KTU NOTES

The learning companion.

KTU STUDY MATERIALS | SYLLABUS | LIVE NOTIFICATIONS | SOLVED QUESTION PAPERS

CST 428 BLOCK CHAIN TECHNOLOGIES

S8 CSE – ELECTIVE

MODULE – 3

Consensus Algorithms and Bitcoin

Consensus Problem

- Distributed systems are classified into two main categories, namely message passing and shared memory
- In the context of blockchain, we are concerned with the message passing type of distributed systems, where participants on the network communicate with each other via passing messages to each other

The Byzantine generals problem

- problem of reaching agreement in the presence of faults or Byzantine consensus
- In distributed systems, a common goal is to achieve consensus (agreement) among nodes on the network even in the presence of faults
- fundamental requirement in a consensus mechanism is that it must be fault-tolerant

The Byzantine generals problem

- it must be able to tolerate a number of failures in a network and should continue to work even in the presence of faults
- Based on the requirement of fault tolerance, consensus algorithms are also called fault-tolerant algorithms, and there are two types of fault-tolerant algorithms
- Crash fault-tolerance (CFT) and the other is Byzantine fault-tolerance (BFT)

CFT algorithms

- One of the most fundamental algorithms in this space is Paxos
- Paxos - developed by Leslie Lamport
  - allowing consensus over a value under unreliable communications
  - makes use of 2F + 1 processes to ensure fault tolerance in a network

Paxos

- Paxos is a two-phase protocol - prepare phase and accept phase
- Participants - proposer and acceptors
- proposer is the replicas or nodes that propose the values and acceptors are the nodes that accept the value
- protocol assumes an asynchronous message-passing network with less than 50% of crash faults

Paxos

- the critical properties of the Paxos consensus algorithm are safety and liveness
- Under safety, we have:
  - Agreement, which specifies that no two different values are agreed on
  - Validity, which means that only the proposed values are decided. In other words, the values chosen or learned must have been proposed by a processor

Paxos

- Under liveness, we have:
  - Termination, which means that, eventually, the protocol is able to decide and terminate. In other words, if a value has been chosen, then eventually learners will learn it
- Processes can assume different roles
  - Proposers, elected leader(s) that can propose a new value to be decided

Paxos

- Acceptors, which participate in the protocol as a means to provide a majority decision
- Learners, which are nodes that just observe the decision process and value

