DISTRIBUTED CONSENSUS PROTOCOLS

Dr. Shyama Prasad Mukherjee International Institute of Information Technology, Naya Raipur

Formal Definition

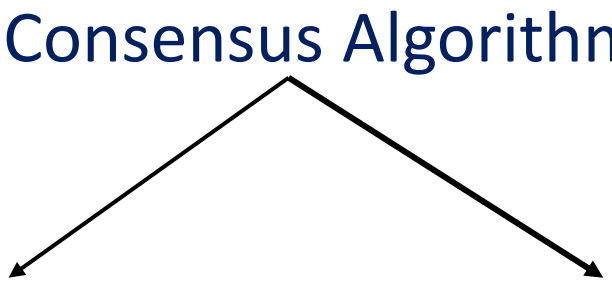
There are “n” nodes, each have an input value. Some nodes are faulty or malicious. A distributed consensus protocol[1ℓ] has the following two properties:

- ☞ The protocol terminates and all honest nodes are in agreement on the same value.

- ☞ This value must have been proposed by some honest node.

1ℓ : In the context of blockchain, consensus is the valid agreement for adding the new blocks in the blockchain network.

Division of Consensus Algorithm



Probabilistic Finality Algorithm (Chain-based)
E.g. Proof-of-work and its variants

Absolute Finality Algorithm (exact Finality)
E.g. Practical byzantine fault tolerance

Proof-of-work (PoW) Consensus

- Proof-of-work (PoW) is a combination of cryptography and computational power which ensure consensus and authenticity of the data recorded in the blockchain framework.
- The core idea of PoW is a solution that difficult to find but, very easy to verify.
- Bitcoin network use (PoW) as consensus protocol in implementation of the bitcoin network in 2009 by Satoshi Nakamoto.
- All participants of the blockchain network keep on calculating hash values using different nonce every time until, the target is achieved 2ℓ.
- When a peer is successful in computing the required hash value, all other participants must mutually agree on the correctness of the hash value.

2ℓ : prefix of current hash value is equivalent or lesser than the specific target value.

