Research on Practical Byzantine Fault Tolerant Consensus Algorithm Based on Blockchain

Xiandong Zheng and Wenlong Feng*

School of Information and Communication Engineering, Hainan University, Haikou, China

*Corresponding author: wenlong_feng@hainanu.edu.cn

Abstract. Practical Byzantine Fault-tolerant Algorithm (PBFT) is the most widely used consensus algorithm in alliance chain, which has the advantages of fault tolerance and large throughput. However, PBFT also has some problems that can't be ignored in specific blockchain applications, such as bad behavior of master node, high network communication overhead and low system flexibility. In this paper, the improvement measures of PBFT consensus algorithm are summarized from many aspects, and then compared with traditional PBFT algorithm. Finally, the development direction and trend of PBFT algorithm are prospected, hoping to provide reference for the research and innovation of PBFT consensus algorithm in the future.

Keywords: Blockchain, consensus algorithm, PBFT algorithm.

1. Introduction

Blockchain is a decentralized distributed ledger technology realized by the deep integration of distributed data storage, consensus algorithm, encryption algorithm and other technologies [1], which provides a way to conduct information value transfer and exchange in an untrusted network. Among them, the consensus algorithm is a core part of the blockchain technology, and it is the basis for building the trust characteristics of the blockchain [2]. At present, consensus algorithms are divided into three categories: public blockchain consensus, private blockchain consensus and consortium blockchain consensus according to the type of blockchain [3]. The comparison is shown in Table 1. In the specific application of blockchain, consortium blockchain is more suitable for various application scenarios, so this article mainly focuses on the PBFT algorithm used by the consortium blockchain.

Table 1. Blockchain classification comparing.

|                      | Public blockchain | private blockchain | consortium blockchain |
|----------------------|-------------------|--------------------|-----------------------|
| Centralization       | No                | Yes                | Yes                   |
| Consensus            | POW               | RAFT               | PBFT                  |
| Participant          | Everyone          | Control centre decision | Pre-defined members |
| Advantage            | 1. Open to any user | 1. No danger of attack | 1. Access control |
|                      | 2. Easy to deploy | 2. Low energy consumption | 2. High scalability |
| Disadvantage         | High energy consumption | Node limited | Conspiracy attack |
PBFT algorithm is the most popular algorithm in the consortium blockchain. Compared with other consensus algorithms, PBFT is widely used in the consortium blockchain because of its fault tolerance and good effect [4]. However, the PBFT algorithm also has its own limitations, such as poor system flexibility, nodes cannot join and exit freely; the three-stage broadcast process consumes bandwidth, which is likely to cause network congestion; the selection of master nodes is arbitrary, which has security risks [5]. In view of the above shortcomings, many scholars have done related researches [6-8].

The article first introduces the principle and process of the traditional PBFT algorithm. Then, in view of the above-mentioned shortcomings of the PBFT algorithm, different improvement algorithms are described in detail from multiple improvement directions. Finally, according to the current research hot issues, the prospect of the future research direction of the PBFT algorithm is proposed.

2. Overview of PBFT Algorithm

The PBFT algorithm is a consensus algorithm for solving Byzantine problems first proposed by Miguel Castro and Barbara Liskov [9] in 1999. The time complexity of the original Byzantine algorithm is reduced from the exponential level to the polynomial level, thus making it possible to solve the Byzantine problem. The problem has moved from the theoretical level to the practical application level. PBFT is an algorithm that solves the consistency of state machine copies in a distributed scenario [10].

The core concept in the PBFT algorithm has three parts, view, replica, role [11]. The view represents the global state of the current system. The participating nodes in the system become replicas. In each view, the roles in replicas are divided into two categories. One of the replicas serves as the primary, and the other replicas serve as backups.

The PBFT consensus algorithm [12] is mainly composed of a consensus protocol, a view replacement protocol and a checkpoint protocol. The consistency protocol is used to ensure the consistency of the data saved by all nodes in the entire network, which is realized through the mutual communication between the three-stage nodes; the view replacement protocol is used to replace the faulty node to ensure the normal operation of the system; the checkpoint protocol is used for regular cleaning Expired interactive data reduces node storage pressure, regularly checks whether the system is unified, and synchronizes inconsistent nodes.

