Scalability Challenges & Proof of Stake

Fan Long

University of Toronto

Scalability

Scalability Problem

PayPal
VISA

Transactions per Second:	~7	~30	~200	~3000

Undesirable user experience, long processing delay, and skyrocketing transaction fees!

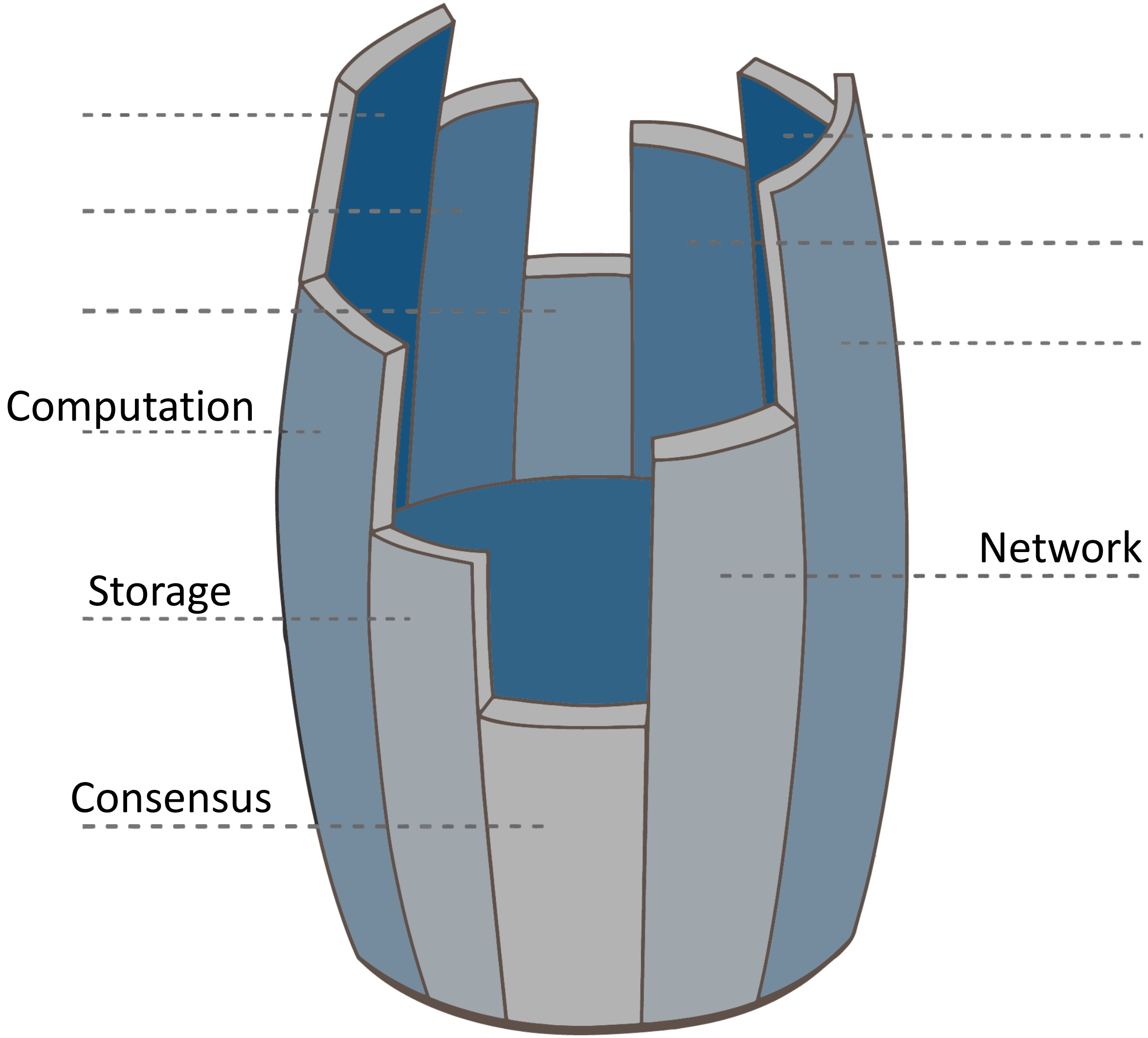
Why standard blockchain systems have very low throughput?