Paxos

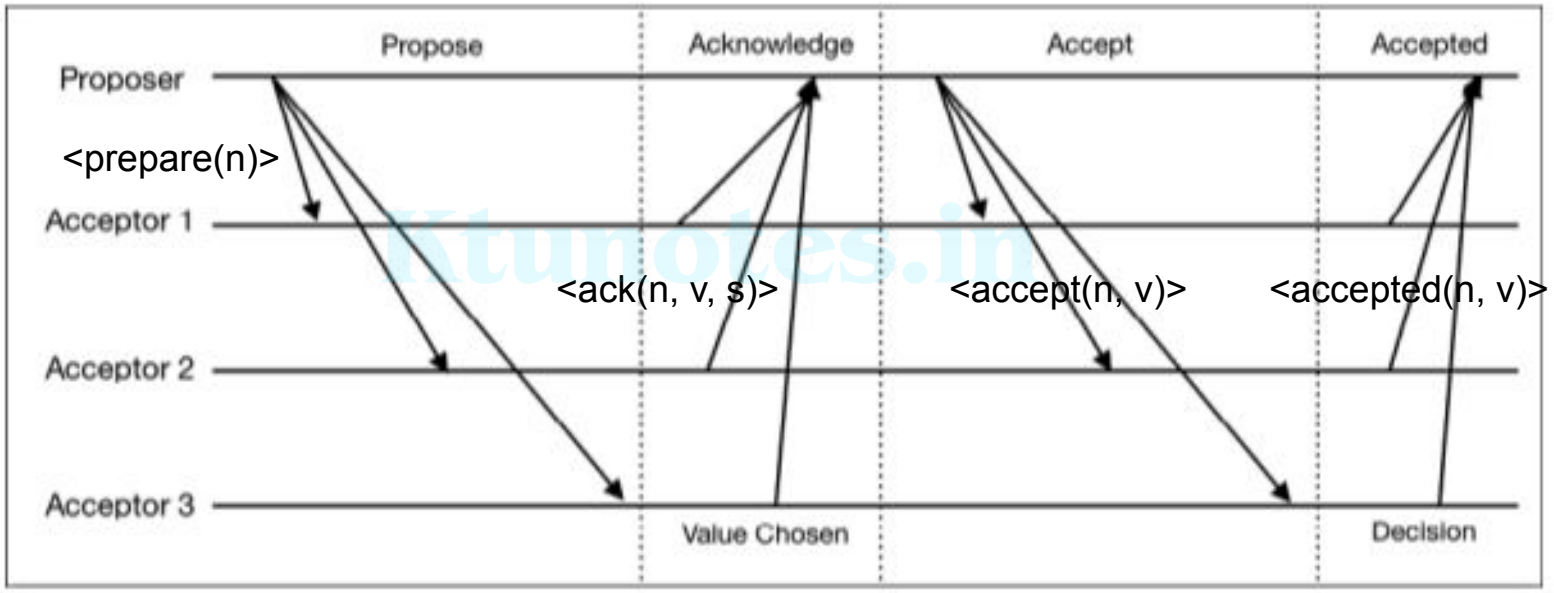


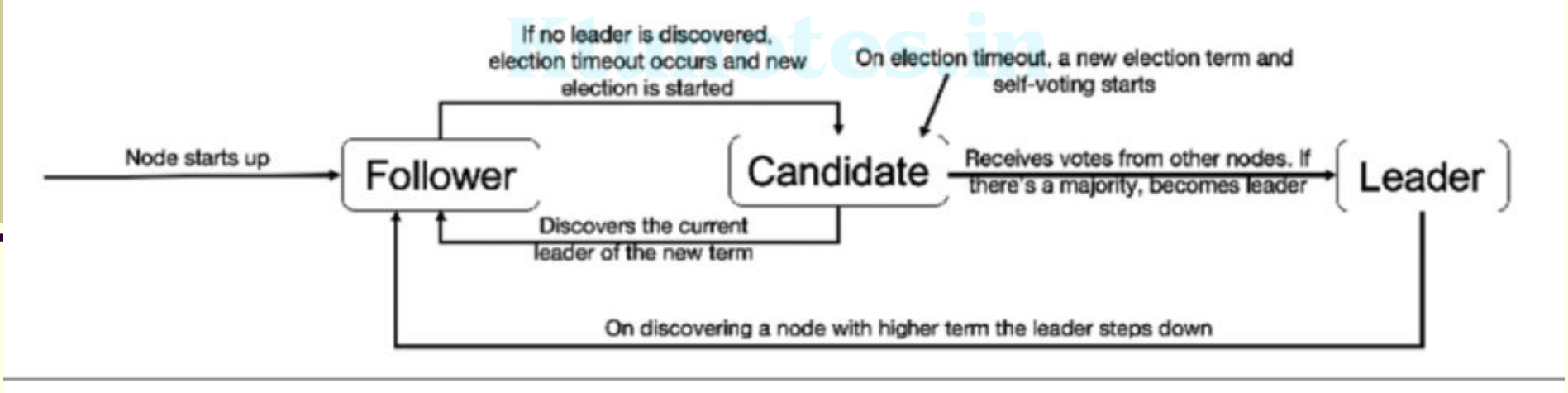
Figure 5.1: How Paxos works

RAFT

- developed by Diego Ongaro and John Ousterhout at Stanford University
- composed of three sub-problems:
  - Leader election (a new leader election in case the existing one fails)
  - Log replication (leader to follower log synch)
  - Safety (no conflicting log entries (index) between servers)

RAFT

- Each server in Raft can have either a follower, leader, or candidate state



Log replication

Log replication logic can be visualized in the following diagram. The aim of log replication is to synchronize nodes with each other.

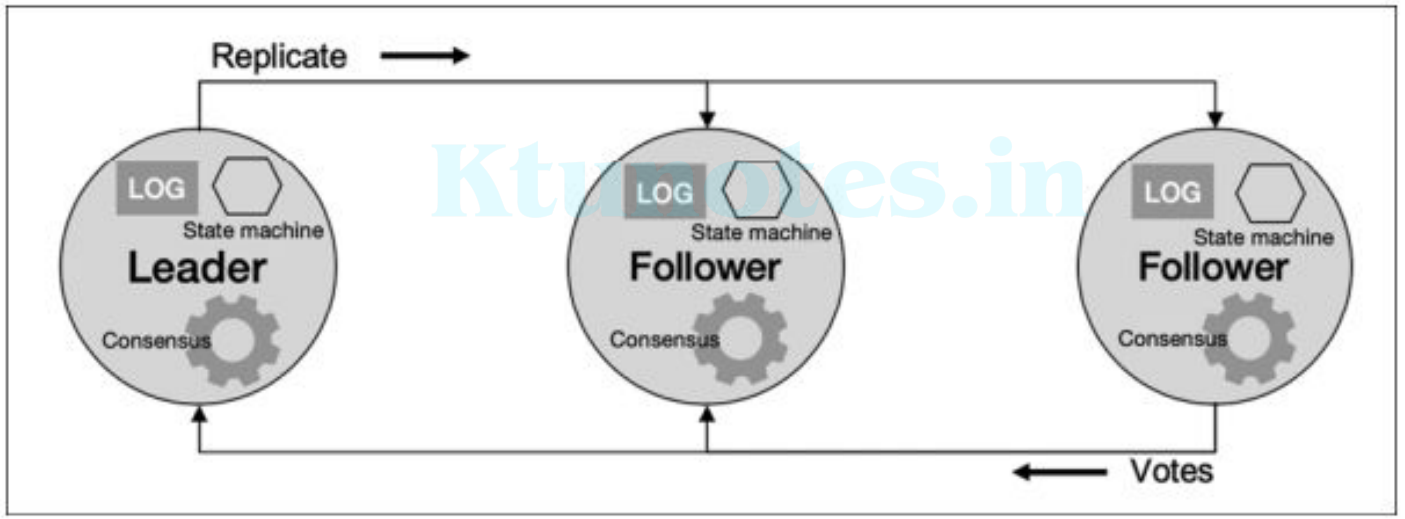


Figure 5.3: Log replication mechanism

RAFT

- the log (data) is eventually replicated across all nodes
- leader is responsible for log replication
- Once the leader has a new entry in its log, it sends out the requests to replicate to the follower nodes