Cont…

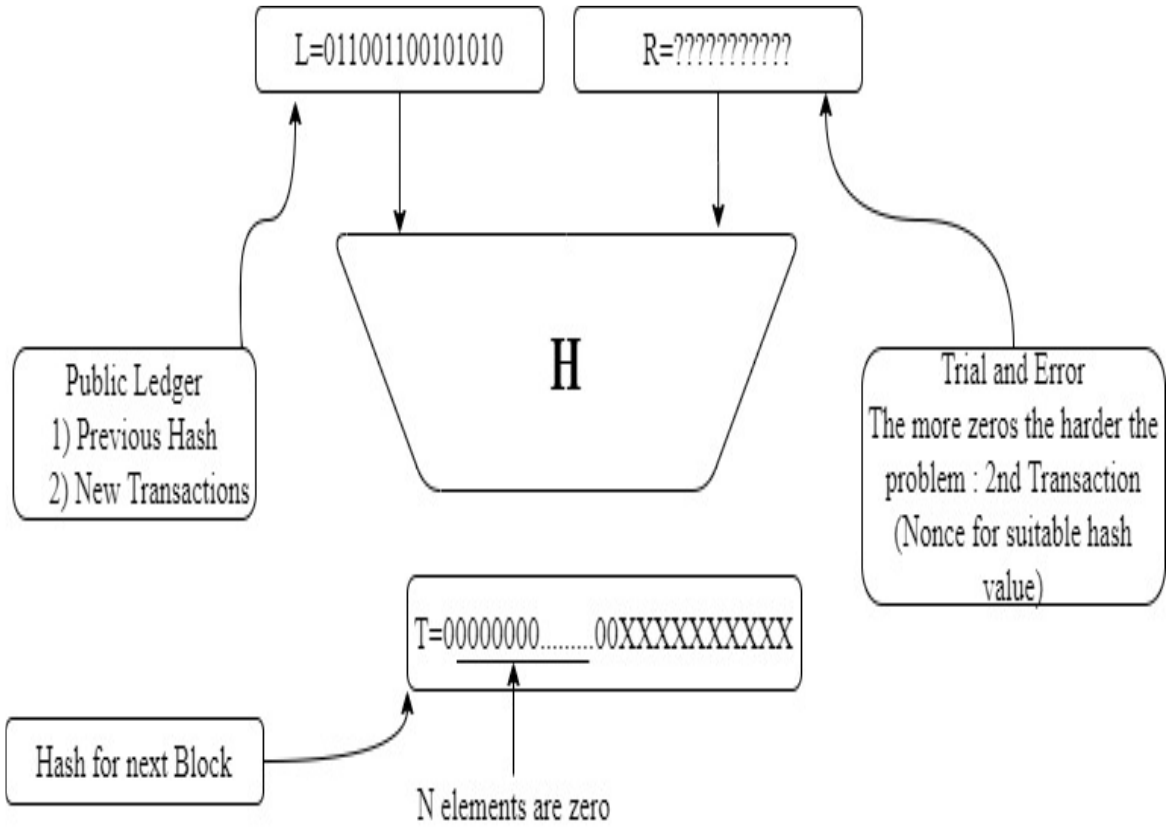


Fig. PoW Puzzle 3ℓ

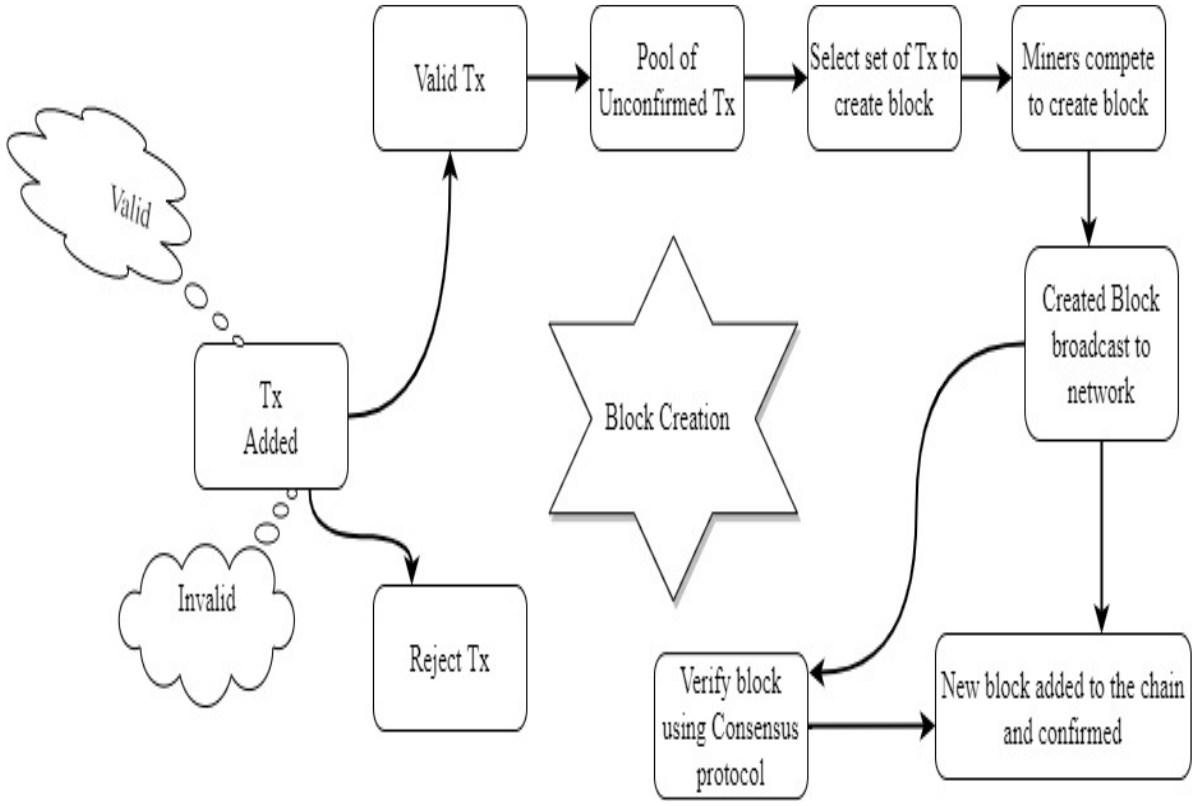


Fig. PoW Working Flow

3ℓ : $H(nonce \mid\mid prev\_hash \mid\mid t_x \mid\mid t_x \mid\mid \ldots\ldots t_x) < target.$

Cont…

- The collection of transactions used for the calculation of hash is considered as authenticated transactions, the nodes that compete to mine the blocks are called miners, and the PoW process is known as the mining algorithm.
- Calculation of the hash is a time-consuming process. Therefore to motivate the miners, an incentive mechanism is proposed.
- There is a possibility that two competing nodes may compute the hash and create a new block at the same instant. However, it is impossible that two contending forks will produce the next block at the same time. In such a case, the Longest chain becomes an authentic one.
- Example: Bitcoin.

