Improved PBFT Algorithm Based on Reputation and Voting Mechanism

Junhua Chen, Xia Zhang* and Pengfei Shangguan

Key Laboratory of Industrial Internet of Things & Networked Control, Ministry of Education, Chongqing University of Posts and Telecommunications, Chongqing, 400065, China

*Corresponding author’s e-mail: zhangxiall@163.com

Abstract. Aiming at the problems of large communication cost of the consistency process, low efficiency of view switching and the random selection of the master node in the Practical Byzantine Fault Tolerance (PBFT), an Improved Practical Byzantine Fault Tolerance (IPBFT) is proposed. Firstly, This paper proposes a credit model and voting mechanism, that the master node is generated by voting with nodes reputation values, which ensures the reliability of the master node and reduces the possibility that the abnormal node acts as master node. Secondly, the consistency process is optimized to decrease the frequency and time of communication between nodes. Finally, the view switching protocol is optimized, that classification process for abnormal node and malicious node further reduces communication time. The experimental results show that compared with PBFT, IPBFT increases the average throughput, reduces the delay, and improves the operating efficiency and security of the system.

1. Introduction

Blockchain technology originated in Bitcoin and was proposed in 2008 by a scholar named "Zhongben Cong". The blockchain is essentially a distributed database with characteristics of decentralization, tamper-proof and traceability [1][2]. With the emergence and popularity of cryptocurrencies such as Bitcoin, the new decentralized architecture and distributed ledger technology of the blockchain has received attention in various fields including finance and Internet of things [3][4]. In order to better protect users’ privacy and data, blockchain technology has been developed from the public chain to the alliance chain, and has been widely used in public and government services [5].

Consensus algorithm is the core of blockchain technology. Currently, many consensus algorithms are applied in blockchain. For example, Proof of Work (PoW) [6], Proof of Stack (PoS) [7], Delegated Proof of Stack (DPoS) [8] and Practical Byzantine Fault Tolerance (PBFT).

To some extent, the reliability of blockchain is guaranteed by the above algorithm. However, data security depends on computing power, so it can not be widely used in alliance chain. The PBFT algorithm [9] was proposed by Miguel Castro and Barbara Liskov in 1999 to solve the problem of consistency of Byzantine error nodes in distributed systems. The PBFT algorithm is applied to the consensus mechanism of the blockchain, which has high reliability and security, and has been widely used in the alliance chain. However, there are some problems in PBFT algorithm, such as the exception node serving as the master node, high communication overhead in the consistent process, and low efficiency of view switching, which affect the operating efficiency and security of the system.

This paper proposes an improved consensus algorithm IPBFT (Improved Practical Byzantine Fault Tolerance) based on PBFT. The credit model and voting mechanism are introduced to effectively reduce
the situation where the abnormal node acts as the master node, and optimize the consistency process and the view switching protocol to improve the operating efficiency and security of the system.

2. IPBFT algorithm design

2.1 Overall thinking
In order to reduce the probability that the abnormal node acts as the master node and reduce the wasted communication resources in the consensus process, this paper proposes an improved consensus algorithm IPBFT based on the PBFT algorithm. The main improvement of IPBFT mechanism includes the following aspects.

1) Define a credit model. The state of the node is divided by the reputation value, and the state transition mechanism is formulated by the performance of the node in the consistency process.
2) Optimize the consistency protocol. Remove the commit phase.
3) Optimize the master node acknowledgment protocol. The primary node is confirmed by the credit model and the voting mechanism, and the possibility that the abnormal node acts as the primary node is reduced.
4) Optimize the view switching protocol. According to the abnormal situation of the node, the view switching is carried out to reduce the number of node broadcasting.

2.2 Credit model and voting mechanism
In the IPBFT consensus algorithm, the master node is responsible for the broadcast request proposal. In order to continuously and effectively send the correct proposal to the slave node and reduce the probability that the abnormal node acts as the master node, this paper introduces the credit model and voting mechanism. The state of the node is divided by the reputation value, and the state transition mechanism is formulated according to the performance of the node in the consistency process, and then the master node is calculated by the voting formula to ensure the reliability of the master node.

