CBS COPENHAGEN BUSINESS SCHOOL
HANDELSHØJSKOLEN

Elective Course on Mastering Blockchain: Foundations to Consensus, session-08

Types of Blockchain and Blockchain Trust and Consensus

Raghava Mukkamala

Associate Professor & Director, Centre for Business Data Analytics
Copenhagen Business School, Denmark
Email: rrm.digi@cbs.dk, Centre: https://cbsbda.github.io/

Course Coordinator at SRMIST:

Prof. K. Shantha Kumari

Associate Professor
Data Science and Business Systems Department,
SRM Institute of Science and Technology, India
Shanthak@srmist.edu.in

EQUIS ACCREDITED AACSB ACCREDITED ASSOCIATION AMBA ACCREDITED CEMS PIM

Outline

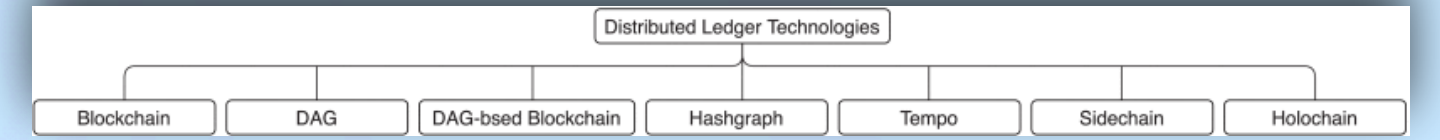
- Types of Distributed Ledgers/Blockchains
- Permissioned versus Permissionless Blockchains
- Blockchain Consensus: Proof-of-Work/Nakamoto Consensus Protocol
- Drawbacks and Vulnerabilities of Nakamoto Consensus

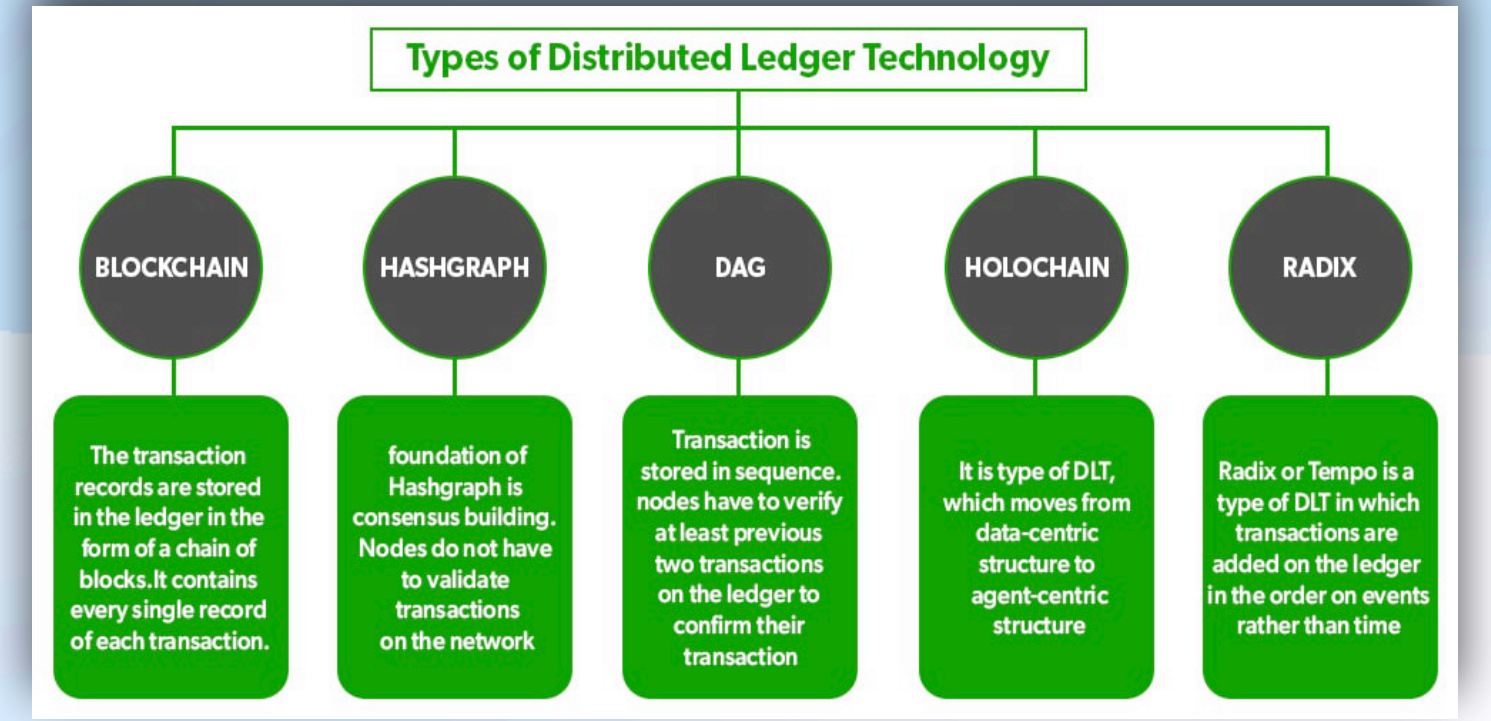
Types of Distributed Ledgers/ Blockchains

