Proof-of-Stake

Instructor: Nitin Awathare

October 19, 2024

Proof-of-Stake (PoS) is a consensus mechanism used in blockchain systems to achieve agreement on the state of the blockchain. Unlike Proof-of-Work (PoW), which relies on computational power to validate transactions and create new blocks, PoS selects validators based on the amount of cryptocurrency they hold and are willing to "stake" as collateral. Below are the detailed steps of how PoS works:

1. Validator Selection Stake Requirement: In a PoS system, individuals who want to participate in block validation must "stake" a certain amount of cryptocurrency (i.e., lock it up as collateral). The more cryptocurrency a participant stakes, the higher their chances of being selected as a validator. Randomized Selection Process: Validators are chosen to propose new blocks or validate transactions through a pseudo-random process, which takes into account: The size of the stake (larger stakes increase chances of selection). The length of time the stake has been locked. Potential randomness to prevent manipulation (some PoS systems add elements of randomness to avoid favoritism toward larger stakeholders).

2. Block Proposal and Validation Block Proposer: The selected validator (often called the block proposer) creates a new block of transactions and proposes it to the network. Block Propagation: The proposed block is broadcast to all other validators in the network. Validation by Other Validators: Other validators check whether the proposed block is valid. Validation includes verifying the authenticity of the transactions, ensuring there are no double spends, and confirming the validity of the proposer's stake.

3. Consensus Voting Voting on the Proposed Block: Once validators receive the proposed block, they participate in a voting process to agree on whether the block should be added to the blockchain. Weighted Voting: In PoS, the voting power of each validator is proportional to their stake. This means that validators with larger stakes have more influence over the decision. However, this is balanced by the protocol to prevent centralization risks. Finalizing the Block: If a majority of validators (often more than two-thirds) agree that the block is valid, the block is finalized and added to the blockchain.

4. Slashing and Rewards Rewards: Validators are rewarded for successfully proposing or validating blocks. The reward typically includes transaction fees and, in some cases, new coins minted in the block. Slashing Penalties: Validators that act maliciously or fail to validate properly may be penalized. This penalty is known as slashing, where a portion of their staked cryptocurrency is confiscated. Slashing can occur in cases such as: Proposing invalid or malicious blocks. Participating in multiple conflicting votes (also called double signing). Being offline when expected to validate a block (depending on the protocol).

5. Chain Finality Finality in PoS: Once a block is added to the blockchain and validators have agreed upon it, the block achieves finality—meaning that the block cannot be reversed or changed without significant cost. Checkpoints: Some PoS systems use checkpoints or epochs (groupings of blocks) to solidify finality. Validators must agree on these checkpoints, which act as reference points for the blockchain's state, ensuring that previous blocks are immutable.

6. Long-Term Participation Staking Duration: Validators must commit to staking their cryptocurrency for a fixed period, during which they cannot withdraw their funds. This ensures long-term commitment to the network and reduces the risk of short-term manipulation. Unstaking Process: Validators who wish to stop participating in the consensus process can initiate an unstaking process, which may involve a waiting period before they can access their staked funds. This waiting period reduces the likelihood of validators exiting the system immediately after engaging in malicious behavior.

7. Fork Choice Rule Chain Selection: In the event of multiple conflicting chains (or forks), validators follow a fork choice rule to determine which chain is valid. In PoS systems, the common fork choice rule is the Longest Chain Rule or the Heaviest Chain Rule, where validators prioritize the chain that has the most

accumulated stake behind it. Preventing Forks: PoS systems are generally designed to discourage forks by penalizing validators who attempt to validate conflicting chains.

8. Key Characteristics of Proof-of-Stake

- Energy Efficiency: PoS is more energy-efficient than PoW since it doesn't require extensive computational power for block validation.
- Security: The cost of an attack on the PoS system is proportional to the amount of cryptocurrency that would need to be staked by the attacker, making it costly and risky to mount a successful attack.
- Reduced Centralization Risk: While validators with larger stakes have more influence, PoS systems are often designed to mitigate centralization risks by incorporating randomness and slashing penalties.
- Incentives for Honest Behavior: Validators are incentivized to act honestly, as malicious behavior can lead to slashing, and honest behavior results in rewards.

