CONSENSUS ALGORITHMS IN BLOCKCHAIN: A COMPARATIVE STUDY

Ramesh Chandra Tripathi*

* Faculty of Engineering, Teerthanker Mahaveer University,
Moradabad, Uttar Pradesh, INDIA
Email id: tripathi.computers@tmu.ac.in

DOI: 10.5958/2249-7137.2021.02477.0

ABSTRACT

Blockchain is a distributed ledger that has garnered widespread interest in a variety of fields. Many sectors have begun to use blockchain technology in their products and services. To comprehend blockchain's effect and applicability to different applications, it is necessary to grasp its main components, functional features, and architecture. Bitcoin, a cryptocurrency, is the most well-known use of blockchain. Because a blockchain network is a distributed ledger, it requires a consensus mechanism among peer nodes to guarantee that it functions properly. In the literature, a variety of consensus algorithms have been suggested, each with its unique set of performance and security features. Every application's needs cannot be met by a single consensus method. It is critical to evaluate the various consensus algorithms on a technical level in order to identify their strengths, limitations, and applications. We have discussed metrics that are relevant to blockchain consensus performance and security. In terms of these factors, the consensus methods are evaluated and contrasted. A research need is identified in terms of developing an efficient consensus method and assessing current techniques. This review paper will serve as a reference for developers and academics who are evaluating and designing a consensus method.

KEYWORDS: *Bitcoin, Block chain, Consensus, Distributed ledger, Security.*

1. INTRODUCTION

Blockchain technology was created to address the risks and inefficiencies associated with commercial transactions. It has changed the way sectors and companies are structured. A distributed ledger shared across the nodes of a business network is known as blockchain **[1], [2].** Transactional data is kept in blocks that are connected together to create a chain. It allows a company's assets to be tracked more easily. Any valuable item, physical or intangible, may be considered an asset. Cars, land, houses, and cash are examples of physical assets. Intellectual property, such as copyrights, properties, patents, and trademarks, are examples of intangible assets. The block size determines how transactions are clustered together in a block. A timestamp is used by Block to validate the order of transactions. Each block keeps a hash of the preceding block, which aids in backtracking while verifying new blocks. Because no one can insert a malicious or corrupted block between legitimate blocks, the whole blockchain is more secure. Immutability refers to the fact that no existing block may be changed. When a transaction happens, each node updates the ledger. Each complete node keeps a copy of the whole ledger (blockchain). The blockchain architecture is shown in Figure 1.

When compared to conventional distributed databases, blockchain offers major benefits such as lower costs and faster processing times, no dependence on a third party, and asset protection. Each node in a distributed database system maintains its own ledger of records and depends on an intermediate to complete transactions. This approach is wasteful due to delays in agreement implementation and extra fees imposed by middlemen. This architecture is susceptible to security threats since a central middleman might be hacked, putting the whole company at risk. Blockchain transactions are both verifiable and safe. These problems have been addressed by the blockchain design. It validates information via a consensus method among network nodes, obviating the necessity for intermediaries. The consensus mechanism is a fundamental idea in blockchain that guarantees a tamper-proof ecosystem in which all nodes agree on just one version of the truth. In a decentralized network, all nodes must agree on the current state of the blockchain. An attacker will have a hard time inserting a modified block into the blockchain as a result of this. When it comes to establishing a blockchain system, choosing the appropriate consensus algorithm is critical. Furthermore, while developing a consensus method, certain important factors must be taken into account.

