- **Transaction Fees**
    - All Transactions Include a Transaction Fee
        - An Incentives for the miners to mine and include the transaction in the blockchain.
        - For maintaining the blockchain.
    - Transaction Fee is paid to the miner who successfully adds the candidate block to the blockchain.
        - All will mine the transaction and all will create the candidate block.
        - But, only one candidate block will be added to the blockchain.
        - The miner who created the candidate block will get the incentives for mining the transaction.
    - Who will fix the transaction fee ?
        - Default transaction fee is set by the network first.
            - It is dynamic and market dependent
        - Wallets takes care of this.

    -
    - Transaction fees serve two essential purposes when it comes to blockchain networks. They reward miners or validators who help confirm transactions and help protect the network from spam attacks.
    - Transaction fees are and have been an essential part of most blockchain systems since their inception. You are most likely to have come across them when sending, depositing, or withdrawing crypto
    - The majority of cryptocurrencies use transaction fees for two important reasons. First of all, fees reduce the amount of spam on the network. It also makes large-scale spam attacks costly and expensive to implement. Secondly, transaction fees act as an incentive for users that help verify and validate transactions. Think of it as a reward for helping the network.
    - For most blockchains, transaction fees are reasonably cheap, but they can get quite expensive depending on network traffic. As a user, the amount you choose to pay in fees determines your transaction's priority in being added to the next block. The higher the fee paid, the quicker the confirmation process
    - How transaction fee is calculated ?
        - It is based on no. of transactions or size of the transaction.
        - Not the value included in transaction
        - Single transaction with Rs. 500 has less transaction fee
        - Whereas multiple transactions with Rs. 300 will have more transaction fee.

Case 1:

P

TxID:000001, Amount = 50, Owner = P

TxID:000023, Amount = 2, Owner = P

TxID:000056, Amount = 75, Owner = P

TxID:000089, Amount = 100, Owner = P, Key = *&*$#, Transfer to R

TxID:000092, Amount = 5, Owner = P

- Only one transaction is required to mine.
- Transaction Fee will be default transaction fee.

R

Case 2:

P

TxID:000001, Amount = 50, Owner = P, Key = *&*$#, Transfer to R

TxID:000023, Amount = 2, Owner = P , Key = *&*$#, Transfer to R

TxID:000056, Amount = 75, Owner = P

TxID:000089, Amount = 100, Owner = P

TxID:000092, Amount = 5, Owner = P, Key = *&*$#, Transfer to R

- Three transactions are required to mine.
- Transaction Fee will be more than the default transaction fee.

R

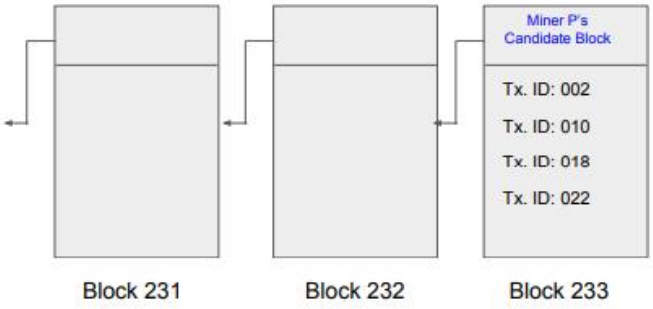
- Case 1: Total transaction Value Rs. 100 - Default Transaction Fee
- Case 2: Total transaction Value Rs. 57 - More Transaction Fee
- Though the transaction value is less in Case 2, because of more transactions to be mined,
- transaction fee is more.
- Wallet should be intelligent enough to understand all these parameters.
- Accordingly it should fix the transaction fee for the transactions.
- What if the wallet gives the default transaction fee for all the transactions ?
- Transaction Fee might affect the processing priority.
- Eventually all transactions will be mined.

Miners and Transaction Fee

- Miners wants to earn more - More Greedy.
  - Tries to include more transactions in the candidate block to get more transaction fee.
- Hence, they will prioritize the transactions.
  - Less transactions to mine - Default fee : They may take it.
  - More transactions to mine - Less fee : They will neglect it
  - More transactions to mine - More fee : Crowd Puller, All Miners will eye on it.
- When a miner is about to complete the candidate block
  - Short of some x bytes.
  - Takes any transaction without looking at the transaction fee.
  - Its priority is to complete the candidate block
  - Mining block will fetch it more reward than the transaction fee.

Transaction Fee Vs Block Rewards

- Block Reward : Bonus for successfully adding one block to the blockchain.
  - It is over and above the transaction fees.
- If the candidate block is added to the blockchain,
- The corresponding miner (owner of the candidate block) will get
  - Transaction fee of the total transactions included in the candidate block
  - And the block reward.



Miner P will get

Transaction Fees of Tx IDs: 002, 010, 018, 022 + The Block Reward

If you Do Not have Exact Change ?

P

TxID:000001, Amount = 50, Owner = P

TxID:000023, Amount = 2, Owner = P

TxID:000056, Amount = 75, Owner = P

TxID:000089, Amount = 100, Owner = P

TxID:000092, Amount = 5, Owner = P, Parent Transaction

- You Need Rs. 78
- Best Transactions : TxIDs:000056 (75) and 000092(5) = Total Rs. 80
- What about remaining Rs. 2?
  - Create a new transaction with Rs. 2 to yourself. Child Transaction
  - If you forget it, miners will take it as a reward.
    - It is over and above the transaction fee !!!

Transaction Chain

- Grand Parent Transaction, Parent transaction, Child transaction,
  - We can have a Generation tree of related transactions.
- Transaction Propagation
  - All Transactions are disseminated to all other miners.
  - C of CAP theorem.
- It may happen
  - A child transaction reaches the miners first than the parent miner.
    - Due to network related issues or compromised miners.
    - Child transactions can not be processed until parent transaction reaches the miner.
    - Such transactions are called as Orphan Transactions.

Orphan Transaction

- Transaction without a parent
- Orphan Transactions are maintained in the Orphan transaction pool or Orphan Pool
- UTXO is different from Orphan pool
- Each miner should maintain both the pools
- Once the parent transaction reaches a miner, its linked child transaction will be searched in the Orphan pool.
  - If child is found, both will be mined.

Orphan Transaction - DoS Attack

- Child Transactions are created by the adversaries.
- Miner receives child transaction and understands
  - Parent Transaction is yet to receive.
  - Miner puts the child transaction in the Orphan Pool
- Parent Transaction never appears
  - Denial of Service (DoS) attack
  - But childs keeps waiting for the parent transaction

- To Prevent Denial of Service (DoS) attack,
- Orphan Pool - fixing the size.
- If the orphan pool reaches the Maximum size which transactions to be removed ?
  - Based on the age of the transaction
  - Randomly removing the transaction

- **Locking and Unlocking Script**
  - A script used to sign the transaction
  - E.g. P transfer transaction ownership to Q, he lock the transaction using locking script and kept in UTXO pool, because of that know one (other wallets) can see that transaction. It is call creating encumbrance on transaction. Q want to spend that transaction for that he should remove encumbrance by singing with private key he has.

    - Transactions
      - X Pays Y
      - P Transfers Ownership to Q
      - Pay/Transfer Ownership to Public Key of Q
    - Locking and Unlocking Script.
    - Locking Script is placed on UTXO
      - aka encumbrance
      - So that no one can decode it
    - Unlocking Input Transaction
      - to Spend
  -

Case Study 2: A Finance Related Activity - Fundraising for School

P

TxID:000001, Amount = 50, Owner = P

TxID:000023, Amount = 2, Owner = P

TxID:000056, Amount = 75, Owner = P

TxID:000089, Amount = 100, Owner = P

TxID:000092, Amount = 5, Owner = P

P locks these transaction using Locking Script

-

P

TxID:000001, Amount = 50, Owner = P

TxID:000023, Amount = 2, Owner = P, Key = *&*$#, Transfer to R

TxID:000056, Amount = 75, Owner = P, Key = *&*$#, Transfer to R

TxID:000089, Amount = 100, Owner = P

TxID:000092, Amount = 5, Owner = P

Input TxIDs : 000023, 000056

Remove encumbrance for these Input TxIDs.

Sign with Private Key and then spend.

R

-

Why Encumbrance is Required ?

