Proceedings of the 5th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/18/20230954

The advance of consensus algorithm in blockchain

Runze Wei

Faculty of Innovation Engineering, Macau University of Science and Technology, Avenida Wai Long, Taipa, Macau

2009853ei011002@student.must.edu.mo

Abstract. As a distributed ledger technology, blockchain has found widespread use in a variety of industries, including finance, the Internet of Things (IoT), healthcare, and manufacturing. This technology addresses the trust issue by converting a low-trust centralized ledger into a highly trusted distributed ledger maintained by various entities. Consensus algorithms are one of the fundamental building blocks of the blockchain, controlling how nodes cooperate and synchronize data to perform secure and reliable activities in a decentralized setting. This paper examines the extant mainstream consensus algorithms, introduces six representative consensus algorithms, analyses their benefits and drawbacks, and discusses the application scenarios and suitability of each consensus algorithm in various blockchain platforms.

Keywords: blockchain, consensus algorithm, byzantine fault tolerance, bitcoin, Ethereum, financial platform.

1. Introduction

The origins of blockchain technology can be traced back to the paper "Bitcoin: A Peer-to-Peer Electronic Cash System." In the paper, Satoshi Nakamoto described a decentralized electronic cash system that utilized blockchain technology to store transaction records [1]. Then, blockchain technology started to become more well-known progressively.

As a decentralized and distributed ledger, blockchain can encrypt data storage, transfer, and access using cryptography. Its chain structure comprises chronologically connected blocks containing data about individual transactions. While the block body contains a list of transactions, the block header maintains information like the Merkle root, Timestamp, and the previous block's Hash value. The connection between blocks is made via the hash value of the parent block in each header, serving as the foundational data structure of the blockchain. Blockchain is an infinitely redundant, decentralized, and shared ledger-style database with immutable characteristics. As such, it is suitable for many decentralized and secure systems that protect user privacy and prevent malicious data tampering. Consensus algorithms play a crucial role in blockchain technology, determining its effectiveness in real-world applications.

Consensus algorithm is one of the core concepts in blockchain technology, which plays a crucial role in ensuring that all nodes in the blockchain network reach consensus on the data, thereby ensuring the security and reliability of the network. The evolution and improvement of consensus algorithms have also driven blockchain technology's widespread application and development. Consensus algorithms can be classified as Byzantine fault tolerance (BFT) or Crash fault tolerance algorithms (also called non-

DOI: 10.54254/2755-2721/18/20230954

BFT) based on the presence of malicious nodes and as Proof-based or Voting-based algorithms based on the node's proof method [2, 3]. Different consensus algorithms, such as PoW, PoS, PBFT, Paxos, and Raft, may suit different scenarios and requirements.

This paper compares multiple consensus algorithms to examine their applicability, benefits, and drawbacks in diverse situations. By contrasting the underlying concepts and traits of various consensus algorithms, user can better comprehend them, which will help people choose and build them for real-world applications. The paper's second section introduces the fundamental characteristics and technical implementation of blockchain. The paper's third section discusses the primary consensus algorithms used in current mainstream platforms, their applications, and future development directions in various fields. The fourth section summarizes the rules for selecting several consensus algorithms. Finally, the results of the earlier investigation will be used to demonstrate the conclusions.

2. Overview of mainstream consensus algorithms

2.1. Proof of work

The Proof of Work (PoW) consensus technique was initially developed to prevent spam email attacks. This design raises the cost of spam-mail sending and decreases attacks by solving a mathematical problem before mail can be effectively sent to the recipient's inbox. Later, Satoshi Nakamoto used the Bitcoin blockchain system to implement this process, giving the PoW algorithm more concrete and useful uses and further refining it. The PoW algorithm was first used publicly on an open, public blockchain network in the Bitcoin whitepaper and remains one of the most indispensable algorithms in public chains today [4].