Standard Nakamoto Consensus

- **Proof-of-work**: participants perform computation tasks to generate blocks
- **Longest-chain**: all participants agree on the longest chain as the valid transaction history
- **Slow/small** block generation
  - Bitcoin: **1MB block per 10 minutes**
  - Ethereum: **~100k block per 15 seconds**
  - Each transaction takes 250-300B at least
  - Very limited throughput

Blockchain Scalability Barrel

- High throughput requires:
  - More storage for blockchain data
  - More network communication
  - More computation to process txs
- Consensus algorithm is the most limiting factor right now
- But to build a scalable blockchain, we need to address all of them



Possible Solutions

- **Large** blocks / **fast** block generation

- **Consensus** with a **small** number of selected nodes
  - Elect a potentially rotating committee to generate blocks
  - Often use BFT algorithms to improve throughput
- Different chain structures to process **concurrent blocks**
  - Organize blocks in Direct Acyclic Graph (DAG)

- Partition of the state – **sharding**

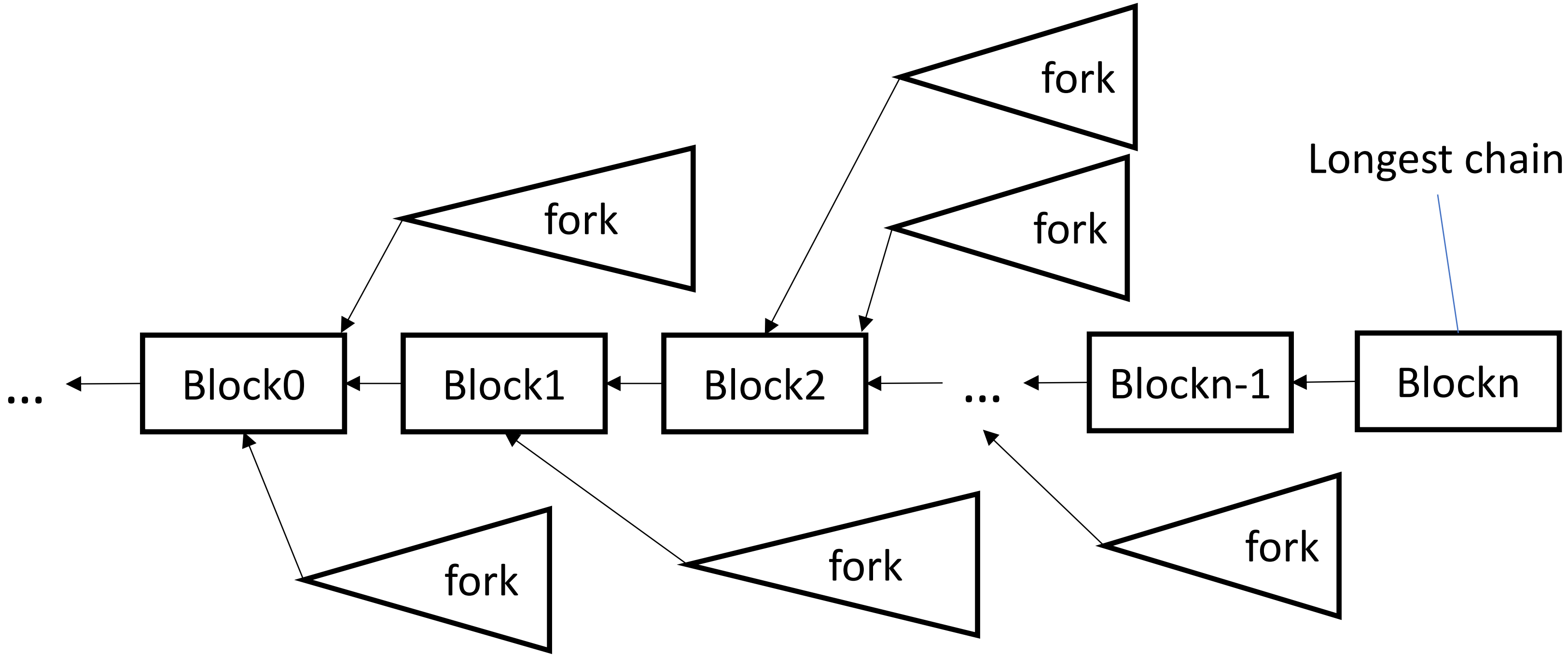
Eternal Fights on Bitcoin Block Size

- Bitcoin starts to get congested around year 2013-2014
- Some minor optimizations like SegWit implemented, but not enough
- Miners and developers fighting on whether to increase the block size
  - Arguments for: Solve congestion
  - Arguments against: Make the chain too large and harm decentralization

- Hard fork of Bitcoin Cash, which uses 8MB block
- Proposals on even removing block limit

