Ethereum network. Figure 1 illustrates these events along with their corresponding timestamps, and is intended to serve as a reference for the remainder of this empirical analysis.

The auction for the block of slot n begins in slot $n-1$ (at $t = -12000$ms), during which builders submit blocks alongside bids to relays. This competitive process between block builders determines the right to construct the block for slot n and secures potential MEV-derived profits (block building profit equates to extracted MEV minus bid value). For each bid, the relay logs the timestamps of events at which the bid was received by the relay (`receivedAt`). After some validity checks are completed by the relay, the bid is made available to the proposer (`eligibleAt`). When the proposer chooses to propose a block [2], the proposer requests `getHeader` to receive the highest bidding, eligible block header from the relay. Upon receiving the header associated with the winning bid, the proposer signs it and thereby commits to proposing this block built by the respective builder in slot n. The signed block header is sent to the relay, along with a request to get the full block content from the relay (`getPayload`). Finally, the relay receives the signed block header (`signedAt`)

[2] An honest participant will request the block header shortly before slot n such that the block can be released on time, at the beginning of slot n ($t = 0$ms).

and publishes the full block contents to the peer-to-peer network and proposer. As soon as peers see the new block, validators assigned to the slot can attest to it. This cycle completes one round of consensus repeating every slot.

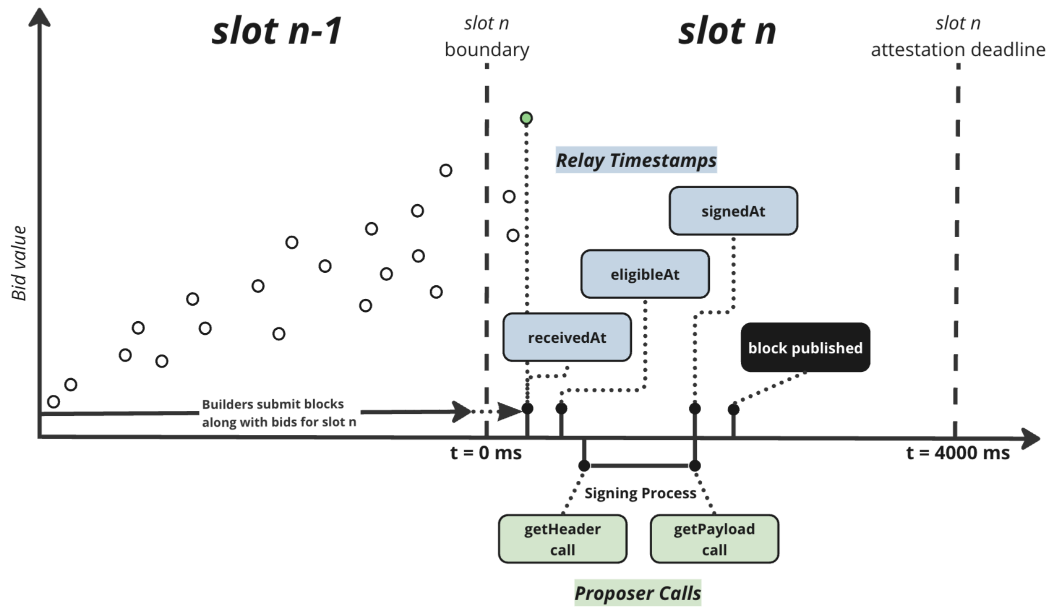


Figure 1 Logical representation of the block production process for slot n. Builder bids begin streaming in during slot $n-1$, after which the proposer and relay interact through requests and responses.

4.3 Data sets

The analysis utilizes data provided by the *ultra sound relay* from March 4, 2023, to April 11, 2023. This covers just under 185,000 slots, interspersed from slot 5,965,398 to slot 6,282,397, and includes all bids placed by block builders through this relay. There were over 800 bids per slot, for a total of over 150 million bids. The winning block originated from the *ultra sound relay* for nearly 85,000 of these slots, and so we measure timestamps and other properties for those slots when investigating winning bids specifically. Finally, we augmented the winning slots with various on-chain measures from the execution layer (EL) and consensus layer (CL), such as attestations and aggregations, using a combination of analytical tools like Dune and direct observation of the peer-to-peer network.