PoW is known as the work-based consensus algorithm, and its core idea is to ensure data consistency and consensus security through computational competition. Its consensus process is referred to as "mining." Miners compete for accounting rights and are recognized and accepted by network nodes. Moreover, they calculate the hash function to find the " nonce " value that satisfies certain conditions. The setting of these conditions ensures that each block in the blockchain network requires a certain amount of time and computational power to be confirmed. Once miners find a nonce that meets the conditions, they can package it with corresponding transactions into a block and broadcast it to the entire network. Other miners will verify the block's validity, and if confirmed, they will add it to their local blockchain, synchronizing the entire network. The main principle of PoW is illustrated in Figure 1.

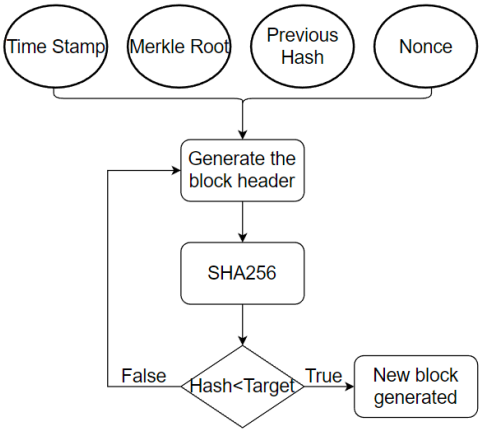

Figure 1. Workflow of the PoW.

The PoW algorithm provides strong security guarantees because breaking the system requires enormous resources and time. If someone wants to cheat in the blockchain successfully, then they must own 51% of the nodes and overpower most of the computational power, which is hard to achieve. Furthermore, PoW can achieve a certain degree of decentralization because this consensus algorithm

DOI: 10.54254/2755-2721/18/20230954

fairly distributes accounting rights to different nodes based on their computational effort, which is also why it is called 'Proof of Work'.

However, the disadvantages of PoW are also evident. First, since the Proof of Work requires many hash calculations, it demands enormous electricity consumption. According to statistics from the BBC, bitcoin mining activities using PoW consensus algorithms will produce over 370,000 tons of carbon dioxide annually. PoW's efficiency is also significantly low because miners can only find the nonce through continuous hash calculations, resulting in a slow block generation speed. The maximum transaction processing throughput is around 7 transactions per second, which cannot meet the needs of large-scale commercial applications [5].

Although PoW provides strong security guarantees and can achieve decentralization, its drawbacks include high electricity consumption and low efficiency, limiting its scalability and application in large-scale commercial environment.

2.2. Proof of stake

In response to the evident drawbacks of the PoW consensus algorithm, the PoS algorithm was introduced in 2011. Unlike PoW, where nodes compete for accounting rights through computational power, PoS grants accounting rights to nodes with the highest stake. PoS introduces the concept of 'Token', where a node's stake is determined by the number and length of time that they hold tokens [6]. As a node's stake increases, they find it easier to discover the next block and efficiently find the target random number. Once a node obtains the accounting rights for a block, its stake is cleared and starts accumulating stake again in the next round. The main principle of PoW is illustrated in Figure 2.

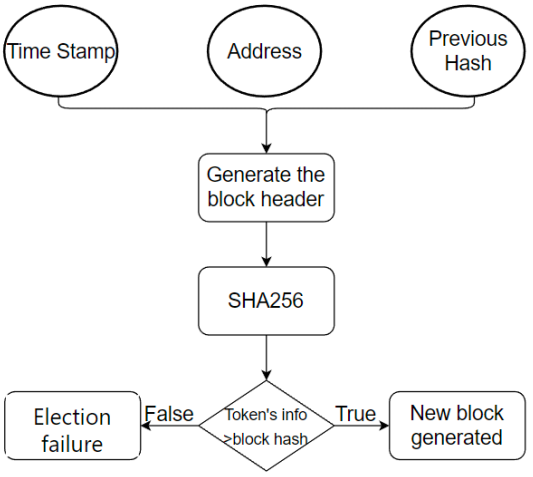

Figure 2. Workflow of the PoS.

One advantage of PoS is that it replaces the tedious and wasteful hash calculation of PoW with a comparison of tokens, reducing computational requirements and improving efficiency. Furthermore, the distribution mechanism of stake allocation in PoS means that attackers would need to control most of the tokens in the system to launch an attack, making PoS more secure than PoW. Additionally, since nodes must hold tokens to participate in the accounting process, they have a greater incentive to protect the system's security.