What if we run Nakamoto Consensus with **large/fast** blocks generation?

Forks in Nakamoto Consensus



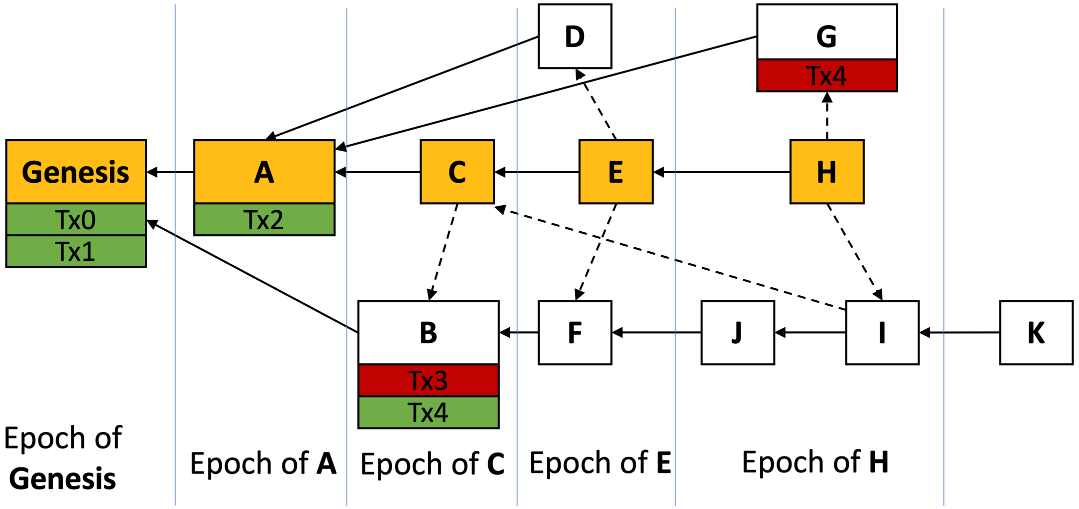
- Forks waste network/processing resources
- Downgrade safety
    - Attacker needs less resource to beat the longest chain

Run Consensus in a Small Group

- Manually elected nodes
  - For example, 21 full nodes in EOS
  - Permissioned network like Ripple/Stellar
  - Sacrifice decentralization completely for efficiency

- Have a rotating committee
  - Recent PoW miners form the committee [ByzCoin, HybridConsensus]
  - Use a verifiable random function to elect a committee [Algorand]
  - Run BFT among committee members to determine each block

Process Concurrent Blocks with DAG



- Allow multiple predecessors for the block to form a direct acyclic graph (DAG)
  - GHOST / Conflux
  - Specter and Phantom
- **Challenge**: How to obtain a total order of blocks

Sharding

- Each node in the network verifies every transaction in the blockchain
- **Idea:** Partition the state and each node verifies part of transactions
    - Each partition is called a shard
- Many security problems and challenges
    - How to partition the blockchain state?
    - How to handle inter-shard transactions?
    - Each shard becomes more vulnerable. Attacker can focus on one shard
- Trading security for efficiency
    - OmniLedger
    - ETH Casper

Proof of Stake

Proof of Work Alternatives

- Energy inefficient
- Tons of power wasted on securing the Bitcoin network
  - Computing useless hash results