RAFT

- When the leader receives enough confirmation votes back from the follower nodes indicating that the replicate request has been accepted and processed by the followers, the leader commits that entry to its local state machine
- At this stage, the entry is considered committed

Byzantine fault- tolerance (BFT) algorithms

1. Practical Byzantine Fault Tolerance (PBFT)

- developed in 1999 by Miguel Castro and Barbara Liskov
- to provide consensus in the presence of Byzantine faults
- comprises three sub-protocols called normal operation, view change, and checkpointing
  - Normal operation is executed when everything is running normally and no errors are in the system
  - View change is runs when a faulty leader node is detected in the system
  - Checkpointing is used to discard the old data from the system

1. Practical Byzantine Fault Tolerance (PBFT)

- comprises three phases or steps
- pre-prepare, prepare, and commit
- The protocol runs in rounds
- in each round, an elected leader node, called the primary node, handles the communication with the client
- the protocol progresses through the three previously mentioned phases

1. Practical Byzantine Fault Tolerance (PBFT)

- participants in the PBFT protocol are called replicas
- one of the replicas becomes primary as a leader in each round, and the rest of the nodes acts as backups
- in order to tolerate Byzantine faults, the minimum number of nodes required is N = 3F + 1, where N is the number of nodes
- and F is the number of faulty nodes
- PBFT ensures Byzantine fault tolerance as long as the number of nodes in a system stays N >= 3F + 1

1. Practical Byzantine Fault Tolerance (PBFT)

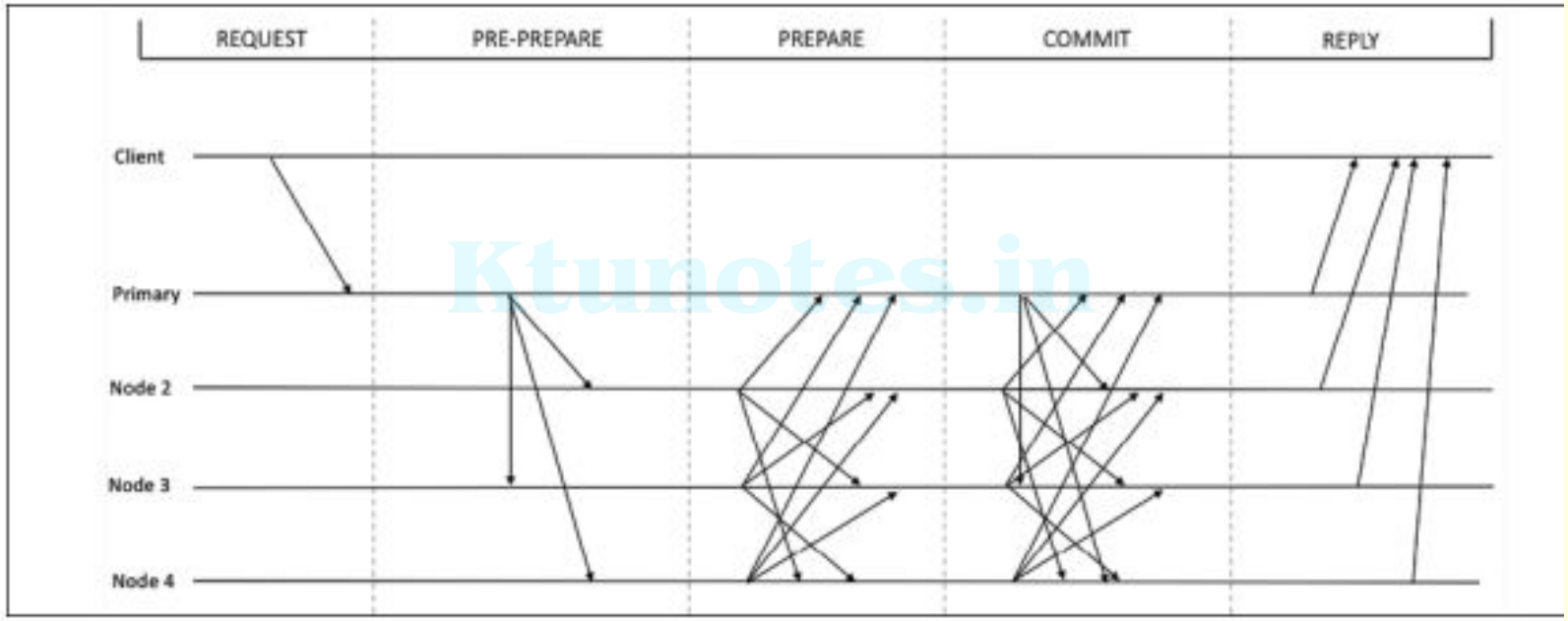


Figure 5.4: PBFT protocol

1. Practical Byzantine Fault Tolerance (PBFT)

- The algorithm for the view-change protocol is shown as follows:

  1. Stop accepting pre-prepare, prepare, and commit messages for the current view

  2. Create a set of all the certificates prepared so far

  3. Broadcast a view-change message with the next view number and a set of all the prepared certificates to all replicas

the collection of 2F + 1 messages of a particular type is a certificate

1. Practical Byzantine Fault Tolerance (PBFT)

- view-change protocol

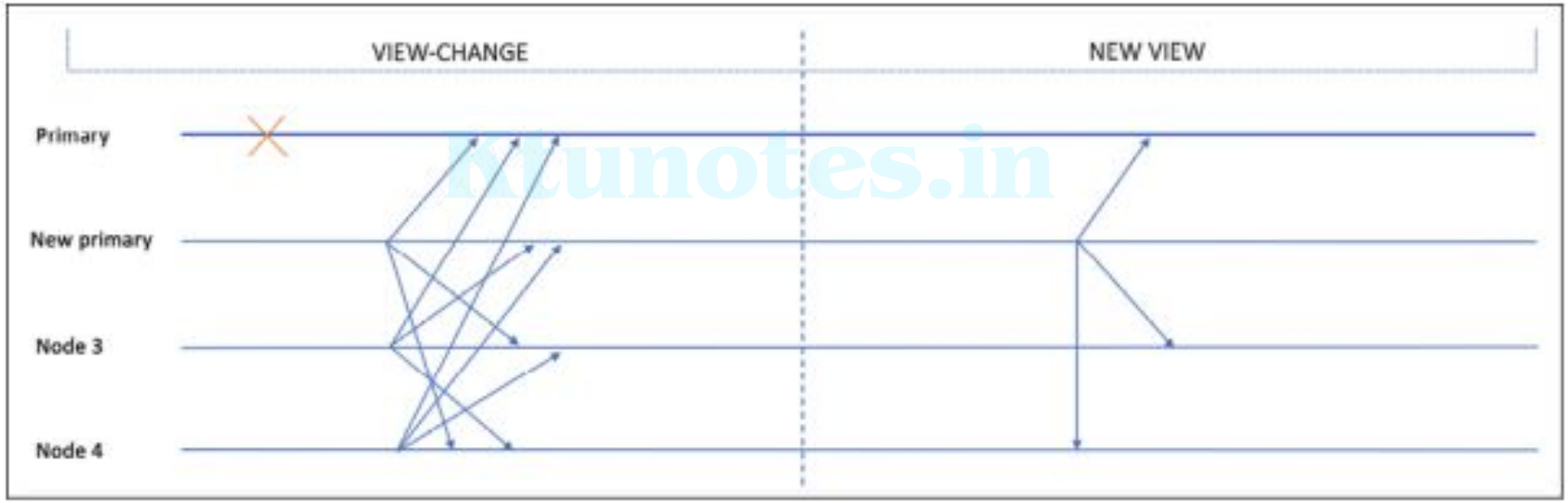


Figure 5.5: View-change sub-protocol

1. Practical Byzantine Fault Tolerance (PBFT)

- Check Pointing
  - to discard old messages in the log of all replicas
  - replicas agree on a stable checkpoint that provides a snapshot of the global state at a certain point in time
  - variable called low watermark is used to record the sequence number of the last stable checkpoint

1. Practical Byzantine Fault Tolerance (PBFT)

- Check Pointing
  - This checkpoint is then broadcast to other nodes
  - As soon as a replica has at least 2F+1 checkpoint messages, it saves these messages as proof of a stable checkpoint
  - It discards all previous pre-prepare, prepare, and commit messages from its logs

1. Practical Byzantine Fault Tolerance (PBFT)

- Advantages
  - provides immediate and deterministic transaction finality
  - energy efficient as compared to PoW
- Disadvantages
  - not very scalable - more suitable for consortium networks, instead of public blockchains
  - faster than PoW protocols
  - Sybil attacks can be carried out

Nakamoto consensus or PoW

- first introduced with Bitcoin in 2009
- Longest-running blockchain network
- the PoW mechanism is designed to mitigate Sybil attacks, which facilitates consensus and the security of the network

A Sybil attack is a type of attack that aims to gain a majority influence on the network to control the network. Once a network is under the control of an adversary, any malicious activity could occur. A Sybil attack is usually conducted by a node generating and using multiple identities on the network. If there are enough multiple identities held by an entity, then that entity can influence the network by skewing majority-based network decisions. The majority in this case is held by the adversary.

