Proof-of-Stake Mining Games with Perfect Randomness

Matheus V. X. Ferreira, S. Matthew Weinberg

PRINCETON UNIVERSITY

Chat and Contact Information

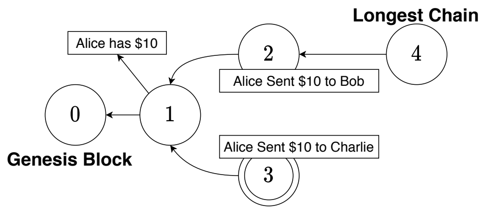
PDF: https://matheusvxf.github.io/files/tapia.pdf
Zoom Chat: https://princeton.zoom.us/j/91423773580
Website: https://www.cs.princeton.edu/~mvxf
Email: mvxf@cs.princeton.edu

Background

- **Blockchains.** Distributed decentralized ledgers.
  - **Main Challenge.** How to reach consensus?
- Bitcoin's solution is:
  - **Proof-of-Work (PoW).** Compute the next Leader through crypto-puzzles.
  - **Longest Chain Rule.** For conflict resolution.
  - *However,* PoW requires a lot of energy!!!
    - 0.21% of world's energy (2019)
- **Proof-of-Stake (PoS).** An alternative to PoW
  - Use an energy-efficient protocol to compute the Leader Election.
  - Winning probability proportional to Wealth!
  - *However*, more complex than PoW.
- **Research Question.** Can PoS preserve miner's incentive guarantees from PoW?

Model

- **Study a two-player game.** Miner 1 (Strategic and owns α fraction of the currency) and Miner 2 (Honest).
- **Asynchronous communication.**
- **Perfect Randomness.** There is an unbiased, unpredictable source of randomness.
- **Objective.** Miner 1 wishes to maximize their fraction of blocks in the longest path subject to only following *undetectable deviations*.

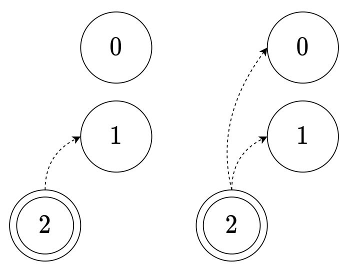


Blockchain Ledger. Miners create blocks at discrete time steps. Initially, Alice owns $10 dollars and attempts to double spend by sending the same coins to both Bob and Charlie in different transactions. The *Longest Chain Rule* states that only the longest path is valid. By introducing a confirmation delay, Alice can only double spend if she can fork the longest chain after the confirmation time.

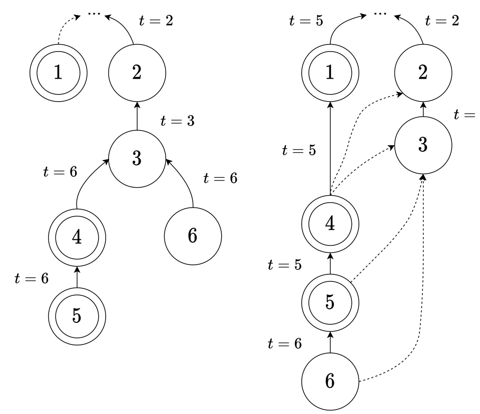
$$X_n \coloneqq Winner\ of\ Round\ n$$

$$Pr[X_{n+1} = Miner\ 1 | X_n] = \alpha$$

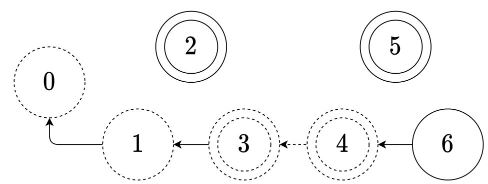
Perfect Randomness. The winner of (n+1)th round is *unpredictable* and *unbiased* until the nth round begins. To be *unbiased* the source of randomness should be independent of the current state of the blockchain. To be *unpredictable*, randomness is introduced in all rounds. The winner of the n-th round should create at most one block with timestamp n pointing to any block $< n$.



Proof-of-Work vs Proof-of-Stake. In the 2nd round, Miner 1 attempts to mine block 1. In PoW, Miner 1 choose to either mine block 2 pointing to 0 or 1. In PoS, Miner 1 wins the right to create a block with timestamp 2 pointing to any block with a lower timestamp. However, Miner 1 can only publish a single block with timestamp 2 since publishing two blocks with the same timestamp is a proof that Miner 1 is deviating from the protocol. Withholding blocks is an undetectable deviation because the communication is asynchronous.



Selfish-Mining with Nothing-at-Stake. This examples highlights the distinction between PoW and PoS and the reason PoS allows profitable undetectable deviations when honest mining is optimal for PoW. Miner 1 wins in rounds 1, 4 and 5. Miner 2 wins in rounds 2, 3 and 6. In the left, Miner 1 uses the selfish mining strategy. In round 3, Miner 1 loses his advantage and abandons block 1 treating block 3 as the genesis block. In the right, instead of publishing block 4 pointing to 3, Miner 1 publishes (1, 4, 5) pointing to block 0. That's because Miner 1 does not need to commit to publish block 4 pointing to either 3 or 1. In fact, Miner 1 had the option to publish block 4 pointing to any block in the set (0, 1, 2, 3). This is strategy is not possible in PoW: Miner 1 must first *commit* to create block 4 pointing to either 0, 1, 2, or 3 even before the 4-th round begins.



Checkpoints. Dashed blocks are checkpoints. We define the genesis block as a checkpoint. Whenever Miner 1 publishes more than half of all the blocks he could have published since the last checkpoint, define the longest chain as a checkpoint. In the example, block 1 is a checkpoint because Miner 1 has no unpublished blocks < 1. Block 6 is not a checkpoint because Miner 1 has on unpublished (block 5) since the last checkpoint (block 4).

Results

- We introduce new mathematical techniques to study blockchain mining games.
- PoS does not preserve the incentive guarantees of PoW even with perfect randomness.
  - When α ≅ 0.32 honest mining is optimal in PoW but **not in** PoS.
- Honest mining is optimal when α < 0.307 in PoS.

Proof Technique

- **Simplifying the Action Space.**
  - Introduce assumptions on the actions of strategic miners.
  - Prove there is an optimal strategy that satisfy these assumption (using reductions).
- **Simplifying the State Space.**
  - **Checkpoints.** Define an increasing sequence of blocks in the longest path $P_0, P_1, P_2, \ldots$
  - Prove there is an optimal strategy that never forks blocks in this sequence.
    - This implies the strategy space resets whenever a new P_i is defined.
  - Prove this strategy is optimal only if the sequence grows fast: $E[P_{i+1} - P_i]$ is small.
    - This implies the optimal strategy is a Positive Recurrent Markov Chain.

Conclusion

- PoS protocols are only incentive compatible when it has access to perfect randomness.
- Even with perfect randomness, PoS does not preserve incentive guarantees from PoW.
- However, PoS can approximate the incentive guarantees of PoW given perfect randomness.

References

- Sapirshtein at al. Optimal Selfish Mining Strategies in Bitcoin.
- Kiayias at al. Ouroboros: A Provably Secure Proof-of-Stake Blockchain Protocol
- Brown-Cohen at al. Formal Barriers to Longest-Chain Proof-of-Stake Protocols
- Bitcoin Energy Consumption Index. Digiconomist