4.4 Are timing games worth playing?

Marginal value of time

Timing games offer potential for substantial profit due to the increased MEV opportunities they provide. First, we assess whether timing games are worth playing for proposers, by estimating the incremental MEV gained per second. We utilize all bids submitted by builders from the *ultra sound relay* to examine the relationship between the timestamp at which the relay received a bid (`receivedAt` timestamp relative to the slot boundary) and the bid value, residualized against slot fixed effects to account for differences between low- and high-MEV regimes and other unobservable forms of heterogeneity. We then fit a regression line to this relationship, obtaining a slope with a coefficient of 0.0065 ETH per second, an estimate for the marginal value of time. Figure 2 depicts the linear increase in median bid values over the slot duration on a point-by-point basis, and the distribution of bid receival times, indicating that most bids are submitted between four seconds before the slot boundary to one second after. This analysis shows there exists a linear positive marginal value of time, indicating that a rational block proposer would participate in timing games.

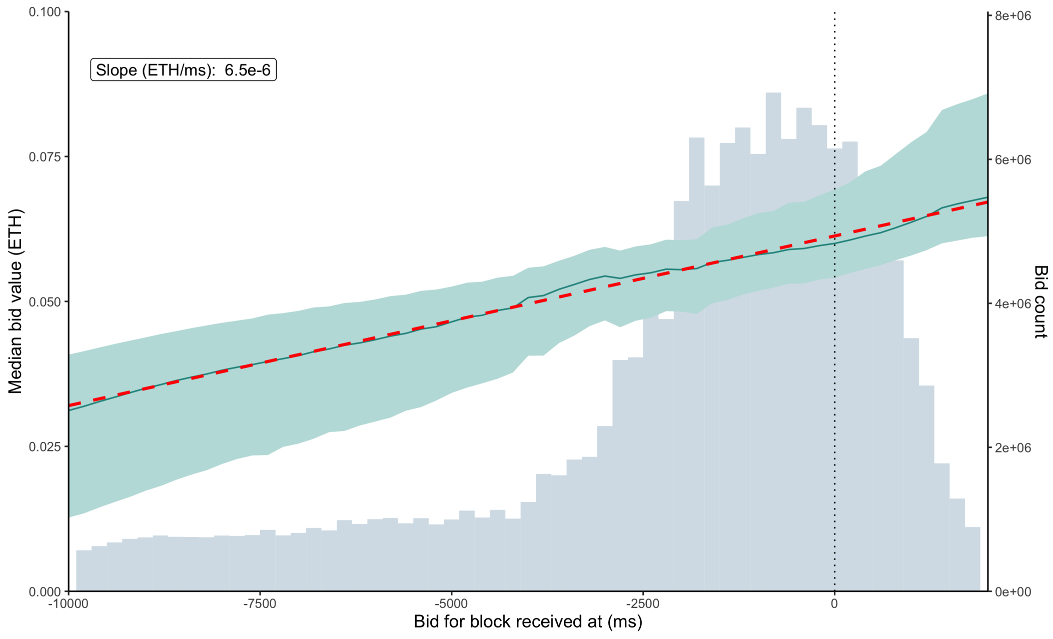


Figure 2 Analysis of bid values and their distribution over slot duration. The histogram (in blue) shows the distribution of bid counts across time. The dark green line represents the median bid value in Ether (ETH) for each time bin (with its associated IQR in green), residualized against the slot fixed effects that are estimated in a linear regression of bid on timestamp (dashed red line). The x-axis shows time in milliseconds relative to the slot boundary, the left y-axis displays the residualized bid value in ETH, and the right y-axis displays the count of bids.

4.5 Are block proposers playing timing games?

Having shown that timing games are worth playing, we turn our attention to whether proposers are currently taking advantage of the opportunity to accumulate more MEV by committing to a bid later than foreseen by the honest validator specifications.

