Improved Practical Byzantine Fault Tolerance Consensus Mechanism with Two-stage Verification

Nigang Sun, Qiaosheng Hu*, Lidong Yao

Chanzhou University, Changzhou, Jiangsu Province, 213000, China
**Corresponding author*

Abstract: *Blockchain technology is widely used in finance, supply chain, Internet of Things and other fields because of its advantages of anti-tampering, decentralization, and traceability. As the core factor affecting the performance of blockchain, consensus algorithm with good performance is the current research focus and goal. Aiming at the problem of insufficient performance and scalability of Practical Byzantine Fault Tolerance (PBFT), a two-stage verification algorithm is proposed. The algorithm improves the three-stage verification of PBFT into the confirmation stage and the review stage. The block contains the confirmation information of the previous block, and the block release and information confirmation are carried out synchronously, which saves the communication cost and reduces the number of communications between nodes, so that the As the system throughput increases, the impact of network scale on performance becomes smaller. The simulation shows that the performance of the algorithm is improved by 50% compared with the PBFT algorithm. After the node reaches the maximum number of connections, the algorithm is limited by the size of the node and becomes smaller, and the scalability of the system is improved.*

Keywords: *Block chain; Consensus; Practical Byzantine Fault Tolerance; Distributed systems*

1. Introduction

The concept of blockchain technology first appeared in "Bitcoin: A Peer-to-Peer Electronic Cash System" published by Satoshi Nakamoto [1]. It integrates technologies such as cryptography, peer-to-peer connection, timestamp technology, distributed system, etc. It is a chain storage technology with the characteristics of decentralization, tamper-proof, traceability, etc which solve the centralization and trust problems that exist in real life. Since blockchain technology can build a decentralized and efficient trust system, it is widely used by many commercial organizations. The technological innovation and development of blockchain is also a strategic development goal at the national level [2]. In October 2019, the Political Bureau of the Central Committee of the Communist Party of China emphasized the important role of blockchain in a collective study.

In 1999, Castro et al. proposed the Practical Byzantine Fault Tolerance (PBFT) algorithm [3], which reduced the complexity of the Byzantine algorithm to the polynomial level, making it into the practical stage. Although this algorithm increases the throughput of the system to a certain extent, as the scale of the blockchain increases, the delay for the system to reach a consensus increases exponentially, which affects the efficiency of the algorithm [4]. In response to the system efficiency and scalability issues of PBFT, Delegated Byzantine Fault Tolerance (DBFT) algorithm was proposed in the Onchain white paper in 2016[5]. The algorithm follows the rules of the PBFT algorithm, but does not require all nodes participate in the voting process, and some nodes are selected as representatives to participate in the voting, which effectively reduces the system consensus delay, but at the same time brings the problem of centralization. In 2019, Li Junqing et al. proposed a reputation-based dynamic authorization PBFT consensus mechanism [6], Chen Zihao et al. proposed an improved PBFT consensus mechanism based on K-medoids [7], and in 2020 Wang Juan et al. proposed blockchain dynamic authorization. The consensus mechanism [8] is to dynamically elect the consensus nodes participating in the PBFT algorithm and reduce the proportion of participating consensus nodes to improve performance and scalability. In 2018, Lei K, Zhang et al. proposed Reputation-Based Blockchain Byzantine Fault Tolerance (RBFT) [9], which randomizes the selection of master nodes in PBFT and adds the function of node voice, which makes the system performance and security. However, RBFT needs to introduce a trusted NTP (Network Time Protocol) server in actual operation, which reduces the decentralized characteristics of the blockchain. Attacks on the server by malicious nodes may lead to system paralysis, and it is impossible to generate new blocks; the sharding consensus scheme of Zamani M et al [10] is to divide the blockchain system into multiple sub-partition systems, each of which can independently verify information and reach a consensus. By increasing the number of shards in the network, linearly increases the throughput of the entire blockchain network [11]. In 2019, Xu Zhi et al. proposed a scheme that simplifies the PBFT consensus protocol and cooperates with credit points to rate nodes to reduce communication overhead [12]. Most of the follow-up research on blockchain performance is the fusion of existing consensus algorithms and the pluggability of consensus algorithms, which has not made a breakthrough in blockchain performance [13][14].