- Others should not see your transactions.
- Others ?
  - Other Users' Wallets
- Transactions are in the blockchain but no one can see its contents.
  - To whom it belongs to ?
- Caution
  - Do not leave the transaction unlocked and unspent in the UTXO.
  - Others can see that and take advantage of it.
- Encumbrance - Worries For Adversaries

-

Encumbrance - A worry for the Adversaries

- Case Study: Wants to sell a product with duplicate ownership
  - First find the particular transaction in the UTXO
  - Second, remove the encumbrance
  - Third, spend it by signing with the private key.
- For each unsuccessful transaction attempts
  - A valid transaction is valid for all miners.
  - A Wrong Transaction is wrong for everyone. Inform the owner
  - Keep track the address which initiated this transaction
  - If it repeats, blacklist that address.

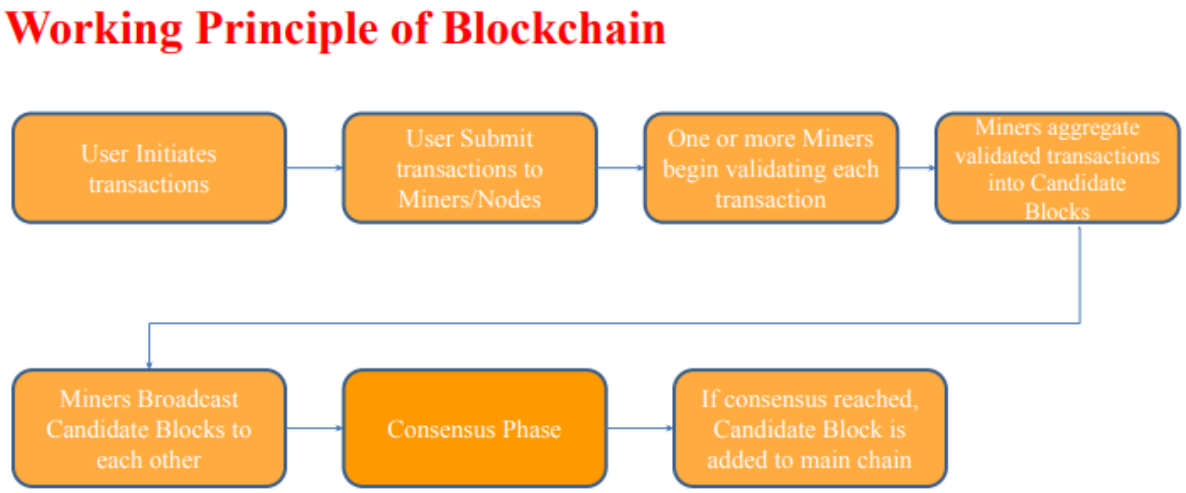
- https://www.youtube.com/watch?v=QNf3yX9lw7w&ab_channel=SatolearnBitcoinTutorials
- A locking script is an encumbrance placed on an output, and it specifies the conditions that must be met to spend the output in the future.
- the locking script was called a scriptPubKey, because it usually contained a public key or bitcoin address.
- An unlocking script is a script that “solves,” or satisfies, the conditions placed on an output by a locking script and allows the output to be spent. Unlocking scripts are part of every transaction input, and most of the time they contain a digital signature produced by the user’s wallet from his or her private key
- Historically, the unlocking script is called scriptSig, because it usually contained a digital signature
- Every bitcoin client will validate transactions by executing the locking and unlocking scripts together. For each input in the transaction, the validation software will first retrieve the UTXO referenced by the input. That UTXO contains a locking script defining the conditions required to spend it. The validation software will then take the unlocking script contained in the input that is attempting to spend this UTXO and execute the two scripts
- First, the unlocking script is executed, using the stack execution engine. If the unlocking script executed without errors (e.g., it has no “dangling” operators left over), the main stack (not the alternate stack) is copied and the locking script is executed. If the result of executing the locking script with the stack data copied from the unlocking script is “TRUE,” the unlocking script has succeeded in resolving the conditions imposed by the locking script and, therefore, the input is a valid authorization to spend the UTXO. If any result other than “TRUE” remains after execution of the combined script, the input is invalid because it has failed to satisfy the spending conditions placed on the UTXO. Note that the UTXO is permanently recorded in the blockchain, and therefore is invariable and is unaffected by failed attempts to spend it by reference in a new transaction. Only a valid transaction that correctly satisfies the conditions of the UTXO results in the UTXO being marked as “spent” and removed from the set of available (unspent) UTXO

- In Bitcoin
  - Locking Script is called as ScriptPubKey
  - Unlocking Script is called ScriptSig

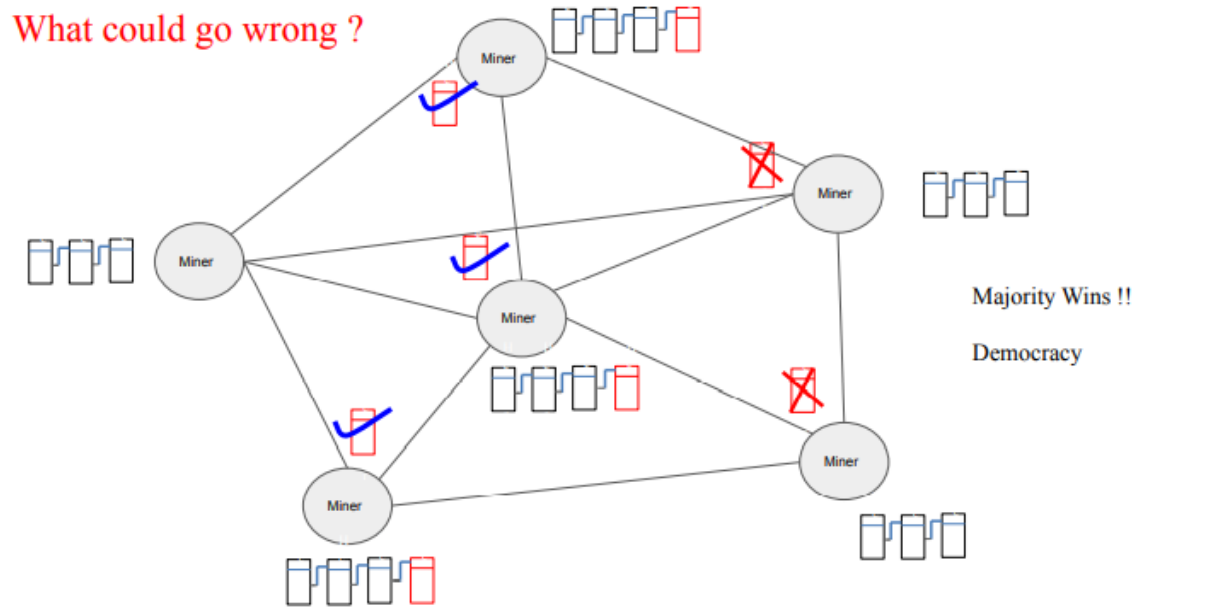
- **Standard Transactions.**
  - Pay to Public Key Hash
    - Hash of the Public Key
    - Adversaries should not change the address itself.
    - P to Y should not be changed on the fly to P to X
  - Pay to Public Kay
  - Multi Signature
    - Just Like Joint Account.
  - OP_RETURN
-
  - The five standard types of transaction scripts are pay-to-public-key-hash (P2PKH), public-key, multi-signature (limited to 15 keys), pay-to-script-hash (P2SH), and data output (OP_RETURN), which are described in more detail in the following sections.
  - **pay-to-public-key-hash (P2PKH) :** The vast majority of transactions processed on the bitcoin network are P2PKH transactions. These contain a locking script that encumbers the output with a public key hash, more commonly known as a bitcoin address. Transactions that pay a bitcoin address contain P2PKH scripts. An output locked by a P2PKH script can be unlocked (spent) by presenting a public key and a digital signature created by the corresponding private key.
  - **Pay-to-Public-Key:** Pay-to-public-key is a simpler form of a bitcoin payment than pay-to-public-key-hash. With this script form, the public key itself is stored in the locking script, rather than a public-key-hash as with P2PKH earlier, which is much shorter. Pay-to-public-key-hash was invented by Satoshi to make bitcoin addresses shorter, for ease of use. Pay-to-public-key is now most often seen in coinbase transactions, generated by older mining software that has not been updated to use P2PKH
  - **Multi-Signature:** Multi-signature scripts set a condition where N public keys are recorded in the script and at least M of those must provide signatures to release the encumbrance. This is also known as an M-of-N scheme, where N is the total number of keys and M is the threshold of signatures required for validation. For example, a 2-of-3 multi-signature is one where three public keys are listed as potential signers and at least two of those must be used to create signatures for a valid transaction to spend the funds. At this time, standard multi-signature scripts are limited to at most 15 listed public keys, meaning you can do anything from a 1-of-1 to a 15-of-15 multi-signature or any combination within that range
  - **Data Output (OP_RETURN):** Bitcoin's distributed and timestamped ledger, the blockchain, has potential uses far beyond payments. Many developers have tried to use the transaction scripting language to take advantage of the security and resilience of the system for applications such as digital notary services, stock certificates, and smart contracts. Early attempts to use bitcoin's script language for these purposes involved

creating transaction outputs that recorded data on the blockchain; for example, to record a digital fingerprint of a file in such a way that anyone could establish proof-of-existence of that file on a specific date by reference to that transaction