The consensus protocol is the core protocol for the PBFT algorithm to complete the consensus process. The execution process is shown in Figure 1. The consensus protocol is mainly divided into the following three processes:

1. Pre-prepare: After the master node receives the service request message and verifies it is correct, it generates a pre-prepare message according to the service request message and broadcasts it to the slave nodes.
2. Prepare: After receiving the pre-prepare message from the master node, the slave node verifies whether the message content has been tampered with. After the content is verified correctly, the slave node will generate a prepare message according to the pre-prepare message and broadcast it to all replica nodes.
3. Commit: Broadcast commit messages to other nodes. After receiving $2f + 1$ commit messages (including themselves), consensus has been reached, and the nodes will execute the request and write data.

3. Improved PBFT Algorithm

In the consortium blockchain application scenario, the PBFT algorithm is the most commonly used consensus method. Compared with other consensus algorithms, the PBFT algorithm does not need to consume a large amount of computing resources, and the advantages of faster consensus speed have been widely used [13-14]. However, the PBFT algorithm also has its own shortcomings. Therefore, many scholars continue to propose improved algorithms to overcome these shortcomings. At present,
the improvement methods of the PBFT algorithm mainly focus on the following directions: reducing network traffic, selecting master nodes, and improving system flexibility.

3.1. Reduce Network Traffic

It can be seen from Figure 1 that when the PBFT algorithm executes the consensus protocol, there will be a lot of communication between nodes. With the increase in the number of nodes and the number of transactions, the network traffic will increase rapidly, and the pressure of increasing bandwidth will easily cause network congestion, thereby affecting consensus efficiency of the PBFT algorithm. As early as 2007, Ramakrishna Kotla [15] proposed the Zyzzyva consensus optimization protocol in the absence of Byzantine nodes, as shown in Figure 2, reducing the complexity of the consensus process from $O(N^2)$ to $O(N)$. This also provides a direction for the improvement of the traditional PBFT algorithm. Based on the characteristics of the alliance chain, Wang Ziyue et al. [16] believe that the nodes in the consortium blockchain system are authorized and authenticated, and there is no Byzantine node, and use an optimized consensus protocol to reduce the amount of communication between nodes.

\[
d(v_i, v_k) = \sqrt{\sum_{j=1}^{p} (v_{ij} - v_{kj})^2}
\]  

(1)

![Figure 1. PBFT algorithm execution process.](image1)

![Figure 2. The Zyzzyva protocol execution process.](image2)
The large-scale network nodes participating in the blockchain consensus are clustered and grouped according to their characteristics. Figure 3 shows the clustering results.

![Figure 3. The clustering results. (a) k=4; (b) k=5.](image)

The clustered nodes are finally grouped for consensus to achieve the purpose of reducing the amount of communication.

Literature [20] introduces the concept of reputation value to evaluate the behaviour of consensus node, and uses it to measure whether a node is credible and reliable enough, and then the consensus node is hierarchically based on reputation value to reduce network communication between nodes. The reputation value is calculated as follows:

\[ R_{v+1} = R_v + a \log(R_v), 0 < a \leq 0.1 \]  
(2)

\[ R_{v+1} = bR_v, 0 < b < 1 \]  
(3)

Here, (2) represents the calculation formula when the reputation value increases, and (3) represents the calculation formula when the reputation value decreases, where \( R_v \) represents the reputation value of node \( v \), and \( a \) and \( b \) represent adjustment factors.

Once the wrong node is found, the network structure is dynamically adjusted. Compared with the traditional PBFT algorithm, the above-mentioned improved algorithm reduces the overall scale of the consensus by grouping and layering the consensus nodes of the entire network, reducing the number of communications between the three-phase communication between nodes, thereby reducing the bandwidth and computing resource overhead.

3.2. Optimize the Leader Node Selection

In the consensus process of the traditional PBFT algorithm, all consensus nodes have equal chances of being elected as the leader node. If the newly elected leader node is a dishonest node, it will cause continuous selection of the leader node, and the external services of the entire blockchain system during the selection of the leader node will be greatly reduced or even unable to provide services normally [21].

Literature [22] introduced a voting mechanism for the random selection of leader nodes in the traditional PBFT algorithm, and the leader node with the most votes was elected. Literature [23] believes that the voting mechanism requires enough nodes to participate in voting, so it adds a reward and punishment mechanism to improve it, rewarding nodes that vote correctly, and punishing nodes that vote incorrectly. Thereby improving the initiative of trusted nodes and reducing the participation of abnormal nodes. Literature [24] believes that there may be collusion among nodes participating in voting, which affects the fairness of voting. Blind signature technology is used to protect the privacy of voting nodes and threshold signatures to ensure the correctness of the counting process and ensure...
the high credibility of the results. Literature [25-27] introduced a credit model [28] to examine the reliability of consensus nodes participating in the blockchain network. In this model, the trust value of each node can be effectively evaluated, and security can be improved by monitoring dishonest nodes. It obtains a unique trust value for each node in the system by recording historical transactions.