Although the above algorithm improves the performance of the PBFT algorithm to a certain extent, there are still the following problems: (1) The efficiency of the algorithm: a faulty node will destroy the security of the system. Therefore, in order to ensure the security of the system, most of the PBFT and its improved algorithms are the three-stage verification, and requires two-thirds of the nodes to agree each time, the communication cost is extremely high, and the overall efficiency of the system is not high. (2) The scalability of the system: Each stage of the PBFT algorithm requires the consent of two-thirds of the verification nodes. With the increase of the number of nodes, the communication cost of the system will increase at a polynomial level, and the scalability of the system is low.

Aiming at the above problems, this paper proposes a two-stage verification KCBFT (K-confirmation Byzantine Fault Tolerance) algorithm. The algorithm is obtained by improving the three-stage verification method of PBFT and the threshold of confirmation information. The application scenario of the algorithm is the alliance chain. The alliance chain restricts the participating nodes, which is beneficial to the security of the blockchain network. KCBFT improves the verification threshold of each stage from one-third of all nodes to an integer K, and improves the verification of each block K times in the three-stage verification to forward verification of K blocks. The Prepare phase and Commit phase in the verification are carried out at the same time, forming the K confirmation phase of the algorithm. This stage can reduce the communication cost, improve the efficiency of the system, and make the impact of network scale on performance smaller. The algorithm is equipped with a review phase to safeguard against security and liveness issues that may arise after threshold changes.

2. Related Technology

Byzantine Fault Tolerance (BFT) originates from the Byzantine Generals Problem and is a fault-tolerant technology in the field of distributed computing. The blockchain network environment conforms to the Byzantine generals problem model. There are normal nodes, faulty nodes and malicious nodes in the network. The normal nodes are usually called non-Byzantine nodes, and the faulty nodes are called Byzantine nodes. The core of the consensus algorithm is to communicate between normal nodes. A consensus is reached on the state of the network.

The PBFT algorithm is the most representative consensus algorithm in BFT. The algorithm is divided into four stages: the request stage, the pre-preparation stage, the preparation stage, and the submission stage. In the algorithm, all nodes must participate in the voting process in order to add the next block. PBFT can tolerate one-third of the wrong behavior [15]. Compared with the proof of work, the consensus speed is faster, and the throughput per second can reach thousands. The steps of the algorithm [16] are shown in figure 1:

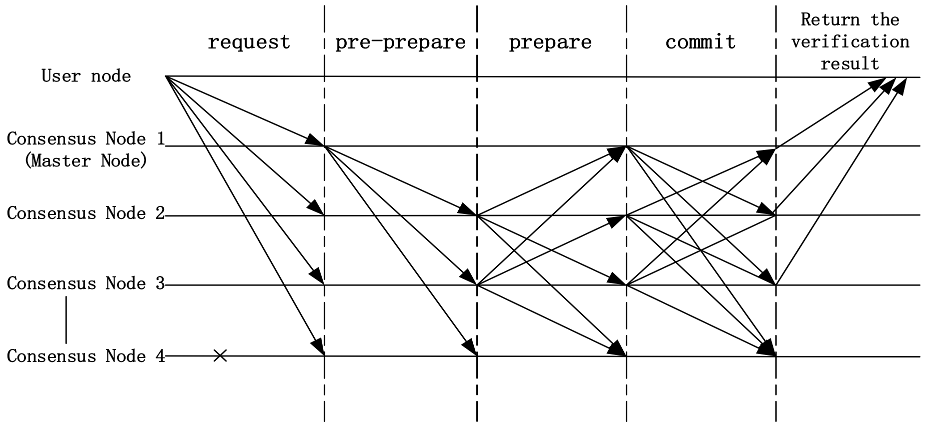

Figure 1: PBFT algorithm steps

Step 1: Request phase: The user node sends a transaction request to the consensus node set.