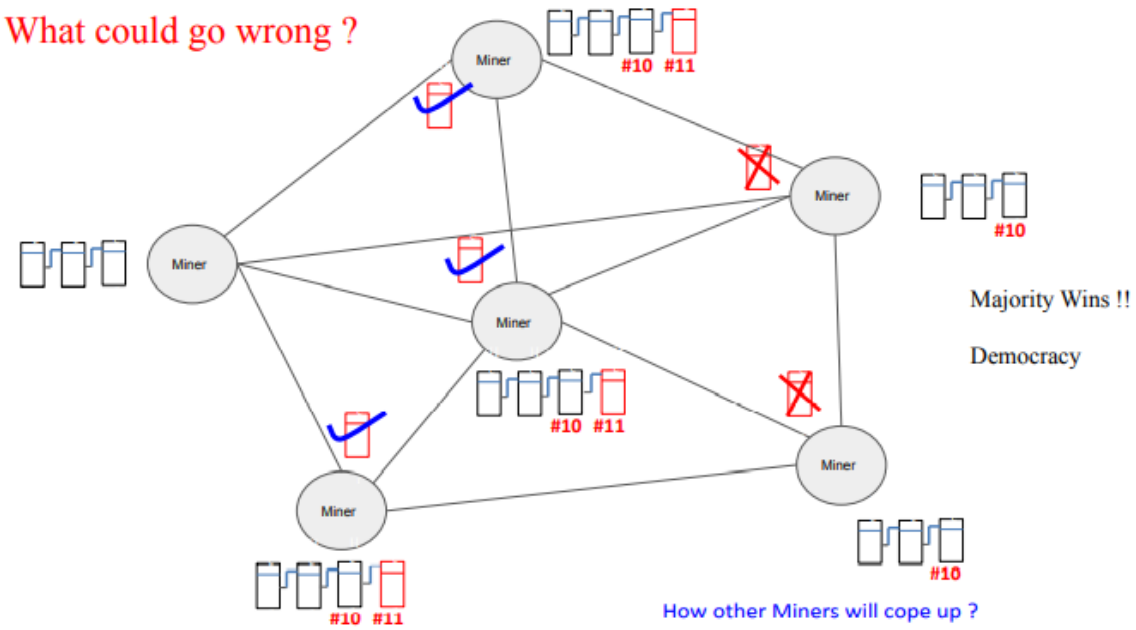
- **Consensus**
  - Why consensus: because decision taken at many site.

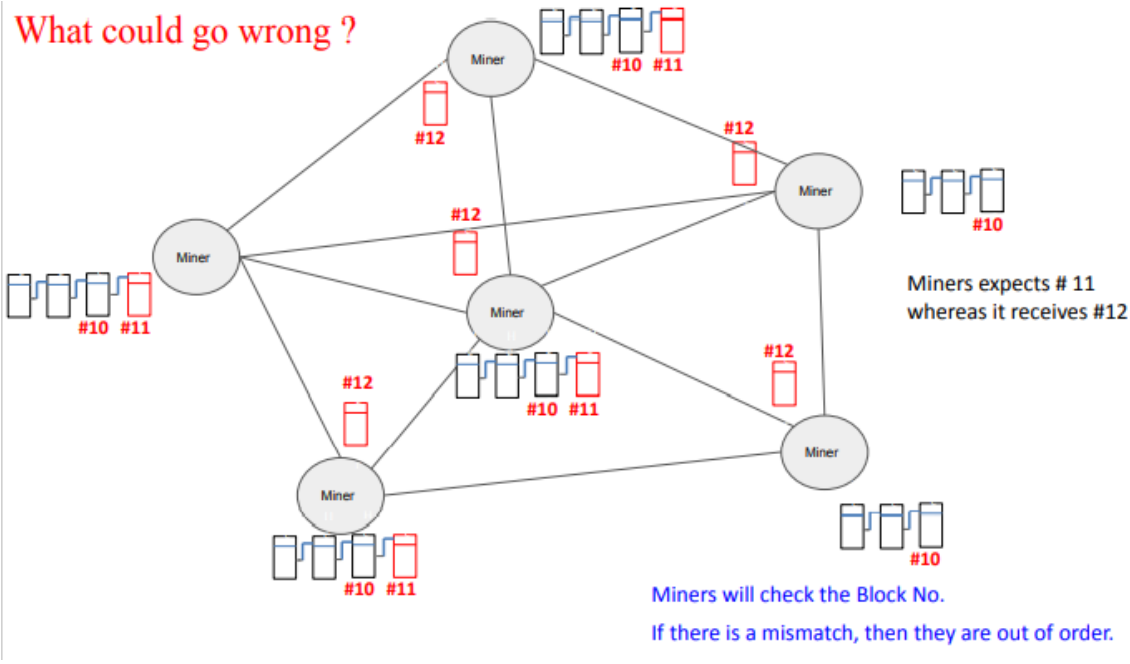


Place of Consensus Algorithm in the Blockchain Workflow

  - 

  -





- In that case longest chain rule apply, out of order blockchain will get discard and take copy from neighbor.
- Why can't we use traditional distributed consensus algorithm in blockchain ?
  - Traditional Distributed System
    - Synchronous behavior.
    - Identity is known to some extent.
  - Blockchain Network
    - Blockchain is a asynchronous system.
    - Identity Unknown (in public and consortium blockchain)
    - Can not expect node behavior

- What type of blockchain ?
  - Private, Public, or Consortium blockchain
- Public Blockchain - Bitcoin
- Private Blockchain - Blockchain Solution For Inventory Tracking, Academic Chain, etc.
- Consortium Blockchain - Smart Grid For Power Supply

-
- There are four main types of blockchain networks: public blockchains, private blockchains, consortium blockchains and hybrid blockchains. Each one of these platforms has its benefits, drawbacks and ideal uses.
- **Public blockchain:**
  - The first type of blockchain technology is public blockchain. This is where cryptocurrency like Bitcoin originated and helped to popularize distributed ledger technology (DLT). It removes the problems that come with centralization, including less security and transparency. DLT doesn't store information in any one place, instead distributing it across a peer-to-peer network. Its decentralized nature requires some method for verifying the authenticity of data. That method is a consensus algorithm whereby participants in the blockchain reach agreement on the current state of the ledger. Proof of work (PoW) and proof of stake (PoS) are two common consensus methods.
  - Public blockchain is non-restrictive and permissionless, and anyone with internet access can sign on to a blockchain platform to become an authorized node. This user can access current and past records and conduct mining activities, the complex computations used to verify transactions and add them to the ledger. No valid record or transaction can be changed on the network, and anyone can verify the transactions, find bugs or propose changes because the source code is usually open source.
  - Use cases. The most common use case for public blockchains is mining and exchanging cryptocurrencies like Bitcoin. However, it can also be used for creating a fixed record with an auditable chain of custody, such as electronic notarization of affidavits and public records of property ownership.
- **Private blockchain:**
  - A blockchain network that works in a restrictive environment like a closed network, or that is under the control of a single entity, is a private blockchain. While it operates like a public blockchain network in the sense that it uses peer-to-peer connections and decentralization, this type of blockchain is on a much smaller scale. Instead of just anyone being able to join and provide computing power, private blockchains typically are operated on a small network inside a