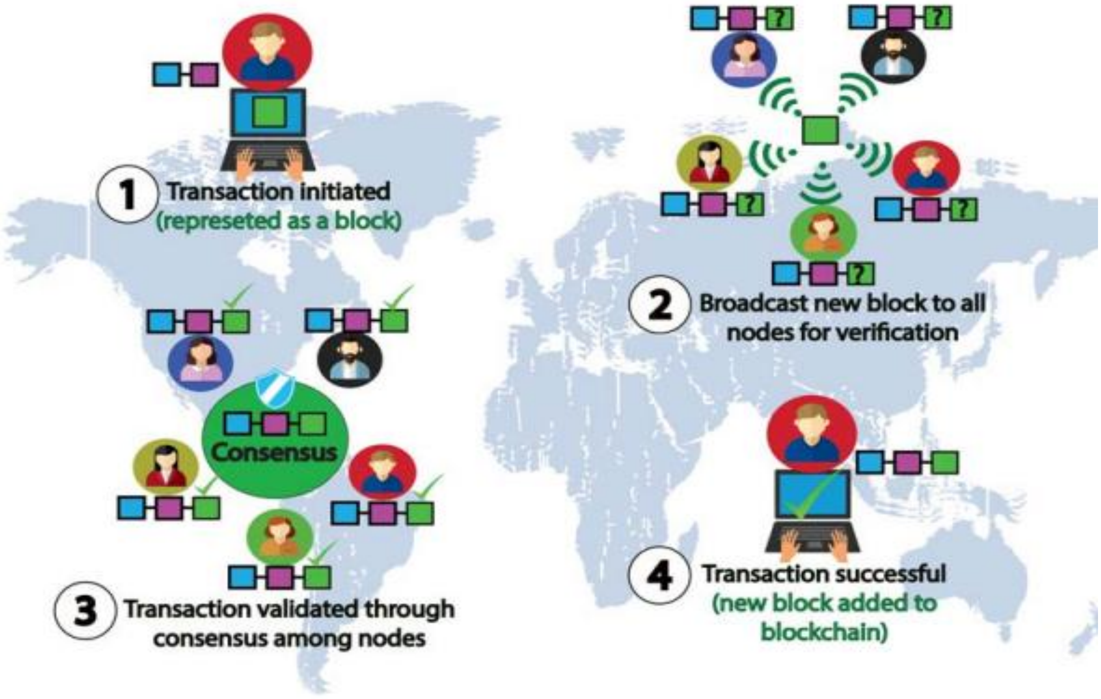

Figure 1: Blockchain architecture.

The importance of consensus methods in distributed systems is discussed in this paper. Various consensus methods for distributed environments in general and blockchain in particular are examined. The developers will be aided in assessing and developing a new consensus algorithm by identifying and discussing parameters important to consensus algorithms.

1.1. Blockchain Technology and Cryptocurrencies:

A distributed system in which a common ledger of transactions is maintained and shared among network members is known as blockchain. Transactions are stored in the form of a chain of blocks, each of which uses a hash value to refer to the preceding block. A list of transactions must be agreed upon by the parties. Divergence will occur in either scenario, resulting in forks. Each miner has a unique blockchain state that varies from miner to miner. When two or more miners mine two distinct blocks at the same time, the local states will be different. Two blocks will then point to the same preceding block. The global state of a blockchain is created by adding all local states together. The moment at which distinct blocks share the same preceding block in a global state is referred to as a fork. Various blockchain implementations, such as Bitcoin and Ethereum, utilize different methods to identify the main branch of the blockchain in order to resolve forks **[3], [4].** Bitcoin resolves forks by using the deepest branch of the blockchain as the primary branch. For this, it employs Nakamoto's consensus mechanism. The branch with the greatest number of nodes is chosen as the main branch. Greedy Heaviest Observed Subtree (GHOST), on the other hand, is an Ethereum consensus method that chooses the heaviest subtree as the primary branch.

2. DISCUSSION

2.1. Various Algorithms:

This section covers the different blockchain consensus algorithms that have recently been suggested. There are also reviews of research s that compare consensus methods in blockchain.

2.1.1. Secure Sharding Algorithm for Open Blockchain:

ELASTICO is a permission-less blockchain agreement mechanism that is claimed to be scalable **[5].** The transaction rate grows linearly with the amount of computing required throughout the mining process. In other words, as computing power increases, more transaction blocks are handled. This method can deal with opponents with a quarter of the overall computing capacity. The basic concept is to break down a network into tiny groups called committees. Each committee deals with a separate set of transactions, and the whole process is parallelized. The agreement requirement is implemented in a probabilistic manner, which distinguishes this method from the traditional byzantine algorithm. Each honest process uses a constraint function to match its agreed-upon value and validates it. Only if the constraint function is satisfied is the solution approved. A typical byzantine consensus is used by all committees. The outcomes (shards) of all committees are combined in the final committee. Some security characteristics are specified, and the algorithm is tested against each of them, highlighting its security strength. Experiments with up to 1600 nodes are performed on Amazon EC2. Additional code is added to the publicly accessible Bitcoin source in order to implement this algorithm.

2.1.2. SCP: A Computationally-Scalable Byzantine Consensus Algorithm for Blockchain:

SCP is a novel byzantine-based blockchain consensus method that is computationally scalable. The term "computationally scalable" refers to an algorithm's ability to modify bandwidth usage by changing computational factors such as the degree of difficulty in the proof of work (PoW) method **[6].** Increased computing power is anticipated to result in increased throughput. Miners with less computing capacity are able to join the committee and participate in the voting process in SCP. Furthermore, transactions are distributed evenly across data blocks, resulting in network decentralization. There is no need for a central certificate, which eliminates the danger of a single point of failure. The SCP algorithm has been translated to SCoin, a scalable crypto currency. Additionally, the Merkel tree is utilized to prevent the issue of duplicate spending. Locally, double spending is avoided by determining if a transaction has just one output. SCP has been shown to grow linearly with increased computing capacity rather than quadratically with increased bandwidth. To test computational scalability, experiments are performed on Amazon EC2 with 80, 40, 20, and 10 cores.