Image by Tumisu from Pixabay

CBS

Types of Distributed Ledgers Technologies



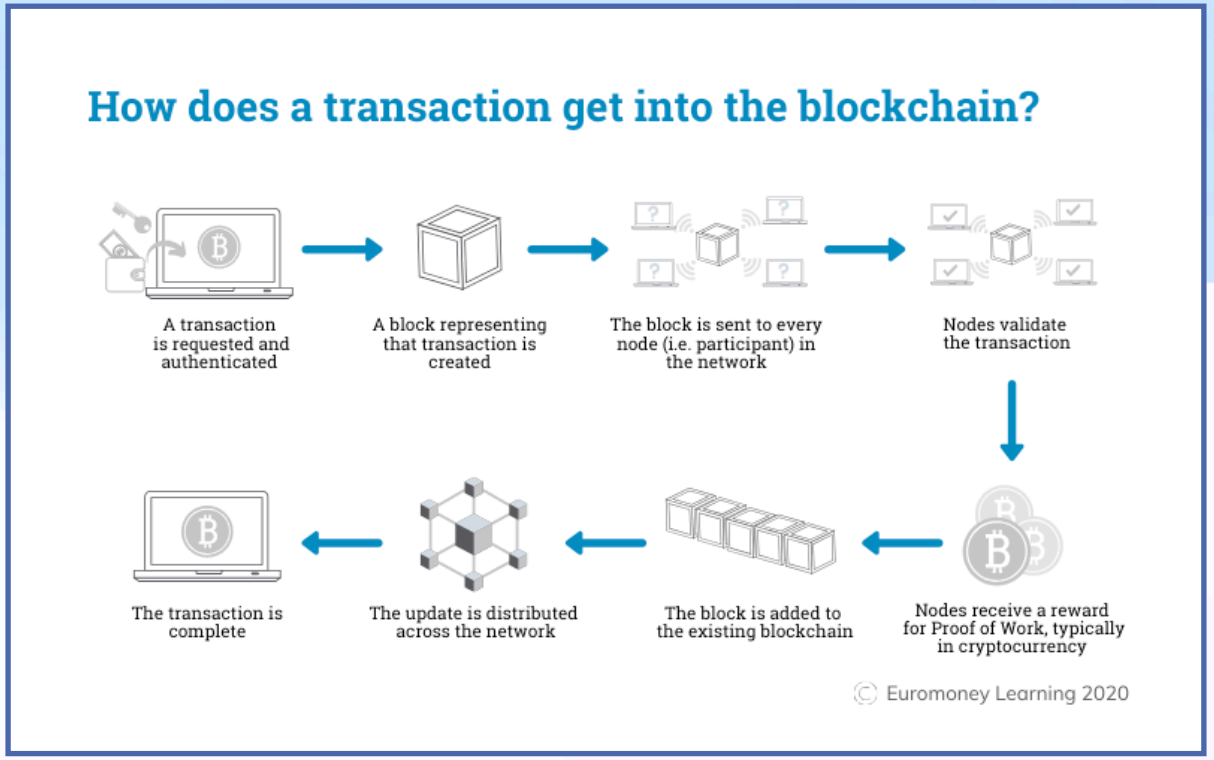


- Hedera is a hashgraph-based, fully open-source, proof-of-stake public network for decentralized applications. It has three primary services: Solidity-based smart contracts, consensus, and token services.
- Holochain is focused on an agent-centric approach, data privacy, and scalability potential, making it well-suited for a wide range of applications, such as social networking.
- Radix DLT is a decentralized finance (DeFi) platform that provides a full stack for Web3 development. It offers a user-friendly Radix Wallet, Scrypto programming language, and Radix Engine for secure dApp development.

Lashkari, B., & Musilek, P. (2021). A comprehensive review of blockchain consensus mechanisms. *IEEE Access*, *9*, 43620-43652.
https://www.geeksforgeeks.org/blockchain-and-distributed-ledger-technology-dlt/

Blockchain

- Immutability: remains irreversible since any transaction cannot be altered, deleted, or reversed unless more than 51% of the nodes agree with the modification.
- Security: Exceptional security measures. Public and private keys, cryptographic hashing functions, and Merkle tree endure integrity on top of validation procedures that hinder manipulation.
- We often consider a practical blockchain network partially synchronous, just like most distributed networks overlaying on the Internet

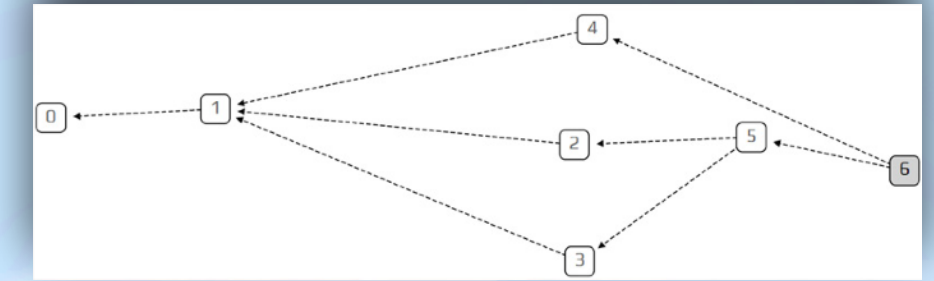


https://www.euromoney.com/learning/blockchain-explained/how-transactions-get-into-the-blockchain

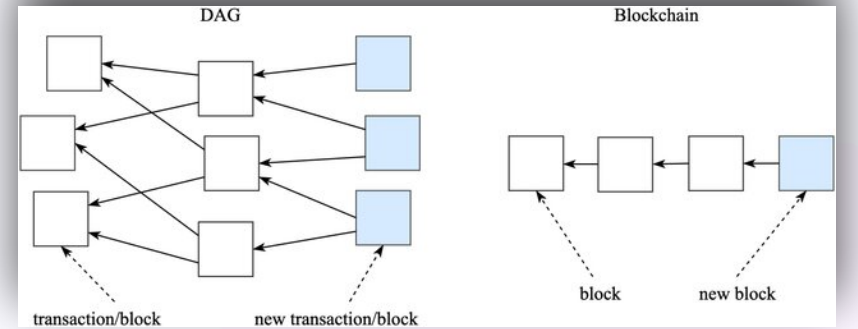
Lashkari, B., & Musilek, P. (2021). A comprehensive review of blockchain consensus mechanisms. *IEEE Access*, *9*, 43620-43652.

Directed Acyclic Graph (DAG)

- Directed Acyclic Graph (DAG) is a variant of DLT that has been proposed as an alternative to blockchain, e.g., IoTA
- Tangle, the underlying data structure for IOTA
- Rather than transactions created by users being incorporated into blocks by miners, users function as both the miners and the creators of transactions.
- Creators of transactions must verify two previous transactions within the network, and each transaction requires a small PoW computation on behalf of the user.



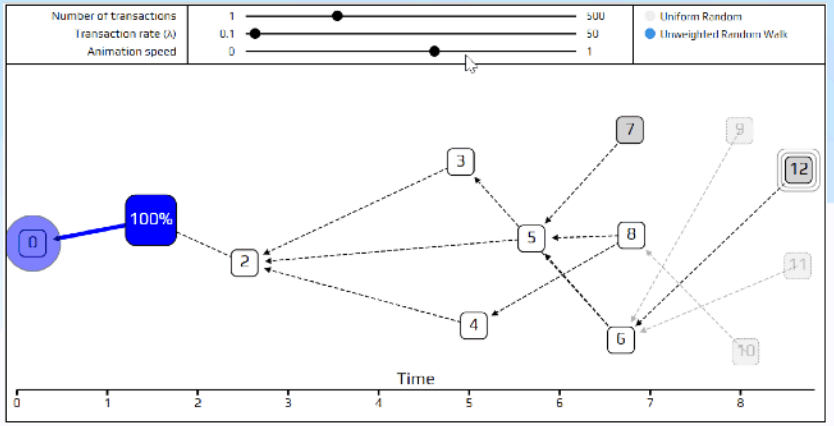
https://blockonomi.com/iota-tangle/



Fan, C., Ghaemi, S., Khazaei, H., Chen, Y., & Musilek, P. (2021). Performance analysis of the IOTA DAG-based distributed ledger. *ACM Transactions on Modeling and Performance Evaluation of Computing Systems*, *6*(3), 1-20.

Directed Acyclic Graph (DAG)

- As a decentralized channel, DAG enables the participants to make instant micro- or even nano-transactions by incorporating fee-free transactions.
- Resilience: DAG is quantum-resistant, and its underlying distributed ledger is less susceptible to quantum computers with higher-level computing properties using Winternitz's one-time signature scheme.
- The cumulative weight property of the Tangle effectively mitigates the double spending problem.
- Transaction Validation: users/miners play a crucial role in the DAG system by not approving invalid transactions. This ensures that all subsequent transactions that approve invalid ones are also considered invalid, maintaining the integrity of the network.



https://blockonomi.com/iota-tangle/

Lashkari, B., & Musilek, P. (2021). A comprehensive review of blockchain consensus mechanisms. *IEEE Access*, *9*, 43620-43652.