company or organization. They're also known as permissioned blockchains or enterprise blockchains.
        - The controlling organization sets permission levels, security, authorizations and accessibility. For example, an organization setting up a private blockchain network can determine which nodes can view, add or change data. It can also prevent third parties from accessing certain information.
    - **Hybrid blockchain**
        - organizations will want the best of both worlds, and they'll use hybrid blockchain, a type of blockchain technology that combines elements of both private and public blockchain. It lets organizations set up a private, permission-based system alongside a public permissionless system, allowing them to control who can access specific data stored in the blockchain, and what data will be opened up publicly.
        - Typically, transactions and records in a hybrid blockchain are not made public but can be verified when needed, such as by allowing access through a smart contract. Confidential information is kept inside the network but is still verifiable. Even though a private entity may own the hybrid blockchain, it cannot alter transactions.
        - When a user joins a hybrid blockchain, they have full access to the network. The user's identity is protected from other users
        - Use cases. Hybrid blockchain has several strong use cases, including real estate. Companies can use a hybrid blockchain to run systems privately but show certain information, such as listings, to the public
    - **Consortium blockchain**
        - The fourth type of blockchain, consortium blockchain, also known as a federated blockchain, is similar to a hybrid blockchain in that it has private and public blockchain features. But it's different in that multiple organizational members collaborate on a decentralized network. Essentially, a consortium blockchain is a private blockchain with limited access to a particular group, eliminating the risks that come with just one entity controlling the network on a private blockchain
        - In a consortium blockchain, the consensus procedures are controlled by preset nodes. It has a validator node that initiates, receives and validates transactions. Member nodes can receive or initiate transactions.
        - Use cases. Banking and payments are two uses for this type of blockchain. Different banks can band together and form a consortium, deciding which nodes will validate the transactions. Research organizations can create a similar model, as can organizations that want to track food. It's ideal for supply chains, particularly food and medicine applications.

- Consensus algorithm for public blockchain - tedious job.
  - Many will be involved.
  - Big boundary.
  - Unknown Identity
- Consensus algorithm for private blockchain - less restrictions.
  - Limited Members
  - Small Boundary
  - Known Identity

- Challenges in building Consensus algorithm
  - Crash Failure
  - Byzantine Faults
  - Partitioned Faults
  - Double Spending
- Attacks
  - 51%
  - Sybil
  - DoS

- Crash Failure:
  - A distributed system faces a number of threats. Processes may crash, devices at some location may fail or a network connection may stop working. For enterprise use, the consensus algorithm must have resilience against a variety of threats.
  - Crash fault tolerance (CFT) builds a degree of resiliency in the protocol, so that the algorithm can correctly take the process forward and reach consensus, even if certain components fail.
  - CFT is a good solution when a component of the system fails. However, when blockchain is used by enterprises, there may be different companies, divisions or teams controlling separate components of the system. These different companies, divisions or teams may have different objectives, which leaves the system vulnerable to the threat of malicious activity. CFT is unable to reach consensus if any entity acts maliciously.
- Byzantine Fault : Byzantine general problem

- Partition Fault : the cluster must continue to work despite any number of communication breakdowns between nodes in the system.
- Double Spending: Double-spending is a problem that arises when transacting digital currency that involves the same tender being spent multiple times. Multiple transactions sharing the same input broadcasted on the network can be problematic and is a flaw unique to digital currencies.

Types of consensus algorithms

- Based on methodology [2]
  - Proof -of-Concept based
  - Voting based
- Based on the application
  - Permission-ed consensus algorithm.
  - Permission less consensus algorithm.

- Proof -of-Concept Based
  - Proof of Work
  - Proof of Stake
  - Proof of Importance
  - Proof of Trust
  - Proof of Burn
  - Proof of Elapsed Time
  - Proof of Activity
  - Proof of Intelligence (AI Based)
- Voting Based
  - PBFT
  - Ripple
  - Stellar
  - Raft
  - Delegated PoS

Proof of Work (PoW)

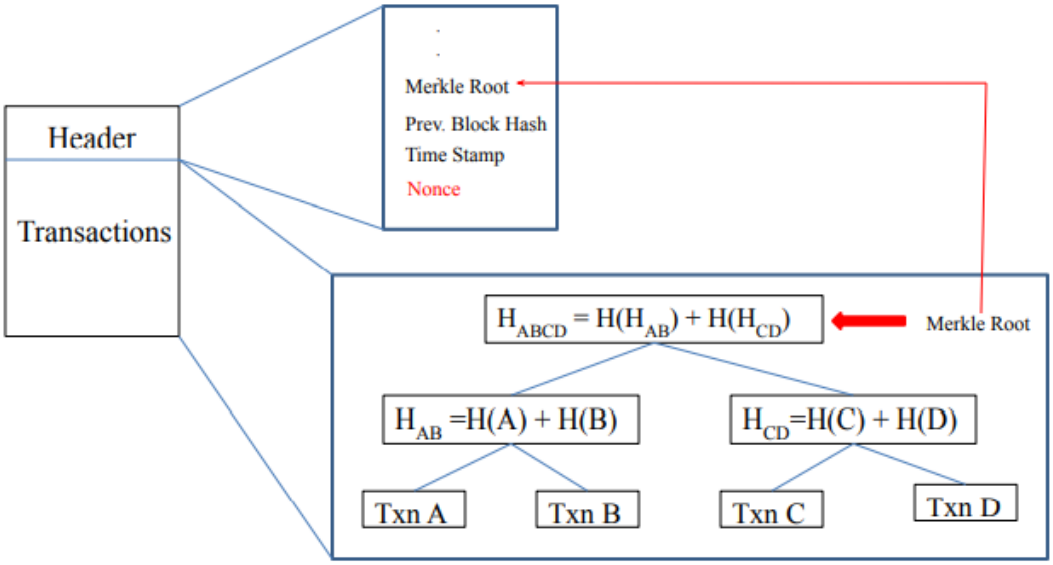
- Used in Bitcoin.
- Miners should prove that they have worked.
  - Uses Hash Cash Algorithm as a proof of work.
- Nonce (Number used only once) value.
  - Nonce, a special variable which is used only in case of PoW.
- Target Value
  - Ex. leading 5 zeros.
- Random Approach.

Nonce

- An Integer Value
- 32 bit value
  - Ranges 4 billion values !!
- Random numbers.
- Finding the particular nonce value is tough, but once it is found, evaluating the nonce is very easy.
- With each possibilities of nonce,
  - Place the nonce in the header of the candidate block
  - Hash (Header + Nonce) < target Value