Step 2: Pre-preparation stage: After the master node receives the transaction request, it verifies the correctness of the signature. If the verification is successful, the request will be numbered, and then the transaction and its number will be broadcast to the consensus node set.

Step 3: Preparation phase: The consensus node judges the received information, checks the correctness of the signature, transaction and view information, and if it is correct, agrees to the received transaction and its number, and then broadcasts the node's preparation message for the transaction and number. When the number of corresponding preparation messages received by the consensus nodes is not less than (2*n)/3 (n is the total number of nodes in the consensus set), the preparation phase ends, and the verifier enters the submission phase.

Step 4: Commit stage: The consensus node broadcasts the submission message to the consensus set, and other nodes verify the correctness of the message after receiving the message, retain the legal message, and discard the illegal request. If the verifier receives (2*n)/3 +1 (including itself) commit messages, the consensus node records the verification result of the transaction on its own blockchain.

There is a concept of view under the PBFT algorithm. In the view protocol view, there is a master node, and the other nodes are slave nodes. The master node is responsible for ordering the client's requests to the slave nodes. If the master node is down or the master node is a Byzantine node, the slave node will trigger an attempt to change the protocol view change protocol to elect a new master node.

3. Algorithm Design

3.1. KCBFT algorithm design

The Byzantine fault-tolerant nodes that the KCBFT algorithm can tolerate is (N-1)/3, where N is the total number of nodes in the system, and the time complexity is O(N^2). Its process is shown in the following figure:

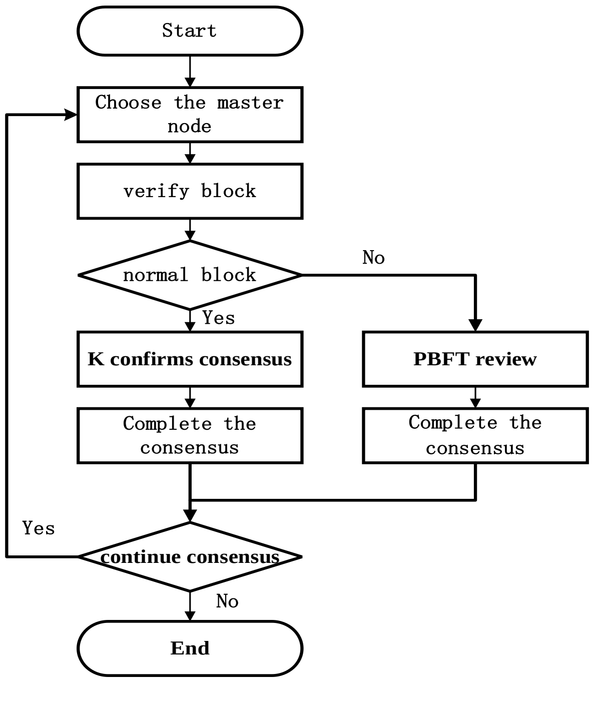

Figure 2: Flow chart of each round of KCBFT

The KCBFT algorithm is inspired by the threshold optimization of the three-stage verification of the PBFT algorithm: in the alliance chain, nodes do not join at will, but have certain restrictions, so the number of Byzantine nodes in the alliance chain is less than one-third of the total nodes , in this relatively safe situation, the consensus process of nodes does not require ((2 N /3) +1) confirmation information, and non-Byzantine nodes only need to confirm that the number of information reaches a certain threshold to be considered that most nodes approve In order to confirm the information, the nodes reach a consensus.

The threshold is an integer less than (N-1)/3, and the integer is set as K in the algorithm. Further, the algorithm optimizes the confirmation of each block K times to include the verification of the previous K blocks when each block is released, and combines the confirmation information with the release of the block to reduce the communication cost between nodes. When a block has K sub-blocks for the first time, the node believes that the block has reached a consensus and adds it to the local chain, and the block is jointly responsible by the K block publishers behind. In order to ensure the security of the system, the algorithm is equipped with a review stage. After a block reaches a consensus, the block can still be questioned by other nodes. After receiving the questioning information, the master node in the rear will initiate a review of the questioned block. The review process is determined as a PBFT consensus for all nodes in the network. The security of the PBFT algorithm has been proven, so it can be determined that the block under review is secure, and the result of the review is the final result of transaction verification.