Cont…

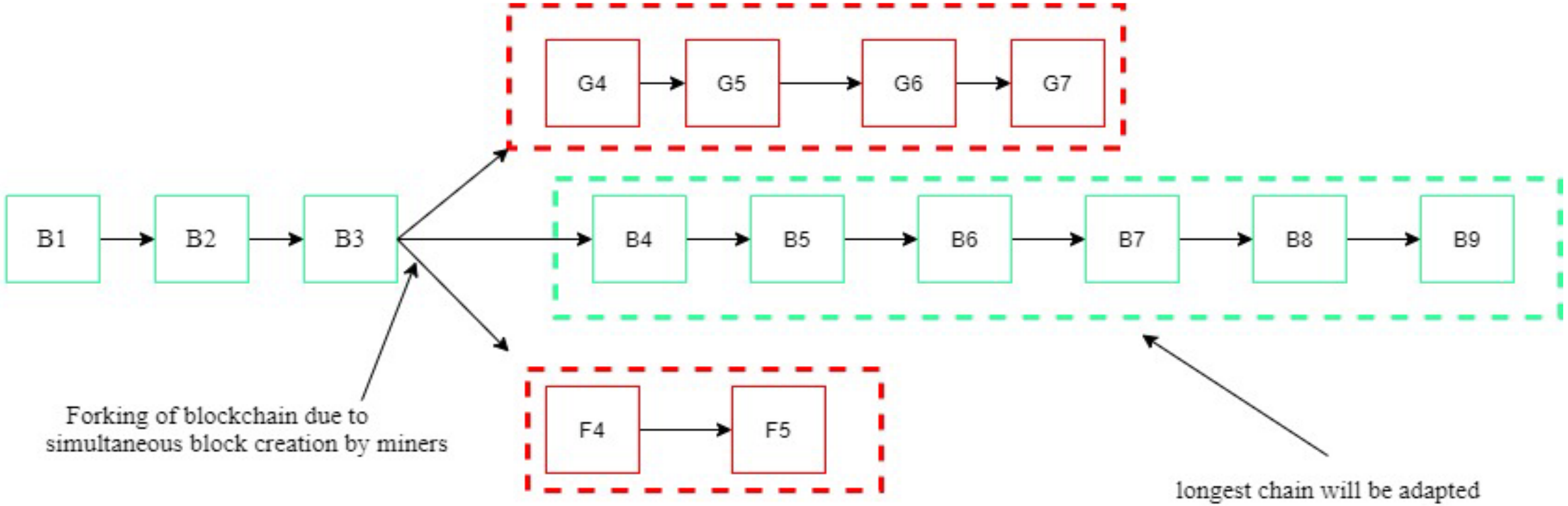


Fig. Blockchain Forking[4ℓ]

4ℓ : In Bitcoin blockchain, when around six blocks are axed, the applicable blockchain is seen as unchanging and credible, and each block is generated every 10 minutes.

PoW simplified:

- ☞ New transactions are broadcast to all nodes.
- ☞ Each node collects new transactions and verify this transaction (unspent and valid signature) .
- ☞ A random node collects the transaction (from pool of unconfirmed but valid transaction) and create a new block (solve hash puzzle).
- ☞ A random node broadacst this new block to all the peers nodes.
- ☞ Nodes express the acceptance of the block (verify the hash using nonce) by including its hash in the next block.

PoW : Advantage & Disadvantage

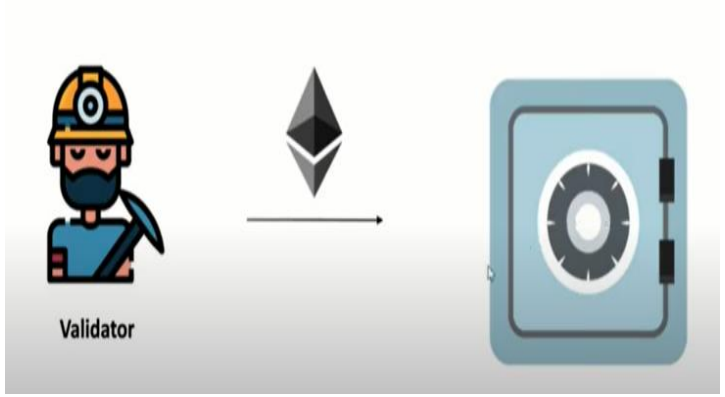
Advantages:

- High level of security.
- Provides a decentralized method of verifying transactions.
- Allows miners to earn crypto rewards.

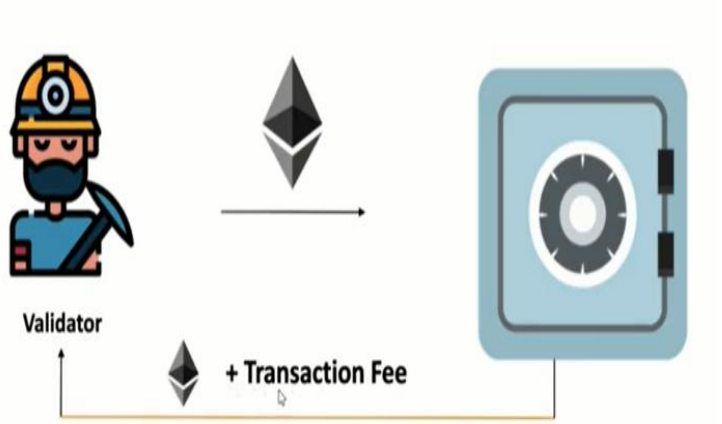
Disadvantages

- Inefficient with slow transaction speeds and expensive fees.
- High energy usage.
- Mining often requires expensive equipment

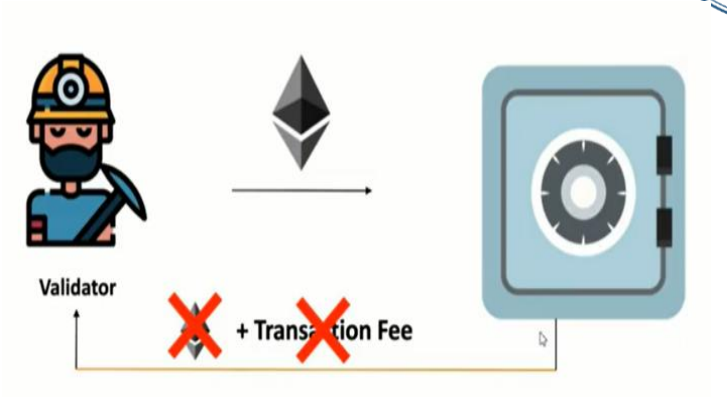
Proof-of-stake (PoS) Consensus



- To join as validator, the node has to provide ETH (min. 32 ETH) to the system.
- The system now given permission that node to act as validator node which can now validate the block and can add the block to the network.