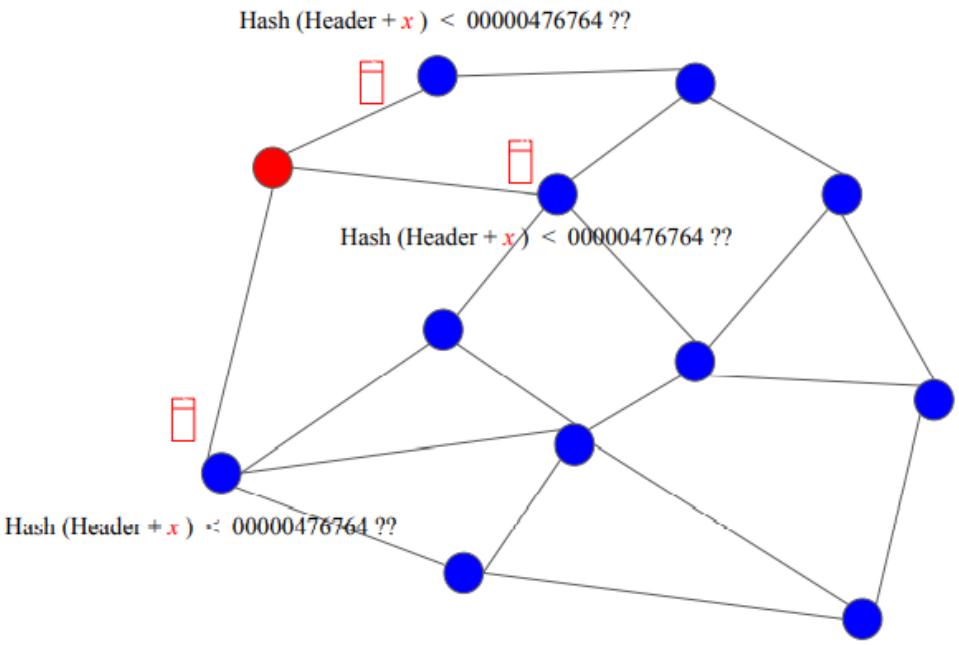
Candidate Block

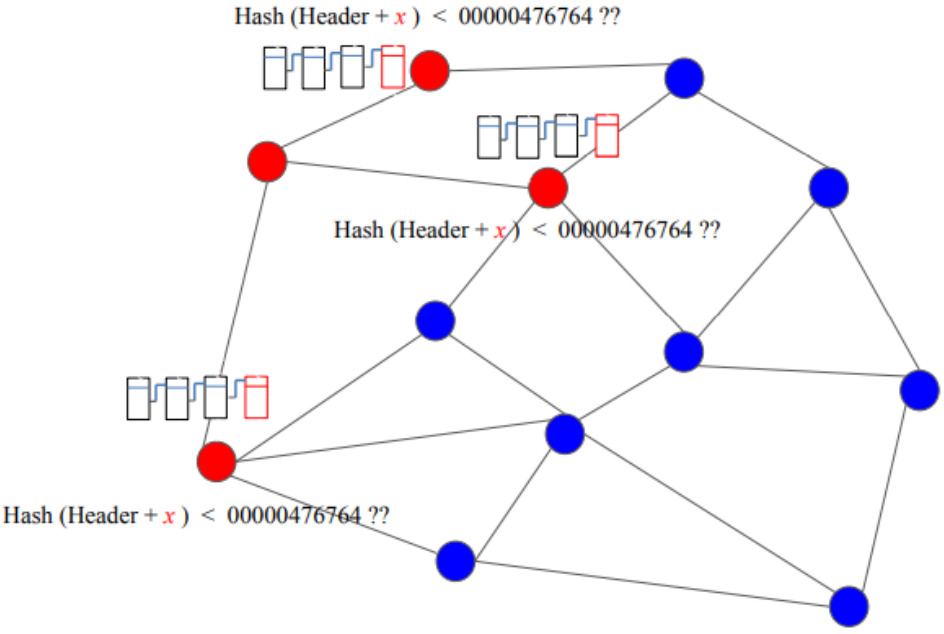
- UTXO – Unspent Transaction Output
- Take Transactions from UTXO- construct candidate block
- Construct the header except nonce.
- Find the nonce.
  - Get the hash, check for the target value.
- The Block Structure now includes a nonce variable in the header part.

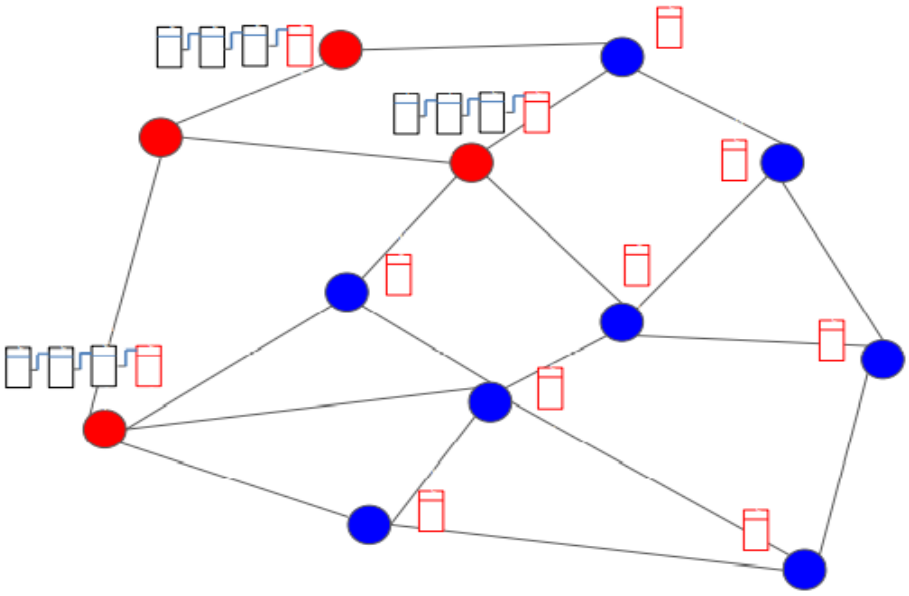
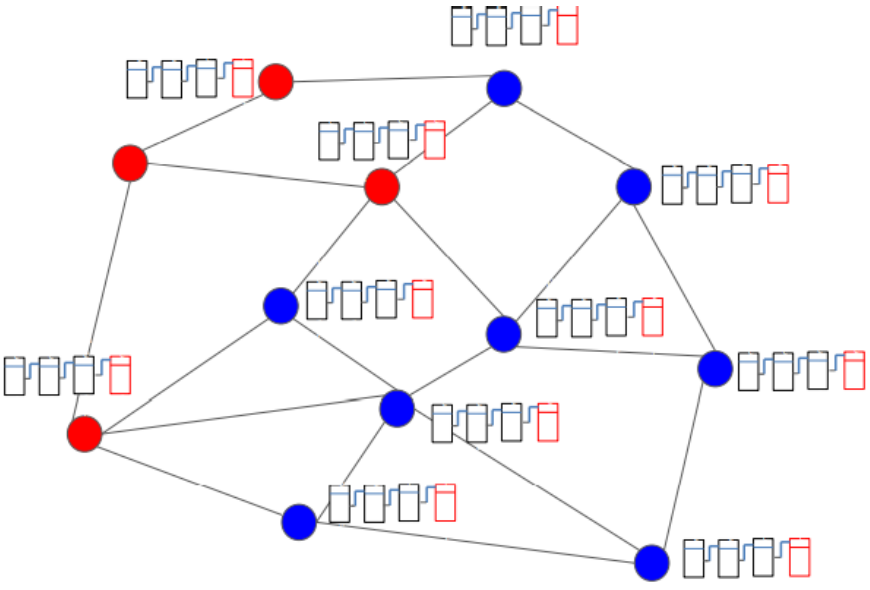


- Hash (Header + Nonce) < Target Value
  - Hash (Header + 001) < 00000476764
  - Hash (Header + 002) < 00000476764
  - Hash (Header + 003) < 00000476764
  - Hash (Header + 004) < 00000476764
  - ….
  - Hash (Header + *x*) < 00000476764
    - Disseminate the Candidate Block for the consensus among the miners.

- Once you find nonce as “x” then this block go for verification to all other miner. All miner just check this x value is valid for nonce, if yes then add. Now here block is verify and every block contain transaction which is verify by that miner itself who found nonce.


Hash (Header + x) < 00000476764 ??
Hash (Header + x) < 00000476764 ??
Hash (Header + x) < 00000476764 ??

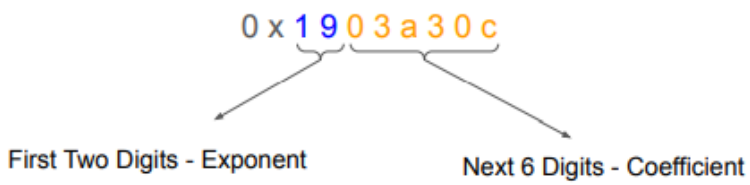

Hash (Header + x) < 00000476764 ??
Hash (Header + x) < 00000476764 ??
Hash (Header + x) < 00000476764 ??

- 
- 

Target Value

It is represented in Hash, called as “Target Hash”

For ex. 0x1903a30c



Target Formula = Coeff *2 $^{(8*exp-3)}$

= 0x03a30c * $2^{(8 * 0x19 - 3)}$

= …….

= 15 leading zeros (In Decimal).

Setting Target Hash

- All Full Nodes should reset the target Hash.
- It should be done dynamically
- For every 2016 blocks, all Full Nodes will reset the target hash.
- Formula to reset the target hash
  - New Difficulty = Old Difficulty * (time taken for last 2016 blocks / 20160)

- The difficulty is a measure of how difficult it is to mine a Bitcoin block, or in more technical terms, to find a hash below a given target. A high difficulty means that it will take more computing power to mine the same number of blocks, making the network more secure against attacks. The difficulty adjustment is directly related to the total estimated mining power estimated in the {hashrate} chart.
- The difficulty is adjusted every 2016 blocks (every 2 weeks approximately) so that the average time between each block remains 10 minutes.
- The difficulty comes directly from the confirmed blocks data in the Bitcoin network.