Example of Proof-of-Stake is Ethereum 2.0. Ethereum's transition from PoW to PoS in Ethereum 2.0 (called "The Merge") illustrates how PoS works in a real-world blockchain. The summary of this transition is as follows:

- Validators must stake at least 32 ETH to participate.
- Validators are randomly chosen to propose blocks, and the voting process ensures consensus is reached.
- Validators are rewarded with transaction fees, and those who act dishonestly or go offline are penalized through slashing.

Given this summary, lets discuss the detailed working and security analysis of the PoS further.

1 Ethereum 2.0

Ethereum 2.0, also known as Eth2 or Serenity, represents a major upgrade to Ethereum, transitioning from a Proof of Work (PoW) consensus mechanism to Proof of Stake (PoS). This upgrade is aimed at improving scalability, security, and energy efficiency. The transaction flow in Ethereum 2.0 involves several new elements like sharding and the Beacon Chain. Lets first discuss the key components that Ethereum 2.0 offers before devel into the details.

Ethereum 2.0 Key Components:

- Beacon Chain: Central PoS chain that manages validators, coordinates shard chains, and maintains consensus across Ethereum 2.0.
- Shard Chains: Divided blockchains that operate in parallel to distribute the network's load and increase scalability.
- Validator Nodes: Participants that stake ETH and are chosen to propose and validate new blocks. Validators replace miners in PoS.
- Staking: Validators are required to lock up a certain amount of ETH (32 ETH minimum) to participate in the validation process.
- Eth1 (Execution Layer): Still manages smart contracts and transactions, integrated into the Eth2 framework as part of its overall architecture.

Given these key features lets talk about each in details in the further discussion. We will begin the detail exploration with the transaction flow.

Ethereum 2.0 Transaction Flow:

- Transaction Creation: Similar to Ethereum 1.0, users create transactions using wallets or decentralized applications (dApps). The user signs the transaction using their private key and submits it to the network. This transaction contains details like the sender, receiver, value, gas fee, and any smart contract data.
- Transaction Propagation: The transaction is broadcasted to the Ethereum network, where it is propagated to validator nodes responsible for validating new blocks.

- Shard Assignment: Transactions are assigned to one of the shard chains for processing. Ethereum 2.0 uses 64 shard chains running in parallel, distributing the workload across multiple chains instead of relying on a single main chain. This allows for greater scalability, as each shard can process transactions independently.
- Validator Selection for Shards: The Beacon Chain coordinates which validators are responsible for proposing and validating blocks on each shard chain. Validators are selected randomly by the Beacon Chain through a mechanism called RANDAO, ensuring security and reducing the risk of manipulation.
- Block Proposal and Validation: A chosen validator (from the pool of stakers) for a shard proposes a block that includes the user's transaction. The validator broadcasts the proposed block to the other validators assigned to that shard chain for attestation (i.e., validation).
- Consensus via Proof of Stake: Validators on the shard chain validate the proposed block. This process is much faster and more energy-efficient than the Proof of Work consensus mechanism used in Ethereum 1.0. If the block is validated, it is broadcasted to the network and added to the shard chain.
- Cross-shard Communication: Ethereum 2.0 employs a Crosslinking mechanism to synchronize data between different shard chains. This ensures that the state of the entire blockchain remains consistent across all shards. Validators responsible for the Beacon Chain take a snapshot of the shard chain's latest state and include it in the Beacon Chain.
- Finalization: The Beacon Chain periodically finalizes blocks from shard chains. Finalization ensures that once a block is added, it cannot be reversed, providing strong security guarantees. Ethereum 2.0 uses Casper FFG (Friendly Finality Gadget) to finalize blocks. Validators cast votes on blocks, and when enough votes are collected, the block is marked as final.
- Transaction Confirmation: Once the transaction is included in a shard chain block and finalized by the Beacon Chain, the transaction is considered confirmed. The state change (e.g., balance transfer, smart contract execution) is reflected in the ledger. The transaction status is then updated, and the user can view the confirmed transaction in their wallet.
- Validator Rewards and Penalties: Validators are rewarded with ETH for their participation in the validation and attestation process. Conversely, validators who act maliciously or fail to perform their duties face penalties (e.g., slashing, where a portion of their staked ETH is lost).