Characterizing late block signing behavior

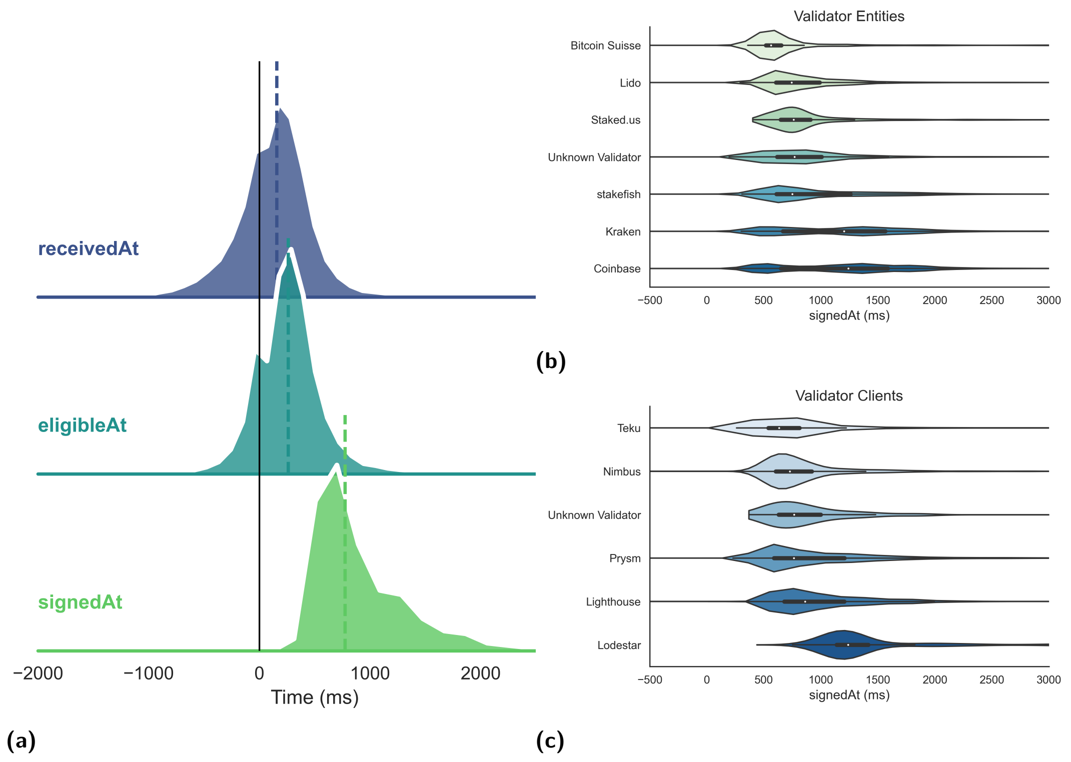
First, we investigate whether block proposers sign headers and associated bids later than the slot boundary ($t = 0$), the time stipulated by the honest protocol specifications to broadcast their block to the network. We observe that meadian winning bid is signed 774 ms after the slot boundary ($t_{(111573)} = 575.5$, $p < 1 \times 10^{-20}$, using a two-tailed paired Student t-test) and about 513 ms after the relay made the bid eligible ($t_{(111573)} = 472.6$, $p < 1 \times 10^{-20}$, using a two-tailed paired Student t-test). Figure 3a displays the distribution of timings for winning bids, based on *ultra sound relay* timestamps for bid reception from the builder (`receivedAt`, median = 157ms), eligibility for proposer signing (`eligibleAt`, median = 260ms), and the actual signing by proposers (`signedAt`, median = 774ms). To better understand the reasons behind late-signing behavior by proposers, we map validator public keys to their staking entities and CL clients, see Figure 3b and 3c respectively). Validator to staking entity mappings were obtained via a combination of open source data sets [3], and validator to client mappings were obtained using blockprint [2], an open source tool assigning client labels to validators based on their attestation packing on the Ethereum beacon chain. We found that staking entities such as Kraken ($t = 38.9$, $p < 1 \times 10^{-20}$) and Coinbase ($t = 67.6$, $p < 1 \times 10^{-20}$), as well as proposers using the Lodestar client ($t = 44.9$, $p < 1 \times 10^{-20}$) sign block headers significantly later than other block proposer types (results were obtained using two-tailed unpaired Student t-tests). Notably, additional analysis is required to differentiate the interdependencies between validator entities and clients. This analysis shows that proposers are signing blocks significantly later than expected, but it does not yet clarify the underlying reasons, which could include participation in timing games or increased latency for independent reasons, e.g., longer signing processes.