The higher the trust value of the node, the lower the probability of error, that is, the higher the reliability, the greater the probability of being selected as a consensus node in the next stage [29]. Compared with the traditional PBFT algorithm, the above improved algorithm uses the node credit model to calculate the reliability of the node. It is believed that the higher the node credit value, the more reliable the node, which effectively avoids the possibility of the Byzantine node becoming the leader node, thereby it can reduce the number of view changes and improve consensus efficiency.

3.3. Improve system flexibility
At present, traditional PBFT is widely used in consortium blockchain, and the most important problem facing the application is the dynamic nature of nodes. The number of nodes in the initial system is fixed, and the increase or decrease of nodes requires the initialization of the consensus, which greatly reduces the flexibility of the system [30]. Aiming at the problem that nodes cannot dynamically join and exit, literature [31] uses the technical characteristics of smart contracts that cannot be changed once deployed and triggered to apply them to the consensus process. It deploys a smart contract in the consensus network. When some nodes in the system meet the contract requirements, such as the node's contribution to the blockchain, the node's hardware performance, etc., he can enter, so that it can be implemented to control the number of PBFT consensus nodes. The algorithm of this document improves the dynamic performance of the network. Nodes can be added or removed at any time in the blockchain network without restarting the network. Literature [32] proposed a ring signature scheme based on ElGamal digital signature algorithm. According to the ring signature scheme, the PBFT algorithm is improved to support the dynamic joining and exiting of nodes.

Literature [33-34] introduced the digital certificate issuing agency CA, which can effectively verify the identity of the node and whether the key is legal, and can also ensure that the key is not tampered with by other institutions, thereby ensuring that the system can run well. Literature [34] classifies nodes on the basis of CA institutions. Assuming that the total number of nodes in the network is represented by \( N_{\text{alt}} \), ordinary nodes are certified as voting nodes with real names. The number of voting nodes is \( N_v \). The production nodes are selected by voting nodes from candidate nodes, the number of production nodes is \( N_p \), the candidate node is generated by the ordinary node application, the number of candidate nodes is \( N_c \), the number of ordinary nodes is \( N_o \). They are satisfied

\[
N_{\text{alt}} = N_v + N_p + N_c + N_o
\] (4)

Compared with the PBFT algorithm, in the above improved algorithm, there is a certain quantitative relationship between various types of nodes. When the node joins or exits, the relevant parameters can be adjusted according to the quantitative relationship and no need to restart the system, which improves the flexibility of the system. However, the algorithm also has shortcomings, such as the limited ability of nodes to process data after node classification, which requires more in-depth research.

3.4. Hybrid Consensus Algorithm
In order to make up for the shortcomings of the PBFT algorithm and improve the performance of the traditional PBFT consensus algorithm, some researchers have proposed combining the PBFT algorithm with the public blockchain algorithm and private blockchain to become a hybrid consensus algorithm. Currently, more hybrid consensus algorithms such as POS+PBFT algorithm, DPOS+PBFT algorithm, RAFT+PBFT algorithm.

POS+PBFT algorithm is a typical design and is currently used on the EOS [35] platform. Holders elect representatives they support by voting and other methods, and the witness network composed of
these representatives reaches consensus through PBFT. Literature [36] through this hybrid consensus algorithm, compared with the traditional PBFT algorithm, the throughput, delay and network communication times have been significantly improved. DPOS+PBFT algorithm is a consensus algorithm adopted by NEO [37]. It is a consensus protocol that achieves large-scale participation through proxy voting. NEO holders can vote for the agent he supports, and then the agent can reach a consensus through PBFT. Literature [38-39] adopts this kind of algorithm, which is faster and better than the traditional PBFT consensus algorithm, improves the consensus efficiency, and is more suitable for blockchain application scenarios. RAFT+PBFT algorithm uses the leader election method of the private blockchain to select the main node, to avoid the failure of the main node and reduce the number of attempts to switch.

Compared with traditional PBFT, the hybrid consensus algorithm combines the good points of the two algorithms. This type of hybrid consensus algorithm not only has the low latency and high throughput of the consortium blockchain, but also has the scalability and fairness of the public chain algorithm. It effectively improves the efficiency of consensus and has higher practicability in practical applications in the future.