Nakamoto consensus or PoW

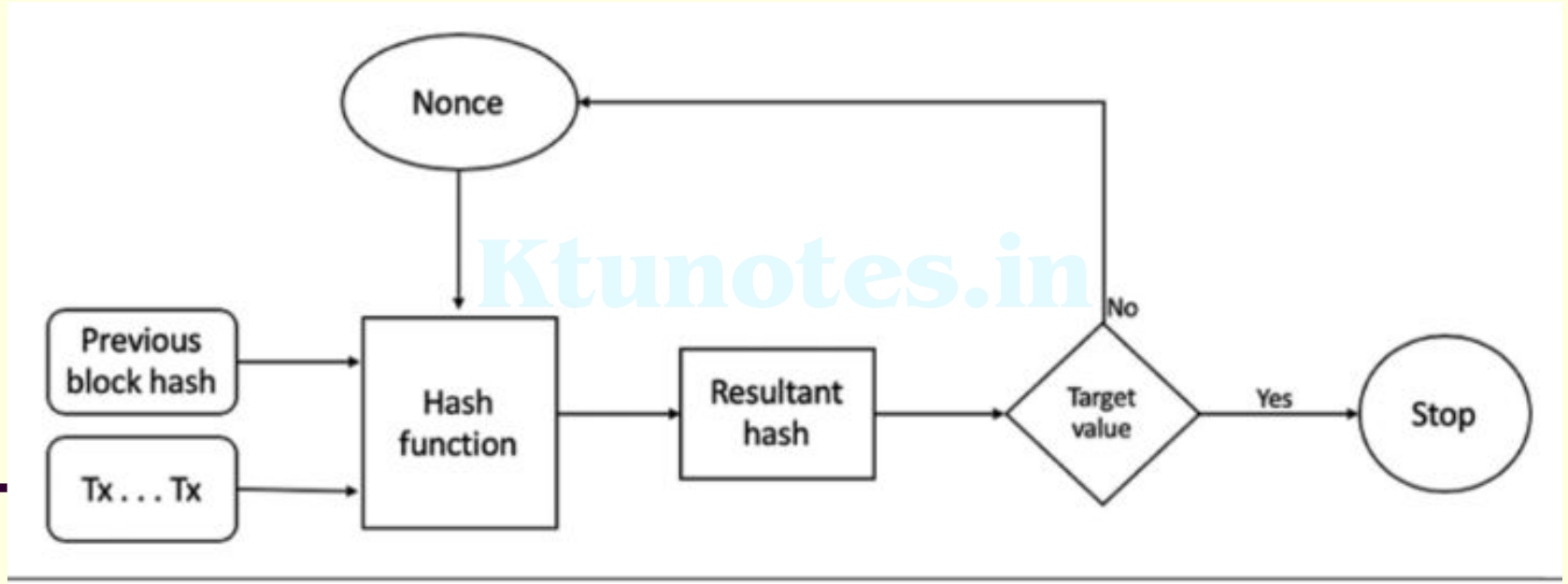


Figure 5.10: PoW diagram

PoW makes use of hash puzzles.

Nakamoto consensus or PoW

- New transactions are broadcast to all nodes on the network.
- Each node collects the transactions into a candidate block.
- Miners propose new blocks.
- Miners concatenate and hash with the header of the previous block.
- The resultant hash is checked against the target value, that is, the network difficulty target value.
- If the resultant hash is less than the threshold value, then PoW is solved, otherwise, the nonce is incremented and the node tries again. This process continues until a resultant hash is found that is less than the threshold value.

Nakamoto consensus or PoW

- key idea behind PoW as a solution to the Byzantine generals problem is that all honest generals (miners in the Bitcoin world) achieve agreement on the same state (decision value)
- As long as honest participants control the majority of the network, PoW solves the Byzantine generals problem

Nakamoto consensus or PoW

- two main variants of PoW algorithms, based on the type of hardware used for processing
  - CPU-bound PoW - processing required to find the solution to the cryptographic hash puzzle is directly proportional to the calculation speed of the CPU or hardware such as ASICs

Nakamoto consensus or PoW

- Memory-bound PoW - performance is bound by the access speed of the memory or the size of the memory

Proof of stake (PoS)

- energy-efficient alternative to the PoW
- first used in Peercoin, and now, prominent cryptocurrency blockchains such as EOS, NxT, Steem, and Tezos are using PoS algorithms
- stake represents the number of coins (money) in the consensus protocol staked by a blockchain participant

Proof of stake (PoS)

- key idea is that if someone has a stake in the system, then they will not try to sabotage the system
- the chance of proposing the next block is directly proportional to the value staked by the participant
- Variations of PoS - chain-based PoS, committee-based PoS, BFT-based PoS, delegated PoS, leased PoS, and master node PoS

Proof of stake (PoS)

- PoS miner is called either a validator, minter, or stakeholder
- right to win the next proposer role is usually assigned randomly
- Proposers are rewarded either with transaction fees or block rewards
- control over the majority of the network in the form of the control of a large portion of the stake is required to attack and control the network

Proof of stake (PoS)

