Research and development of blockchain consensus algorithms

Xinkun Ma*[,a], Yingna Li[b,1], Jinyu Wang[a,2]
[a] Faculty of Information Engineering and Automation, Kunming University of Science and Technology, 727 Jingming South Rd., Chenggong District, Kunming ,China 650500; [b] Computer Technology Application Key Lab of the Yunnan Province, 727 Jingming South Rd., Chenggong District, Kunming ,China 650500.

ABSTRACT

Blockchain technology is a paradigm of distributed systems, and its consistency problem is solved by consensus algorithms. This paper summarizes the existing consensus algorithms and divides them into three types: proof, voting, and Paxos-like. The implementation methods and advantages and disadvantages of some representative algorithms among the three types of consensus algorithms are introduced. Finally, the research hotspots and development directions of consensus algorithms in terms of performance and scalability are pointed out.

Keywords: Consensus algorithms; blockchain; PoW; PBFT; paxos

1. INTRODUCTION

In 2008, "Satoshi Nakamoto" published the paper "Bitcoin: A Peer-to-Peer Electronic Cash System" [1], in which he described the ledger concept and technical details of decentralized digital currency transactions. In 2009, the Bitcoin system was officially released, and then blockchain technology entered the public's field of vision.In recent years, the tremendous development of blockchain technology has provided the possibility for its application in more fields. The research and application of blockchain technology have shown explosive growth. It is considered to be the fifth disruptive innovation in the field of information technology. , is the fourth milestone in the history of human credit evolution after blood relative credit, precious metal credit, and central bank banknote credit. Today, blockchain, artificial intelligence, cloud computing, and big data are known as the four major black technologies in the information industry. Development of blockchain technology[2].

After the emergence of a distributed system, the first fundamental problem that needs to be solved is how to achieve the unity of data on different nodes within the cluster and the consistency of specific operating states (such as "commit" or "rollback") between different nodes, that is, Through the consensus mechanism, the distributed nodes can achieve consensus or achieve a stable state. Different from traditional distributed systems, the blockchain system establishes trust between nodes through consensus algorithms in a complex and open Internet environment lacking trust mechanisms (especially public chains), and finally achieves the consistency of ledgers on different nodes and data security[3]. As the core component of the blockchain system, the consensus mechanism provides a guarantee for the decentralization of the system. The quality of the consensus algorithm directly affects the performance efficiency, security and scalability of the overall blockchain system[4]. In general, different blockchain frameworks use different consensus algorithms. Since blockchain technology is a distributed database from a macro perspective, blockchain network nodes must autonomously maintain the entire ledger according to certain rules. data consistency. Usually, the key technology to solve the consistency problem between node ledgers is the consensus algorithm. A good consensus algorithm can greatly save the time required for blockchain network nodes to achieve consistent ledger data synchronization, thereby improving the operating efficiency of the entire blockchain system[5].

*emptycity.stu@gmail.com; [1]2032727859@qq.com ; [2]1712547473@qq.com

Seventh International Conference on Mechatronics and Intelligent Robotics (ICMIR 2023), edited by Srikanta Patnaik,
Tao Shen, Proc. of SPIE Vol. 12779, 1277915 · © 2023 SPIE · 0277-786X · Published under a
Creative Commons Attribution CC-BY 3.0 License · doi: 10.1117/12.2688759