On the contrary, one of the main drawbacks of the PoS consensus algorithm is the initial allocation problem, which significantly impacts the decentralization and security of the network. If the initial allocation needs to be better designed, it can lead to a concentration of wealth among a few individuals or entities, undermining the network's security and untrustworthiness. Moreover, PoS can also lead to centralization, as wealthier validators have more voting power and, therefore, more control over the system. This can make the system less secure and potentially prone to collusion among a small group of validators.

DOI: 10.54254/2755-2721/18/20230954

PoS has shown to be an alternative to upgrading PoW due to its increased efficiency and security. However, the centralization of interests is still an issue that needs to be addressed.

2.3. Delegated proof of stake

Although PoS largely addressed the PoW problem of resource waste, a perfect consensus mechanism is still required. Delegated Proof of Stake (DPoS) was introduced by the Bitshares project in August 2013 and is based on the idea of "board decision-making." The DPoS algorithm's fundamental idea is to choose a set number of representative nodes to take part in decision-making. A trustee committee made up of these representative nodes oversees approving transactions, packaging blocks, and maintaining the ledger.

The DPOS consensus algorithm lets nodes vote for the candidate nodes they trust as part of the election process. Each node can vote for more than one candidate, and the amount of votes each candidate gets determines how many people will take charge of accounting for each block. This feature makes it possible for every node in the system to become a decision-maker. Compared to other consensus algorithms, this makes DPoS more decentralized. Also, if a decision maker breaks the law of blockchain, their ability to make decisions can be taken away. This makes sure that all nodes that use DPoS are safe. The main idea of DPoS is shown in Figure 3.

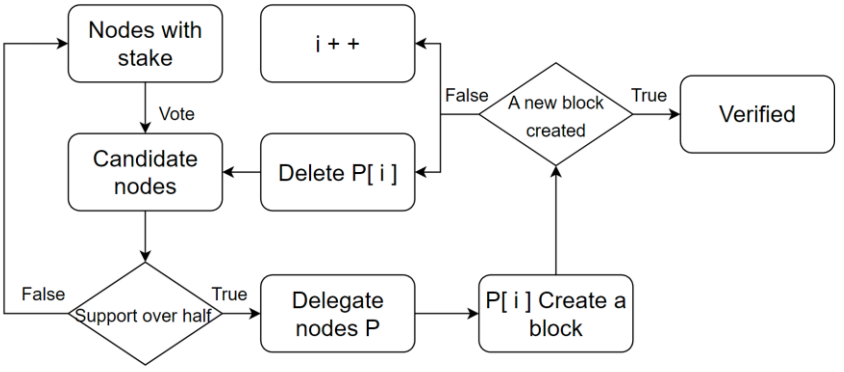

Figure 3. Main workflow of DPoS.

The DPoS consensus algorithm offers several advantages over both PoS and PoW. First, it does not rely on significant computing resources. Second, by utilizing multiple decision-makers with accounting rights, the number of delegate nodes is reduced, leading to a simpler network topology between delegate nodes. The chosen delegate nodes are typically high-performance nodes that increase the speed of communication between nodes, which significantly improves the transaction throughput of the system. It enhances the decentralization degree and the throughput of the DPoS network by increasing the number of delegate nodes, but this relationship is basing on various factors rather than linear, such as the network size and communication costs, etc.

However, it is worth noting that DPoS also has limitations. One major issue is that it is susceptible to security attacks when super nodes are compromised or conspire to act maliciously. Additionally, while the DPoS algorithm ensures node fairness using open-source algorithms, the sparse distribution of super nodes could still cause issues in terms of centralization, which means the centralization issue still not being well-solved.

2.4. Practical byzantine fault tolerance

As a byzantine fault tolerance consensus algorithm, the Practical Byzantine Fault Tolerance (PBFT) algorithm was proposed by Barbara Liskov in 1999 [7]. PBFT predates the PoW, PoS, and DPoS consensus algorithms and is rooted in Lamport's Byzantine Generals' Problem, which was introduced in 1982. The problem is ensuring that loyal generals act simultaneously and accurately if some generals may be traitors and communication is restricted to messengers.