3.2. Algorithm Details

The KCBFT algorithm needs to go through two stages. The first stage is the preparation stage (K confirmation stage). The preparation stage and the preparation stage in PBFT are carried out at the same time. When the block has K sub-blocks, the preparation stage is completed. The second stage is the submission stage (review stage). The submission stage is to ensure the security and activity of the algorithm. After T confirmed blocks in the submission stage, the submission stage is completed and the block is written into the blockchain. A block goes through four states from being published to being successfully submitted. When the number of confirmation messages is 1~K-1, the block is called a pre-prepared block. After there are K verification messages, the state of the block is is called a preparation block. When the confirmation information of a block is K~K+T-2, the block is called a prepared block. When there are K+T-1 confirmation information in a block, the block status becomes a submitted block. After submitting the block information and receiving a confirmation message, the user can consider the block to be a legal block and write the block into the blockchain.

Preparation stage (K confirmation stage): The preparation stage is the stage when the transaction is packaged and published by the master node, and the verification information is released after being verified by other nodes until there are K verification information. The consensus process in the K verification phase:

Step 1: Pick the master node among all nodes.

Step 2: The user node sends a transaction request and broadcasts it to all nodes.

Step 3: All consensus nodes receive transaction information, check the validity of the transaction, keep valid transactions, and discard invalid transactions. The master node packages the local transaction and links the latest block with the pre-prepared block. The link successfully puts the new block in the cache, updates the status of the earliest pre-prepared block to the pre-prepared block, and broadcasts the zone to all nodes yuan.

Step 4: The master node is updated. The new master node verifies the received block after receiving the block. If the verification is successful, it will be added to its own cache and marked as a pre-prepared block. If the verification is invalid, the block will be blocks are discarded.

Step 5: Update the status of each block and write the block to the local blockchain.

Submission phase (review phase): After the preparation phase is completed, the node can confirm that there are K nodes that have not only acknowledged the correctness of the block, but also completed the confirmation of the updated block, that is, the pre-preparation phase and the preparation phase have been completed at the same time. The commit phase is to deal with security problems that may arise after the modification of the threshold, and to give users fault tolerance time to ensure the consistency of the system. If no user has challenged the block after T blocks, the block can no longer be challenged in the blockchain, and the submission phase is completed. If there is a user's question during this period, the review stage will be executed.

Review phase consensus process:

Step 1: The node questions the transaction information in a prepared block and broadcasts the questioning information to the network.

Step 2: The current master node receives the questioning information of the transaction information, locates and broadcasts the censored block, and broadcasts the questioned transaction to the whole network.

Step 3: The verification node performs two-stage verification after receiving the broadcast. The steps are the same as the preparation stage and the submission stage in PBFT. After the submission stage, the result is recorded in the current block. At this time, all nodes reach a consensus result.

3.3. Algorithm Analysis

The security of a distributed system means that all bad things will not happen. In the KCBFT algorithm, it means that all non-Byzantine nodes will eventually be consistent. The security analysis of the algorithm is mainly divided into two types: the master node is Security issues when the Byzantine node and slave nodes are Byzantine nodes.

3.3.1. Security Analysis

When the master node is a Byzantine node: node A, node B, node C (hereinafter referred to as A, B, C) are the master nodes of the three rounds of consensus in turn, where A is a Byzantine node, B and C are non-Byzantine nodes. Since K blocks are confirmed forward, the communication between each node is closed. When broadcasting a block, multiple conflicting blocks may be broadcast at the same time, so as to achieve the purpose of attack. When A is the main node, broadcasting different blocks to different nodes, broadcasting block X to B, and broadcasting block X' to C, B cannot know the existence of X' after receiving X. When B becomes the master node, block Y will be published after the received block. When C serves as the master node, it will receive two chains at the same time, the X' chain issued by A and the X and Y chains issued by B. At this time, C can easily verify the correctness of X and make a choice, and if X is correct, it can be concluded that A It is a Byzantine node when it publishes a conflicting block when serving as the master node; if X is forged, it can be concluded that B is a Byzantine node, C selects a longer chain to publish block Z and broadcasts the information of the Byzantine node at the same time, and others receive the conflict. The node of the message discards the original block X' and adds the longest legal block chain to its own block, so as to ensure the correctness of the block chain. The process is shown in Figure 3.