- Could we find better solutions to defend against **Sybil attacks**?
  - Proof of Work: Voting power = Computation power
  - **Proof of Stake: Voting power = Your stake in the system**

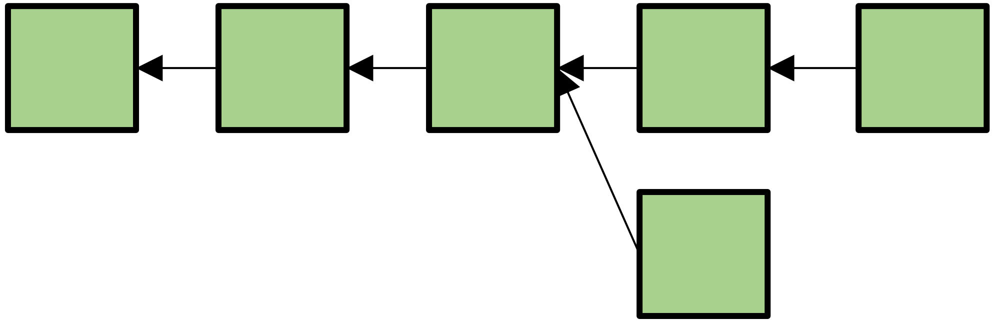
- **Proof of Stake** does not waste energy and is safe against Sybil attacks

Proof of Stake Challenges

- Economic Problem: **Rich gets richer**
  - Rich can generate more blocks and claim mining more rewards in PoS

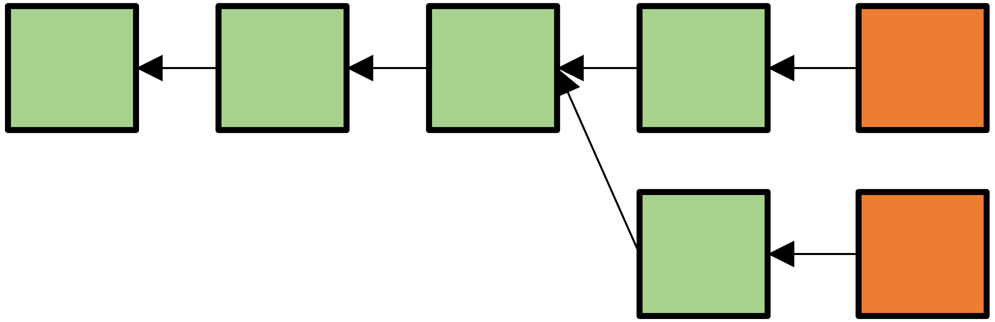
- Security Problem 1: **Nothing at stake**
  - When facing forks in the chain, nothing stops a node to work on multiple branches
- Security Problem 2: **Long range attack**
  - Malicious majority in history can create alternative chains
  - No good solution so far

Nothing At Stake



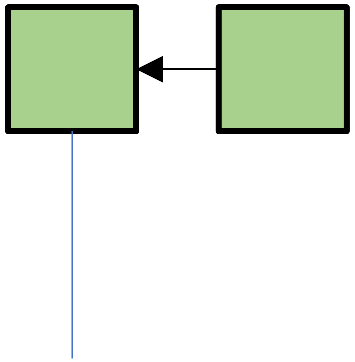
- Forks appear during the process of Nakamoto Consensus
- In PoW, a miner appends his new block to one fork to break tie
- He can only generate one block at a time
  - because each block costs computation power

Nothing At Stake



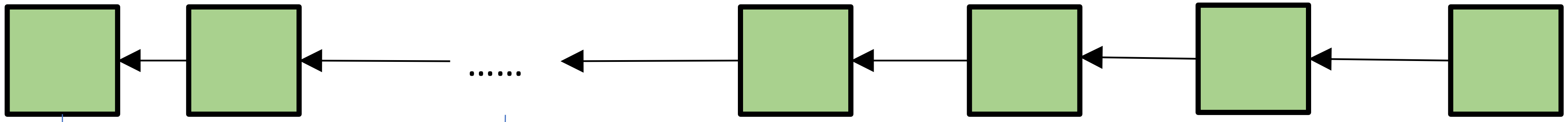
- In PoS, generating a block is cheap as long as you own stake
- Incentivized to generate multiple blocks, one for each branch
- No matter which branch wins, the miner will be able to get reward
- **Consequence:** Forks may never get resolved!
- **Solution**: Design penalties to such behavior