4. Conclusions
Since the PBFT algorithm was proposed, scholars have continuously proposed improved algorithms to enrich the content of the PBFT algorithm. The PBFT algorithm is a consensus algorithm widely used in consortium blockchain, and its future applications will become more and more common. For some shortcomings in the traditional PBFT algorithm and the improvement methods for these shortcomings, the article has carried out a detailed elaboration. But the improvement of the PBFT algorithm is by no means limited to the directions described in this article. Summarizing the previous experience, the PBFT algorithm can be studied in the following directions in the future: (1) Designing a consensus algorithm with high throughput and low latency suitable for multi-node consortium blockchain is the focus of future research. (2) Blockchain has been a popular topic in recent years, and the research on blockchain consensus algorithms has also made new progress. Combining the blockchain consensus algorithm improvement theory with actual specific application scenarios is the next step direction of the research.

Acknowledgments
This work was financially supported by the National Key Research and Development Project.

References
[1] Weide Cai, Lian Yu, Rong Wang, Na Liu, Enyan Deng. Research on application system development method based on blockchain [J]. Journal of Software, 2017, 28 (06): 1474-1487.
[2] W. Wang, D. T. Hoang, Z. Xiong et al., “A survey on consensus mechanisms and mining management in blockchain networks,” IEEE Access, vol. 7, pp. 22 328 – 22 370, 2019.
[3] Pilkington M. Blockchain Technology: Principles and Applications. Social Science Electronic Publishing, 2016.
[4] Qifeng Shao, Cheqing Jin, Zhao Zhang, Weining Qian, Aoying Zhou. Blockchain: Architecture and Research Progress. Journal of Computer Science, 2018.
[5] Dorri, A.; Steger, M.; Kanhere, S.S.; Jurdak, R. Blockchain: A distributed solution to automotive security and privacy. IEEE Commun. Mag. 2017, 55, 119–125.
[6] L. S. Sankar, M. Sindhu and M. Sethumadhavan, "Survey of consensus protocols on blockchain applications," 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, 2017, pp. 1-5.
[7] Rihong Wang, Lifeng Zhang, Quanqing Xu, Hang Zhou. A Byzantine fault-tolerant consensus algorithm that can be applied to alliance chains [J/OL]. Computer Application Research, 2020, pp. 1-6.
[8] W. Zhao, "Application-Aware Byzantine Fault Tolerance," 2014 IEEE 12th International
Conference on Dependable, Autonomic and Secure Computing, Dalian, 2014, pp. 45-50, doi: 10.1109/DASC.2014.17.

[9] G. S. Veronese, M. Correia, A. N. Bessani, L. C. Lung and P. Verissimo, "Efficient Byzantine Fault-Tolerance," in IEEE Transactions on Computers, vol. 62, no. 1, pp. 16-30, Jan. 2013, doi: 10.1109/TC.2011.221.

[10] G. Bracha, and S. Toueg. Asynchronous Consensus and Broadcast Protocols. Journal of the ACM, 32 (4), 1985.

[11] Castro M, Liskov B. Practical Byzantine fault tolerance//OSDI. 1999: 173-186.

[12] Gervais, A., et al., On the Security and Performance of Proof of Work Blockchains. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016; pp. 3–16.

[13] Hou, R., et al., Research on secure transmission and storage of energy IoT information based on Blockchain. PEER-TO-PEER NETWORKING AND APPLICATIONS, 2020.

[14] Chen, Z., et al., A Security Authentication Scheme of 5G Ultra-Dense Network Based on Block Chain. IEEE ACCESS, 2018. 6: p. 55372-55379.

[15] Ramakrishna Kotla. Zyyzyva: Speculative Byzantine Fault Tolerance [C] // Acm Sigops Symposium on Operating Systems Principles. ACM, 2007.

[16] Ziyue Wang. Research and optimization of blockchain system for supply chain traceability [D]. Beijing Jiaotong University, 2019.

[17] S. Jeon, I. Doh and K. Chae, "RMBC: Randomized mesh blockchain using DBFT consensus algorithm," in 2018 International Conference on Information Networking (ICOIN), Chiang Mai, Thailand, 2018 pp. 712-717.

[18] ANDROULAKI E, BARGER A, BORTNIKOV V, et al. Hyperledger fabric: a distributed operating system for permissioned Block-chains [C] // Proceedings of the Thirteenth EuroSys Conference (EuroSys 2018). 2018: 1-15.