2. DEVELOPMENT OF EXISTING CONSENSUS ALGORITHMS

The earliest distributed consensus algorithm Paxos was proposed by Lamport[6-7], and then various other types of consensus algorithms were based on this and were continuously proposed by scholars. Since the Paxos algorithm knowledge theory cannot be realized in production practice, practical algorithms such as Raft emerged in the industry, and then Lamport proposed the classic Byzantine general problem in 1982[8]. In 1985, Michael Fisher, Nancy Lynch and Michael Paterson jointly proposed and proved the "FLP Impossibility Theorem", which plays an important guiding role in the design of distributed system consensus algorithms[9]. The theorem points out: In an asynchronous communication system with a reliable network, when there is a node failure (even if there is only one), there is no protocol that can guarantee the system to reach a consensus within a finite time. "FLP Impossibility Theorem" points out that in a distributed asynchronous communication system where there may be node failure, theoretically there is no consensus algorithm that can make the system reach a consensus within a limited time. Therefore, researchers adjust the problem model to avoid the "FLP Impossibility Theorem" to find an engineering-feasible consensus algorithm. For example, in the Bitcoin system, the network is assumed to be weakly synchronized, that is, the network nodes can be quickly synchronized, and the miners work in a To circumvent the "FLP Impossibility Theorem" by investing a finite amount of time in blocks etc. Until the late 1990s, Castro and Liskov proposed PBFT, which solved the Byzantine general problem better. By analyzing and summarizing the implementation principles of the existing consensus algorithms, they can be classified into three categories: a) Proof classes based on node attributes. The so-called proof class means that the final consensus result is determined by a certain node, such as the PoW algorithm , the miners are selected through the calculation of the workload of the nodes, and the final consensus result is released by the miners; b) Consensus algorithm based on the voting system, the voting system is to select a group of nodes to participate in the consensus, the final result is voted and the number of votes reaches a specific threshold The consensus result will only be passed. For example, PBFT will only pass the consensus result if 33% of the nodes reply; c) Paxos-like consensus algorithm. This type of consensus algorithm directly replicates the leader node when the leader node is in good condition. log, so as to continuously update its own local log, such as the Raft algorithm, the follower node always copies the log of the leader node. Figure 1 lists the development history of some consensus algorithms.

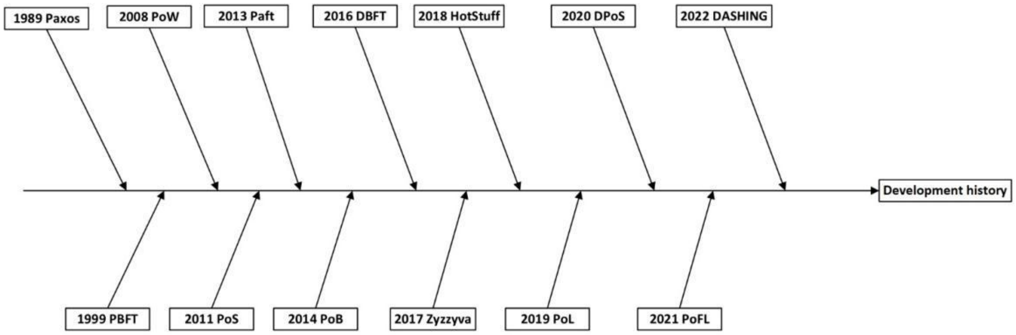

Figure 1. The development history of consensus algorithm

3. PROOF CLASS BASED CONSENSUS ALGORITHM

3.1 PoW algorithm

PoW(Proof of Word) is the neutralization core algorithm of the Bitcoin system. In the Bitcoin system, the sender (certifier) must prove that it has completed a certain amount of computing work within a certain time interval[10]. The idea for this consensus comes from the work in the Dwork paper[11]. PoW is proposed for a peer-to-peer version of an electronic cash system (Bitcoin), where online payments happen directly between two parties without any intermediaries in between. It is believed to solve the double spending problem due to the reversibility of online transactions[12]. Its core formula is shown in (1), where data is the data that combines information such as timestamp, version number, and block height, nonce is a random value, D(d) is the target value, and d is the mining As the difficulty of mining increases, it becomes more difficult to match the target value with the hash value.

$$\begin{cases} SHA256(SHA256(data \mid nonce)) \leq D(d) \\ D(d) = \dfrac{2^{224}}{d} \end{cases} \tag{1}$$

There are three main problems in the PoW algorithm: a) Low efficiency. The block capacity in the Bitcoin system is only 1MB, the transaction volume in the network is about 7 transactions/s, and the block generation time is about 10 minutes. The throughput of this specification is very unsuitable for high-concurrency business scenarios. b) High resource consumption. Since the miner nodes need a large amount of power and stream processors and other mining machines during the mining process, according to the Digiconomist Bitcoin Energy Consumption Index and the Cambridge Bitcoin Power Consumption Index, the electricity consumption of the Bitcoin network in 2018 is equivalent to Electricity consumption in the Philippines or Finland. c) The degree of decentralization is low. The original intention of Satoshi Nakamoto to design Bitcoin mining is that everyone uses CPU to mine, but now most of the computing power is in the hands of computing power mining pools, and mining pools and mining pools will also form alliances to compete for bookkeeping rights.