2.1.3. Leader-Free Byzantine Consensus Algorithm:

For a partly synchronous system, the Byzantine-based consensus issue is addressed by Borran et al **[7]**. A partly synchronous system is an asynchronous system that gradually transitions to synchronous mode. It's difficult to come up with a deterministic leader-free method for a partly synchronous system. A leader-free algorithm, on the other hand, is made up of rounds in which all nodes communicate with each other. To establish consensus in a partly synchronous system, theirpaper presented a leader-free method. For partly synchronous systems, the consensus leader free method for synchronous systems is extended. This 's consensus method for synchronous systems is based on the interactive consistency issue **[7].** Each process generates a collection of values, each of which represents a process element. Correct procedures provide the same set of results. There are two ways in which this algorithm may be improved. One method is to use a parameterized consensus algorithm, which allows various failure models to agree. The paper also proposes making the consensus optimum with rigorous validation.

2.1.4. Proof of Work Consensus Algorithm:

With regard to different network and consensus related characteristics, a quantitative methodology is presented by Asgaonkaret al. to evaluate the security and performance of PoW consensus based blockchains **[8]**. The majority of current blockchains use the PoW consensus. In terms of security evaluation, the variations of PoW have received little attention in the literature. The connection between PoWblockchain performance and security has not been thoroughly investigated. This connection was investigated using a simulator that simulated a blockchain's network and consensus process. The effect of changing the block size and block interval on selfish mining and double spending is investigated. It has been discovered that a large block reward for blockchain will reduce the likelihood of a double-spending assault. The security is unaffected by the block size of 1 MB and the block interval duration of 1 minute.

2.1.5. Implicit Consensus:

A form of implicit consensus is suggested by Ren et al. in which each node has its own blockchain **[9].** The main advantage is that the throughput is unlimited. BFT's termination property is replaced with a self-interest property. For every transaction, consensus is not assured. As a result, given linear message complexity, this method is scalable. Instead of transaction blocks, a specific kind of block called as check point blocks is explored. Theoretically, the performance and other factors are examined.

2.1.6. Proof-of-Stake Consensus Algorithm:

One of the most hotly debated features of Bitcoin is the Proof of Work (PoW). It's a method of choosing a block signer based on miners' computational effort in solving a difficult mathematical problem. Proof of Stake (PoS) was developed to address the problem of energy waste [**10]**. The fundamental concept is that nodes with a higher stake will be able to contribute blocks to the network more often. A fresh block signer is chosen based on a random criterion depending on the amount of stake held by a miner. When compared to brute-force computing in PoW, the power required to mine a block will be considerably reduced. The two most well-known PoS-based consensus methods are Ouroboros and Casper. By using a safe coin flipping algorithm and effective time slots synchronization, Ouroboros selects the stakeholder at random. Casper offers a shakier assurance as to how much stake the attacker has in order to create disruption.

2.2. Analysis of the Comparative Study:

This section covers the factors that should be considered while assessing blockchain consensus methods. The following parameters are identified as critical parameters for comparing the various consensus algorithms for blockchain: blockchain category, transaction rate, extensibility, adversary tolerance prototype, experimental setting up, latency, bandwidth, throughput, communication model, communication complexity, threats, power usage, mining, consensus category, and consensus finality. The following algorithms are compared: ELASTICO, leader-free Byzantine consensus, implicit consensus, blockchain with unbounded throughput, Proof of trust (PoT), DBFT consensus, PoPF, and Proof of Work. The discovered parameters are listed below, along with a comparison of the consensus methods with regard to them.

2.2.1. Blockchain type:

Public, private, and consortium blockchains are the three kinds of blockchain. The membership control in the consensus algorithm is determined by the blockchain type. This must be taken into account when assessing consensus algorithms to determine the kind of membership anticipated in the design. The blockchain type should be determined by the nature of the business application.

2.2.2. Scalability:

In today's world, scalability is a must when dealing with large data. When the number of nodes is increased, more transaction blocks are handled, resulting in scalability. ELASTICO and proof of trust are both scalable. Implicit consensus and proof-of-work aren't scalable. Other methods that are used in comparisons have not yet been verified for scalability.

2.2.3. Adversary Tolerance Model:

The adversary model determines the percentage of the blockchain network that can survive failure or assault without compromising consensus. The suggested method for blockchain consensus includes a threshold value for this attacker scenario. A higher enemy threshold value is preferable. ELASTICO outperforms the other algorithms in terms of adversary control.