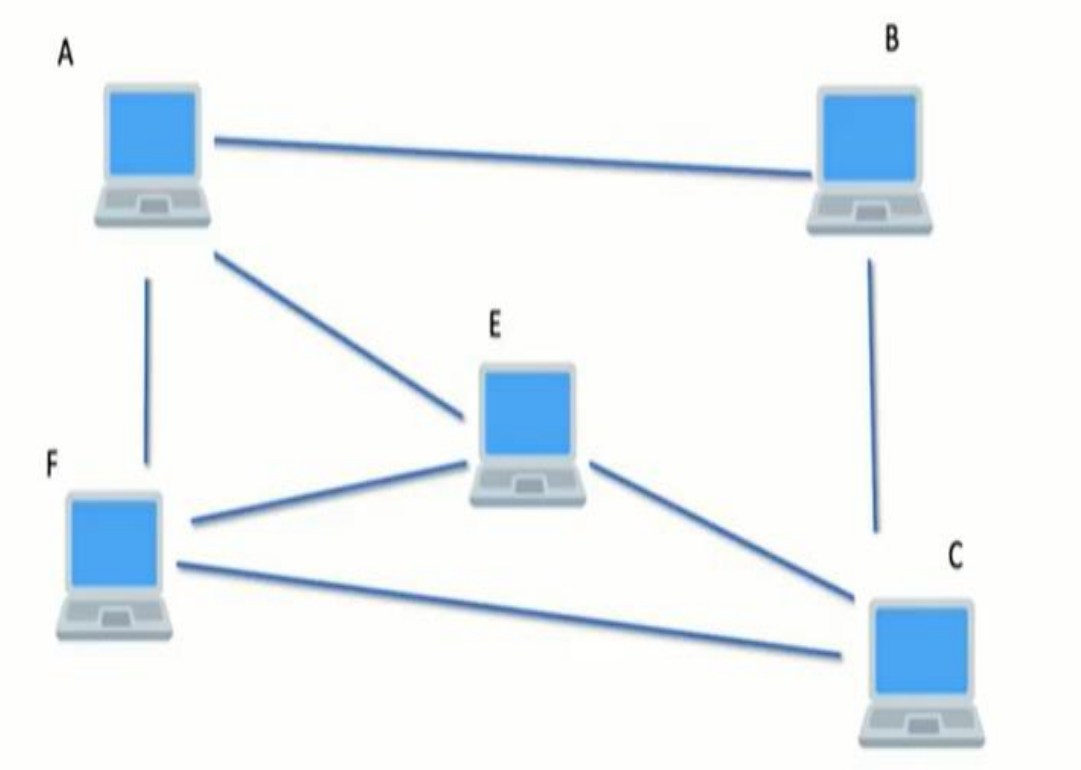


- If the validator node is performing well, the system will return the stake amount as well as transaction fees.



- If the validator node is creating some malicious activities, the system will not return the stake amount as well as transaction fees.

Proof-of-stake (PoS) Consensus



How the system will choose validator node?

Criteria:

1. The more ether you pay the more chances of getting randomly selected you have
2. Random based selection

Proof-of-stake (PoS) Consensus

- PoW requires huge amount of energy (in terms of computational power), and has left the researchers to think for alternative of PoW to attain consensus in the blockchain network.
- Proof-of-stake (PoS) may be one of candidate to solve the energy requirement problem in the blockchain network.
- In PoS protocol, ownership of currency allows peer to participate in mining process (to validate the transaction and generate new blocks).
- In PoW, random node (no leader selection method is present) is create a new block. Whereas, in PoS, leader is selected based on the amount of stake which the miner currently holds proportion to the network capacity.