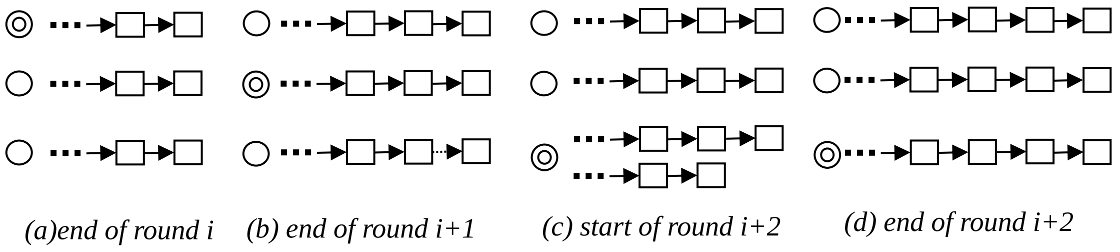

Figure 3: Three consecutive rounds of consensus process

When the slave node is a Byzantine node: the consensus is confirmed by the master node in each round, and whether the slave node is a Byzantine node has no effect on the security of the consensus.

In extreme cases, a situation that may occur is: consecutive K nodes are malicious nodes, and a malicious block confirmed by K is written into the blockchain, so the number of confirmations for each block reaches K There is still a fault tolerance time T after the number of confirmations, that is, within the fault tolerance time T after K confirmations are reached, other nodes can still challenge the block, and the questioned block requires all nodes to conduct PBFT consensus. The results are recorded in the blockchain to achieve a consistent state for all nodes.

3.3.2. Activity Analysis

Liveness means that all good things will happen. The main reason for affecting liveness in the blockchain is that the communication between each network node is delayed, so that all nodes cannot reach a consensus and cannot produce consistent results.

The algorithm sets a confirmation phase and a review phase. The confirmation phase can ensure that the blocks issued by non-Byzantine nodes can reach K verifications and be written into the blocks. If there are no malicious blocks, all nodes will be consistent. The review will only be initiated on blocks with wrong transactions published by malicious nodes, and will not affect the generation of normal blocks. Therefore, within a certain period of time, the correct transaction will be written into the block.

Due to the communication delay between network nodes, in the case of some network crashes, most nodes may not be guaranteed to have the same block height. Although nodes find a shorter chain and broadcast, it is still not sure that most nodes receive the same block height. For this transaction, only the node will review the block published by the Byzantine node, and the block height can be synchronized at this time. In order to cope with this situation, it is set to synchronize every 100 blocks, that is, when the block released by the master node is an integer multiple of 100, a PBFT consensus of the whole network is carried out, so as to ensure that most nodes are not in the review stage. have the same block sequence.

4. Simulation Experiments

Reference [17] summarizes and compares various consensus algorithms and various evaluation methods in the network. The throughput and security of the algorithm are the main factors for evaluating the consensus algorithm. Therefore, the simulation experiments in this paper are mainly to evaluate the performance of the KCBFT algorithm. and safety testing. The simulation test is written in C++, and multi-threading is used to simulate different nodes. In the simulation test, a node does not interrupt sending transactions, and the consensus node and the master node agree on the transaction; based on ns-3, an open source simulator, through the command line Specify the data to complete the simulation of the data. The confirmation time of the transaction is ignored during the simulation. The propagation of blocks of different sizes among multiple nodes can be simulated. After the data is obtained, it is compared with the PBFT algorithm completed in the same configuration environment. The test is mainly divided into two parts. The first part is the performance test, which compares the transaction throughput (transactions per second, TPS) of the algorithm with other algorithms.

Where transactions is the total amount of transactions processed and t is the time taken. The second part is the selection of K value. This part mainly determines the K value from the proportion of malicious nodes and delay.