DOI: 10.54254/2755-2721/18/20230954

The PBFT algorithm is based on the idea that establishing consensus and accuracy across dispersed nodes during decision-making is the key to tackling this issue. The PBFT consensus algorithm consists of the following four phases and the workflow is specifically illustrated in figure 4 as follows.

1) User initializes a primary node.

2) The primary node sends a request to other peer nodes and waits for their replies in each round.

3) When the peer nodes receive the request, they forward it to the other peer nodes and validate and vote according to predefined rules.

4) When the user receives replies from most nodes, it broadcasts the results to all nodes, thus completing the consensus process once.

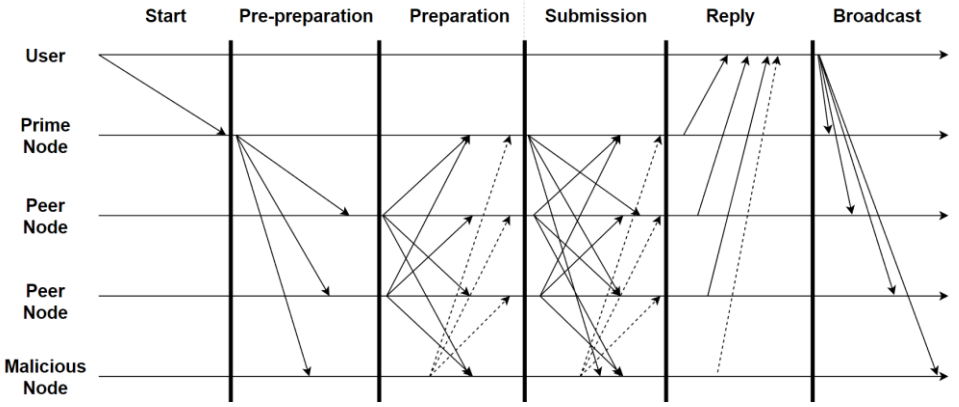

Figure 4. Workflow of PBFT [8].

In this process, nodes communicate through messaging, with the roles of primary and backup nodes alternating to ensure a reliable and fault-tolerant system. It also ensures communication security and data consistency through techniques such as digital signatures and hashes. PBFT allows for a certain degree of node dishonesty, hence the name Byzantine fault-tolerance algorithm.

Compared to the PoW and PoS algorithms mentioned above, the PBFT consensus algorithm reduces algorithm complexity because it only requires one consensus calculation per round. PoW and PoS, on the other hand, require significant computation and verification work, which can lead to transaction delays and congestion in high-concurrency scenarios. Moreover, PBFT's faster consensus speed enables it to process more transactions per unit of time, resulting in higher throughput and better scalability, making it more suitable for high-concurrency scenarios. Additionally, PBFT is a resource-saving consensus algorithm that does not require extensive hash calculations, unlike PoW and PoS.

Nevertheless, implementing the PBFT algorithm is more complex, and it demands a high level of network environment and node behavior, or it may lead to high latency issues.

2.5. Paxos

Unlike the consensus algorithms described above, Paxos is a typically non-Byzantine fault-tolerant consensus algorithm. From a different perspective, Paxos is a voting-based distributed consensus algorithm that ensures data consistency among nodes in a distributed system that is the same as PBFT. Its core principle is based on election and voting mechanisms to achieve consensus, which can be summarized in three steps: the prepare phase, the proposal phase, and the commit phase, as detailed below.

1) Prepare Phase: The proposer sends a prepare request to the acceptors. Upon receiving the prepared request, if the acceptor has yet to commit to any other proposal, it sends a promise to the proposer. It promises not to support any proposal with a smaller proposal number. If the acceptor has already committed to another proposal, it sends the proposal number it has already promised.

2) Proposal Phase: The proposer sends a proposal to the acceptors and requests their votes. Upon receiving the proposal, the acceptors compare it with the proposal they have already promised. If the

DOI: 10.54254/2755-2721/18/20230954

proposal number exceeds the promised amount, the acceptor supports the proposal and sends an acceptance message to the proposer. Otherwise, the acceptor ignores the proposal and sends no message.