DAG-based Blockchain

- Ever since blockchain, many investigations have been performed to address its deficiencies, such as its limitations throughout.
- DAG-based blockchain has been proposed as the next generation of blockchain. It inherits the significant features of both DAG and blockchain.
- The DAG-based blockchain introduces a novel verification procedure. Before it can be added to the blockchain, every new transaction must be validated by at least two earlier transactions.
- This efficient process, inherited from DAG, eliminates the need for miners in transaction authentication, thereby speeding up the entire system.
- The gossip protocol for communication among nodes enhances the linear structure of traditional blockchain.
- Several unknowns and no actual implementations

Lashkari, B., & Musilek, P. (2021). A comprehensive review of blockchain consensus mechanisms. *IEEE Access*, *9*, 43620-43652.

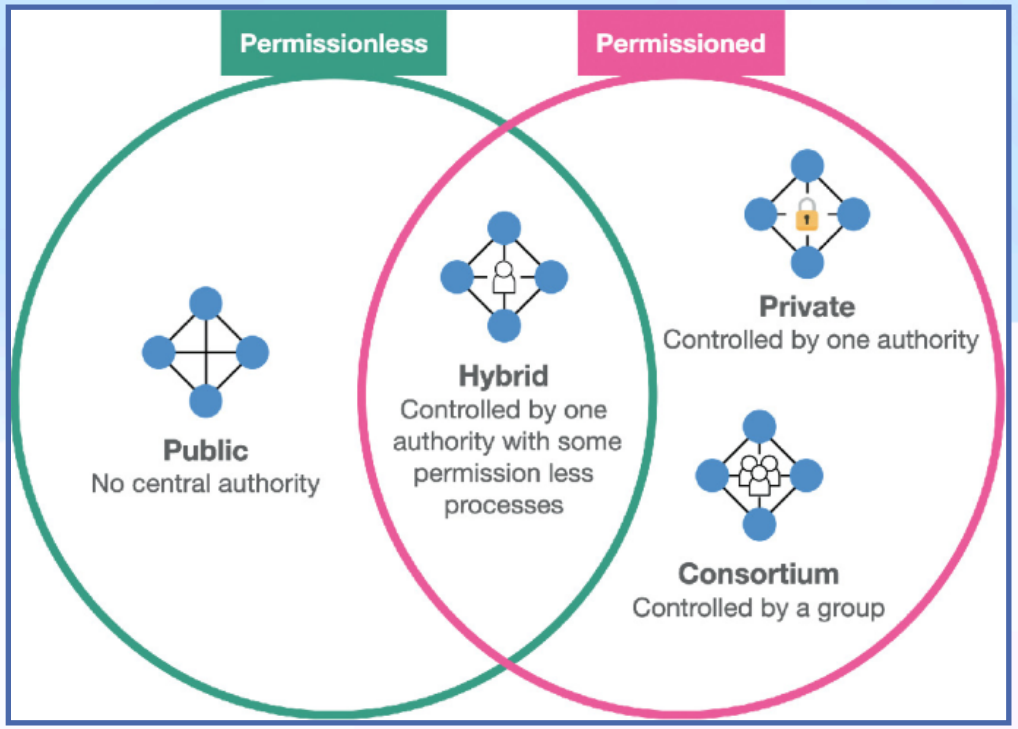
Blockchain Classification

Image by Tumisu from Pixabay

CBS

Blockchain Classification

- Depending on the control of network participation, blockchain networks generally fall into two categories: permissionless and permissioned.
- A permissionless blockchain allows for free to join and leave without any authorization, as long as the node holds a valid pseudonym (account address) and is able to send, receive, and validate transactions and blocks by common rules.
- Permissionless blockchain is also known as public blockchain for there is usually one such blockchain network instance on a global scale which is subject to public governance.
- Specifically, anyone can participate in blockchain consensus, though one's voting power is typically proportional to its possession of network resources, such as computation power, token wealth, storage space, etc.



Yi, X., Yang, X., Kelarev, A., Lam, K.Y., Tari, Z. (2022). Bitcoin, Ethereum, Smart Contracts and Blockchain Types. In: Blockchain Foundations and Applications. SpringerBriefs in Applied Sciences and Technology. Springer,

Xiao, Y., Zhang, N., Lou, W., & Hou, Y. T. (2020). A survey of distributed consensus protocols for blockchain networks. *IEEE Communications Surveys & Tutorials*, *22*(2), 1432-1465.

Blockchain Classification

- The operational environment of permissionless blockchain is often assumed as large-scale and zero-trust, which often cautions the community against increasing transaction processing capacity or using more efficient consensus schemes
- A **permissioned** blockchain requires participants to be authorized first and then participate in network operations with revealed identity.
- The network governance and consensus body can be either the subsidiaries of a single private entity or a consortium of entities
- Compared to permissionless blockchain, the identity-revealing requirement and more effective network governance of permissioned blockchain make it ideal for internal or multi-party business applications.
- Meanwhile, the limited size of a permissioned blockchain's consensus body allows for the deployment of more efficient consensus protocols that achieve higher transaction capacity

TABLE I
A COMPARISON OF PERMISSIONLESS AND PERMISSIONED BLOCKCHAIN

	Permissionless blockchain	**Permissioned blockchain**
Governance	Public	Private / Consortium
Participation	Free join and leave	Authorized
Node identity	Pseudonymous	Revealed
Transparency	Open	Closed / Open
Network size	Large (thousands or more)	Small (tens∼hundreds)
Network connectivity	Low	High (oft. fully-connected)
Network synchrony	Asynchronous / partially synchronous	Partially synchronous / synchronous
Transaction capacity (tps)	Low (oft. sub-ten∼tens)	High (oft. thousands)
Application examples	Cryptocurrency, smart contract, public record, DApp	Inter-bank clearing, business contract, supply chain

Yi, X., Yang, X., Kelarev, A., Lam, K.Y., Tari, Z. (2022). Bitcoin, Ethereum, Smart Contracts and Blockchain Types. In: Blockchain Foundations and Applications. SpringerBriefs in Applied Sciences and Technology. Springer,

Xiao, Y., Zhang, N., Lou, W., & Hou, Y. T. (2020). A survey of distributed consensus protocols for blockchain networks. *IEEE Communications Surveys & Tutorials*, *22*(2), 1432-1465.