2.2.1 Reputation value and node status. In the credit model, the nodal credit value is set as [0,150]. If the effective block is successfully generated from the node, the system will reward it with 2 points of credit. According to the value of the reputation value, the credit model adds a status identifier to each node and sets five different node states, as shown in Table 1.

| Reputation value | [100-150) | [60-100) | [30-60) | [10-30) | 0 |
|------------------|-----------|---------|--------|--------|---|
| Node status      | Credible  | Normal  | Excepted | Untrusted | Invalid |

2.2.2 Node state transition. Node state transition refers to the change of node state, which is mainly related to the performance of the node in the consistency process. When the system just runs, the nodes are in Normal state. When the node generates valid blocks for several times and the credit value reaches the good credit threshold, it can be upgraded to the Credible state. The invalid block is generated from a node in the Credible state or the Normal state, and the node state changes to the Check state. At this time, the node error behavior and reputation value will be reviewed by the system. If the node reputation value is higher than the good threshold and the block failure occurs due to the downtime behavior, the node will become Excepted state. If the node has malicious behavior during the consistency process, the node will become Untrusted state. Nodes in the Excepted state will be downgraded to Untrusted state if there is another downtime or malicious behavior. Nodes in the Untrusted state will be converted to Invalid state if malicious or downtime occurs. In addition, after the node generates effective blocks, the credit value reaches a certain threshold, and the node status will be upgraded accordingly. The node state transition is shown in Figure 1.
2.2.3 Node authority assignment. The nodes in different states in the credit model have different system permissions. The specific rights are assigned as shown in Table 2.

| Node status | Candidate | vote | consensus |
|-------------|-----------|------|-----------|
| Credible    | ✓         | ✓    | ✓         |
| Normal      | ✓         | ✓    | ✓         |
| Excepted    | ×         | ✓    | ✓         |
| Check       | ×         | ×    | ✓         |
| Bad         | ×         | ×    | ✓         |
| Invalid     | ×         | ×    | ×         |

Credible and Normal nodes have all rights to participate in the primary election, voting and consensus process of the primary election. However, Excepted nodes only have the authority to vote for voting and consensus processes. The Check node is a special node that does not have any permissions before the review is completed. And Untrusted nodes can only take part in the consensus process.

2.2.4 Voting mechanism. The main function of the voting mechanism is to establish the master node jointly with the credit model. This paper proposes a new voting calculation formula. The final score calculation formula of the participating nodes is as follows:

\[
Score_n = credit_n \times \lambda + \left( \sum_{k=1}^{N} \frac{credit_k}{150} \times vote_k \right) \times \mu
\]

In the above formula, the total score is mainly divided into the basic score and the voting score. The basis is divided into the product of the credit value of the candidate node and the parameter \( \lambda \), and the voting is divided into the ratio of the credit value and the total credit value of the voting node, and the voting situation and parameter \( \mu \) are calculated. The \( credit_n \) is the reputation value of the selected node \( n \). The \( credit_k \) is the reputation value of the voting node \( k \). The \( vote_k \) is the voting situation of the \( k \)-node voting, support, opposition and waiver correspond to 1, -1 and 0 respectively. In addition, two parameters \( \lambda \) and \( \mu \) \((\lambda + \mu = 1)\) were introduced, which can affect the total score of the participating nodes to some extent. The Credible node and Normal node participating in the primary node election have different \( \lambda \) and \( \mu \). When the node is in the Credible state, it has a higher credibility, so the number of votes required to become the main node should be less than that of Normal node. The smaller the \( \lambda \).
value, the more votes the master node needs, so the value of $\lambda$ should be greater than 0.5, and the corresponding value of $\mu$ should be less than 0.5. This paper sets the $\lambda$ is 0.65 and $\mu$ is 0.35 of the Credible node. For a node in the Normal state, the number of votes required for the master node is required, so the value of $\lambda$ is less than 0.5, and the value of $\mu$ is greater than 0.5. This paper sets the $\lambda$ is 0.45 and $\mu$ is 0.55 of the Normal node. This voting calculation method ensures that the credible node with high credibility becomes the master node and needs fewer votes than the Normal node, which ensures the fairness of the voting election and improves the security of the system.