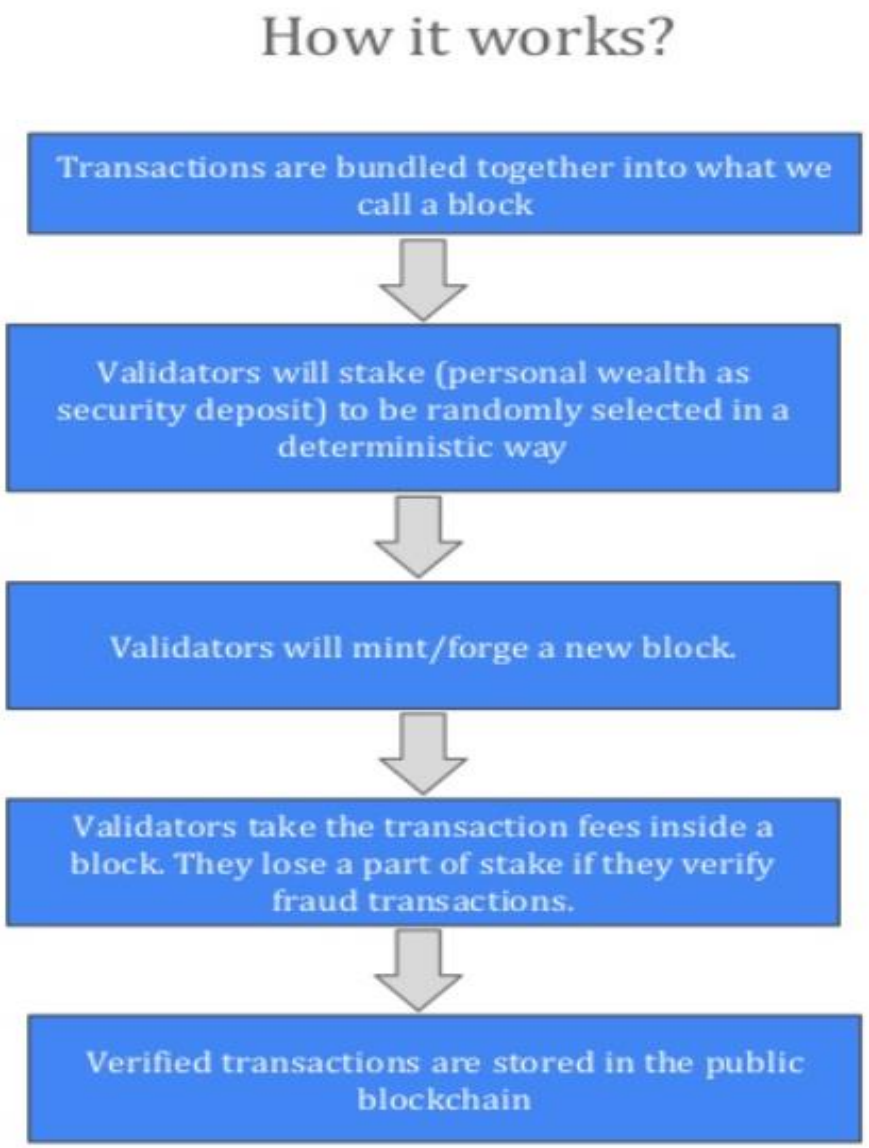


Fig. Working of PoS consensus

Points to be remember

- ☞ Creator of new block is chosen in a deterministic way depending on its wealth (stake).
- ☞ No block reward, so the validator takes the transaction fees.
- ☞ If a node stop being a validator, validator stake plus transaction fees will be released after a certain period of time.
- ☞ Example: Peercoin.