Distributed Consensus Fundamentals

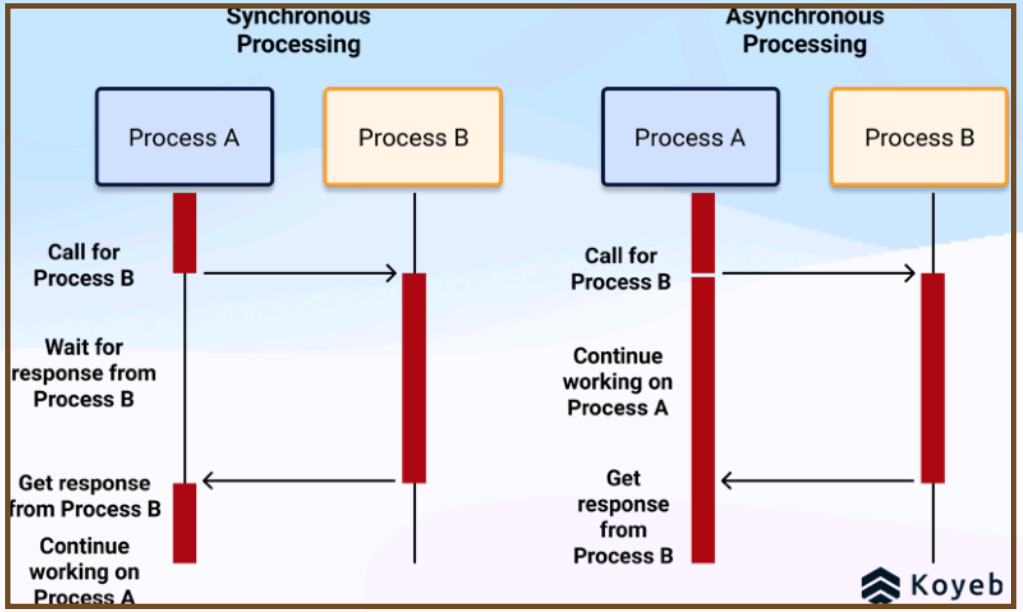
Image by rawpixel.com on Freepik

Xiao, Y., Zhang, N., Lou, W., & Hou, Y. T. (2020). A survey of distributed consensus protocols for blockchain networks. *IEEE Communications Surveys & Tutorials*, *22*(2), 1432-1465.

CBS

Network Synchrony

- Network synchrony defines the level of coordination among all processes. Three levels: synchronous, partially synchronous, and asynchronous.
- In a synchronous network, operations of processes are coordinated in rounds with clear time constraints. E.g. Http client-server
- In a partially synchronous network, operations of processes are loosely coordinated in a way that message delivery is guaranteed but with an uncertain amount of delays
- In an asynchronous network, the operations of processes are hardly coordinated. There is no delay guarantee on a message except for its eventual delivery. And the coordination of processes is solely driven by the message delivery events.

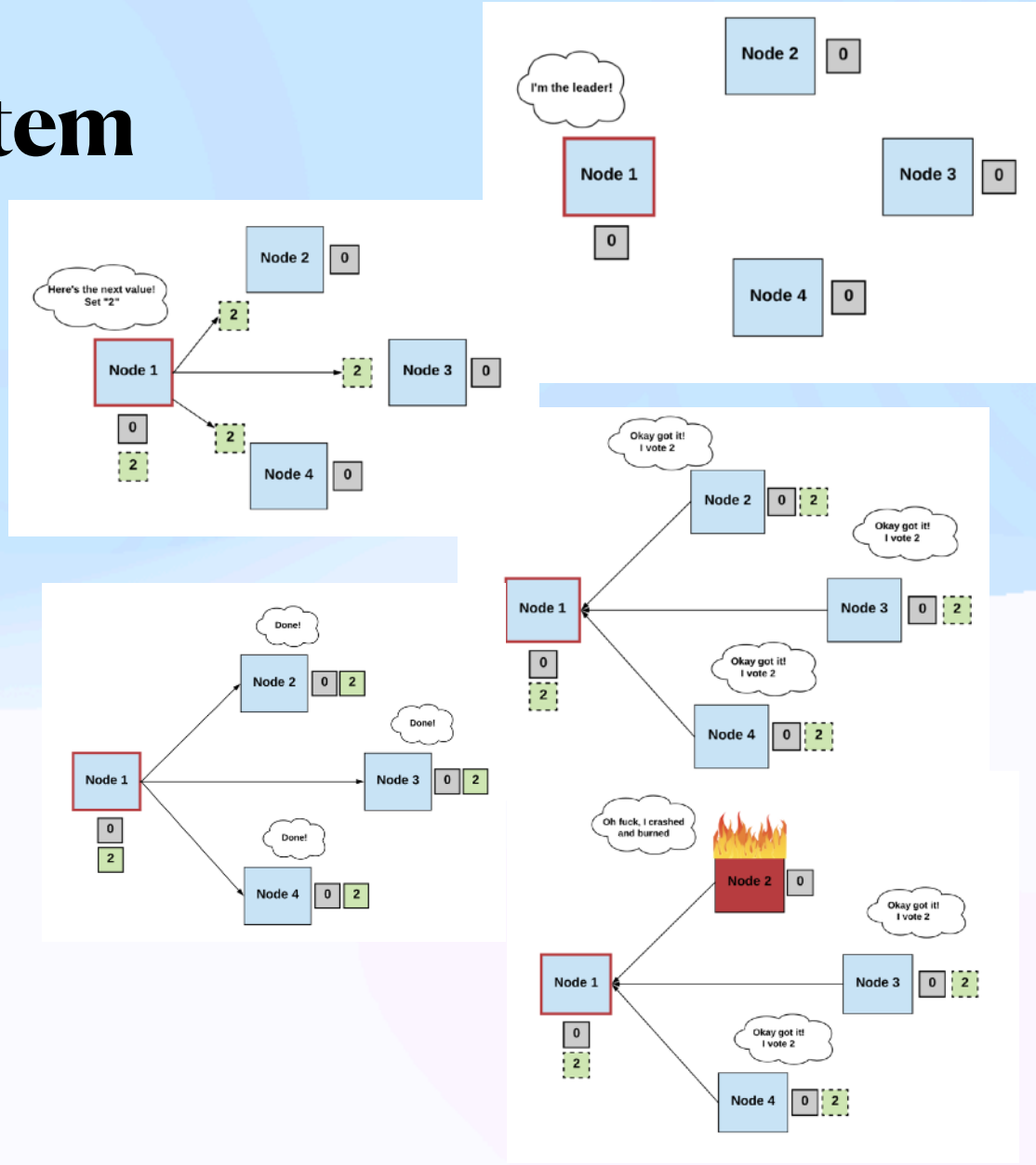


https://www.koyeb.com/blog/introduction-to-synchronous-and-asynchronous-processing

Xiao, Y., Zhang, N., Lou, W., & Hou, Y. T. (2020). A survey of distributed consensus protocols for blockchain networks. *IEEE Communications Surveys & Tutorials*, *22*(2), 1432-1465.

Consensus in a Distributed System

- Achieving consensus in a distributed system is challenging.
- Consensus algorithms should be resilient to failures of nodes, partitioning of the network, message delays, and messages reaching out-of-order and corrupted messages.
- They also have to deal with selfish and deliberately malicious nodes. Several algorithms are proposed in the research literature to solve this.
- A consensus protocol has three key properties based upon which its applicability and efficacy can be determined: Safety, Liveness, and Fault Tolerance



https://www.preethikasireddy.com/post/lets-take-a-crack-at-understanding-distributed-consensus

Xiao, Y., Zhang, N., Lou, W., & Hou, Y. T. (2020). A survey of distributed consensus protocols for blockchain networks. *IEEE Communications Surveys & Tutorials*, *22*(2), 1432-1465.