[19] Zihao Chen, Qiang Li. Improved PBFT consensus mechanism based on K-medoids [J]. Computer Science, 2019, 46 (12): 101-107.

[20] Feng, L., et al., Scalable Dynamic Multi-Agent Practical Byzantine Fault-Tolerant Consensus in Permissioned Blockchain. APPLIED SCIENCES-BASEL, 2018. 8 (19): 191910.

[21] H. Sukhwani, J. M. Mart’irez. X. Chang et al., “Performance modeling of Pbft consensus process for permissioned blockchain network (hyperledger fabric),” in Proceedings of the2017IEEE36thSymposiumonReliableDistributedSystems (SRDS), pp. 253–255, Hong Kong, China, September 2017.

[22] LEEK, JAMES JI, EJETATG, KIM HJ. Electronic voting service using blockchain [J]. The Association of Digital Forensics, Security and Law, 2016, 11 (2): 123-136.

[23] Wang, Y., et al., Study of Blockchains’ Consensus Mechanism Based on Credit. IEEE ACCESS, 2019. 7: p. 10224-10231.

[24] Jing Li, Xu Jing, Huijun Yang. Blockchain electronic ticket counting scheme based on practical Byzantine fault-tolerant algorithm [J]. Computer Applications, 2020, 40 (04): 954-960.

[25] S. Gao, T. Yu, J. Zhu and W. Cai, "T-PBFT: An Eigen Trust-based practical Byzantine fault tolerance consensus algorithm," in China Communications, vol. 16, no. 12, pp. 111-123, Dec. 2019, doi: 10.23919/JCC.2019.12.008.

[26] Lei K, Zhang Q C, Xu L M, et al. Reputation-based Byzantine Fault-Tolerance for consortium blockchain [C] // 2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS), 2018: 604-611.

[27] YEOW K, GANI A, AHMAD R W, et al. Decentralized consensus for edge-centric internet of things: a review, taxonomy, and research issues [J]. IEEE Access, 2018, 18 (8): 15131524.

[28] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, “The Eigen trust algorithm for reputation management in p2p networks,” in Proceedings of the 12th international conference on World Wide Web. ACM, 2003, pp. 640–651.

[29] K. Lu, J. Wang, and M. Li, “An eigen trust dynamic evolutionary model in p2p file-sharing
systems,” Peer-to-Peer Networking and Applications, vol. 9, no. 3, pp. 599–612, 2016.

[30] Ion Stoica, Robert Morris, David. Karger, M. Frans Kaashoek, Hari Balakrishnan. Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications [J]. IEEE Transactions on Networking, 11, Feb.2003.

[31] Peng Dai. Evaluation and improvement of blockchain model based on Practical Byzantine Consensus Algorithm (PBFT) [D]. Beijing University of Posts and Telecommunications, 2019.

[32] Yi Fang, Jianqiu Deng. An improved PBFT blockchain consensus algorithm based on ring signature [J]. Computer Engineering, 2019, 45 (11): 32-36.

[33] YEOW K, GANI A, AHMAD R W, et al. Decentralized consensus for edge-centric internet of things: a review, taxonomy, and research issues [J]. IEEE Access, 2018, 18 (8): 15131524.

[34] Haiyong Wang, Kaixuan Guo, Qiqing Pan. Byzantine fault-tolerant consensus algorithm based on voting mechanism [J]. Computer Applications, 2019, 39 (06): 1766-1771.

[35] LARIMER D. EOS. Io Technical White Paper V2 [EB/OL]. https://github.com/EOSIO/Documentation/blob/master/TechnicalWhitePaper.md, 2018-4-28.

[36] Wu, Y., P. Song and F. Wang, Hybrid Consensus Algorithm Optimization: A Mathematical Method Based on POS and PBFT and Its Application in Blockchain. MATHEMATICAL PROBLEMS IN ENGINEERING, 2020. 2020 (7270624).

[37] ZHANG E. A Byzantine Fault Tolerance Algorithm for Blockchain [EB/OL]. https://docs.neo.org/en-us/basic/consensus/whitepaper.html, 2019-1-7.

[38] Xiaofei Liu. Research on Blockchain Performance Improvement Based on Dynamic Authorization Byzantine Fault Tolerant Consensus Algorithm [D]. Zhejiang University, 2017.

[39] Biryukov, A. and D. Feher, ReCon: Sybil-resistant consensus from reputation. PERVASIVE AND MOBILE COMPUTING, 2020. 61 (101109SI).