4.1. Algorithm throughput and scalability

The throughput test of the algorithm is carried out in an environment where the proportion of malicious nodes is 33%, the total number of nodes are (4, 7, 10, 13, 16, 19, 22). The confirmation time of the transaction is not included in the test, only the consensus time of all non-Byzantine nodes is recorded. After multiple experiments, the TPS of the two algorithms at different nodes is recorded. Figure 4. is part of the experimental data, and the average value is taken.

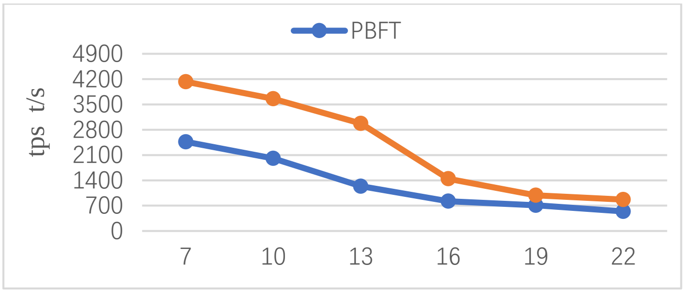

Figure 4: Algorithm throughput test results

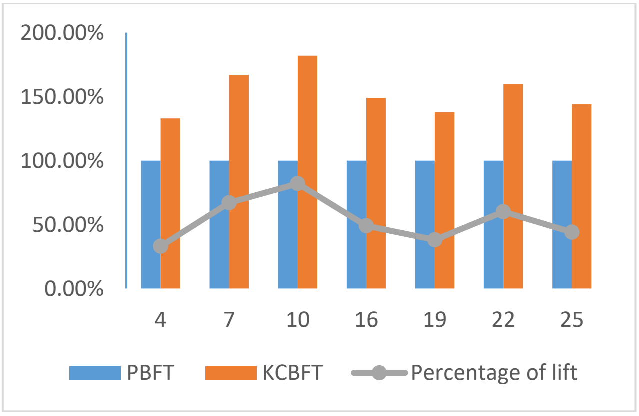

Figure 5: The throughput of KCBFT compared with other improved PBFT consensus mechanisms

Figure 4 shows the TPS comparison of PBFT algorithm and KCBFT algorithm under different scales. It can be seen from the figure that when the network scale increases, the TPS of the two algorithms shows a downward trend, and both are affected by the network scale. Under the same network scale, KCBFT has different degrees of improvement compared with PBFT, and when there are many nodes, The TPS of KCBFT can still maintain a 30% improvement. After many tests, the KCBFT algorithm can improve the TPS by more than 50% compared with the PBFT algorithm. Figure 5 shows the average TPS improvement of each algorithm compared with the PBFT algorithm. It can be seen from the figure that the KCBFT algorithm has improved by more than ten percentage points compared with other algorithms.

Due to the limitation of the hardware environment, when the total number of nodes is 25, the computer can no longer process all the messages of the PBFT algorithm, and the K confirmation stage of KCBFT can still achieve normal consensus. Figure 6 shows that the total number of nodes is (25, 28, 31, 34, 37, 40, 43) system throughput performance. The black dots in the figure represent the scatter plot composed of five experimental data, and the triangles in the figure represent the average data of multiple experiments. It can be seen from the figure that when the total number of nodes is more than 25, the PBFT algorithm cannot process all the data in the test environment. The transaction throughput of the KCBFT algorithm still maintains a transaction throughput of 100 when the number of nodes is 43. In the figure, a trend line is drawn for the average transaction throughput of the KCBFT algorithm. Although the algorithm shows a downward trend as a whole, the downward trend is gradually decreasing, and more nodes can be accommodated to participate in the consensus. Through the test of the participating consensus nodes that can be accommodated by the KCBFT algorithm and the PBFT algorithm in the same environment, it can be concluded that the KCBFT algorithm can accommodate more nodes to participate at the same time, and has better scalability.