3.2 PoS algorithm

PoS(Proof of Stake) was proposed by King and others, and was later adopted by Nextcoin. PoS proposes solutions to the problems of excessive waste of PoW resources and slow block generation time, so that each participating node has two attributes of currency holdings and currency age. The calculation is shown in formula (2). Whenever a new transaction message in the blockchain network begins to appear, the weight is calculated according to the currency age of each node, and the node conducts a round of elections according to the weight. The probability of the node with the greater weight being selected as the bookkeeping node is also The bigger it is, then the bookkeeping node starts to package transaction messages, its currency age will be reduced, and the bookkeeping node will get incentives and some transaction fees. Although PoS solves the problems caused by PoW to a certain extent, the degree of decentralization is fundamentally low, and the bookkeeping rights are still in the hands of a small number of nodes with older currency ages, which is likely to cause polarization among currency holders The phenomenon. In theory, the attacker needs to have 51% of the currency holdings to successfully attack[13].

$$hash(block\_header) \leq t\arg et \cdot coinage \tag{2}$$

The first problem faced by the original PoS algorithm is the non-interest attack (the bookkeeping node maliciously forks to obtain double benefits). The main reason for this problem is that there is no clear equity distribution and punishment measures for the bookkeeping nodes in the application framework . Another problem is long-range attacks. Since most public chains are based on the longest chain principle, if malicious nodes create a pseudo-chain longer than the main chain, all transaction records will be tampered with, and the public chain If there are many nodes that have been offline for a long time or have newly added nodes when synchronizing data, the blockchain will fork. According to the principle of the longest chain, the pseudo-chain will be selected as the main chain.

3.3 PoSpace algorithm

The core idea of PoSpace (proof of space)[14] is to use the user's hard disk space as the cost of proof to replace the computing resources of PoW. The size of the user's workload can be judged by the size of the content downloaded by the node. After the node pays for the disk capacity, subsequent mining There is no need to pay additional costs, and nodes with larger storage space are more likely to become bookkeeping nodes. In the PoSpace algorithm, the participating nodes create a nonce value through the Shabal algorithm. The nonce is very difficult to calculate, so it is necessary to pre-calculate and store the nouce value. Every two hash values constitute a scoop. A nouce is composed of multiple scoops. The more space the user has, The more the nouce value, the node needs to use its available disk space to continuously store the nouce value before starting mining. If there is a hash value that is closest to the puzzle in the network, it wins the mining This algorithm has the characteristics of a high degree of decentralization, low requirements for obtaining tokens, and low energy consumption. A quality function is defined in PoSpace as a competition indicator:

$$Quality(hash, S) = (hash)^{1/S} \tag{3}$$

Among them, hash is the hash value of the ordinary node being verified, and S is the size of the storage space occupied by the data. In each round of competition, the node with the smallest quality function result wins.

The PoSpace consensus algorithm uses storage space instead of computation, saving power resources; also, under this consensus protocol, the storage space size is determined when the node first connects to the network and cannot be expanded afterwards, preventing the emergence of "mining pools" in the PoW consensus algorithm (pooling the computing power of different nodes into one large computing power node) and, to a certain extent, avoiding the problem of increased centralisation and reducing security risks. The problem of increased centralisation is avoided to a certain extent, and security risks are reduced.

4. VOTING CONSENSUS ALGORITHM

4.1 DPoS algorithm

DPoS was designed by Larimer and first implemented in the BitShares project[15]. Its purpose is to solve the problem of centralized accounting in PoS. DPoS introduces a proxy mechanism, and currency holders can elect super nodes as accounting representatives, and elect several representative nodes to keep accounts in turn. Each super node has a cycle. It will be removed when there is an abnormality in the cycle. DPoS consumes less energy, reduces network operating costs, is more decentralized than PoS, and achieves consensus faster, but the participation in community election voting is very low. When dealing with abnormal super nodes, the election system cannot solve abnormalities in time. Problems posed by malicious nodes.