We subsequently collected data for 1241 slots (slot 6,200,251 to 6,204,957 on April 11, 2023) and used the time difference between `getHeader` and `getPayload` calls by proposers as an approximation to estimate the duration of the signing process (see Figure 1). Figure 4 shows that the median difference between `getHeader` and `getPayload` call is 418 ms. Interestingly, this delay, attributable to the signing process, accounts for 75.42% of the overall latency. This percentage was determined by calculating the difference between the signing time and the moment the bid was deemed eligible by the relay on a slot-by-slot basis, using the formula $\frac{\text{median}(\texttt{getPayload}-\texttt{getHeader})}{\text{median}(\texttt{signedAt}-\texttt{eligibleAt})} \times 100$. We conclude that late signing behavior is primarily attributed to latency caused by the signing process, rather than intentional delays to incorporate more MEV in blocks. This finding aligns with the hypothesis that large US-based staking entities, such as Coinbase and Kraken, may prefer utilizing sophisticated remote secure signing mechanisms, resulting in a lengthier signing process compared to other parties.

4.6 The impact of latency on the peer-to-peer network

Our prior results indicate that validators are largely not engaging in timing games to accrue more MEV. Nonetheless, we assess the implications of late signing of consensus messages on the peer-to-peer network. Specifically, we examine the relationship between relay timestamps and the time at which (1) blocks and (2) attestations and aggregations are first seen by the rest of the peer-to-peer network. The consensus layer data was obtained through nodes run by the Ethereum Foundation, for 2643 slots (slots 6,357,601 to 6,363,807 on May 3 and 4, 2023). Figure 5a shows the sequence of these event timestamps over the course of a slot. We subsequently assess the correlations between each of these event pairs, as depicted in Figure 5b. Our analysis reveals high correlations between the time at which blocks are signed by proposers (i.e., the `signedAt` relay timestamp) and the time at which blocks (correlation coefficient = 0.986) and attestations (correlation coefficient = 0.971) are initially observed by the peer-to-peer network. These findings underscore the significance of proposers signing blocks promptly, as it considerably impacts the downstream processes at the consensus layer in the network.

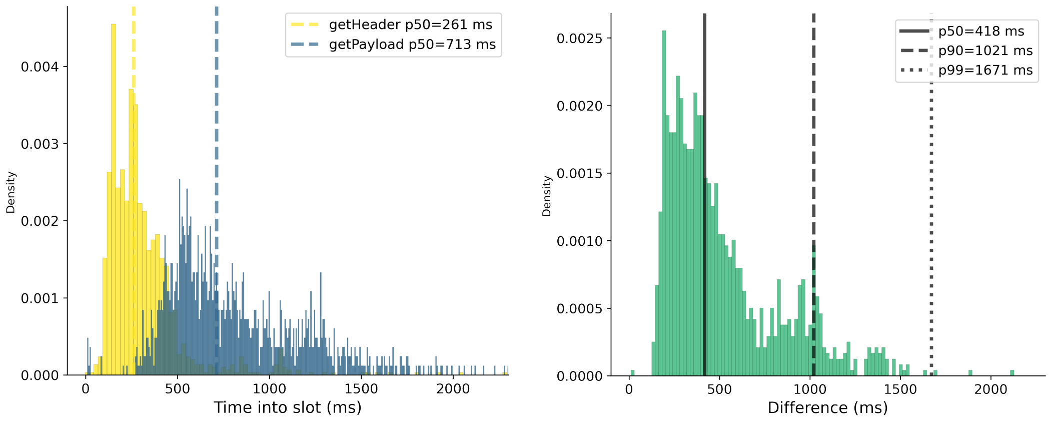
[3] Dune Spellbooks: `https://dune.com/spellbook`, Mevboost.pics Open Data: `https://mevboost.pics/data.html`



■ **Figure 3** Analysis of event timestamps and their distributions among Validator Clients and Entities. (3a) Multiple Kernel Density Estimation (KDE) distributions of event timestamps from the relay data, showing the probability density functions for three event types: `receivedAt` (blue), `eligibleAt` (green), and `signedAt` (light green). (3b-3c) Violin plots comparing the distribution of `signedAt` event timestamps for the top 7 validator entities and clients. The x-axis represents time in milliseconds (ms) relative to the slot boundary, while the y-axis displays validator clients and entities, ordered by the mean `signedAt` time. The width of each violin plot signifies the kernel density estimation of the `signedAt` event timestamps, demonstrating the distribution and frequency of the events within each group.