### 2.3 Optimized consistency protocol

IPBFT consensus algorithm mainly realizes the consensus among the nodes participating in the consensus. The slave node receives the master node broadcast, which only needs to verify the received proposal information in the preparation stage. When the preparation stage is completed, it means that enough slave nodes have verified and passed the proposal initiated by the master node, that is, consensus is reached. The submission stage is only the confirmation of the proposal passing in the preparation stage to ensure that the legal number of nodes complete the verification of the proposal. Therefore, the three-phase protocol can be optimized to two phases, namely Consistency Proposal and Consistency Confirmation, as shown in Figure 2.

![Figure 2. IPBFT algorithm flow](image)

### 2.4 Optimized master node acknowledgment

The main purpose of this protocol is to confirm the selection of the primary node. According to the formula $p=v \mod N$, PBFT algorithm determines the number of the primary node in order. The abnormal node is likely to act as the primary node, which affects the operation efficiency of the system. In the improved IPBFT algorithm, the credit model and the voting mechanism are introduced to optimize the master node acknowledgment protocol. The node with high reputation has a higher probability of becoming the master node by election, which effectively reduces the possibility of abnormality as the master node, improves system security and system operation effectiveness. The specific algorithm is as follows:

```plaintext
algorithm Master()
Input: client request
Output: Master node number
1: view=0
2: p= Scores[view]
3: if trigger change-view
4: view++
5: p= Scores[view]
6: return p
```

Master() is the main node confirmation function, and returns the main node number. The Scores array stores the node numbers sorted by the total score of the participating nodes. When the change-view is true, the view switching operation is performed, the view is incremented by 1, and the next node in the sorting node is selected as the master node.
2.5 Optimized view switching protocol
The main purpose of this protocol is to determine whether the node is abnormal. If so, the view switch is performed that the primary node and view are replaced. In PBFT algorithm, a broadcast is made when the slave node considers that the master node is abnormal and that each broadcast wastes system resources. In the IPBFT algorithm, the downtime and malicious behavior of the abnormal nodes are classified. When the primary node is down, there is no need to perform the inter-node voting process and directly perform the view switching operation. However, When the slave node thinks that the master node is the problem node, it will vote to determine whether the master node has problems. After confirmation, carrying out view switching can reduce the voting link when the master node goes down, save the communication time of each node, and further improve the system operation efficiency. The specific algorithm is as follows:

```
algorithm ViewChange()
Input: <ConsistencyProposal>
Output: Whether to switch views
1: time = CurrentTime
2: if ConsistencyProposal not null
3: if Primary node response time minus time more than the t
4: change-view
5: else
6: return
7: else if Master node evil
8: Broadcast view-change and record feedback from other nodes
9: if Determine the count value of the master node to do evil count greater than 2f
10: change-view
11: else
12: return
13: else
14: return
```

3. Experiment and analysis

3.1 Experimental environment
In order to analyze the performance of IPBFT, a simulation platform built by six virtual machines running Linux operating system was built. The Linux system version is Ubuntu 16.04, and the host system is Windows 10 Enterprise Edition. The main parameters of the computer are as follows: Intel(R) Core (TM) i3-8100, 8G DDR4 memory, SATA2 7200 120G hard disk, 6MB processor cache. The Linux virtual machine is in the same LAN, and the IP addresses are 192.168.242.220, 192.168.242.221, 192.168.242.222, 192.168.242.223, 192.168.242.224 and 192.168.242.214. The blockchain base component is Hyperledger fabric V1.2. This paper will evaluate the performance of IPBFT from two aspects of throughput and delay.

3.2 Throughput test
In the blockchain system, the throughput refers to the number of transactions processed by the system per unit time. It is generally expressed by TPS (Transaction Per Second) and is an important indicator for measuring the concurrent processing capability of the system. The formula is as follows:

\[ TPS = \frac{\text{Transactions}_{\Delta t}}{\Delta t} \] (2)

Among them, \( \Delta t \) is the block time, and \( \text{Transactions}_{\Delta t} \) represents the number of transactions included in the block time.