Queries…

- Who will set up the target value ?
  - blockchain is a decentralized system.
- All nodes will run a formula simultaneously after n blocks have been added.
- The result of the formula is the new target for the next n blocks.
- Target is set in such a way that, the new block is constructed at every 10 minutes in bitcoin.

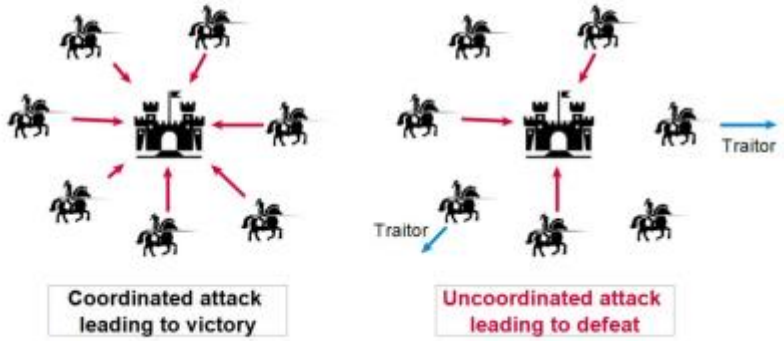
Pros and Cons

- Target Hash Study: A Miner with 1 Tera Hashes/ Sec (1 Trillion Hashes = 1 followed by 12 zeros) takes years together to find nonce
- Hence, To generate more hash more hardware is used - produces heat
  - Non Eco Friendly
  - Energy inefficient
- Expensive - in terms of setup.
- No Fairness among the miners.
  - The richer wins
  - Richer buys more hardwares, finds the nonce sooner than the others.
- Pool of miners a threat - 51% Attack

Byzantine Fault Tolerance (BFT)

- Solves Byzantine General Problem
- Reaches consensus despite malicious nodes.
- Assumptions,
  - N nodes, at most f faulty nodes
  - Closed Environment (identity of the sender is known)
  - Fully connected, Reliable communication medium

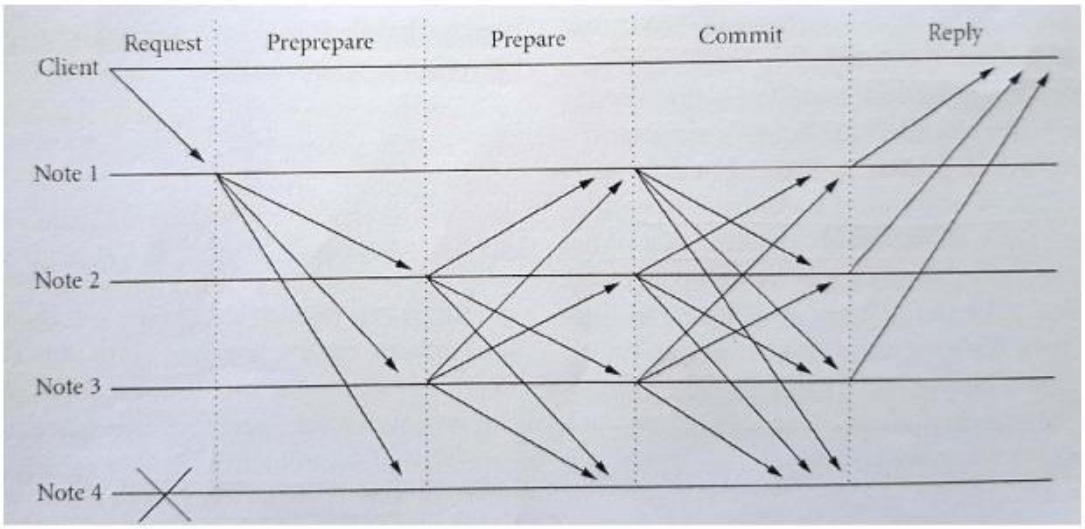
Byzantine General Problem



- The real world systems are asynchronous.
- No guaranteed message delivery.
- Practical BFT (PBFT) was proposed to tackle asynchronous system.
- 3 types of nodes,
  - Clients, Commander (leader), lieutenant (follower)
- PBFT runs in three phases,
  - Pre prepare, prepare and commit phase

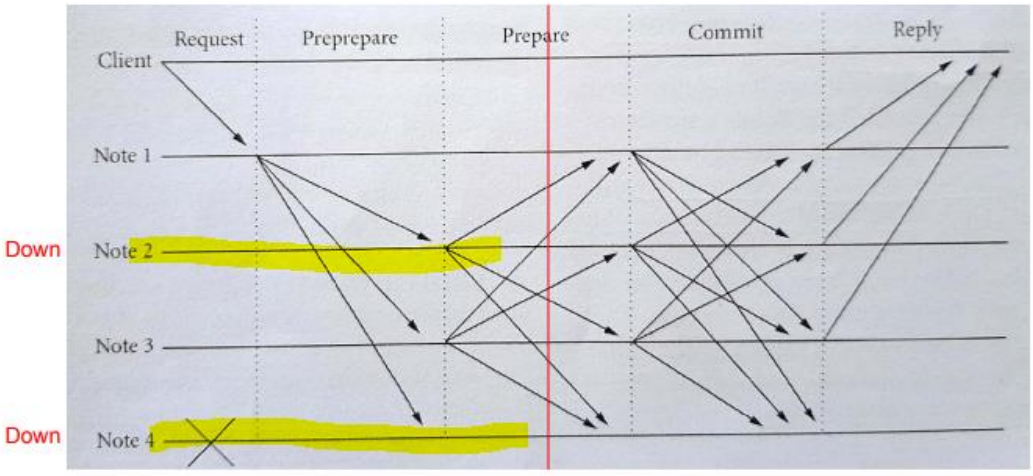
- Pre Prepare Phase
  - Client submits a request to commander
  - Commander assigns a sequence no. to the request along with the digital signature.
    - Any node which receives the request becomes the commander.
  - Commander Multicast the request to the Lieutenants.
- Prepare Phase
  - Lieutenants are validators, once they accept the request, Prepare Phase starts.
  - These lieutenants further multicasts the request in the network.
  - Once the majority of the lieutenants receives request, commit phase starts

- Commit Phase
  - The Commit information is broadcasted.
  - If Majority of the nodes receive commit message, the commit phase will be complete.
  - Finally, the information will be replied back to the client.
- *PBFT is called as Three Phase Protocol.*



Different Cases:

- Case 1: Node 4 is Down.
  - Majority of the nodes are up. Hence, consensus is achieved.
- Case 2: Nodes 4 and 2 are down.
  - Prepare Phase Fails.
  - In Prepare Phase, request should be received from majority of the nodes.
  - In this case, Only nodes 1 and 3 multicasts. Majority rule fails.



Different Cases:

- Case 3: Node 4 is down, Node 2 is malicious.
  - Prepare phase is achieved, but commit fails.
  - As majority rule fails.
  - But, it's a known environment - Private Blockchain.
  - Nodes can be blacklisted.
  - A threat to the miner which forces them to maintain the integrity.
- Case 4: How much time nodes should wait in Prepare Phase ?
  - For each phase, a fixed time interval will be set.
  - In case of time out, the current round of the consensus terminates and the new session starts.

$$TOTmsg = (n-f)[2n-1]$$

Points to Ponder - pBFT

- What is the Max no of faulty nodes a network can tolerate ?
- f = (n - 1) / 3
- f= (4-1) / 3 = 1
  - In a network of 4 nodes, utmost 1 faulty/unreachable/down nodes can be present for the proper functioning of the network.

- Advantages
  - High throughput, low latency, less energy usage
- Disadvantages
  - Multicasting overhead
  - Network Scaling problem

Proof of Stake

- Introduced as an alternative to PoW.
  - By Scott Nadal and Sunny King in 2012, a pseudo names.
- Works on the staked coins
- All miners will stake the coins.
  - More the coin, high will be the win probability.
- Only the winner miner will add new block to the blockchain.
  - The winner miner will earn all the Tx fees.
  - There is No Block Reward in PoS.
  - Rich becomes much richer.

- Proof-of-stake reduces the amount of computational work needed to verify blocks and transactions. Under proof-of-work, it kept blockchain secure.