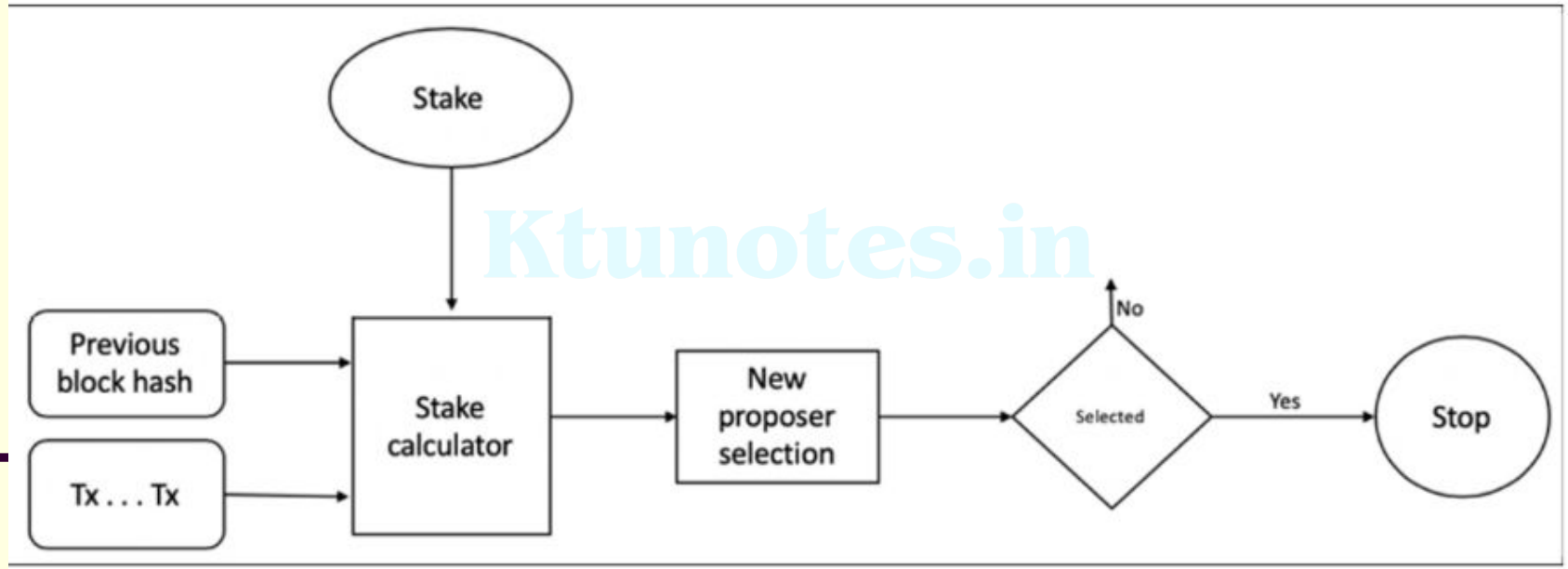


Figure 5.11: PoS diagram

Proof of stake (PoS) types

1	Chain-based PoS	• very similar to PoW • only change is the block generation method • Transactions are picked up from the memory pool and a candidate block is created • Hash (B || clock time) < target x stake value • hashing puzzle in PoS is solved at regular intervals based on the clock tick
2	Committee-based PoS	• group of stakeholders is chosen randomly, using a verifiable random function (VRF) • VRF produces a random set of stakeholders based on their stake and the current state of the blockchain • The chosen group of stakeholders becomes responsible for proposing blocks in sequential order

Proof of stake (PoS) types

3	Delegated PoS	• Instead of using a random function to derive the group of stakeholders, the group is chosen by stake delegation • group selected is a fixed number of minters that create blocks in a round-robin fashion • Delegates are chosen via voting by network users • Votes are proportional to the amount of the stake that participants have on the network • used in Lisk, Cosmos, and EOS

Bitcoin definition

- defined in various ways; it's a protocol, a digital currency, and a platform
- It is a combination of a peer-to-peer network, protocols, and software that facilitates the creation and usage of the digital currency
- Nodes in this peer-to-peer network talk to each other using the Bitcoin protocol

Bitcoin definition

- The double-spending problem arises when, a user sends coins to two different users at the same time and they are verified independently as valid transactions
- resolved in Bitcoin by using a distributed ledger (the blockchain) where every transaction is recorded permanently, and by implementing a transaction validation and confirmation mechanism