Using DPoS can reduce the number of participating verification and accounting nodes, so as to achieve second-level consensus verification. However, the centrality of the blockchain system using the DPoS consensus algorithm is relatively weak, and its security is weaker than that of POW. At the same time, the node agent is manually selected, and its fairness is lower than that of POS. At the same time, the entire consensus mechanism still relies on tokens. Additional issuance to maintain the stability of proxy nodes. DPoS introduces an election mechanism into PoS. DPoS divides nodes into two roles: ordinary nodes and trusted nodes. Ordinary nodes can vote for trusted nodes or be voted as trusted nodes. In DPoS, nodes consume rights and interests as voting rights, and the most trusted N nodes are generated according to the weighted results of rights and interests to become the trust node set $N\{N_1...N_n\}$, and each trust node is given a fixed period to become the master node in turn , after the authorization time is over, the authority is handed over to the next trusted node. After the set N ends the cycle, a new N is re-elected, and malicious nodes will lose trust in this round of elections. The flow of the DPoS consensus algorithm is shown in Figure 2.

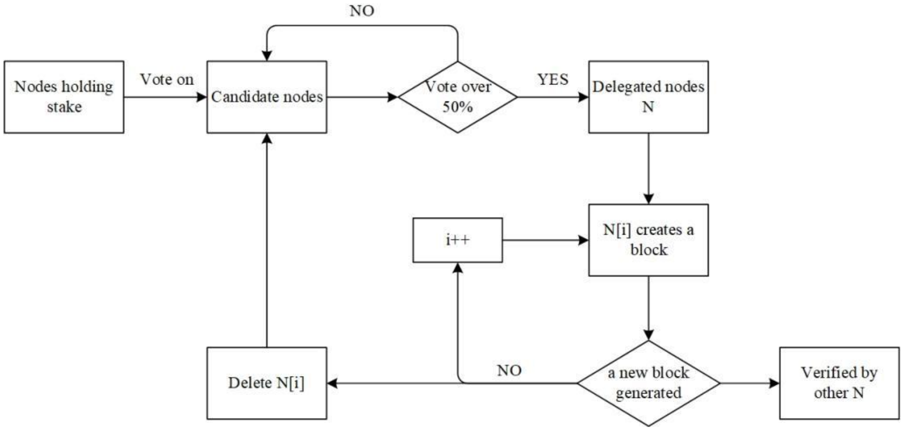

Figure 2. DPoS consensus algorithm flow chart

At present, the blockchain application systems that use the DPoS consensus algorithm mainly include Bitshares[16] (bit shares), Steem[17], and EOS[18], and the number of principals in both Steem and EOS is 21.

4.2 PBFT algorithm

The PBFT consensus algorithm is implemented in Fabric v0.6.0[19]. The consensus algorithm selects a master node from the entire network nodes to be responsible for creating blocks, and then reaches a consensus through three-stage voting: pre-preparation stage, preparation stage, and submission stage. As shown in Figure 3, the main process is as follows:

(1) Pre-preparation stage: select a master node from the whole network; each node broadcasts the transaction information sent by the client to the whole network, the master node collects all transaction information, creates a new block and broadcasts it to the whole network;

(2) Preparation stage: After each node receives the block information sent by the master node, it enters the preparation stage from the pre-preparation stage. The node verifies the block, and broadcasts a preparation message to other nodes after the verification is passed;

(3) Submission phase: The node enters the submission phase after broadcasting the preparation message to the whole network. If the node receives the preparation message of more than 2/3 nodes, it broadcasts a submission message to the whole network; if a node receives the submission of more than 2/3 nodes message, the new block can be submitted to the local blockchain, and a consensus on the latest height block can be reached.