3) Commit Phase: If the proposer receives acceptance messages from most acceptors, it considers the proposal selected and sends a decision message to all learners. Upon receiving the decision message, the learners update the distributed system data with the selected proposal.

The pseudo code for the three stages is shown in table 1 as followed:

Table 1. Basic algorithm of Paxos.

Client (Proposer)	Server (Reciever)
Initialization	
c # waiting for the command	$T_{max} = 0$ # current biggest ticket number
t = 0 # current ticket number	
	C = ⊥ # current command
	$T_{store} = 0$ # tickets used to store C
Prepare Phase	
1: t = t + 1	
2: send message to all servers, and ask for the tickets that number is t	
	3: if t > T_{max} then
	4: T_{max} = t
	5: reply: ok(T_{store}, C)
	6: end if
Proposal Phase	
7: if over half of the servers reply ok then	
8: choose the max (T_{store}, C)	
9: if T_{store} > 0 then	
10: c = C	
11: end if	
12: reply propose (t, c) to the servers reply ok	
13: end if	
	14: if t = T_{max} then
	15: C = c
	16: T_{store} = t
	17: reply: success
	18: end if
Commit Phase	
19: if over half of the servers reply: success then	
20: send execute (c) to all servers	
21: end if	

In simpler terms, Paxos is a consensus protocol that achieves consistency in an asynchronous communication environment, even in situations where some machines are not operational. By agreeing on the same value across multiple replicas, Paxos achieves consistency. Since Paxos does not require identity signature verification in communication between nodes, it significantly reduces the communication overhead and is considered efficient. With rigorous mathematical proof and ingenious system design, Paxos also ensures a certain level of security.

DOI: 10.54254/2755-2721/18/20230954

However, due to its complex system design and rigorous mathematical proof, Paxos is challenging to implement in engineering practice and requires different levels of engineering optimization. Deviations in engineering design can lead to system failures. Moreover, due to the need for an identity verification phase, Paxos can only accommodate faulty nodes and is limited in its use in private chain applications.

2.6. Raft

Similar to Paxos, Raft is a non-Byzantine fault-tolerant algorithm based on leader election mechanisms. The goal is to provide an easy-to-understand and implement consensus algorithm, ensuring that all nodes eventually reach a consistent state by treating log entries as inputs to a state machine. The algorithm primarily involves two following steps:

Leader election (shown in Figure 5): Initially, all nodes are followers. When a node becomes a candidate, it sends an election request to other nodes. Based on the latest log record, other nodes decide whether to vote for the candidate. Once a candidate receives more than half the votes, it becomes the new leader.

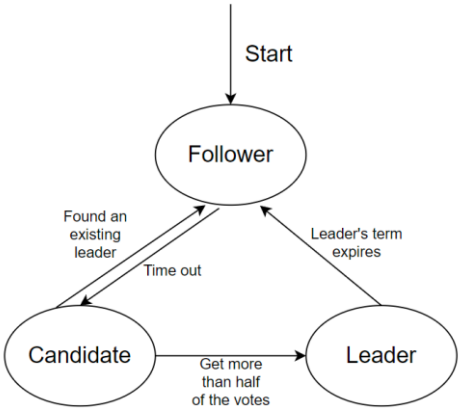

Figure 5. Workflow of leader election.

Log replication (shown in Figure 6): After receiving a client request, the leader converts it into a log entry and sends its followers to other nodes. Other nodes update their state machines based on the information in the log entry.

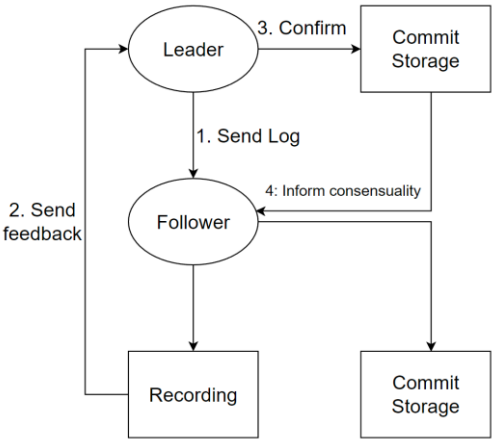

Figure 6. Workflow of log replication.