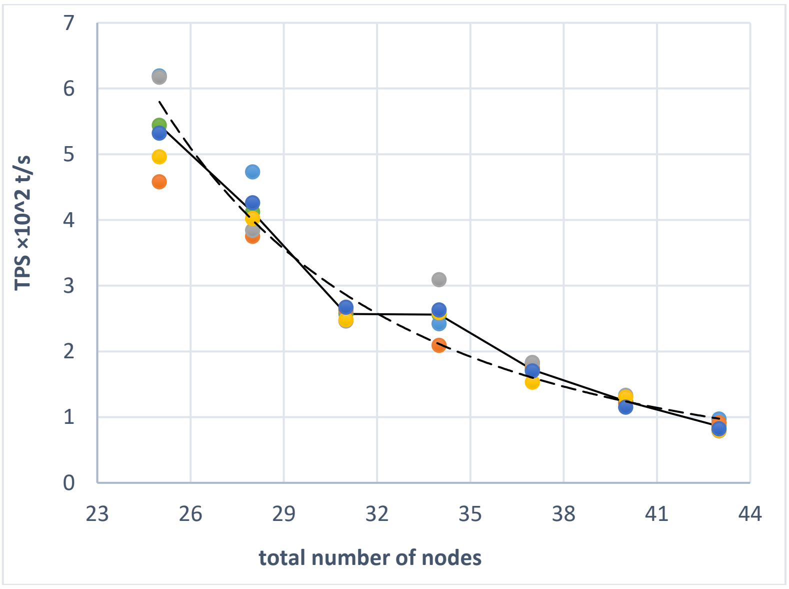

Figure 6: TPS of different nodes in KCBFT

4.2. Security and Latency of Algorithms

In the environment where the number of malicious nodes in the KCBFT algorithm is (33%, 25%, 20%, 16.70%, 14%, 12.50%, 11.10%, 10%, 5%), several simulation tests were carried out, each time the test created one hundred blocks, recording the number of malicious node blocks and the number of longest forks in it. After multiple tests, it is found that the number of blocks generated by malicious nodes is positively correlated with the probability of malicious nodes, and the proportion of the two to the total number is basically the same. The number of longest forks decreases as the proportion of malicious nodes decreases. When the proportion of malicious nodes is 33%, the average length of the longest malicious blockchain is 6.33, and the longest length is 7. At the same time, after experimental testing, the increase of the K value is proportional to the consensus delay of the algorithm. The larger the selected K value, the greater the consensus delay of the system. After considering the influence of K value selection on security and consensus delay, the K value is tentatively set to 12 to ensure the security of the system and prevent malicious nodes from adding malicious blocks to the blockchain. And when the value of K is 12 and there are 22 nodes in total, the consensus delay of the algorithm is 28.51 seconds, which is within an acceptable level.