Proof-of-stake changes the way blocks are verified using the machines of coin owners, so there doesn't need to be as much computational work done. The owners offer their coins as collateral—staking—for the chance to validate blocks and then become validators

- Validators are selected randomly to confirm transactions and validate block information. This system randomizes who gets to collect fees rather than using a competitive rewards-based mechanism like proof-of-work.
- To become a validator, a coin owner must "stake" a specific amount of coins. For instance, Ethereum requires 32 ETH to be staked before a user can become a validator.Blocks are validated by more than one validator, and when a specific number of the validators verify that the block is accurate, it is finalized and closed
- How they select winner?
    - The blockchain algorithm selects validators to check each new block of data based on how much crypto they've staked. The more you stake, the better your chance of being chosen to do the work. When the data that's been cleared by the validator is added to the blockchain, they get newly minted crypto as a reward.
- How Does Proof-of-Stake Solve the Byzantine Generals Problem?
    - Networks governed by the proof-of-stake consensus algorithm don't rely on mining – they rely on staking. To become a network validator, the user must first stake funds in the system. Those who own a greater share can also validate more blocks and earn greater rewards. Those who attempt to tamper with the information are at risk of losing their staked amount.

Points to Ponder

- PoS is used in Ethereum, Peer Coin and other blockchain technologies
- Deterministic - We can guess the winner !!
- No mining Concept here.
    - Solving complex mathematical solution like PoW.
- May lead to Monopoly - Rich may dominate
    - Hence, Many variations have come to avoid the monopoly
-
-
-
-

If Winner Miner is A Byzantine Node…

- Only Winner Miner is creating the candidate block and adding it to the main chain.
- No verification from other Miners like PoW.
- What if Winner Miner includes wrong transactions in to the blockchain ?
  - An Hefty penalty will be imposed
  - May be black listed.

Ripple Consensus Algorithm

- Ripple - "Wave"
- Permission-ed consensus Algorithm
- Targets correctness, agreement.
- Run on all nodes at specific interval.

Components

- Server (n) – runs ripple server, helps in consensus process
- Ledger – book keeps the Txns.
- Last Closed Ledger – Last successfully modified version of the ledger.
- Open Ledger – Local copy containing Txns which are not committed.
- Unique Node List (UNL) – A list of nodes which participate in the consensus.
- Proposer – who proposes the transactions to be included in the ledger.

Working Principle

- All valid Txns will be taken by the servers which are not committed in the closed ledger.
- Candidate Set will be prepared by the servers.
- Voting phase, the nodes should agree on the candidate set to include in the ledger.
- 80% of the UNL should agree to convert open ledger to closed ledger.

-
- How they handle BFT and other problem
-
  - Let us say there are 100 generals (nodes) in the battalion (blockchain network).
  - There is a strategist(RPCA), that aims to ensure that each and every loyal general either comes to the same conclusion, or no conclusion at all. They cannot afford to have a different conclusion.
  - So the strategist asks each of them to select the generals that they trust (this does not guarantee their loyalty. It's just the people they think they can trust.) and make a list (UNL: Unique Node List).
  - Now the strategist asks them to have a consensus with the people in their list. If more than 80% of the people in their UNL come up with the same decision, then it is finalized by the general. This is because the strategist knows that their army can only handle the traitorship of less than 20% of the generals (Byzantine fault tolerance = 20%). Similarly all the generals consult with their UNLs and come to a consensus.

- Now let's say there are 20 generals (all traitors), each of whom have the rest of the 19 in their UNL. And there is another UNL of the same type but with honest generals. So both of these UNLs will independently come to a consensus, which will contradict each other. But the main motive of the strategist was to ensure that every loyal general comes to the same consensus in their UNLs.
- So the strategist made another rule: If we select any two UNL's, they should have at least 20% of the members in common. Hence there will always be enough people in each UNL to stop them from reaching the wrong decision.
- Eg. in the twenty faulty generals, if we replace 4 generals (20% of 20 generals) with honest ones, then the majority would not be more than 80%, thus preventing consensus.
- Hence none of the generals would ever come to the wrong consensus, regardless of who is in their UNL, as long as the number of traitors is less than 20%.

Stellar

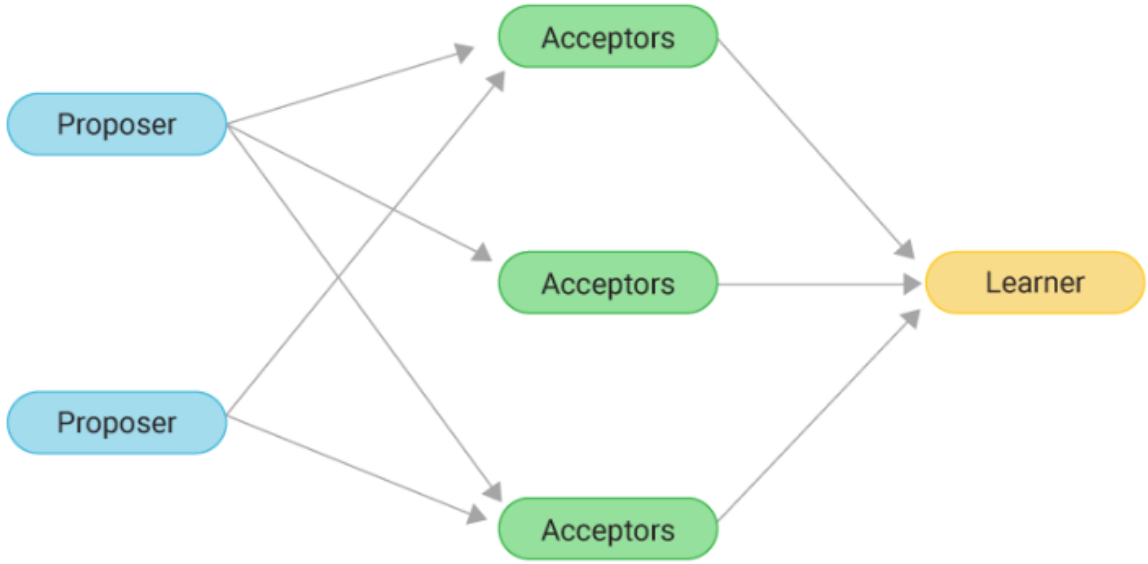
- Based on Federated Byzantine Agreement (FBA)
- FBA – A chain like structure, where each node has a set of nodes that it considers important.
- Each node waits for other nodes (which it has considered as important) to agree on the Txn.
- In turn, these other nodes will wait for its other nodes to agree for the Txn.
- If any one node agrees for a Txn, it will have a cascaded effect in the algorithm.
- Thus, node failure will not affect the algorithm.

- A Federated Byzantine Agreement (FBA) is a form of Byzantine fault tolerance where each byzantine general is responsible for their own blockchain. A Federated Byzantine Agreement (FBA) is used for its high throughput, network scalability, and low transaction costs. Notable cryptocurrencies using the Federated Byzantine Agreement (FBA) include Stellar and Ripple. Stellar was the first cryptocurrency to successfully implement a safe and secure Federated Byzantine Agreement (FBA), even though FBA consensus mechanism was pioneered by Ripple.

Paxos

- Nodes are categorized in to
  - Proposers , Acceptors and Learners.
- Proposer prepares a proposal called *prepare message and sends it to acceptors.*
- Prepare message consists of a (latest) proposal no.
- Each acceptor checks for the latest version of the proposal. If not, rejected.
- Acceptor prepares a *proposal acceptance message*.

- In the voting phase, acceptors will vote for the proposal message.
- Based on the *majority* of the proposal acceptance ratio, proposer prepares a acceptance message to all the acceptors. (In the winning case)
- When a acceptor accepts a proposal, it will inform all other Learners about the proposal acceptance.
- If (N/2 -1) acceptors fails, the algorithm can not reach consensus.



- Paxos is an algorithm that is used to achieve consensus among a distributed set of computers that communicate via an asynchronous network. One or more clients proposes a value to Paxos

and we have consensus when a majority of systems running Paxos agrees on one of the proposed values. Paxos is widely used and is legendary in computer science since it is the first consensus algorithm that has been rigorously proved to be correct.