Baliga, A. (2017). Understanding blockchain consensus models. Persistent, 4(1), 14. (White paper)

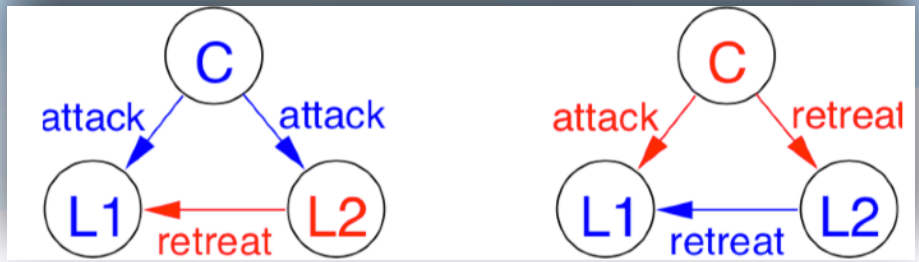
Types of Faults in Consensus Protocol

- Two types of faults in distributed systems.
- 1) Fail-stop faults are caused by hardware or software crashes. They deal with node failures that cause nodes to stop participating in the consensus protocol.
- 2) Byzantine faults cause nodes to behave erratically/maliciously. Leslie Lamport identified and characterized this category of faults as the Byzantine General's Problem.
    - A Byzantine node can lie, provide contradictory responses, or completely mislead other nodes involved in the consensus protocol to sabotage achieving consensus.
- A consensus protocol must operate correctly and reach consensus in the presence of Byzantine nodes as long as the number of Byzantine nodes is limited.

Xiao, Y., Zhang, N., Lou, W., & Hou, Y. T. (2020). A survey of distributed consensus protocols for blockchain networks. *IEEE Communications Surveys & Tutorials*, *22*(2), 1432-1465.
Baliga, A. (2017). Understanding blockchain consensus models. Persistent, 4(1), 14. (White paper)

The Byzantine Generals' Problem

- A group of generals, each commandeering a part of the Byzantine army, surrounded an enemy city.
- All the generals must agree on a battle plan to attack the city and communicate via messengers only.
- The messengers might be captured by the enemy, and the message might never reach the other general.
- The difficulty is that one or more generals might be traitors interested in sabotaging the battle plan.
- To this end, they might send false messages, distort messages, or not send any messages at all.
- All loyal generals will act according to the plan. A few traitors should not cause loyal generals to adopt a bad plan.

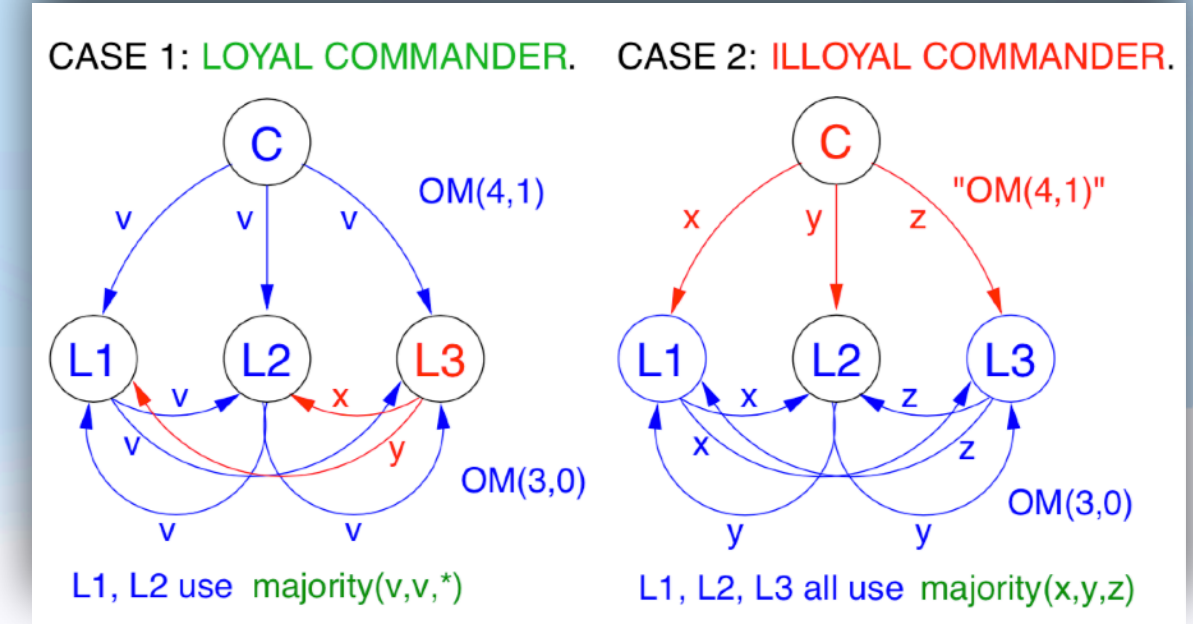


Baliga, A. (2017). Understanding blockchain consensus models. Persistent, 4(1), 14. (White paper)
Lamport, L., Shostak, R., & Pease, M. (2019). The Byzantine generals problem. In Concurrency: the works of leslie lamport (pp. 203-226).
George Coulouris, Jean Dollimore, Tim Kindberg, Gordon Blair. Distributed Systems: Concepts and Design Fifth Edition

Byzantine Fault Tolerant Consensus

- *Termination:* Every non-faulty process decides an output.
- *Agreement:* Every non-faulty process eventually decides the same output $\hat{y}$.
- *Validity:* If every process begins with the same input $\hat{x}$, then $\hat{y} = \hat{x}$.
- *Integrity:* Every non-faulty process' decision and the consensus value $\hat{y}$ must have been proposed by some non-faulty process.

The four requirements provide a general target for distributed consensus protocols. For any consensus protocol to attain these BFT requirements, the underlying distributed network should satisfy the following condition: $N \geq 3f + 1$ where f is the number of Byzantine processes. This fundamental result was first proved by Pease *et al.* [30] in 1980 and later adapted to the BFT consensus framework. The proof involves induction from $N = 3$ and partitioning all processes into three equal-sized groups, with one containing the faulty



- We call a consensus protocol Byzantine fault tolerant (BFT) if it can tolerate a certain amount of crash or Byzantine process failures while keeping normal functioning.

Baliga, A. (2017). Understanding blockchain consensus models. Persistent, 4(1), 14. (White paper)

Lamport, L., Shostak, R., & Pease, M. (2019). The Byzantine generals problem. In Concurrency: the works of leslie lamport (pp. 203-226).

George Coulouris, Jean Dollimore, Tim Kindberg, Gordon Blair. Distributed Systems: Concepts and Design Fifth Edition

Practical Byzantine Fault Tolerance (PBFT)

- Tolerating Byzantine faults, increases the complexity of the consensus protocol by adding several extra layers of messaging into the system.
- VR, Paxos can not tolerate Byzantine faults
- PBFT was the first practical approach that allowed for Byzantine fault-tolerant with low overhead.
- PBFT consists of three sub-protocols: 1) Normal operation, 2) Checkpoint, 3) View-change.
- PBFT can tolerate f crashed nodes when $N \geq 2f + 1$,
- PBFT can tolerate f byzantine nodes when $N \geq 3f + 1$,
- Well-known proposals include Quorum/Update (QU), Hybrid Quorum (HQ), Zyzzyva etc.