We evaluate the impact of the release time of block n on the share of attestations that vote for block n and are included on-chain in block $n+1$. As a reminder, attestations of slot n are only included on-chain one or more slots later (slot $\geq n+1$). We use the time at which blocks are signed by proposers (`signedAt`) as a heuristic for the block release time. In our analysis, we focus on attestations included in the subsequent slot and compute a metric *next-slot attestation share*. This metric refers to the share of attestations voting for block n that are included in block $n+1$, out of the total number of slot n attestations included in block $n+1$ that vote for any block other than n. The argument is that if a proposer releases their block n too late, attesters will not receive it in time to vote for it before their attestation deadline ($t = 4000$ms, see Figure 1, and [4]). Hence, in such scenarios attesters vote for a different block (e.g., the parent block). This is captured by the next-slot shares metric.

Figure 6 shows that late blocks do indeed cause a steep drop-off in the share of attestations voting for that block that are included in the subsequent block. We observe that the next-slot attestation share remains close to one for as long as the block is signed within the first two seconds of the slot. Once the two second threshold is crossed, there is a substantial



(a) **(b)**

Figure 4 Estimating the latency induced by the signing process. (4a) Histogram of `getHeader` and `getPayload` call timestamps relative to slot boundary. The histogram displays the density of events occurring at different times into the slot (in milliseconds) for `getHeader` (yellow) and `getPayload` (blue) calls. (4b) Histogram of the time difference between `getHeader` and `getPayload` calls. The histogram shows the density of time differences (in milliseconds) `getHeader` and `getPayload` calls. Vertical lines represent the 50th (solid), 90th (dashed), and 99th (dotted) percentiles of the distribution.

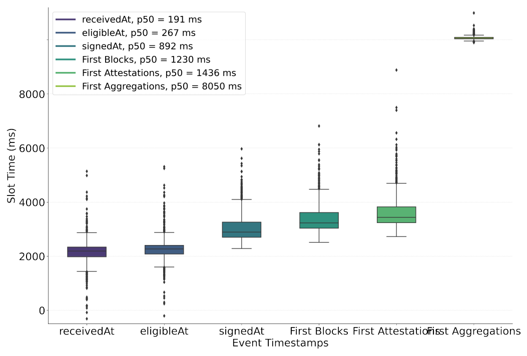
drop-off and many winning blocks earn fewer than half of the next-slot attestation share, which continues to rapidly decrease towards zero as we approach the theoretical $t = 4000$ms attestation deadline. These results demonstrate the impact of untimely block proposals on the rest of the peer-to-peer network and highlight the importance of signing and broadcasting blocks on time in order to prevent fair rewards for honest consensus participation, missed slots and chain reorganizations.

5 Discussion

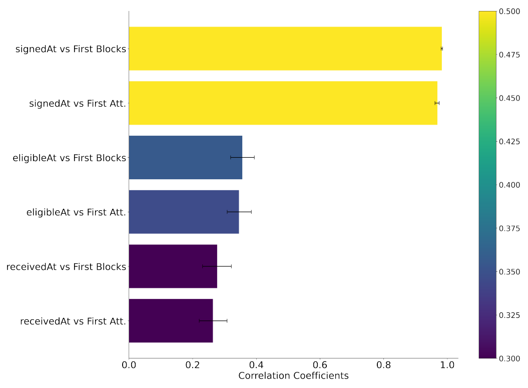
In this paper, we present an argument that consensus participants are subject to exogenous incentives, primarily MEV, that exist outside the consensus mechanism itself. This highlights the imperative for blockchain protocols to ensure economic fairness among all consensus participants. Specifically, it necessitates a design where honest and rational consensus participation become indistinguishable, and honesty within the protocol is the most profitable strategy. This approach ensures that rational participants have no incentive to deviate from honest consensus participation.