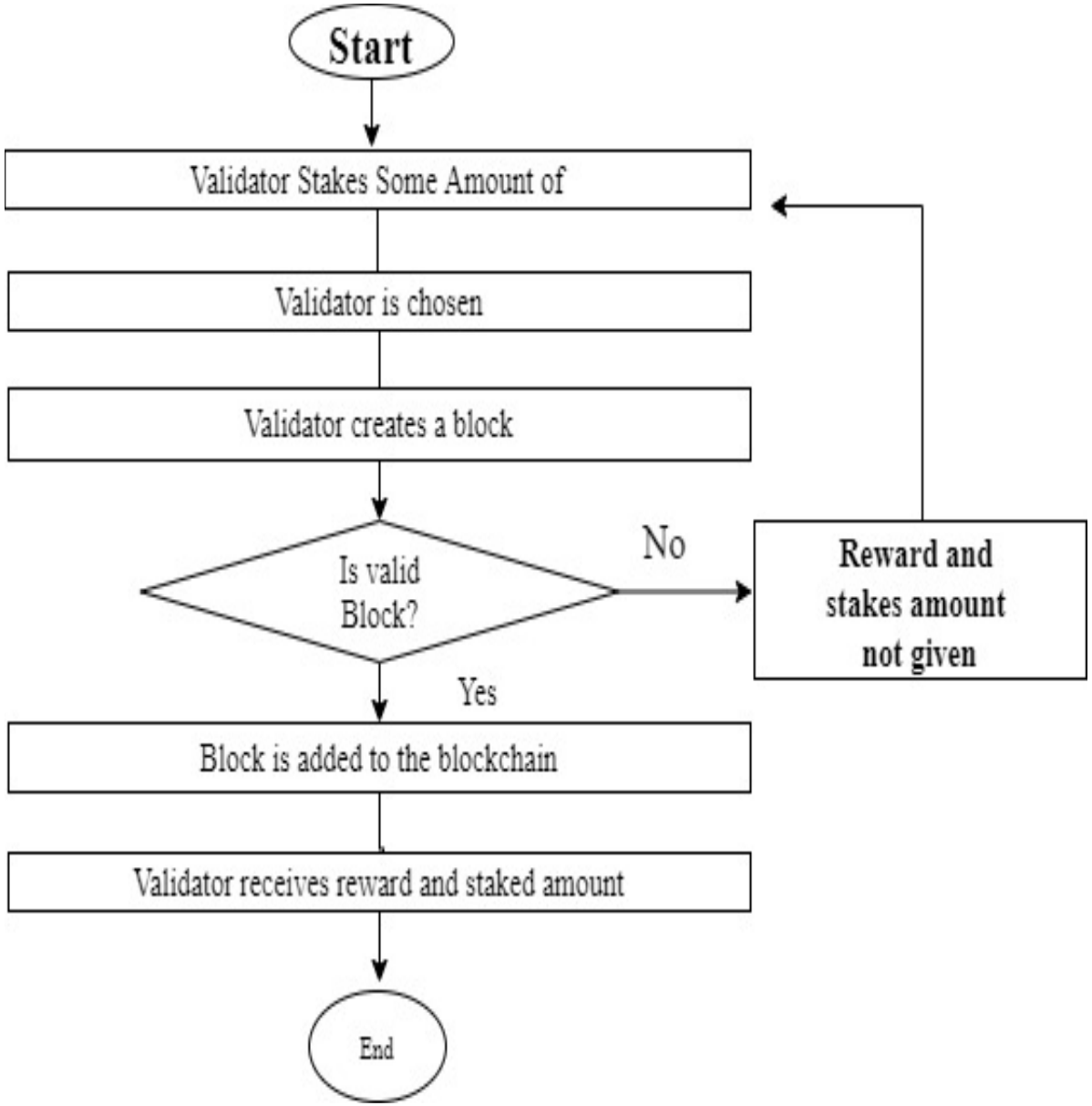


Fig. Flow of PoS consensus

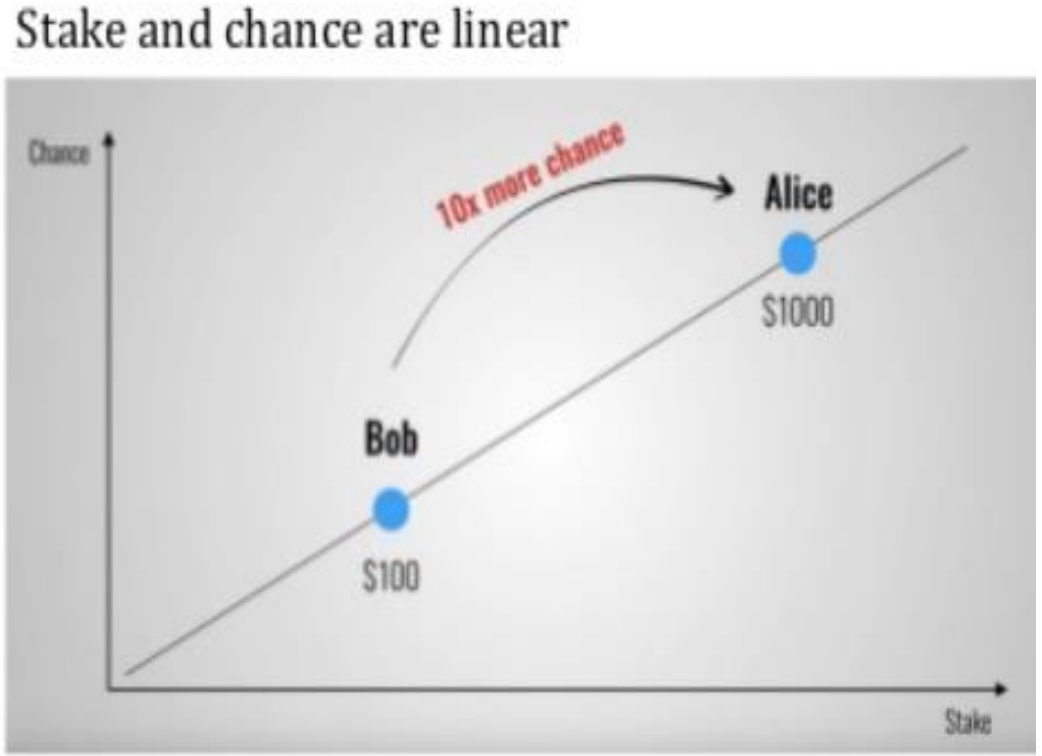


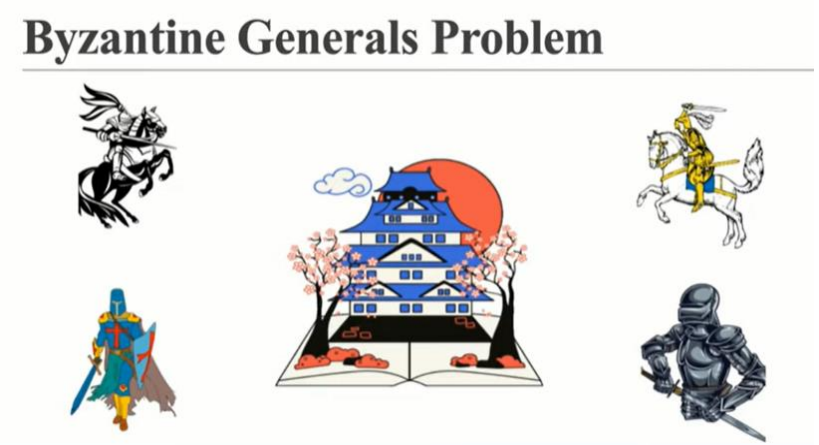
Fig. Relation between stake and its winning chance

POW vs POS

Proof of Work (POW)	Proof of Stake (POS)
Miners	validators
High performance hardware required	Mobile or laptop are enough
Lots of electricity required	Not much electricity required
The more hashing power you have the more blocks you can validate	The more ETH you state the more blocks you can validate
Attack to happen 51% hashing power is required	Attack to happen 51% stake is required
Competition based selection	Random based selection