Long Range Attack



Suppose Alice, Bob, and Charles together own a majority of stakes of the system

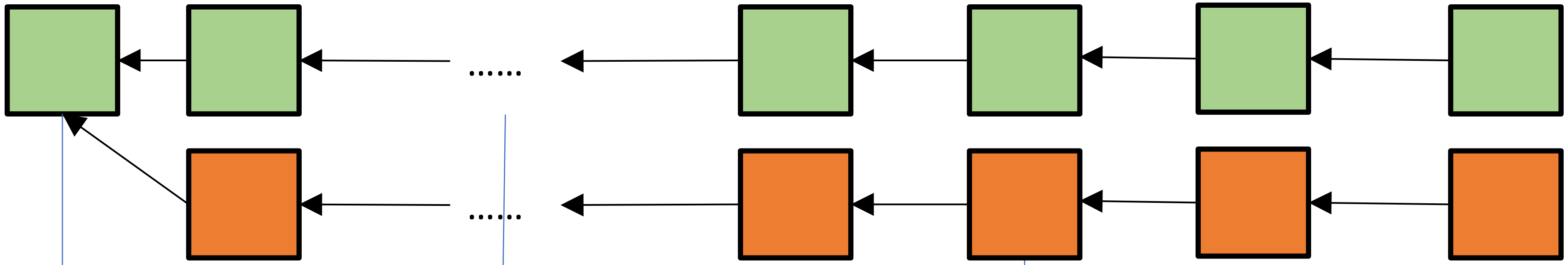
Long Range Attack



Suppose Alice, Bob, and Charles together own a majority of stakes of the system

5 years pass and they spent those stakes in some way

Long Range Attack

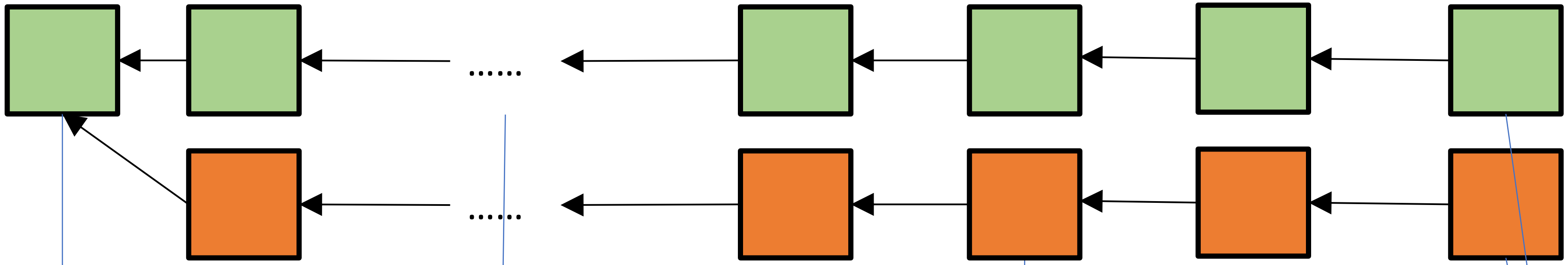


Suppose Alice, Bob, and Charles together own a majority of stakes of the system

5 years pass and they spent those stakes in some way

Now they create an alternative chain from the very beginning together. They can do this because creating the chain requires no computation cost

Long Range Attack



Suppose Alice, Bob, and Charles together own a majority of stakes of the system

5 years pass and they spent those stakes in some way

Now they create an alternative chain from the very beginning together. They can do this because creating the chain requires no computation cost

Now you are an new user joining the system. How could you know which chain is valid?

Discussion

- Many believe that Bitcoin has low throughput because of Proof-of-Work.

- A chain with Proof of Stake will always process more transactions than a chain with Proof of Work.

- Is this true or not?