DOI: 10.54254/2755-2721/18/20230954

Raft uses two safety mechanisms to ensure the correctness and consistency of log replication. The first ensures that each log entry has a unique index, ensuring that nodes submit and replicate logs in the same order. The second is that each log entry contains a term value, indicating which round of elections the log entry belongs to, preventing the leader from changing committed log entries.

Since Raft uses leader election to clarify the roles of nodes in the system and adopts a majority principle, it can ensure that the system continues to operate normally even in the event of node failure or network partition. Therefore, Raft is more suitable for handling consensus issues in distributed systems.

Raft's implementation details are very clear, with the exception of the sections on configuration changes, leader election, and log replication. In contrast, the Paxos algorithm has numerous open ends for developers to work around, resulting in a variety of implementations. Michael Deardeuff of Amazon discovered that many of the Paxos implementations on Github have issues [9]. This is fatal for an infrastructure dependency library since it makes it difficult to locate a reliable reference and almost certainly forces user to go over the same ground repeatedly. Developers began to emulate and make certain optimizations after Raft's algorithm's success in systems like ETCD.

Above all, Raft appears to be more generalizable overall than Paxos.

3. Blockchain platforms and projects

Blockchain technology is a disruptive technology that has a place in most application areas. For example, in the financial sector, blockchain can be used to build a more secure, efficient, transparent, and reliable transaction system, thereby reducing transaction costs, reducing fraud risks, and improving the financial system's stability. Meanwhile, in non-financial fields, blockchain can also be used to establish decentralized applications and digital asset trading platforms, thereby accelerating digital transformation and business model innovation. All these applications rely on consensus algorithms because consensus algorithms are the core mechanism ensuring blockchain systems' security and trustworthiness.

Different consensus algorithms have their own applicable scenarios and advantages and disadvantages. Therefore, when selecting the appropriate consensus algorithm, many factors need to be considered, such as the size of the system, transaction volume, number of nodes, scalability, and security, leading to the differences when it comes to different situations.

3.1. Financial sector

Bitcoin is the first cryptocurrency to apply blockchain technology successfully [1]. The initial purpose of Bitcoin was to solve the many problems brought about by centralized currency issuance and exchanges. By using blockchain technology to achieve decentralization, tamper-proofing, and distributed accounting, among other features, Bitcoin ensures the security and transparency of transactions. This process requires a lot of computing power and energy consumption, so there is a certain relationship between the degree of decentralization of the Bitcoin network and energy consumption. PoW is one of the earliest consensus algorithms applied to the blockchain. It selects the producer of the next block through a competition for computing power. It is still widely used in some cryptocurrency blockchains.

Ethereum is an upgraded financial platform of Bitcoin and mainly applies PoW and PoS, two consensus algorithms [10]. Ethereum has been evolving, and its overall development can be divided into three stages. When Ethereum was just born in the first stage, it used a PoW consensus algorithm like Bitcoin. However, due to the slow block generation speed and the problem of great computing difficulty that gradually emerged in the development of Ethereum, these two factors led to an increase in the mining difficulty based on PoW, ultimately forcing the consensus mechanism of this blockchain to transition to a PoS consensus algorithm gradually. After years of planning and delays, the switch was completed on September 15, 2022, and the core digital machine of Ethereum, the second-largest coin by market value, was changed to a system that uses less energy. This finished the switch from PoW to PoS, which cut energy use by 99.5%.

DOI: 10.54254/2755-2721/18/20230954

3.2. Non-financial sector

Hyperledger Fabric is a comprehensive blockchain application platform launched by the Linux Foundation in 2015 [11]. It is appropriate for a range of industrial applications, including supply chain monitoring, trade finance, loyalty and incentives, and financial asset clearing and settlement, thanks to its modular and generic structure, special identity management, and access control capabilities. It uses multiple consensus algorithms, including Raft, BFT, and PBFT, which are selected and adjusted according to different scenarios and needs. For example, when high throughput and low latency are required, the PBFT algorithm quickly confirms transactions and can tolerate a certain number of malicious nodes. When higher security is needed, the BFT algorithm is used, which requires more nodes to confirm transactions but can tolerate more malicious nodes. For some lightweight applications, the Raft algorithm is used, which is characterized by simplicity, low latency, and relatively weaker security. In summary, the Farbrc platform balances security, scalability, and efficiency by selecting and combining different consensus algorithms to meet the needs of different application scenarios.