2.2.4. Performance Related Parameters:

Some current consensus methods have not been tested in the lab. Only theoretically, using soundness proofs, are they compared. However, a quantitative study of the performance and security of these consensus methods is also required. For each of the consensus methods, the three main performance factors that must be focused on are latency, throughput, and bandwidth. Other algorithms, with the exception of ELASTICO, are not empirically tested in terms of these performance characteristics.

2.2.5. Communication Model and Complexity:

The sender waits for the receiver to acknowledge the request in synchronous communication. Asynchronous communication eliminates the requirement for the sender to wait for a response from the receiver before continuing the conversation. PoW, PoT, Ripple, and implicit consensus may be explored for real-time applications that cannot afford delays. If an application is going to have a lot of read operations, the synchronous model should be used since it responds quickly. In the design of the ELASTICO and leader-free consensus algorithms, a synchronous communication model is assumed. In comparison to ELASTICO and PoT, the leader-free consensus method has a linear and lower communication cost. The remainder of the algorithms' communication costs are yet to be studied in the literature.

3. CONCLUSION

Many apps and companies are adopting blockchain-based solutions as a result of the current blockchain boom. As a result, a thorough examination of blockchain's performance and security features was required. Recently, considerable study has been conducted with the goal of comparing current blockchain consensus algorithms and proposing a new one. In this paper, we go through the many types of consensus mechanisms, as well as their significance in a distributed context. Consensus methods for distributed systems in general and blockchain in particular are explored. We evaluated the number of factors that have a substantial effect on the consensus algorithm of several newly suggested consensus algorithms. Both security and performance issues are covered by the criteria chosen for comparison. The methods, as well as each of the discovered parameters, are then explained. Aside from this, a number of additional factors must be taken into account. Topology of the network (e.g., fully linked graph), transaction rate, consensus solution consistency, concurrency check, transaction verification time, and round complexity (if the consensus algorithm involved multiple rounds or phases).

These characteristics may be used in future study to create a more comprehensive comparison. The comparative perspective presented in this paper has emphasized the factors that need to be evaluated and analyzed for certain current algorithms. It is possible to conduct a thorough qualitative and quantitative comparison that will fill in the gaps in the comparison provided in this paper. Experiments in a cluster setting are required to fully assess the strengths and drawbacks of consensus algorithms in relation to big data requirements.

REFERENCES

1. Kuo TT, Kim HE, Ohno-Machado L. Blockchain distributed ledger technologies for biomedical and health care applications. Journal of the American Medical Informatics Association. 2017 Nov 1;24(6):1211-1220

2. Janowicz K, Regalia B, Hitzler P, et al. On the prospects of blockchain and distributed ledger technologies for open science and academic publishing. Semantic Web 2018;9:545–55.

3. Rouhani S, Deters R. Performance analysis of Ethereum transactions in private blockchain. In: 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS), pp. 70–74. IEEE, Beijing (2017). https://doi.org/10.1109/ICSESS.2017.834286F.

4. Parino F, Beiró MG, Gauvin L. Analysis of the Bitcoin blockchain: socio-economic factors behind the adoption. EPJ Data Sci. 2018;7:38 doi: 10.1140/epjds/s13688-018-0170-8.

5. Luu L, Narayanan V, Zheng C, Baweja K, Gilbert S, Saxena P. A secure sharding protocol for open blockchains. Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. 2016. doi: 10.1145/2976749.2978389.

6. Feng L, Zhang H, Chen Y, Lou L. Scalable Dynamic Multi-Agent Practical Byzantine Fault-Tolerant Consensus in Permissioned Blockchain. Applied Sciences. 2018; 8(10):1919.

7. Borran F, Schiper A. A leader-free byzantine consensus algorithm. 2010, doi: 10.1007/978-3-642-11322-2_11.

8. Asgaonkar A, Palande P, Joshi RS. Is the cost of proof-of-work consensus quasilinear? CoDS-COMAD '18: Proceedings of the ACM India Joint International Conference on Data Science and Management of Data. 2018, doi: 10.1145/3152494.3167978.

9. Ren Z, Cong K, Pouwelse J, Erkin Z. Implicit Consensus: Blockchain with Unbounded Throughput. 2017, [Online]. Available at: http://arxiv.org/abs/1705.11046.

10. Gazi P, Kiayias A, Russell A. Stake-bleeding attacks on proof-of-stake blockchains. 2018, doi: 10.1109/CVCBT.2018.00015. Available at:https://eprint.iacr.org/2018/248.pdf