Bitcoin Payment

Payment request is initiated through mail or SMS with address

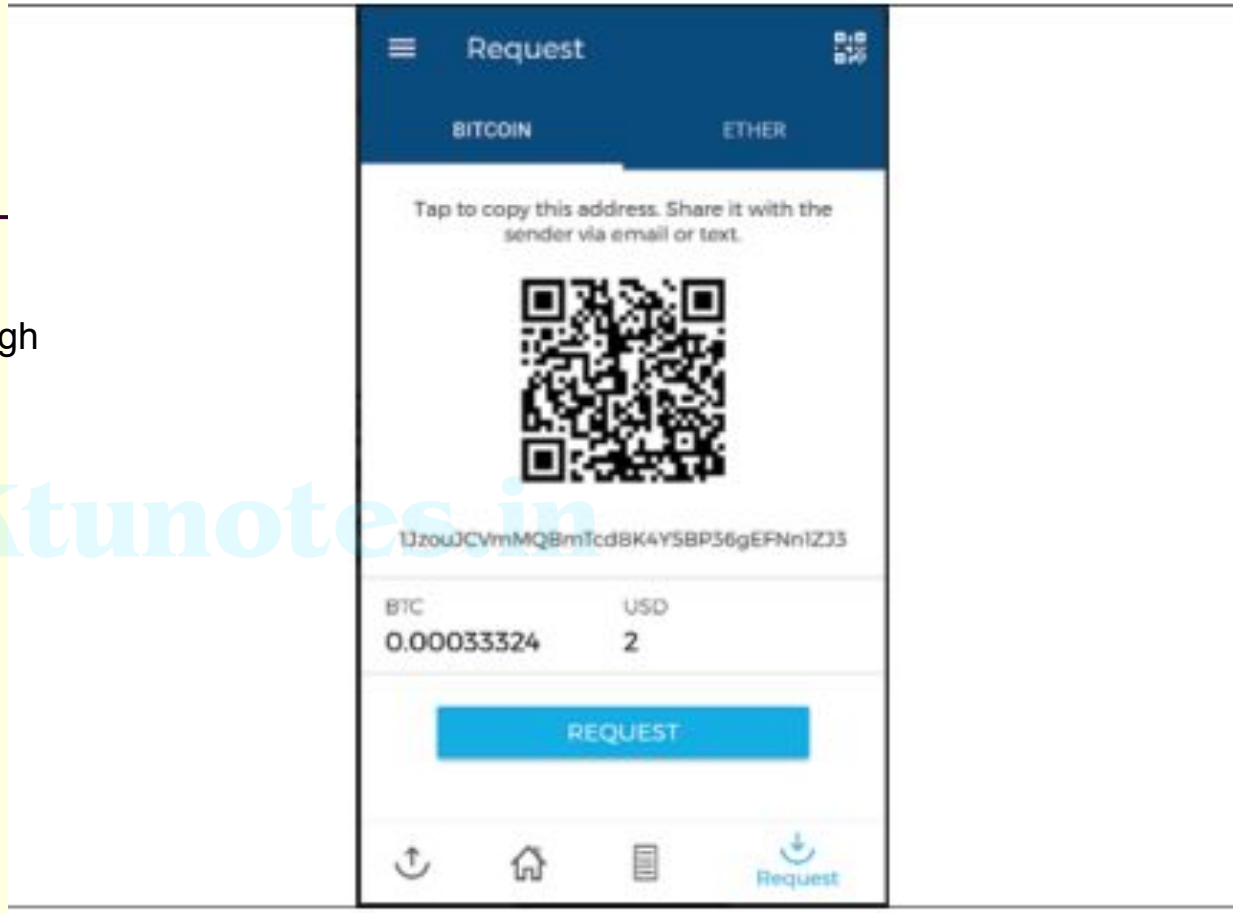


Figure 6.2: Bitcoin payment request (using the Blockchain wallet)

Bitcoin Payment

2. The sender either enters the receiver's address or scans the QR code that has the Bitcoin address, amount, and an optional description encoded in it. The wallet application recognizes this QR code and decodes it into something like:

```
"Please send <amount> BTC to address <receiver's Bitcoin ad
```

3. With actual values, this will look like the following:

```
"Please send 0.00033324 BTC to address 1JzouJCVmMQBmTcd8K4Y
```

4. This is also shown in the following screenshot:

Figure 6.3: Bitcoin payment QR code

Bitcoin Payment

- fee is calculated based on size of the transaction and a fee rate, which is a value that depends on the volume of the transactions in the network at that time
- This is measured in Satoshis per byte
- Bitcoin network fees ensure that your transaction will be included by miners in the block

Figure 6.4: Sending BTC using the Blockchain wallet

Bitcoin Payment

- transaction has been constructed, signed, and sent out to the Bitcoin network
- This transaction will be picked up by miners to be verified and included in the block
- confirmation is pending for this transaction
- These confirmations will start to appear as soon as the transaction is verified, included in the block, and mined
- the appropriate fee will be deducted from the original value to be transferred and will be paid to the miner who has included it in the block for mining