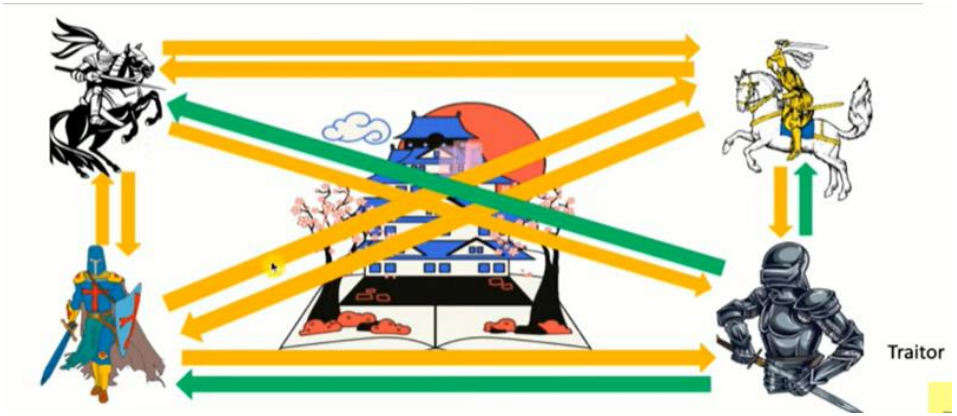
Byzantine Generals Problem

This problem is only in Distributed system



Generals (4), Palace (1)

All Generals agreed to attack



3 Generals Agree, 1 not agree

Solution??

Byzantine Fault Tolerance (BFT) Consensus

Faulty Component

- ☞ Crash Failure
- ☞ Byzantine Failure

To achieve the consensus in the presence of faulty component, the following goals must be satisfied by the system:

- ☞ Validity: Any value decided upon must be proposed by one of the process (proposer).
- ☞ Agreement: All non-faulty processes must agree on the same value.
- ☞ Termination: All non-faulty node eventually decide on the output value.

☞ In a message-passing system with n components, if f components are Byzantine and n ≤ 3f, then it is impossible for the system to reach the consensus goal.

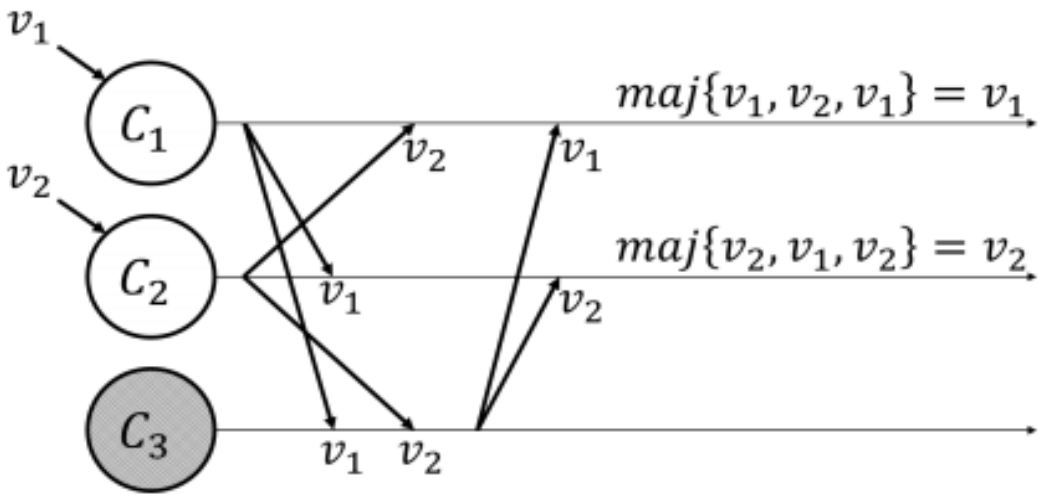


Fig. Three-component message-passing system with one component being Byzantine 5ℓ

Conclusion: In the general case, for any distributed system with N components and f being Byzantine, N ≥ 3f +1 is required to ensure consensus.

5ℓ : Majority rule is applied to achieve the consensus.

Cont…

- The BFT consensus did not perform well for permissionless blockchain network. the reason for this is that in the permissionless system, we did not memorize the identity of peers node. Therefore, the number of nodes currently present in the system is an open question?
- Whereas, to achieve the consensus using BFT, it is desirable that every peers should know the identity of others and also number of nodes present in the system.

Conclusion: BFT consensus works well for permissioned blockchain system.

Practical Byzantine Fault Tolerance (PBFT) Consensus

- The practical Byzantine fault tolerance algorithm is an example of Byzantine fault tolerance (BFT), published by Miguel Castro and Barbra Lisko in 1999.
- It is a replication algorithm to deal with byzantine faults in a distributed network.
- To decide the faulty node, the honest nodes of the system reach a consensus and a system that can conclude is not affected by a malicious/faulty node.
- The communication overhead is more in the PBFT consensus.
- In the PoW Algorithm block is created by the winning miner node. In PoS, the block creator is the richest miner. Unlike PoW and PoS, in PBFT block is not generated by any special node, rather the most agreed block is committed to the chain.