We present a model that highlights the time-dependent value for consensus participants and probes into the strategic timing considerations that block proposers face. Our model uncovers a spectrum of equilibria wherein attesters can enforce any deadline for block proposals to achieve canonical status, thereby emphasizing the crucial role of Schelling points as coordination mechanisms. For instance, in the Ethereum network we observe the emergence of such Schelling points through the default settings of client software. The widespread use of these default settings among consensus participants generally ensures their effectiveness.

We support our theoretical findings by observations of the Ethereum network. Our analysis demonstrates that timing games are indeed worth playing for block proposers, enabling them to capture additional MEV by delaying their block proposals beyond the



(a)



(b)

Figure 5 Analysis of relay and consensus layer timestamps. (5a) Box plot of the time differences between relay and consensus timestamps. The box plots display the distribution of time differences for `receivedAt`, `eligibleAt`, and `signedAt` events, as well as blocks, attestations, and aggregations first seen by the peer-to-peer network. The boxes represent the interquartile range (IQR) from the first quartile (Q1, 25th percentile) to the third quartile (Q3, 75th percentile), while the whiskers extend to the minimum and maximum values within three times the IQR. The horizontal lines within the boxes represent the median values. (5b) Bar plot of Pearson correlation coefficients for each pair of event timestamps. The bars represent the mean correlation coefficient for each relationship, while the error bars represent the 95% confidence intervals obtained via bootstrapping.

timeframe prescribed by the honest validator specification. However, we observe that current instances of delayed block proposals are primarily due to latency in the block signing process, rather than a conscious strategy to maximize profits. The apparent lack of maximal MEV capture by honest proposers could be attributed to either a lack of common knowledge or existing social norms around this practice. It's clear, however, that these are not sustainable safeguards for maintaining economic fairness.

The implications of timing games are manifold and significant. An rational participant who engages in timing games will outperform honest participants, leading to a centralization of stake over time. Hypothetically, this could culminate in a breach of consensus security. In a more practical sense, it may encourage individual stakers to delegate their stake to professional entities adept at these practices, negatively impacting the network's decentralization. Moreover, timing games can overload the messaging system within a short time span, potentially causing cascading failures at the peer-to-peer layer, particularly within client systems.

Essentially, timing games are facilitated by the monopolistic right that block proposers possess for a single round of consensus. Introducing competition in block proposing, similar in effect to the exogenous randomness in Proof of Work (PoW) systems, emerges as a potential solution. However, the challenge lies in deterministically selecting a winning proposer, or reverting to peer-to-peer latency races, which in itself is centralizing. Alternatively, an on-chain heuristic for timely block proposals could incentivize timely participation, yet the allure of MEV rewards might still outweigh any in-protocol consensus rewards. Tackling the root cause of timing games remains an open challenge.

In the Ethereum context, a late-block reorging mechanism has been adopted in the fork choice, effectively imposing a 4-second deadline for block proposers. This constraint significantly limits the extent to which block delays are possible. Looking ahead, the adoption of (block, slot) type of attestations is likely, further refining the protocol. However, it