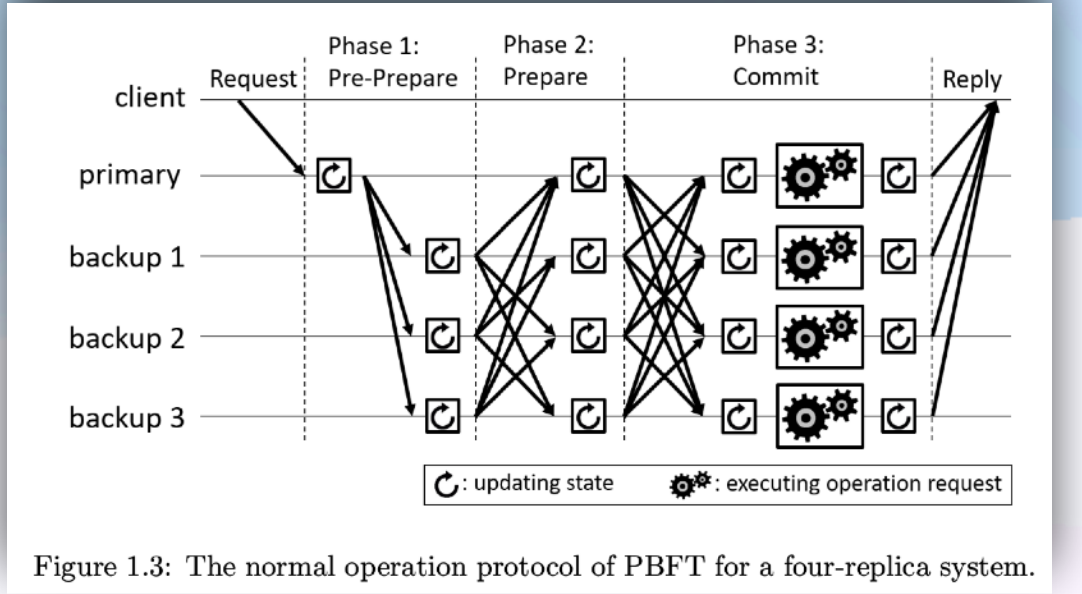


Figure 1.3: The normal operation protocol of PBFT for a four-replica system.

Baliga, A. (2017). Understanding blockchain consensus models. Persistent, 4(1), 14. (White paper)

Lamport, L., Shostak, R., & Pease, M. (2019). The Byzantine generals problem. In Concurrency: the works of leslie lamport (pp. 203-226).

George Coulouris, Jean Dollimore, Tim Kindberg, Gordon Blair. Distributed Systems: Concepts and Design Fifth Edition