Cont…

- PBFT has pre-prepared, prepared, and commit stages to complete the block creation process.
- PBFT system can tolerate up to n faulty node out of 3n+1 node.
- To make any decision, PBFT needs approval of [(3n + 1) – n] = 2n + 1 node from the network which has 3n + 1 node.
- In PBFT, the block is not created by a special miner but is the most agreed block from the network. PBFT protocol will append the most agreed block in the network.
- PBFT is an energy-efficient algorithm because the consensus is achieved without solving complex cryptographical mathematical puzzle and transactions do not require multiple confirmation

Roles

Client: Client nodes that are responsible for sending transaction requests. (D is the client node)

Primary: Main nodes that are responsible for packing transactions into blocks and finalizing blocks. Each consensus-reaching process has one and only one Primary node. (R0 is the primary node)

Replicas: Replica nodes that are responsible for finalizing blocks. Each consensus-reaching process involves multiple Replica nodes, and they all proceed in a similar way. (R1,R2,R3 are replica node)

Both Primary and Replica nodes are consensus nodes.

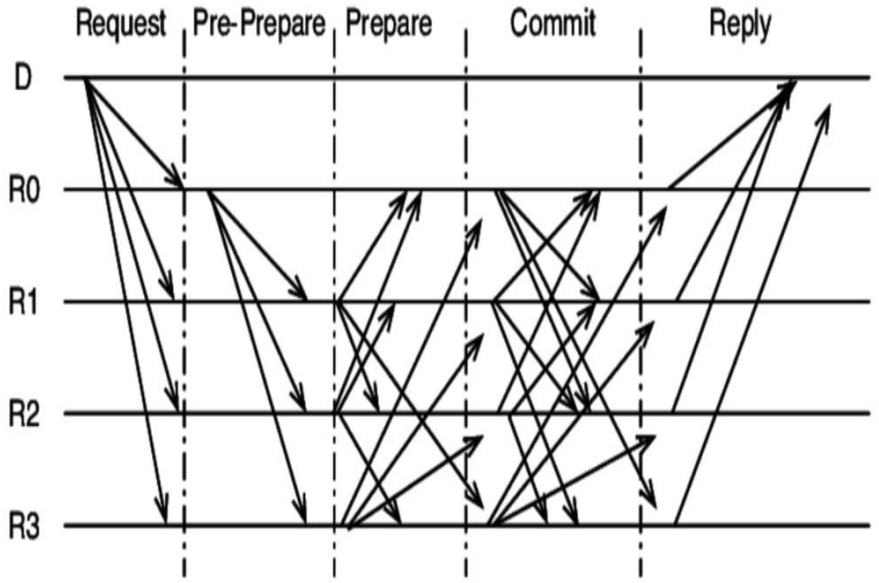


Fig. 3-phase protocol of PBFT

Algorithm Description

1. As shown in Fig.1, Client D sends a request to the system, and consensus nodes (R0, R1, R2, R3) receive the request. After the Primary node (R0 in this case) broadcasts the pre-prepare message, the system starts to execute the three-phase consensus.
2. **Pre-Prepare:** Primary node is responsible for verifying the requests and generating corresponding pre-prepare messages. Then, the Primary node will broadcast pre-prepare messages to all Replica nodes. After receiving the messages, Replica nodes will verify the legitimacy of those pre-prepare messages and then broadcast a corresponding prepare message.
3. **Prepare:** Gathering prepare messages. After a certain node gathers 2f+1 prepare messages, it will announce that it is ready for block submission and start to broadcast commit messages;
4. **Commit:** Gathering commit messages. After a certain node gathers 2f+1 commit messages, it will process the native requests cached locally and make corresponding changes to the system state.

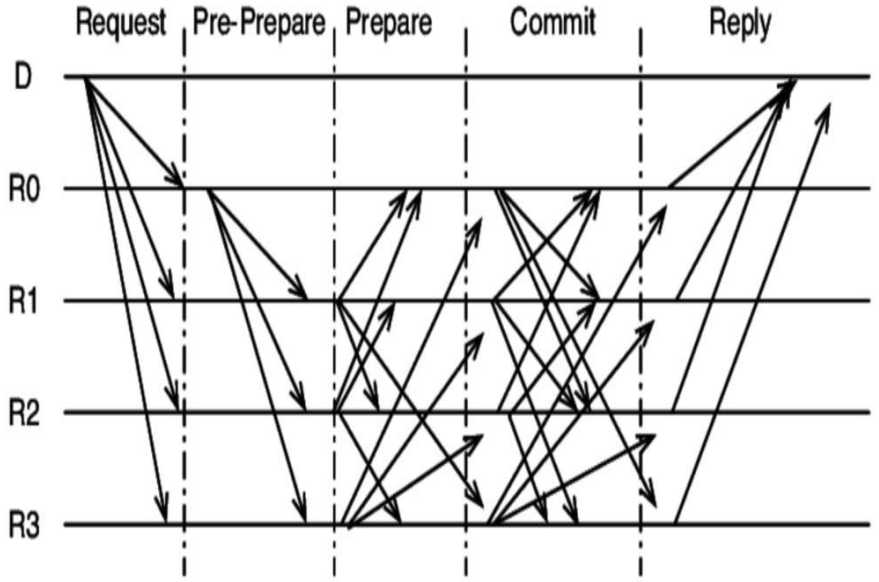


Fig. 3-phase protocol of PBFT

Thank you

Dr. Shyama Prasad Mukherjee International Institute of Information Technology, Naya Raipur