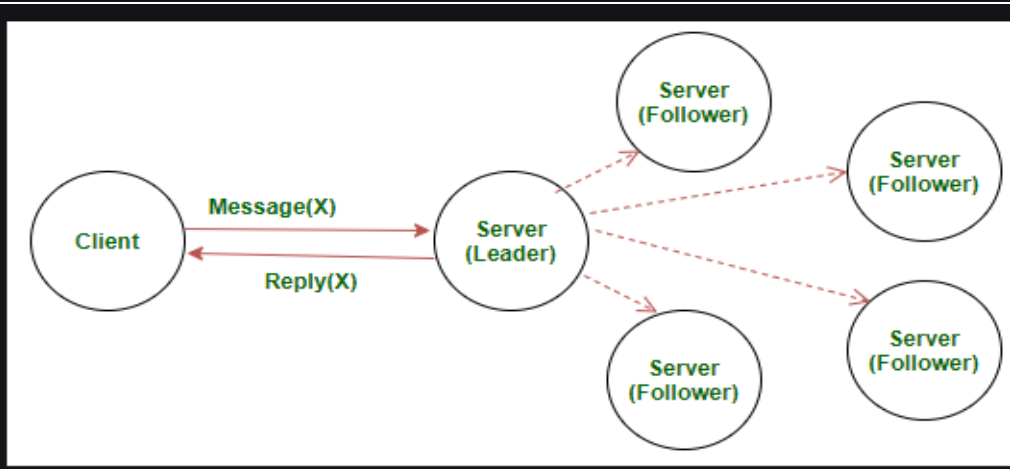
- Paxos simply selects a single value from one or more values that are proposed to it and lets everyone know what that value is. A run of the Paxos protocol results in the selection of single proposed value. If you need to use Paxos to create a replicated log (for a replicated state machine, for example), then you need to run Paxos repeatedly. This is called multi-Paxos. There are some optimizations that could be implemented for multi-Paxos but we will not discuss those here.
- Paxos provides abortable consensus. This means that some processes abort the consensus if there is contention while others decide on the value. Those processes that decide have to agree on the same value. Aborting allows a process to terminate rather than be blocked indefinitely. When a client proposes a value to Paxos, it is possible that the proposed value might fail if there was a competing concurrent proposal that won. The client will then have to propose the value again to another run of the Paxos algorithm.

Raft

- All nodes collectively selects leader, and rest of the nods becomes followers.
- Each node can be at any state - Follower, Candidate, Leader.
- Initially, followers will wait for the leader to follow.
- If time ‘t’ expires, then the election process starts.

- Election Phase.
  - Some of the followers volunteer to become leader.
  - Volunteered leader request vote from the followers through *request vote message*.
  - Followers checks the newest version of the request vote message and votes for the suitable candidate.
  - Based on the majority of the votes, the leader candidate is decided.
  - Record entries flow in only one direction, from leader to follower.
  - Leader sends a "heart beat" signal to all followers at regular intervals to maintain the authority.

- **Leader** – Only the server elected as leader can interact with the client. All other servers sync up themselves with the leader. At any point of time, there can be **at most one leader** (possibly 0, which we shall explain later)
- **Follower** – Follower servers sync up their copy of data with that of the leader's after every regular time intervals. When the leader server goes down (due to any reason), one of the followers can contest an election and become the leader.
- **Candidate** – At the time of contesting an election to choose the leader server, the servers can ask other servers for votes. Hence, they are called candidates when they have requested votes. Initially, all servers are in the Candidate state.



multiple server labelled raft visual

Limitations

- Raft is strictly single Leader protocol. Too much traffic can choke the system. Some variants of Paxos algorithm exist that address this bottleneck.
- There are a lot of assumptions considered to be acting, like non-occurrence of Byzantine failures, which sort of reduces the real life applicability.
- Raft is a more specialized approach towards a **subset of problems** which arise in achieving consensus.
- Cheap-paxos(a variant of Paxos), can work even when there is only one node functioning in the server cluster. To generalise, K+1 replicated servers can tolerate shutting down of/ fault in K servers.

- advantages/Features
    - The Raft protocol has been decomposed into smaller subproblems which can be tackled relatively independently for better understanding, implementation, debugging, optimizing performance for a more specific use case
    - The distributed system following the Raft consensus protocol will remain operational even when minority of the servers fail. For example, if we have a 5 server node cluster, if 2 nodes fail, the system can still operate.
    - The leader election mechanism employed in the Raft is so designed that one node will always gain the majority of votes within a maximum of 2 terms.
    - Any node in the cluster can become the leader. So, it has a certain degree of fairness.

How to build Consensus Algorithm - Private Blockchain ?

- Winner Miner mines the candidate block on behalf of all other miners
    - How many blocks winner miner can mine ?
    - How many messages are exchanged ?
        - Are you sending confirmation of vote received to the miners ?
    - How to prove the integrity of the votes ?
    - A Random Miner who receives the transaction can accept the transaction
    - Elite Group Will mine the candidate block
        - Among the Elite group Who will mine the block first ?
    - Rotating Winner Miner. All should get their turn

Transaction Age

Alert Messages

- A Emergency Broadcast Message
  - Warning Messages to the miners.
- Sent to all full Miners as a warning

Network Discovery Nodes

- When a new node is joined to the network
  - It should find its peers.
  - Ask for a copy of the blockchain
- But, how to know the peers - No information about the Peers.
- Blockchain Application should Provide a “Well Known Port”
- It is a publicly available port
- The new node will establish a TCP connection with this well known port.
- Enters into the Blockchain

How to know other peers ?

- With the well known port a node can get connected to nearest node/miner
- Getting connected to other peers is also important
  - To establish P-2-P network
- Seed Nodes - A special node in each blockchain network.
  - Maintains the details of all the miners in the blockchain.
  - From the very first node, get the details of Seed Node
  - Connect to seed node and get the details of all the peers.
  - Connect to all, establish P-2-P network, get started.
  - All the full nodes by default act as a Seed Nodes.

Candidate Block Construction

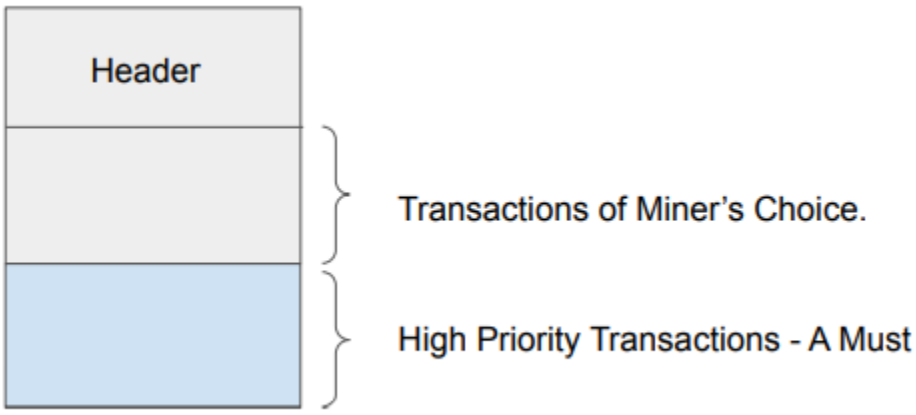
- Transactions from UTXO will be taken and the Candidate Block will be created.
- Which transactions to choose from UTXO
  - Which yields high fees - One end
  - Prioritized transactions - Must
- Every Transaction is prioritized based on its age.
  - Thus, old transactions are given more priority after some age.

Priority of a Transaction

Priority = Sum (Value of input * Input Age) / Transaction Size

- Value of Input = Transaction Amount
- Input Age = The number of blocks that have been added to blockchain since the UTXO was recorded on the blockchain
- Transaction Size = The size of the transaction is measured in bytes.

- Every block divided into two regions the transactions which is select by miners and the priority transactions. This is way all transaction eventually get verified.



Generation Transaction/Coinbase Transaction

- If the blockchain application provides rewards and Transaction Fee to the miners
- How to claim ?
- In the candidate block, very first transaction should be coinbase transaction
- A self transaction which tells how much the miner should get
- Transaction fee + reward (if it is applicable)

- Each Miner selects random transactions from UTXO/Tx Pool in the candidate block
- If block gets accepted all the transactions in the candidate block will be removed from the Tx Pool
- Until the candidate block gets accepted the transactions will not be removed from the UTXO
- Not sure whether the candidate block will gets accepted or not.