If you wish to visualize it diagrammatically, you can look it as follows:

- User $\rightarrow$ Wallet/Client: Initiates transaction and signs it.
- Transaction $\rightarrow$ Validator Network: Broadcast to the network of validators.
- Beacon Chain: Manages validator selection, shard assignments, and overall consensus.
- Shard Chains: Process the transaction in parallel on assigned shards.
- Finalization: Beacon Chain ensures finalized, immutable transactions across all shards.

In the above explanation we have seen that for the validator assignment in the shard, Ethereum 2.0 uses RANDAO. I would like to put more light on how RANDAO works, what are its limitation, and how it has been enhanced for Ethereum 2.0. All this we will discuss next.

2 RANDAO

RANDAO is a decentralized, verifiable randomness generation mechanism used in blockchain protocols, including Ethereum 2.0, to introduce randomness into the network securely. Its primary purpose is to generate random numbers in a distributed, unpredictable, and unbiased manner, which is critical for selecting validators and ensuring fairness in various protocol decisions.

What can we do using RANDAO In blockchain systems, randomness is crucial for many processes, such as:

- Validator Selection: Randomly choosing validators helps to prevent centralization and security vulnerabilities by ensuring that no one can predict or control who the next validator will be.
- Sharding Assignment: In Ethereum 2.0, randomness helps in assigning validators to specific shard chains, which is necessary for maintaining decentralization and security. Fairness in Protocol

- Decisions: Whether it's random committee formation, proof-of-stake validation, or block leader selection, randomness ensures that no single party can game the system.

RANDAO works by aggregating random inputs (also called commitments) from multiple participants in the network, who each submit their own cryptographically secure random numbers. The final random value is generated based on these combined inputs, ensuring that no single participant can predict or control the final result. The step-by-step execution of RANDAO is as follows:

- Commit Phase (Random Number Contribution): Each validator or participant privately chooses a random value (let's call it r_i) and commits to it by submitting a cryptographic hash of the value. The hash is published to the blockchain to indicate that the participant has committed to a specific value, without revealing what the value actually is.
- Reveal Phase: After all participants have committed their random values, each participant reveals the original random number r_i by publishing it to the blockchain. The random value provided must match the hash committed earlier, ensuring that the participant is honest and cannot alter their value after seeing others' contributions.
- Aggregation of Randomness: Once all participants reveal their random values, these values are aggregated, typically by XOR-ing them together (i.e., $r = r_1 \oplus r_2 ... \oplus r_n$), to produce the final random number r. The final result depends on the contribution of all participants, ensuring that no single individual can predict or control the outcome unless they control all contributors.

Limitations of RANDAO : While RANDAO provides an effective way to generate randomness, it is vulnerable to a certain attack known as the last-revealer problem. This occurs when the final participant can decide whether or not to reveal their random value based on previous reveals, thereby potentially biasing the final random result. To mitigate this issue, Ethereum 2.0 enhances RANDAO with additional mechanisms like verifiable delay functions (VDFs), which ensure that the final random value cannot be tampered with or influenced by any participant. VDFs ensure that once the RANDAO random value is created, it cannot be altered or tampered with. This guarantees that even the last revealer in RANDAO cannot manipulate the result. Essentially, VDFs ensure that the final random output is both unbiased and publicly verifiable.