Gossip Protocol

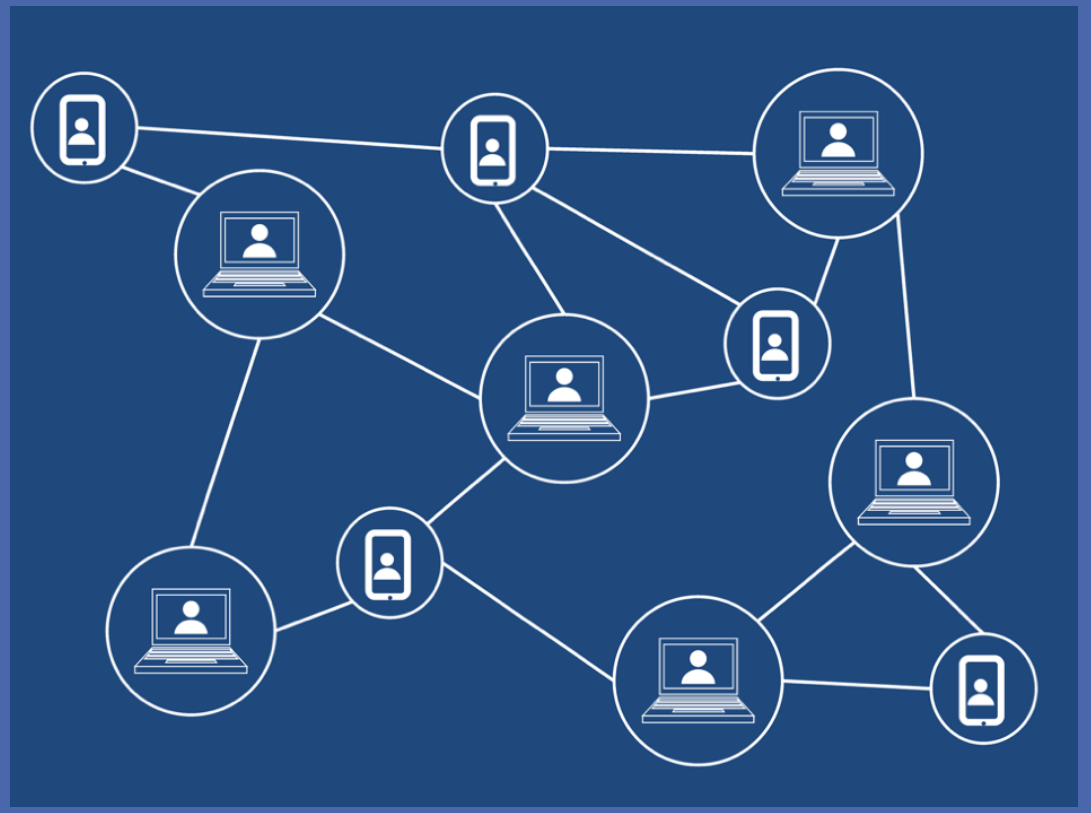


Image by Maicon Fonseca Zanco from Pixabay

CBS

Gossip Protocol in Blockchain Network

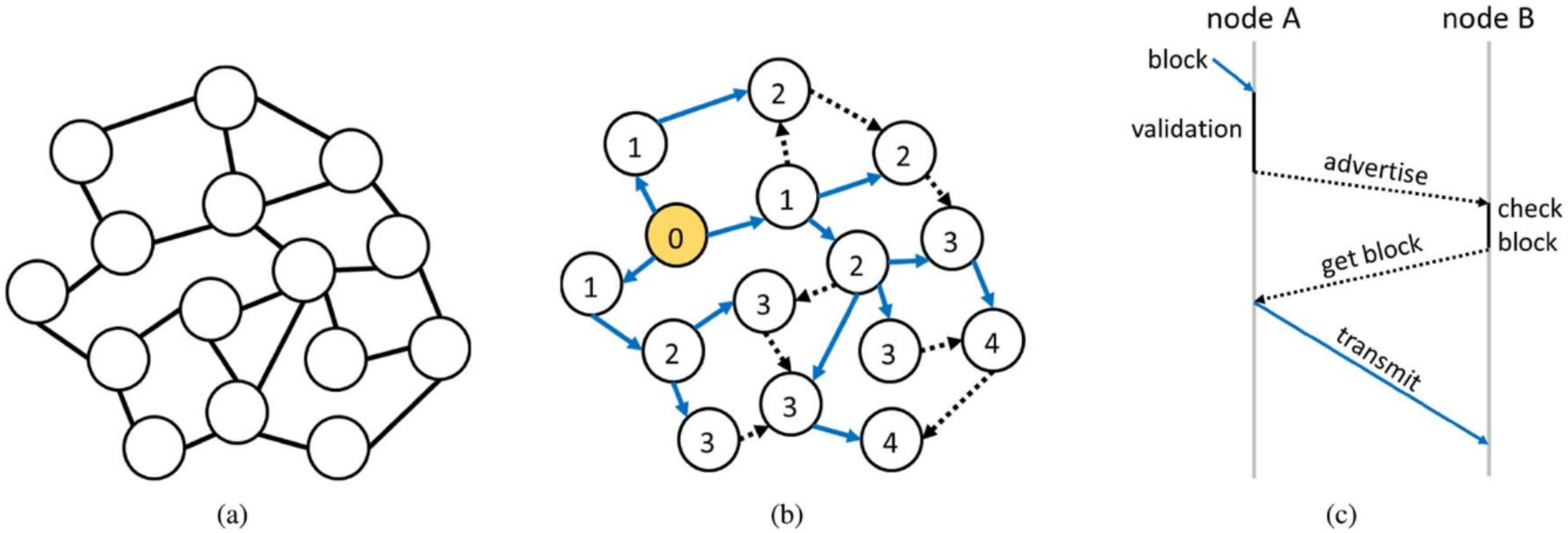


Fig. 5. A toy example of block propagation in the Bitcoin network. (a) The P2P network structure, an undirected graph. (b) The gossiping process. A solid blue arrow represents one-hop block propagation (advertise→get block→transmit), while a dotted black arrow represents only advertise. Number denotes the gossiping hop (0 for the block producer). (c) Block propagation in one hop.

Xiao, Y., Zhang, N., Lou, W., & Hou, Y. T. (2020). A survey of distributed consensus protocols for blockchain networks. *IEEE Communications Surveys & Tutorials*, *22*(2), 1432-1465.

Gossip Protocol in Blockchain Network

- In blockchain networks, block or transaction messages are propagated across the P2P network through gossiping.
- For each new block received and validated, a node advertises it to peers, who will request for this block if it extends their local blockchain.
- The gossiping process continues until every node in the network has this block.
- **Probabilistic finality**: For any honest node, every new block is either discarded or accepted into its local blockchain.
- An accepted block may still be discarded but with an exponentially diminishing probability as the blockchain continues to grow. For example, in Bitcoin, after a block got 6 successor blocks, the chance of discarding a block is very, very small.

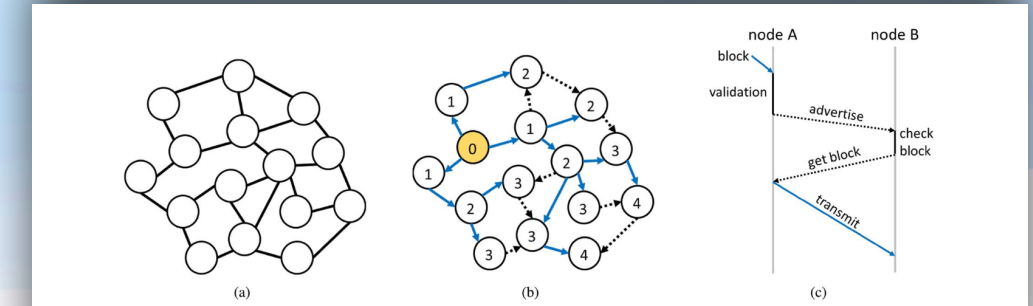


Fig. 5. A toy example of block propagation in the Bitcoin network. (a) The P2P network structure, an undirected graph. (b) The gossiping process. A solid blue arrow represents one-hop block propagation (advertise→get block→transmit), while a dotted black arrow represents only advertise. Number denotes the gossiping hop (0 for the block producer). (c) Block propagation in one hop.

Xiao, Y., Zhang, N., Lou, W., & Hou, Y. T. (2020). A survey of distributed consensus protocols for blockchain networks. *IEEE Communications Surveys & Tutorials*, *22*(2), 1432-1465.

Proof-of-Work/ Nakamoto Consensus Protocol

https://alexey-shepelev.medium.com/hierarchical-key-generation-fc27560f786

CBS

Components of Blockchain Consensus Protocol

The following are the five key components of a blockchain consensus protocol

- Block Proposal: Generating blocks and attaching generation proofs.
- Information Propagation: Disseminating blocks and transactions across the network.
- Block Validation: Checking blocks for generation proofs and transaction validity.
- Block Finalization: Reaching agreement on the acceptance of validated blocks.
- Incentive Mechanism: Promoting honest participation and creating network tokens.

Baliga, A. (2017). Understanding blockchain consensus models. Persistent, 4(1), 14. (White paper)
Lamport, L., Shostak, R., & Pease, M. (2019). The Byzantine generals problem. In Concurrency: the works of leslie lamport (pp. 203-226).
George Coulouris, Jean Dollimore, Tim Kindberg, Gordon Blair. Distributed Systems: Concepts and Design Fifth Edition

Nakamoto Consensus Protocol

- The Nakamoto consensus protocol is the key innovation behind Bitcoin and inspired many other cryptocurrency systems, such as Ethereum and Litecoin
- The Nakamoto consensus protocol is summarized by the following rules:
- Block Proposal: Proof of Work (PoW)
    - Block generation requires finding a preimage to a hash result that satisfies a difficulty target, which is dynamically adjusted to maintain an average block generation interval.
- Information Propagation: Gossiping Rule
    - Any newly received or locally generated transaction or block should be immediately advertised and broadcast to peers.

Xiao, Y., Zhang, N., Lou, W., & Hou, Y. T. (2020). A survey of distributed consensus protocols for blockchain networks. *IEEE Communications Surveys & Tutorials*, *22*(2), 1432-1465.
Baliga, A. (2017). Understanding blockchain consensus models. Persistent, 4(1), 14. (White paper)

Nakamoto Consensus Protocol

- Block Validation: Validation Rule
  - A block or transaction must be validated before being broadcast to peers or appended to the blockchain. The validation includes a double-spending check on transactions and proof-of-work validity check on the block header.
- Block Finalization: Longest-Chain Rule
  - The longest chain represents network consensus, which should be accepted by any node that sees it. Mining should always extend the longest chain.
- Incentive Mechanism: Block Rewards and Transaction Fees
  - Generator of a block can claim a certain amount of new tokens plus fees collected from all enclosed transactions in the form of a coinbase transaction to itself.