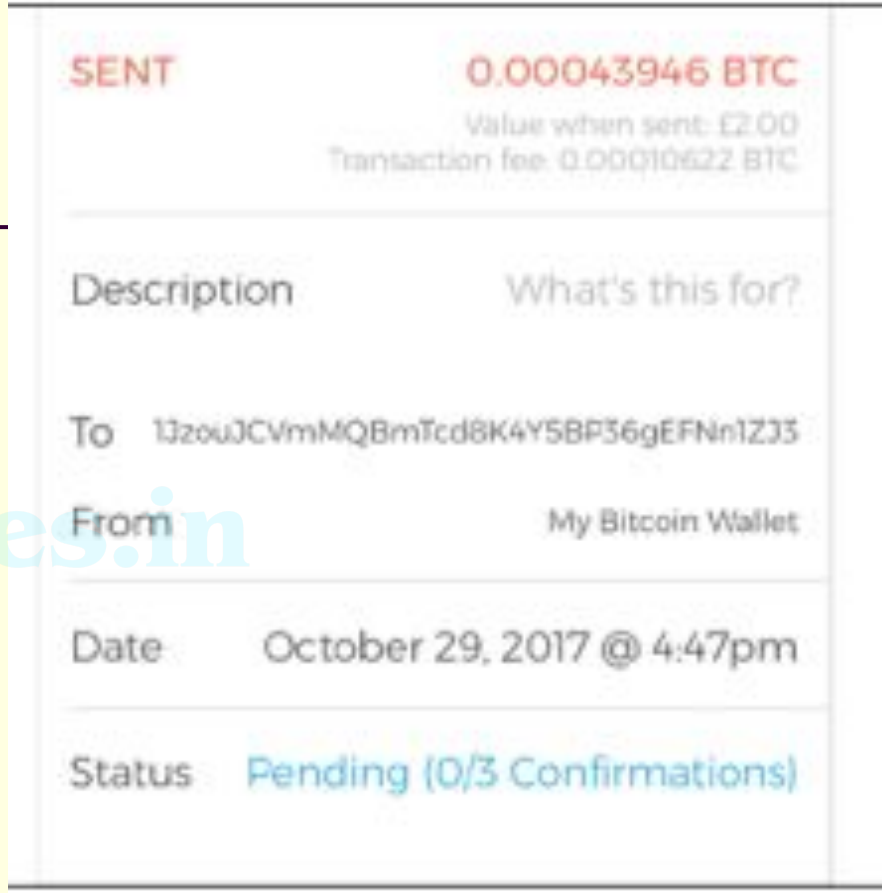


Figure 6.5: Transaction sent

Bitcoin Payment

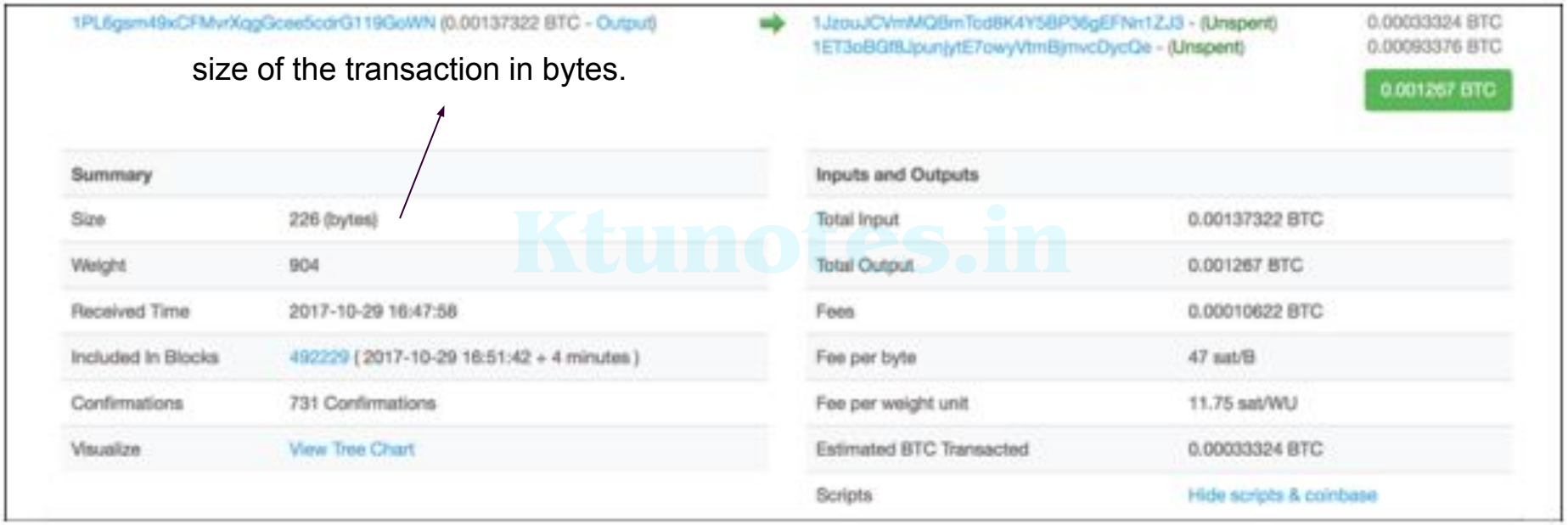


Figure 6.7: Snapshot of the transaction taken from blockchain.info

Bitcoin Payment

In summary, a payment transaction in the Bitcoin network can be divided into the following steps:

1. The transaction starts with a sender signing the transaction with their private key.
2. The transaction is serialized so that it can be transmitted over the network.
3. The transaction is broadcast to the network.
4. Miners listening for transactions pick up the transaction.
5. The transaction is verified for its legitimacy by the miners.
6. The transaction is added to the candidate/proposed block for mining.
7. Once mined, the result is broadcast to all nodes on the Bitcoin network.
8. Usually, at this point, users wait for up to six confirmations to be received before a transaction is considered final; however, a transaction can be considered final at the previous step. Confirmations serve as an additional mechanism to ensure that there is probabilistically a very low chance for a transaction to be reverted, but otherwise, once a mined block is finalized and announced, the transactions within that block are final at that point.

Bitcoin Payment

DENOMINATION	ABBREVIATION	FAMILIAR NAME	VALUE IN BTC
Satoshi	SAT	Satoshi	0.00000001 BTC
Microbit	µBTC (uBTC)	Microbitcoin or Bit	0.000001 BTC
Millibit	mBTC	Millibitcoin	0.001 BTC
Centibit	cBTC	Centibitcoin	0.01 BTC
Decibit	dBTC	Decibitcoin	0.1 BTC
Bitcoin	BTC	Bitcoin	1 BTC
DecaBit	daBTC	Decabitcoin	10 BTC
Hectobit	hBTC	Hectobitcoin	100 BTC
Kilobit	kBTC	Kilobitcoin	1000 BTC
Megabit	MBTC	Megabitcoin	1000000 BTC

Figure 6.8: Bitcoin denominations