Till now we have discussed about the working of PoS and its popular example, i.e Ethereum 2.0. However, similar to the PoW, PoS is also a probabilistic. So we will delve into the analysis of PoS next, like what we did for PoW.

3 Analysis of Proof-of-Stake (PoS) Protocol

In this analysis, we will mathematically explore some of the core concepts of PoS, focusing on validator selection, security, and rewards.

1. Validator Selection Probability: Validator Selection Probability In a PoS system, each participant stakes a certain amount of tokens, and the probability of being selected as a validator is proportional to the stake they hold relative to the total amount of tokens staked in the system.

Let:S_i be the amount of stake held by validator i, S_{total} be the total amount of tokens staked in the network, P_i be the probability that validator i is selected to propose the next block.
The probability P_i that validator i is selected is:

$P_i = \frac{S_i}{S_{total}}$

This equation shows that the more tokens a participant stakes, the higher their chance of being selected as the validator.

In PoS, validators are chosen probabilistically based on their stake. The expected number of rounds until validator i is selected to propose a block is inversely proportional to their selection probability P_i. This can be modeled using the geometric distribution, where the expected time $E(T_i)$ for validator i to be selected is:

$E(T_i) = \frac{1}{P_i} = \frac{S_{total}}{S_i}$

This means that validators with a larger stake will, on average, be selected more frequently than those with smaller stakes.

2. Security Analysis (Attacker's Stake): An important aspect of PoS security is analyzing how much stake an attacker needs to control the network. Suppose an attacker holds $S_{attacker}$ tokens, and the total staked tokens are S_{total}. The attacker's probability $P_{attacker}$ of being selected to propose a block is:

$P_{attacker} = \frac{S_{attacker}}{S_{total}}$

For an attack, the attacker might aim to control a significant fraction of block proposals. To compute the probability that an attacker controls k out of n blocks in a row (or within a certain window), we can use the binomial distribution:

$P(X = k) = \binom{n}{k}(\frac{S_{attacker}}{S_{total}})^k(1 - \frac{S_{attacker}}{S_{total}})^{n-k}$

This gives the probability that the attacker is selected k times out of n block proposals, helping us assess how much stake is required to have a meaningful probability of compromising the network.

3. Reward Distribution: Validators in PoS protocols receive rewards for creating valid blocks. The reward R_i earned by validator i in a single round is typically proportional to their stake and the total reward available R_{total} for that block. The reward R_i for validator i is given by:

$R_i = P_i * R_{total} = \frac{S_i}{S_{total}} * R_{total}$

Thus, validators with larger stakes earn larger rewards, and the reward distribution follows the proportion of the stake held.

Given this, the long-term profitability for validators depends on both their stake and the network's economic conditions. Assuming that the rewards are compounded (reinvested as stake), a validator's stake grows over time. Let $S_i(t)$ as validator i's stake at time t, r as the reward rate per round.

The evolution of $S_i(t)$ is given by:

$S_i(t+1) = S_i(t) + r * \frac{S_i(t)}{S_{total}} * R_{total}(t)$

This is a recursive equation showing how the validator's stake grows over time due to rewards. The growth rate is directly tied to their existing stake and the total reward in the system.

4. 51% Attack in PoS: Similar to PoW, in PoS, a 51% attack occurs when a validator or group of validators control more than 50% of the total stake in the network. This allows them to have a high probability of being selected to validate most blocks, leading to the possibility of double-spending or disrupting consensus. Mathematically, the condition for a 51% attack is:

$S_{attacker} > 0.5 * S_{total}$

The next prominent change in from Ethereum 2.0 is sharding, which we will discuss next.