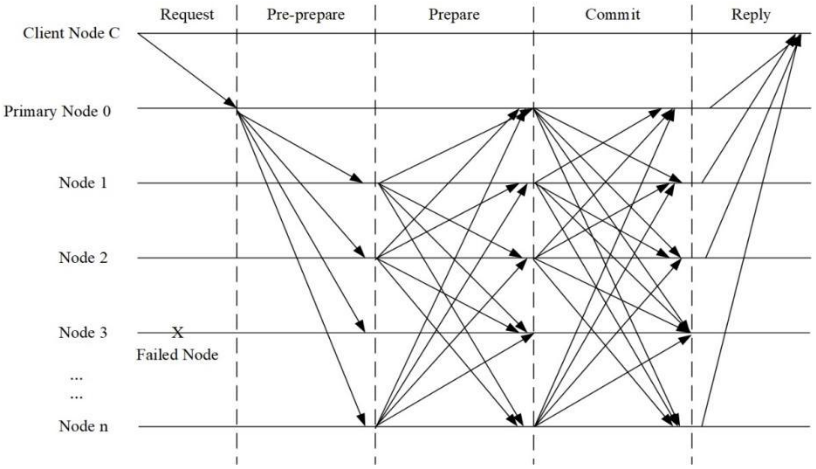

Figure 3. PBFT consensus process

PBFT can provide strong consistency and overcome the shortcomings of the original BFT algorithm, which is not efficient. The application of this consensus algorithm in the blockchain system can greatly improve the block generation speed of the system, and at the same time provide a certain degree of security, which can ensure the safe operation of the system when less than 1/3 of the nodes fail. However, this algorithm also has some shortcomings. Since the three-stage protocol requires nodes to broadcast messages to the entire system, if the number of nodes in the system is too large, the number of messages in the network will increase significantly, resulting in network congestion, so it is not suitable for large-scale Large-scale blockchain system.

4.3 DBFT algorithm

The DBFT algorithm is proposed in the NEO project[20]. DBFT is improved from the PBFT algorithm. The PBFT algorithm consensus process requires the participation of all nodes, and requires three rounds of network request confirmation to reach a consensus. The complexity of network communication is *O(N2)*, where N is the number of nodes in the entire network, and the scalability is poor. As the network size increases, it is difficult to reach consensus quickly. NEO's solution is to vote for some nodes to participate in the consensus, thereby reducing network communication consumption, improving scalability, and increasing transaction processing speed. The DBFT formula process is shown in Figure 4.

DBFT consensus process: All nodes will vote for a bookkeeping node (that is, a credible node, here is done according to the original PBFT algorithm), while other nodes are not bookkeeping, not Will participate in the consensus process, but accept the final consensus. Then, the speaker is selected from the collection of bookkeeping nodes (the speaker is the main node responsible for establishing blocks), and the remaining bookkeeping nodes are councilors; after each consensus, the speaker will propose a new block, and then Confirmation and broadcast by MPs. When a node receives the confirmation information sent by more than 2/3 of the bookkeeping nodes, it will add a block locally, that is, reach a consensus on the latest block; after that, select a new master node and carry out A new round of consensus. The advantages of the DPoS consensus algorithm are: no need to consume huge energy, higher operating efficiency, faster block generation, and less prone to forks. The disadvantage is: the degree of decentralization is not high, and the problem of bribery is prone to occur.