MedicalChain is a representative decentralized healthcare platform based on blockchain technology that enables the secure, fast, and transparent exchange and use of medical data [12]. It makes a user-centric electronic health record with a single, true version of user data using private chain technology. The information in a patient's medical records, which includes things like blood test results, CT scans, and doctor diagnostic letters, is frequently dispersed and kept in several institutions. It arranges all these records chronologically and filters them into the categories mentioned above to help with data processing. Patients can access and comprehend these records more easily because of this categorization, which also makes it easier for researchers to look up relevant information. To accomplish these functions, MedicalChain uses the consensus algorithm based on PBFT, which ensures the consistency and reliability of medical data, thereby ensuring data security and privacy.

Chubby uses the Paxos algorithm to ensure consistency of logs across replicas, ensuring that the local logs of each replica have the same content. A highly fault-tolerant distributed database layer is then added on top of this. Chubby has been used in a number of Google projects, including Bigtable and DFS. ZooKeeper's design philosophy is similar to Paxos', with a few minor modifications, and it can be seen as an open-source implementation of Chubby [13]. Hadoop, which is based on ZooKeeper, can provide a strongly consistent distributed file system. Hbase and Yahoo! Message are the only two projects at Yahoo! where ZooKeeper has been successfully used. The virtual computing platform's foundation, the Nutanix Distributed File System (NDFS) created by VMware, is in charge of handling all metadata and data. When nodes fail, NDFS's extremely high fault tolerance ensures data availability and consistency. Strong data consistency and arbitration (Quorum) for leader node elections are ensured by the system using Paxos.

PowerLedger is a prominent example of a successful blockchain application in the energy industry for the Internet of Things (IoT) [14]. PowerLedger uses a specialized permissioned Solana blockchain and was created as software to track, trace, and sell renewable energy. Solana's architecture, which combines a history-based proof and a stake-based proof (PoS) consensus mechanism, makes it faster and more energy-efficient than current proof-of-work blockchains (processing over 50,000 transactions per second). Because of the PoS nature of Solana, only validators who have been approved by PowerLedger are allowed to participate. PowerLedger also takes into account the type of energy that each validator uses, including renewable energy.

4. Issues in practical applications

This paper discussed several blockchain-based platforms and projects that utilize different consensus algorithms in the previous section. This leads to an important question: What is the most appropriate consensus algorithm?

When choosing the right consensus algorithm, it is necessary to analyse the characteristics of different scenarios and business requirements. In scenarios where security and reliability are the most critical factors, such as in the financial sector, platform developers tend to use PoW and PoS consensus algorithms. Although some drawbacks of energy transition consumption and high node requirements

DOI: 10.54254/2755-2721/18/20230954

exist, these algorithms ensure a relatively high level of security and reliability of the system, effectively resisting malicious attacks and tampering behaviour. Where platforms require security, reliability, and high throughput, consensus algorithms such as PBFT, Paxos, and Raft are more suitable. This is because these algorithms can confirm transactions quickly and ensure the system's efficiency while keeping it secure and reliable.

If the number of nodes on the platform is small, DBFT and other consensus algorithms are a better choice. These algorithms can confirm transactions quickly without requiring many computing resources and are suitable for scenarios with few nodes. In scenarios where energy conservation and environmental protection are required, other consensus algorithms, such as PoS, are a more suitable choice. These algorithms can save energy while ensuring system efficiency and security.

In conclusion, when choosing a consensus algorithm for practical applications, various factors such as security, throughput, number of nodes, and energy efficiency must ultimately be considered to select the most appropriate consensus algorithm.

5. Discussion

Having introduced several basic consensus mechanisms and blockchain application platforms in the previous sections, we see that a single basic consensus algorithm may sometimes meet only a single requirement. And no consensus algorithm is perfect. Typically, there are two ways to improve the consensus mechanism for platform development. First, multiple consensus algorithms can be utilized simultaneously. For example, when dealing with different types of tasks, different algorithms can be selected or combined depending on the characteristics of the task, such as throughput, number of nodes, and security. Secondly, the basic algorithm can be upgraded, which can be in the direction of reputation evaluation mechanisms, node election mechanisms, traceability quality evaluation, and verification mechanisms.