Under the same conditions, the throughput comparison of the two algorithms is shown in Figure 3.
In a blockchain system, the performance of the system is greatly affected by the number of nodes. When the number of nodes increases, the block capacity will increase and the throughput will increase accordingly. However, the system consensus time and network load will also increase, and the system performance will decrease when the threshold is exceeded. In the IPBFT algorithm, the consistency process is optimized, and only a certain proportion of nodes participate in the consensus by voting. When the number of nodes is large, the consensus mechanism can run stably. In the simulation, the throughput of PBFT and IPBFT was tested with 5, 10, 15, 20 and 25 nodes respectively. As shown in Figure 3, the throughput of the PBFT algorithm decreases as the number of nodes increases, while the throughput of IPBFT gradually increases at the beginning. When the number of nodes exceeds 15, the IPBFT performance remains stable.

3.3 Delay test
The delay represents the time required from block generation to confirmation of new blocks, which is mainly generated by the consistency process in the consensus mechanism and the view switching process. The low latency indicates that the consensus process execution time is short, so the blockchain is not prone to bifurcation and the system security is higher. The delay formula is expressed as follows:

\[ Delay = T_{confirm} - T_{commit} \]  

(3)

Among them, \( T_{confirm} \) represents the block confirmation time, and \( T_{commit} \) represents the block generation time.

Under the same conditions, the delay of the two algorithms is shown in Figure 4.

Figure 4 illustrates the variation of the delay of the two algorithms over time. The delay of the PBFT algorithm is stable at about 200ms, and the delay of the IPBFT algorithm is just about 215ms higher.
This is because the primary node needs to be determined through the credit model and voting mechanism when the system is just running, which takes a certain amount of time. However, as time goes by, the optimized consistency process and view switching process reduce the number of node broadcasts, save the broadcast transmission time of nodes, and thus reduce the system delay. Therefore, the delay of the IPBFT algorithm is stable at around 180ms, which is significantly lower than the PBFT algorithm.

4. Conclusion
This paper proposes an Improved Practical Byzantine Fault Tolerance (IPBFT), which designs a credit model and a voting mechanism to optimize the master node confirmation protocol thus reduces the possibility of the exception node acting as the master node. Meanwhile, consistency protocol is optimized, which cancels the submission process in the consistency process and reduces the network communication time of the node broadcast. In addition, Optimized view switching protocol is optimized, which abnormal node classification processing, further reducing network communication time. The simulation results show that compared with the PBFT algorithm, the IPBFT algorithm has a significant improvement in throughput and delay, which improves the performance, security and reliability of the system.

Acknowledgments
This work was supported by the National Key Research and Development Program (No.2017YFE0123000).

References
[1] Yuan, Y., Wang, F.Y. (2016) Blockchain: the state of the art and future trends. Acta Automatica Sinica., 42(4): 481-494.
[2] Shao, Q.F., Jin, Q.Q., Zhang, Z., et al. (2018) Blockchain Technology: architecture and progress. Acta Computerica Sinica., 41(5): 969-988.
[3] Wu, J., Dong, M., Ota, K., Li, J., & Guan, Z. (2018) Big data analysis-based secure cluster management for optimized control plane in software-defined networks. IEEE Transactions on Network and Service Management., 1-1.
[4] Hooker, C. N. H. (1996) Haccp as an international trade standard. American Journal of Agricultural Economics., 78(3): 775-779.
[5] Cai, W.D., Yu, L., Wang, R., Liu N., et al. (2017) Blockchain Application Development Techniques. J. Journal of Software., 28(6): 1474–1487.
[6] Wang, X.G. (2017) Summary of the Consensus Algorithm of Block Chain Technology. China Computer & Communication., 9: 72-74.
[7] Sunny, K., Scott, N. (2012) PPCoin: Peer-to-peer crypto-currency with proof-of-stake. https://decred.org/research/king2012.pdf.
[8] Daniel, L. (2014) Delegated Proof-of-Stake (DPoS). Bitshare whitepaper.
[9] Castro, M., Liskov, B. (1999) Practical Byzantine fault tolerance. In: Proc.3rd Symp. Oper. Syst. Design Implement., New Orleans. pp. 173-186.