Xiao, Y., Zhang, N., Lou, W., & Hou, Y. T. (2020). A survey of distributed consensus protocols for blockchain networks. *IEEE Communications Surveys & Tutorials*, *22*(2), 1432-1465.
Baliga, A. (2017). Understanding blockchain consensus models. Persistent, 4(1), 14. (White paper)

Nakamoto Consensus Algorithm

Algorithm 3: Nakamoto Consensus Protocol General Procedure

```
  /* Joining network                                   */
1 Join the network by connecting to known peers;
2 Start BlockGen();
  /* Main loop                                         */
3 while running do
4     if BlockGen() returns block then
5         Write block into blockchain;
6         Reset BlockGen() to the current blockchain;
          /* Gossiping rule                            */
7         Broadcast block to peers;
8     end
      /* Longest-chain/validation rule                 */
9     if block received & is valid & extends the longest
        chain then
10        Write block into blockchain;
11        Reset BlockGen() to the current blockchain;
12        Relay block to peers;
13    end
14 end
  /* PoW-based block generation                        */
15 Function BlockGen():
16    Pack up transactions (including coinbase);
17    Prepare a block header context 𝒞 containing the
        transaction Merkle tree root, hash of the last block
        in the longest chain, timestamp, and other essential
        information reflecting blockchain status;
      /* PoW hashing puzzle                            */
18    Find a nonce that satisfies the following condition:

                    Hash(𝒞|nonce) < target

        wherein more preceding zero bits in target indicate
        a higher mining difficulty;
19    return new block;
20 end
```

Xiao, Y., Zhang, N., Lou, W., & Hou, Y. T. (2020). A survey of distributed consensus protocols for blockchain networks. *IEEE Communications Surveys & Tutorials*, *22*(2), 1432-1465.
Baliga, A. (2017). Understanding blockchain consensus models. Persistent, 4(1), 14. (White paper)

Fork Resolution in Nakamoto consensus

Fork resolution: Ideally, the 10-minute block interval should be enough to ensure the thorough propagation of a new block so that no block of the same height is proposed. However due to the delay during the message propagation and the probabilistic nature of the hashing game, the possibility of two blocks of the same height being propagated concurrently in the network can not be ignored. This situation is called a "fork", detectable by any node. Correspondingly, the longest-chain rule provides the criterion for fork resolution. Assume a miner receives two valid blocks B_1^k, B_2^k of the same block height k sequentially, then a fork is detected by this miner. It chooses B_1^k (the first arrived) to continue and may encounter the following cases:

- *Case 1:* If receiving or successfully generating a block B_1^{k+1} confirming B_1^k, accept B_1^k, B_1^{k+1} and orphans B_2^k.
- *Case 2:* If receiving a block B_2^{k+1} confirming B_2^k, switch to B_2^{k+1} and accept B_2^k, then orphans B_1^k.
- *Case 3:* If simultaneously receiving two blocks B_1^{k+1} and B_2^{k+2} confirming respectively B_1^k and B_2^k, choose one to follow and continue until case 1 or 2 is met.

Wherein to "orphan" a block means to deny it into the main chain. Because of the randomized nature of PoW mining, the likelihood of encountering case 3 drops exponentially as time elapses, reflecting the probabilistic finality of Nakamoto consensus.

Xiao, Y., Zhang, N., Lou, W., & Hou, Y. T. (2020). A survey of distributed consensus protocols for blockchain networks. *IEEE Communications Surveys & Tutorials*, *22*(2), 1432-1465.
Baliga, A. (2017). Understanding blockchain consensus models. Persistent, 4(1), 14. (White paper)

Fork Resolution in Nakamoto consensus

Bitcoin Fork Resolution Explained: Everything You Need to Know! : https://www.youtube.com/watch?v=JIwAgWfX4Pw

Security Analysis of Nakamoto Consensus Protocol

- The fault tolerance of Nakamoto consensus is characterized by the percentage of adversarial hashing power the system can tolerate.

- As long as less than 50% of total hashing power is maliciously controlled, the blocks produced by honest miners are timely propagated, and the main chain by the honest majority can eventually outgrow any malicious branch.

- From the perspective of classical distributed consensus, Nakamoto consensus cleverly circumvents the fundamental 1/3 BFT bound by adopting probabilistic finality.

Xiao, Y., Zhang, N., Lou, W., & Hou, Y. T. (2020). A survey of distributed consensus protocols for blockchain networks. *IEEE Communications Surveys & Tutorials*, *22*(2), 1432-1465.
Baliga, A. (2017). Understanding blockchain consensus models. Persistent, 4(1), 14. (White paper)

Drawbacks and Vulnerabilities of Nakamoto Consensus

Image by rawpixel.com on Freepik

Slides based on Xiao, Y., Zhang, N., Lou, W., & Hou, Y. T. (2020). A survey of distributed consensus protocols for blockchain networks. *IEEE Communications Surveys & Tutorials*, *22*(2), 1432-1465.

CBS

Tight Tradeoff Between Performance and Security

- The Nakamoto consensus is widely criticized for its low transaction throughput. For instance, Bitcoin can process up to 7 TPS meanwhile, the VISA payment network processes 10 000 - 20 000 TPS on average.
- The limited performance of Bitcoin follows from the security implication of its probabilistic finality and two protocol parameters: block interval and block size.
- As discussed, the 10-minute block interval ensures that every new block is sufficiently propagated before a new block is mined.
- Reducing the block interval increases the transaction capacity but leaves new blocks insufficiently propagated and caused more fork incidents, undermining the security of the main chain.

Xiao, Y., Zhang, N., Lou, W., & Hou, Y. T. (2020). A survey of distributed consensus protocols for blockchain networks. *IEEE Communications Surveys & Tutorials*, *22*(2), 1432-1465.
Baliga, A. (2017). Understanding blockchain consensus models. Persistent, 4(1), 14. (White paper)

Tight Tradeoff Between Performance and Security

- Note that although any fork can be resolved given enough time, the higher the fork rate, the larger the amount of honest mining power is wasted.

- On the other hand, increasing the block size (currently 1 MB) has the same effect since larger block sizes lead to higher block transmission delays and insufficient propagation.

- According to the measurement and analysis by Croman et al. [3] in 2016, given the current 10-minute block interval, the maximum block size should not exceed 4 MB, which yields a peak throughput of 27 TPS.

Xiao, Y., Zhang, N., Lou, W., & Hou, Y. T. (2020). A survey of distributed consensus protocols for blockchain networks. *IEEE Communications Surveys & Tutorials*, *22*(2), 1432-1465.
Baliga, A. (2017). Understanding blockchain consensus models. Persistent, 4(1), 14. (White paper)

Energy Inefficiency

- As of November 2019, an average Bitcoin transaction consumes 431 KWh of electricity, which can power 21 U.S. households for a day.
- The Digiconomist's Bitcoin Energy Consumption Index estimated in 2022 that one bitcoin transaction takes 1,449 kWh to complete, or the equivalent of approximately 50 days of power for the average US household.
- This enormous energy consumption is directly caused by the PoW-based block proposing a scheme of Nakamoto consensus.
- In response, the blockchain community has developed various block-proposing schemes, such as proof of stake (PoS), proof of authority (PoA), and proof of elapsed time (PoET), as energy-saving alternatives to PoW.

Xiao, Y., Zhang, N., Lou, W., & Hou, Y. T. (2020). A survey of distributed consensus protocols for blockchain networks. *IEEE Communications Surveys & Tutorials*, *22*(2), 1432-1465.
Baliga, A. (2017). Understanding blockchain consensus models. Persistent, 4(1), 14. (White paper)

Thank you!

Raghava Mukkamala

rrm.digi@cbs.dk

https://www.cbs.dk/staff/rrmdigi

https://raghavamukkamala.github.io/

https://cbsbda.github.io/