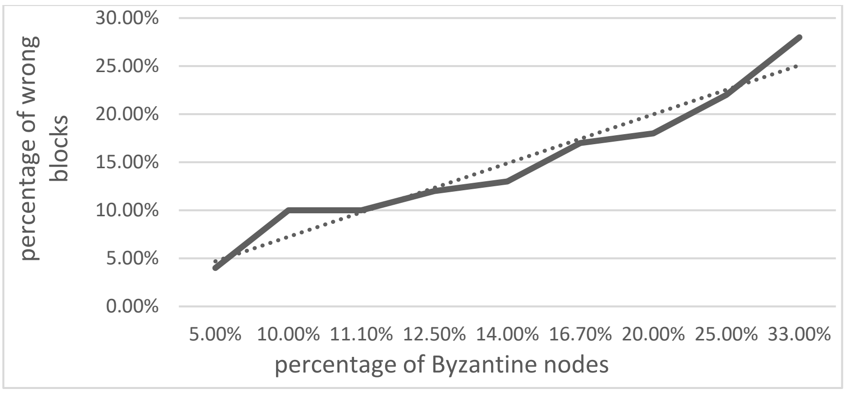

Figure 7: Longest bifurcation length

5. Conclusions

This paper proposes a fast-confirmation consensus algorithm KCBFT mainly for the low performance of the alliance chain consensus algorithm. By improving the consensus threshold of the PBFT algorithm and the three-stage verification method, the network resource consumption during consensus is reduced and the system is improved. Scalability, through the forward K confirmations, the message confirmation in PBFT is converted from breadth to depth, and a small confirmation delay is added to improve the performance of the algorithm. The simulation test shows that the KCBFT algorithm can resist the attack of malicious nodes, and the TPS increases by more than 50% in the system where the proportion of malicious nodes accounts for 33%. The K value selection in the KCBFT algorithm is positively correlated with the confirmation delay. The larger the K value, the higher the confirmation delay. At the same time, the proportion of malicious nodes in all nodes will affect the efficiency of the system. Therefore, the follow-up work mainly focuses on the selection and design of the K value in different scenarios, and A reward and punishment mechanism filters nodes, so as to reduces Byzantine nodes in the nodes,and to improve the performance of the algorithm.

References

[1] NAKAMOTO S. Bitcoin: a peer to peer electronic cash system. [EB/OL], [2022-03-04]. https:// bitcoin.org/bitcoin.pdf
[2] PISA, MICHAEL, MATT J. "Blockchain and economic development: Hype vs. reality." [EB/OL], [2022-03-04] https://www.cgdev.org/publication/blockchain-and-economic-development-hype-vs-reality.
[3] LIU Y H, CHEN K. New progress of blockchain consensus mechanism [J]. Application Research of Computers, 2020, 37(S2): 6-11.
[4] LIN I C, LIAO T C. A survey of blockchain security issues and challenges [J]. IJ Network Security, 2017, 19(5): 653-659.
[5] NEO White Paper, [EB/OL], [2022-03-04] https://docs.neo.org/v2/docs/zh-cn/basic/ whitepaper. html
[6] LI J Q, XIN Y S, SONG C Q, et al. PBFT consensus mechanism for dynamic authorization based on reputation [J]. Software, 2019, 40(05): 1-7.
[7] CHEN Z H, LI Q. Improved PBFT consensus mechanism based on K-medoids [J]. Computer Science, 2019, 46(12): 101-107.
[8] WANG J, LIU W B, JI L L. A consensus mechanism for blockchain dynamic authorization [J]. Journal of Heilongjiang University of Science and Technology, 2020, 30(02): 193-199.
[9] LEI K, ZHANG Q, XU L, et al. Reputation-based byzantine fault-tolerance for consortium blockchain [C]. IEEE, 2018: 2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS). Piscataway, NJ, 2018: 604-611.
[10] ZAMANI M, MOVAHEDI M, RAYKOVA M. Rapidchain: Scaling blockchain via full sharding [C]. CCS, 2018: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM, 2018: 931-948.
[11] BUTERIN V, GRIFFITH V. Casper the friendly finality gadget [J]. arXiv preprint arXiv: 1710.09437, 2017.
[12] XU Z L, FENG H M, LIU B. An improved PBFT efficient consensus mechanism based on credit [J].

Application Research of Computers, 2019, 36(09): 2788-2791.
[13] ZHENG Z, XIE S, DAI H, et al. An overview of blockchain technology: Architecture, consensus, and future trends[C]. IEEE, 2017: 2017 IEEE international congress on big data (BigData congress). Piscataway, NJ, 2018: 557-564.
[14] SUKHWANI H, MARTÍNEZ J M, CHANG X, et al. Performance modeling of pbft consensus process for permissioned blockchain network (hyperledger fabric) [C]. Piscataway, NJ: IEEE, 2017: 2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS). 253-255.
[15] RANA T, SHANKAR A, SULTAN M K, et al. An Intelligent approach for UAV and drone privacy security using blockchain methodology[C]. IEEE, 2019: 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence). Piscataway, NJ, 2019: 162-167.
[16] GAO L F, HU Q G. Byzantine Algorithm of Blockchain Consensus Mechanism [J]. Digital Communication World, 2019(01): 43-49.
[17] Fu X, H Wang, Shi P. A survey of Blockchain consensus algorithms: mechanism, design and applications[J]. Science China Information Sciences, 2021, 64(2):1-15.