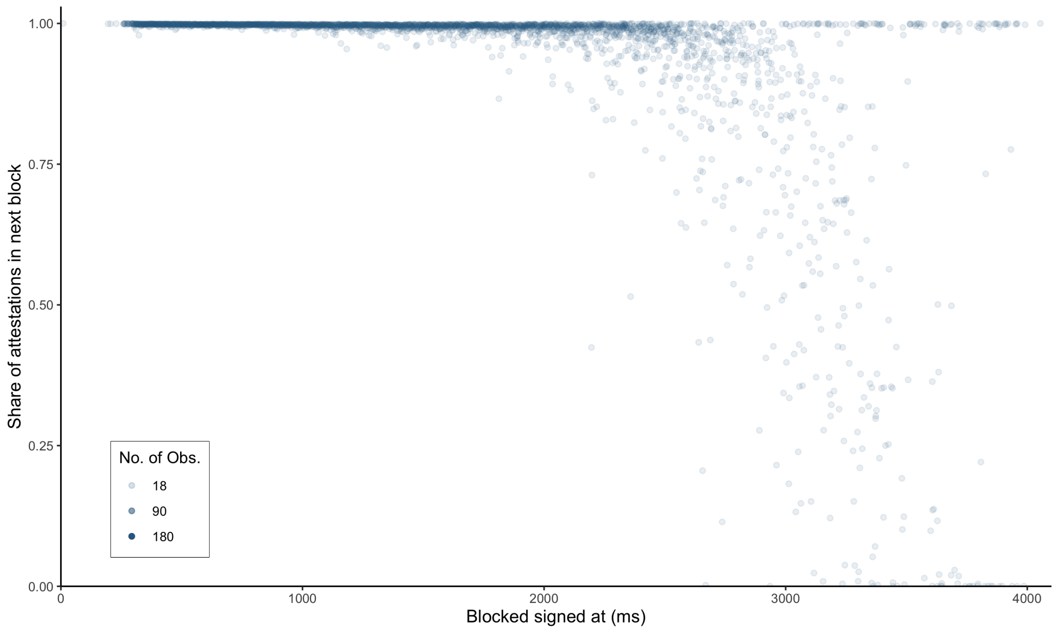


Figure 6 Effect of block release time on share of attestations that vote for said block and are included in subsequent block. The `signedAt` timestamps are used as a heuristic for release time. Each point on the graph corresponds to the time (in milliseconds) at which the winning block was signed within the slot and the average share of attestations voting for it that are included in the next block.

remains challenging to address the root cause of timing games, as it is deeply intertwined with the fundamental workings of Proof of Stake (PoS). Although limiting the length of the proposer's interval is feasible, completely eliminating the monopolistic market structure of block proposers proves to be a difficult task.

Consequently, it may prove valuable to find a more general abstraction for PoS type of protocols and further explore the implications of consensus participants being exposed to incentives outside of consensus itself, such as MEV. More generally, assuming rational as opposed to honest type of consensus participation should prove significant in designing economically fair blockchain protocols. Revisiting the existing literature on consensus mechanisms through the lens of rational as opposed to honest consensus participation should prove valuable.

References

1 (block, slot) fork choice. `https://github.com/ethereum/consensus-specs/pull/2197`. Accessed: 2023-10-05.

2 blockprint. Accessed: 2023-10-05. URL: `https://github.com/sigp/blockprint`.

3 Ethereum consensus specifications - beacon chain. Accessed: 2023-10-05. URL: `https://github.com/ethereum/consensus-specs/blob/dev/specs/phase0/beacon-chain.md`.

4 Ethereum consensus specifications - honest validator. Accessed: 2023-10-05. URL: `https://github.com/ethereum/consensus-specs/blob/dev/specs/phase0/validator.md`.

5 Mev-boost. Accessed: 2023-10-05. URL: `https://github.com/flashbots/mev-boost`.

6 Kushal Babel, Philip Daian, Mahimna Kelkar, and Ari Juels. Clockwork finance: Automated analysis of economic security in smart contracts. *arXiv preprint arXiv:2109.04347*, 2021.

7 Vitalik Buterin. Proposer/block builder separation-friendly fee market designs. Accessed: 2023-10-05. URL: `https://ethresear.ch/t/proposer-block-builder-separation-friendly-fee-market-designs/9725`.

8 Vitalik Buterin. Two-slot proposer/builder separation. Accessed: 2023-10-05. URL: `https://ethresear.ch/t/two-slot-proposer-builder-separation/10980`.

9 Vitalik Buterin and Virgil Griffith. Casper the friendly finality gadget. *arXiv:1710.09437 [cs.CR]*, 2019. URL: `https://arxiv.org/abs/1710.09437`.

10 Vitalik Buterin, Diego Hernandez, Thor Kamphefner, Khiem Pham, Zhi Qiao, Danny Ryan, Juhyeok Sin, Ying Wang, and Yan X Zhang. Combining ghost and casper. *arXiv preprint arXiv:2003.03052*, 2020.

11 Christian Cachin and Marko Vukolić. Blockchain consensus protocols in the wild. *arXiv preprint arXiv:1707.01873*, 2017.