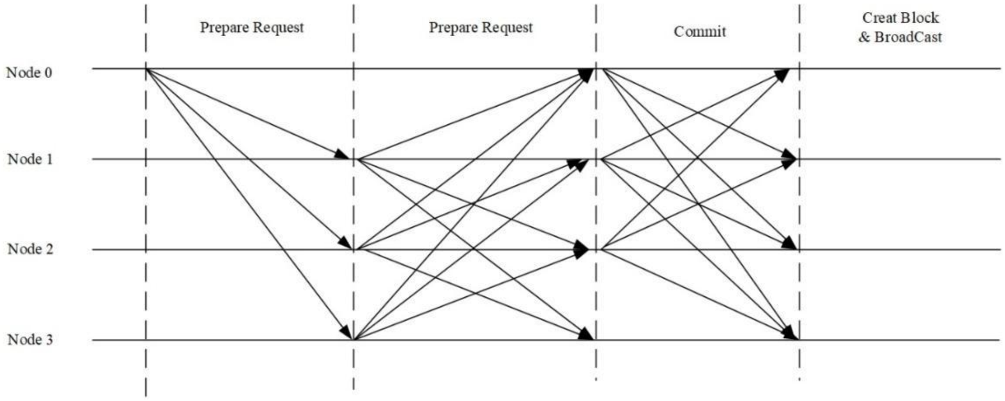

Figure 4. DBFT consensus process

5. PAXOS-LIKE CONSENSUS ALGORITHM

Raft[21] is a new and easy-to-understand distributed consensus replication protocol proposed by Diego Ongaro and John Ousterhout of Stanford University as the central coordination component in the RAMCloud project. As a replication state machine, Raft is the core and most basic component in a distributed system. It provides orderly replication and execution of commands among multiple nodes. When the initial states of multiple nodes are consistent, the state between nodes is guaranteed to be consistent. As long as the majority of nodes survive, the system can process normally. It allows message delay, discard and disorder, but does not allow message tampering (non-Byzantine scenario).

The core idea of Raft is to divide nodes into three categories: leaders, candidates, and followers. Usually, followers will copy the log of the leader node. The leader has its own term and continuously sends heartbeat packets to followers. If followers If no empty log heartbeat packet is received, the follower will change its status to a candidate. The candidate will increase the number of the previous leader and start voting. When more than N/2+1 nodes agree, it will become the leader node. As shown in Figure 5.

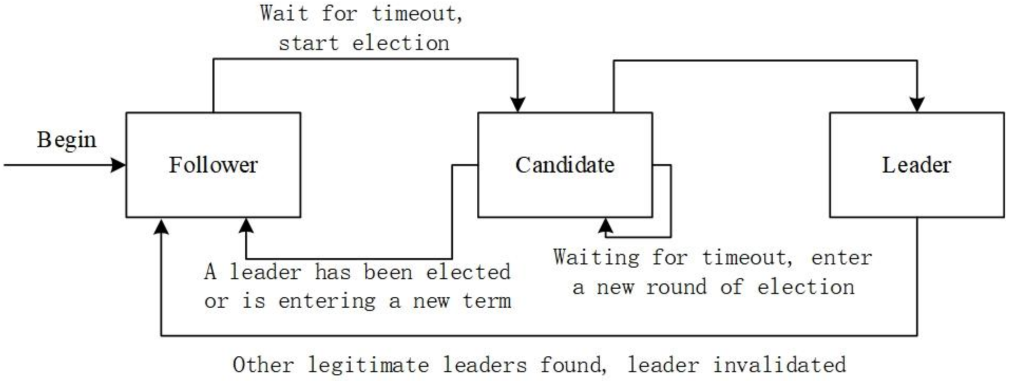
Wait for timeout,
start election
Begin
Follower
Candidate
Leader
A leader has been elected
or is entering a new term
Waiting for timeout, enter
a new round of election
Other legitimate leaders found, leader invalidated

Figure 5. Raft consensus process

The Raft algorithm is a consensus algorithm for a strongly consistent distributed system. It is improved on the basis of the Paxos algorithm, taking into account both efficiency and security. However, similar to Paxos, Raft only provides crash-tolerant performance. It cannot cope well with Byzantine errors in system nodes, which makes the Raft-based blockchain system vulnerable to network attacks.

6. CONCLUSION

This article compares 8 different consensus algorithms from 8 aspects, including degree of centralization, resource consumption, application scenarios, fault tolerance probability, consensus efficiency, and security. The comparison results are shown in Table 1.

Table 1. Comparison of Consensus Algorithms

	Proof			**Vote**			**Paxos-like**
	PoW	PoS	PoSpace	DPoS	PBFT	DBFT	Raft
degree of centralization	lower	lower	high	lower	middle	middle	high
fault tolerance probability	high	low	low	low	low	low	low
Resource consumption	49%	49%	49%	49%	33%	33%	49%
consensus efficiency	low	high	high	lower	higher	higher	high
safety	high	middle	high	higher	higher	higher	high
application scenarios	Public Blockchain	Public Blockchain	Public Blockchain	Public Blockchain	Consortium Blockchain	Consortium Blockchain	Consortium Blockchain