4 Sharding in Ethereum 2.0

Sharding is one of the central upgrades in Ethereum 2.0 (Eth2) aimed at improving the network's scalability and capacity without compromising decentralization. It is part of Ethereum's roadmap to solve the limitations of the existing Ethereum network, which faces issues like high gas fees and slow transaction processing times due to its monolithic architecture. Ethereum 2.0's transition to sharding is occurring in multiple phases:

- Phase 0 (Beacon Chain): Launched in December 2020, the Beacon Chain introduced Ethereum's PoS system but did not yet include sharding. Validators began staking ETH and participating in the network, but this phase was only about setting up the PoS infrastructure.
- Phase 1 (Sharding for Data Availability): In Phase 1, Ethereum will implement 64 shards to improve the network's scalability. Initially, shards will not process transactions or smart contracts directly but will instead be used for storing and retrieving data (data availability). This design allows the network to scale more efficiently, especially when combined with Layer 2 solutions like rollups. These shards will provide a secure way for Ethereum to scale transaction throughput by offloading storage and data verification.
- Phase 1.5 (The Merge): This phase involves merging Ethereum 1.0 (the current Ethereum mainnet) with Ethereum 2.0. The existing Ethereum chain will transition from Proof of Work (PoW) to Proof of Stake (PoS) and become one of the shards in the Ethereum 2.0 ecosystem. This is a critical step that solidifies Ethereum's full transition to PoS.
- Phase 2 (Execution Sharding): In Phase 2, shards will become fully functional and able to process both transactions and smart contracts. Shards will then act like mini-blockchains that can handle decentralized applications (dApps) and process transactions in parallel. This will enable a significant increase in Ethereum's transaction capacity and improve the overall user experience.

Advantages of Sharding in Ethereum 2.0:

- Scalability: Sharding increases Ethereum's capacity by allowing it to process transactions in parallel across multiple shards. This addresses Ethereum's current bottleneck of processing only around 30 transactions

per second (TPS). With sharding, Ethereum 2.0 could theoretically handle thousands of TPS, depending on the number of shards and their design.

- Decentralization: Sharding ensures that Ethereum 2.0 remains decentralized. Validators only need to maintain the state of the shard they are assigned to, reducing the hardware requirements for running a node. This allows more individuals to participate as validators, promoting decentralization.
- Energy Efficiency: By replacing Proof of Work with Proof of Stake, sharding in Ethereum 2.0 reduces the energy consumption needed to secure the network. PoS requires significantly less computational power compared to PoW, making the network more environmentally sustainable.
- Data Scaling: Ethereum 2.0 will offload much of the data storage to shards, reducing the burden on Layer 1. This, combined with Layer 2 solutions, will allow Ethereum to scale efficiently while keeping transaction fees manageable.

Challenges and Considerations:

- Cross-Shard Communication: Implementing efficient cross-shard communication is complex. Ensuring that transactions involving multiple shards are secure, fast, and consistent is a major challenge that Ethereum developers are working to address.
- Security of Individual Shards: Protecting individual shards from attacks, such as single-shard takeovers, requires strong security mechanisms like randomized validator assignment. If a malicious group gains control over a shard, they could disrupt its operations or attempt to forge transactions.
- Coordination via Beacon Chain: The Beacon Chain is critical to the overall operation of sharded Ethereum. It must efficiently coordinate validators, maintain consensus, and synchronize shard states, which requires robust design and testing.
- Rollups and Layer 2 Compatibility: While sharding will increase the base layer's scalability, Ethereum will continue to rely on Layer 2 solutions like rollups (Optimistic and ZK-Rollups) for further scalability. These solutions will use shards as data availability layers and will process transactions off-chain, offering greater scalability when combined with Ethereum 2.0's sharding.

In summary, we began by discussing the detailed steps of Proof-of-Stake (PoS). Then, we explored PoS in the context of Ethereum 2.0. Finally, we concluded the lecture with an overview of sharding.

References:

1. Foundations of Blockchains `https://timroughgarden.github.io/fob21/`.
2. Blockchain gets better: moving beyond Bitcoin
`https://www.comp.nus.edu.sg/features/2018-blockchain-gets-better/`

Disclaimer :