12 Miles Carlsten, Harry Kalodner, S Matthew Weinberg, and Arvind Narayanan. On the instability of bitcoin without the block reward. In *Proceedings of the 2016 acm sigsac conference on computer and communications security*, pages 154–167, 2016.

13 Benjamin Y Chan and Elaine Shi. Streamlet: Textbook streamlined blockchains. In *Proceedings of the 2nd ACM Conference on Advances in Financial Technologies*, pages 1–11, 2020.

14 Philip Daian, Steven Goldfeder, Tyler Kell, Yunqi Li, Xueyuan Zhao, Iddo Bentov, Lorenz Breidenbach, and Ari Juels. Flash boys 2.0: Frontrunning, transaction reordering, and consensus instability in decentralized exchanges. *CoRR*, abs/1904.05234, 2019. `arXiv:1904.05234`.

15 Francesco D'Amato, Joachim Neu, Ertem Nusret Tas, and David Tse. No more attacks on proof-of-stake ethereum? *arXiv preprint arXiv:2209.03255*, 2022.

16 Francesco D'Amato and Luca Zanolini. Recent latest message driven ghost: Balancing dynamic availability with asynchrony resilience. *arXiv preprint arXiv:2302.11326*, 2023.

17 Francesco D'Amato and Luca Zanolini. A simple single slot finality protocol for ethereum. *arXiv preprint arXiv:2302.12745*, 2023.

18 Amir Dembo, Sreeram Kannan, Ertem Nusret Tas, David Tse, Pramod Viswanath, Xuechao Wang, and Ofer Zeitouni. Everything is a race and nakamoto always wins. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 859–878, 2020.

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

20 Kose John, Thomas Rivera, and Fahad Saleh. Economic implications of scaling blockchains: Why the consensus protocol matters. *NYU Stern Working Paper*, 2023.

21 Jae Kwon. Tendermint: Consensus without mining. *Draft v. 0.6, fall*, 1(11), 2014.

22 Kevin Liao and Jonathan Katz. Incentivizing blockchain forks via whale transactions. In *Financial Cryptography and Data Security: FC 2017 International Workshops, WAHC, BITCOIN, VOTING, WTSC, and TA, Sliema, Malta, April 7, 2017, Revised Selected Papers 21*, pages 264–279. Springer, 2017.

23 Barnabé Monnot. Timing games in proof-of-stake. Accessed: 2023-10-05. URL: `https://ethresear.ch/t/timing-games-in-proof-of-stake/13980`.

24 Joachim Neu, Ertem Nusret Tas, and David Tse. Ebb-and-flow protocols: A resolution of the availability-finality dilemma. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 446–465. IEEE, 2021.

25 Joachim Neu, Ertem Nusret Tas, and David Tse. Two more attacks on proof-of-stake ghost/ethereum. In *Proceedings of the 2022 ACM Workshop on Developments in Consensus*, pages 43–52, 2022.

26 Fahad Saleh. Blockchain without waste: Proof-of-stake. *Review of Financial Studies*, 34(3):1156–1190, 2021.

27 Thomas C Schelling. *The Strategy of Conflict: with a new Preface by the Author*. Harvard university press, 1980.

28 Caspar Schwarz-Schilling, Joachim Neu, Barnabé Monnot, Aditya Asgaonkar, Ertem Nusret Tas, and David Tse. Three attacks on proof-of-stake ethereum. In *Financial Cryptography and Data Security: 26th International Conference, FC 2022, Grenada, May 2–6, 2022, Revised Selected Papers*, pages 560–576. Springer, 2022.

29 Yonatan Sompolinsky and Aviv Zohar. Secure high-rate transaction processing in Bitcoin. In *International Conference on Financial Cryptography and Data Security*, pages 507–527. Springer, 2015.

30 Sen Yang, Fan Zhang, Ken Huang, Xi Chen, Youwei Yang, and Feng Zhu. Sok: Mev countermeasures: Theory and practice, 2022. doi:10.48550/ARXIV.2212.05111.

31 Maofan Yin, Dahlia Malkhi, Michael K Reiter, Guy Golan Gueta, and Ittai Abraham. Hotstuff: Bft consensus with linearity and responsiveness. In *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing*, pages 347–356, 2019.