In recent years, although the consensus algorithm cannot be optimized in terms of security, efficiency, and decentralization, it can be explored and studied from the following aspects: (1) Use the idea of sharding to divide and conquer and group transactions Segmentation to achieve faster consensus speed. (2) The existing consensus algorithms tend to be mixed, and different consensus algorithms are used in a mixed manner, that is, the fusion of proof-type consensus algorithms and BFT-type consensus algorithms, and the combination of consensus algorithms with credit mechanisms and deep learning. (3) The results will be stored on the chain, and frequent operations will not be carried out on the blockchain, and offline networks such as the Lightning Network will be developed. The key to the innovation and application of blockchain technology is the consensus algorithm, which requires continuous exploration and research, and the existing consensus algorithm also needs to prove its effectiveness in continuous practice.

REFERENCES

[1] Nakamoto, S., Bitcoin: A peer-to-peer electronic cash system[J]. Decentralized business review, 21260 (2008).

[2] Deshpande, A., Stewart, K., Lepetit, L., et al., Distributed Ledger Technologies/Blockchain: Challenges, opportunities and the prospects for standards[J]. Overview report The British Standards Institution (BSI), 40: 40 (2017).

[3] Wang, Q., Li, F. J., Wang, Z. L., et al., Blockchain Principles and Key Technologies [J]. Computer Science and Exploration, 2020, 14(10): 1621 (2020).

[4] Yuan, Y., Ni, X., Zeng, S., et al., Blockchain consensus algorithms: the state of the art and future trends[J]. Acta Automatica Sinica, 44(11): 2011-2022 (2018).

[5] Deng, X. H., Wang, Z. Q., Li, J., et al., Comparative Research on Mainstream Blockchain Consensus Algorithms [J]. Computer Application Research, 39(1): 1-8 (2022).

[6] Lamport, L., The part-time parliament[M]//Concurrency: the Works of Leslie Lamport. pp. 277-317 (2019).

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

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

[9] Fischer, M. J., Lynch, N. A., Paterson, M. S., Impossibility of distributed consensus with one faulty process[J]. Journal of the ACM (JACM), 32(2): 374-382 (1985).

[10] Jakobsson, M., Juels, A., Proofs of work and bread pudding protocols[C]//Secure Information Networks: Communications and Multimedia Security IFIP TC6/TC11 Joint Working Conference on Communications and Multimedia Security (CMS'99) September 20–21, 1999, Leuven, Belgium. Springer US, pp.258-272 (1999).

[11] Dwork, C., Naor, M., Pricing via processing or combatting junk mail[C]//Advances in Cryptology—CRYPTO'92: 12th Annual International Cryptology Conference Santa Barbara, California, USA August 16–20, 1992 Proceedings 12. Springer Berlin Heidelberg, pp. 139-147 (1993).

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

[13] King, S., Nada,l S., Ppcoin: Peer-to-peer crypto-currency with proof-of-stake[J]. self-published paper, August, 19(1) (2012).

[14] Dziembowski, S., Faust, S., Kolmogorov, V., et al., Proofs of space[C]//Advances in Cryptology--CRYPTO 2015: 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part II. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 585-605 (2015).

[15] Larimer, D., Delegated proof-of-stake (dpos)[J]. Bitshare whitepaper, pp. 81: 85 (2014).

[16] Bitshares. What is bitShares? [EB/OL].(2019-01-01) [2021-10-02]. https://how.bitshares.works/en/master/technology/what_bitshares.html.

[17] Steem. Steem: an incentivized, blockchain-based, public content platform [EB/OL].(2019-06-01) [2021-10-02]. https://www.steem.com/steem-whitepaper.pdf.

[18] EOSIO.WhatisEOS?[EB/OL].[2021-10-02].https://eos.io/eos-public-blockchain/.

[19] Fabric official website. https://get.fabric.io/. Jan 2023.

[20] NEO offical website. https://docs.neo.org/v2/docs/zh-cn/tooldev/consensus/consensus_algorithm.html. Jan 2023.

[21] Ongaro, D., Ousterhout, J., The raft consensus algorithm[J]. Lecture Notes CS, pp. 190: 202 (2015).