Blockchain technology is developing rapidly and is being actively adopted by many traditional industries. Targeted improvement and innovation of algorithms to meet the diverse needs of different industries or platforms is a long and unfinished process.

6. Conclusion

This paper describes the consensus algorithms, including its specific features and applications. And they are divided into four parts. The first part discusses the background and history of blockchain and the importance of consensus algorithms. The second part introduces the basic principles of the classical consensus algorithms used in blockchains and analyses the advantages and disadvantages of each according to their principles. There are some Byzantine fault-tolerant algorithms include PoW, PoS, DPoS, and PBFT. In addition, the paper also introduces two non-Byzantine algorithms: Paxos and Raft. In Section three, the paper lists the mainstream blockchain platforms and analyses the main algorithms within them. In the fourth part, the paper analyses the application scenarios and applicability of various consensus algorithms. It concludes by discussing how blockchain technology can be upgraded for different situations. In the future, the author will further investigate updating and upgrading consensus algorithms and explore the possibility of combining some artificial intelligence techniques.

References

[1] Nakamoto, S., & Bitcoin, A. (2008). A peer-to-peer electronic cash system. Bitcoin.–URL: https://bitcoin. org/bitcoin. pdf, 4(2).

[2] Lamport, L. (2001). Paxos made simple. ACM SIGACT News (Distributed Computing Column) 32, 4 (Whole Number 121, December 2001), 51-58.

[3] Driscoll, K., Hall, B., Sivencrona, H., & Zumsteg, P. (2003). Byzantine fault tolerance, from theory to reality. In Computer Safety, Reliability, and Security: 22nd International Conference, SAFECOMP 2003, Edinburgh, UK, September 23-26, 2003. Proceedings 22 (pp. 235-248). Springer Berlin Heidelberg.

[4] Nakamoto, S. (2008). Bitcoin whitepaper. URL: https://bitcoin. org/bitcoin. pdf-(: 17.07. 2019).

DOI: 10.54254/2755-2721/18/20230954

[5] Yuan, Y., Zhao, Y., Su, M., Wang, G., & Liu, X. (2022, August). A New PoW Consensus of Blockchain Based on Legendre Sequence. In 2022 IEEE International Conference on Blockchain (Blockchain) (pp. 187-193). IEEE.
[6] Ge, L., Wang, J., & Zhang, G. (2022). Survey of Consensus Algorithms for Proof of Stake in Blockchain. Security and Communication Networks, 2022.
[7] Castro, M., & Liskov, B. (1999, February). Practical byzantine fault tolerance. In OsDI (Vol. 99, No. 1999, pp. 173-186).
[8] Xiong, H., Chen, M., Wu, C., Zhao, Y., & Yi, W. (2022). Research on progress of blockchain consensus algorithm: a review on recent progress of blockchain consensus algorithms. Future Internet, 14(2), 47.
[9] Wang, Z., Zhao, C., Mu, S., Chen, H., & Li, J. (2019, July). On the Parallels between Paxos and Raft, and how to Port Optimizations. In Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing (pp. 445-454).
[10] Wood, G. (2014). Ethereum: A secure decentralised generalised transaction ledger. Ethereum project yellow paper, 151(2014), 1-32.
[11] Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., ... & Yellick, J. (2018, April). Hyperledger fabric: a distributed operating system for permissioned blockchains. In Proceedings of the thirteenth EuroSys conference (pp. 1-15).
[12] SA, M. Medicalchain whitepaper 2.1, 2018.
[13] Hunt, P., Konar, M., Junqueira, F. P., & Reed, B. (2010, June). ZooKeeper: wait-free coordination for internet-scale systems. In USENIX annual technical conference (Vol. 8, No. 9).
[14] Kim, G., Park, J., & Ryou, J. (2018, January). A study on utilization of blockchain for electricity trading in microgrid. In 2018 IEEE International Conference on Big Data and Smart Computing (BigComp) (pp. 743-746). IEEE.