Blockchain

Paolo Bettelini

Contents

1 Components

1.1 Wallet

Each user owns a pair of private and public key. Transactions must be signed by the private key. Users can send their money to another address, which is given by the public key.

1.2 Block

All the transactions broadcasted to the network are grouped into blocks, which contain

- Transactions hash
- Timestamp
- difficulty (PoW)
- Nonce (PoW)
- Previous block hash
- Number of transactions

With each block being confirmed, the blockchain is created and transactions are confirmed.

1.3 Proof of Work

Proof-of-Work (PoW) is a cryptographic proof that a party has spent a certian amount of computational effort.

When a miner solves the puzzle, the current block is archived, a new block is generated and all the transactions in the previous block are confirmed. The miner is then rewarded by the system.

1.4 Proof of Stake

Proof-of-Stake (PoS) is a consensus mechanism that help choose witch participants are rewarded.

When blockchain participants verify that a transaction is legitimate and add it to the blockchain, participants have archived consensus.

1.5 Smart Contracts

Smart contracs are programs associated with an address and run on the blockchain. The nodes run code from the contract program at a relevant event, such as a received transation.

Users can interact with the contract via transactions. Contracts can often interact with other contracts and some of them are Turing-complete.

1.6 Difficulty (PoW)

If we want the time gap between two mined blocks to be approximately N units of time, we need adjust the difficulty of the mining process every M units of time such that

$$\text{difficulty}_{\text{current}} = \text{difficulty}_{\text{previous}} \cdot \frac{M}{\Delta(\frac{M}{N})}$$

where $\Delta(x)$ is the units of time to mine the last x blocks.

Or we could adjust the difficulty N blocks

$$\text{difficulty}_{\text{current}} = \text{difficulty}_{\text{previous}} \cdot \frac{N \cdot k}{\Delta(k)}$$

where k is the number of last blocks on which you want to base the adjustment on. These two formulas are the same.

1.7 Mempool

The mempool is the place where uncofirmed transactions wait to be confirmed. When a transaction is broadcasted and received by the nodes, if valid, it is put in the mempool. When a new block is created, up to X transactions are removed from the mempool and will be included in the next block.

1.8 Deployment

To deploy a transaction a node has to broadcast it to his peers. To avoid flooding of the network, a node will only broadcast the same transaction once (as long as the transaction is still in the mempool).