Summary on Ethereum

Ethereum is an ambitious altcoin that introduces a Turing-complete programming language for writing smart contracts, allowing users to execute decentralized applications on the Ethereum Virtual Machine (EVM).

Smart Contract Programming Model

Ethereum enables the deployment of smart contracts, which are programs stored and executed on the blockchain. These contracts are written by users and triggered through transactions, operating within Ethereum's specific virtual machine.

GAS, Incentives, and Security

Ethereum incorporates a GAS mechanism to regulate computational costs, ensuring that transactions requiring excessive computation are rejected to prevent infinite loops. However, Ethereum's security is less robust compared to Bitcoin due to its complexity.

State and Account Balance

Unlike Bitcoin's UTXO model, Ethereum employs an account-based system where each account's balance is stored within the blockchain. Each block records the current state of all addresses and smart contracts.

Data Structure and State Tracking

Ethereum optimizes data retrieval by using a **Patricia Merkle Trie** instead of a traditional Merkle Tree. This allows efficient tracking of accounts and contract states. Ethereum prevents transaction replay by maintaining a nonce counter that increments with each transaction.

Ethereum's state tracking diverges from Bitcoin:

- **UTXO Model (Bitcoin):** Transactions spend unspent transaction outputs (UTXOs), creating new ones, with only unspent UTXOs stored.
- **Account-Based Model (Ethereum):** Tracks the state of accounts, allowing for more complex operations beyond token transfers.

Blockchain Storage and Organization

Ethereum's block structure is similar to Bitcoin's but includes additional complexity due to its account-based system. Each block contains:

- **Block Body:** Transaction list and uncles (stale blocks referenced for rewards).
- **Block Header:** Parent block hash, nonce (proof of work solution), and root hashes of various tries (state trie, storage trie, transactions trie, and receipts trie).

Transaction Processing and Block Size

Ethereum limits block sizes based on GAS consumption, ensuring that the sum of transaction GAS within a block does not exceed the miner-set limit. The account nonce ensures sequential transaction execution and mitigates replay attacks.

State Trie and Data Organization

Ethereum employs **Merkle Patricia Tries** to efficiently manage state changes and optimize storage operations such as search, insertion, and deletion. Multiple tries exist for different purposes:

- **State Trie:** Tracks accounts and balances.
- **Storage Trie:** Manages smart contract data.
- **Transaction Trie & Receipts Trie:** Store transaction history and outcomes.

Memory Storage and Pruning

Ethereum retains all transactions indefinitely but prunes obsolete state trie vertices after 5000 blocks. This ensures efficient data retention while limiting excessive storage requirements.

Updating the Blockchain

Blockchain updates involve complex consensus mechanisms. Ethereum's Turing-complete execution model demands higher computational power, necessitating block size restrictions based on computation costs. Ethereum requires more storage due to its complex state model.

Conflict Resolution & Uncles

Ethereum uses a modified **GHOST (Greedy Heaviest-Observed Subtree) protocol** to resolve conflicts. Unlike Bitcoin, where stale blocks are discarded, Ethereum allows miners to reference up to two stale blocks (uncles), which are rewarded to improve network security.

Ethereum's Transition to Proof of Stake (PoS)

Since 2022, Ethereum has adopted Proof of Stake (PoS) to enhance security and reduce energy consumption. Validators replace miners, requiring a **32 ETH deposit** to participate. They validate and propose blocks based on a random selection mechanism.

Proof of Stake Transaction Execution

Transaction processing in PoS involves:

1. A user submits a transaction with GAS fees.
2. The execution client verifies its validity.
3. Valid transactions are stored in mempools and broadcasted.
4. Execution clients bundle and locally execute transactions.
5. Validators confirm block validity and finalize transactions.
6. Transactions are finalized once a supermajority is reached.

Crypto-Economic Security & Fork Choice

Ethereum penalizes dishonest validators by slashing their stake if they propose multiple conflicting blocks. Fork choice is managed using **LMD-GHOST**, selecting the heaviest-weight chain.

Comparison: PoS vs. PoW

PoS provides multiple advantages over PoW:

- **Lower energy consumption**
- **Reduced centralization risk**
- **Easier validator participation**
- **Stronger protection against 51% attacks**

Future Upgrades & Scalability Improvements

Ethereum is actively working on scalability improvements, including:

- **Proto-Danksharding:** A temporary blob storage solution reducing transaction costs.
- **Danksharding:** Enables data availability sampling for efficient data verification.
- **Decentralized Rollups:** Distributes sequencers and provers across a broader network to improve decentralization.

Ethereum continues to evolve with advancements aimed at improving transaction costs, scalability, and decentralization, positioning itself as a robust platform for decentralized applications.