RESEARCH ON DYNAMIC PBFT CONSENSUS ALGORITHM

Cao Xiaopeng and Shi Linkai

School of Computer Science and Technology,
Xi'an University of Posts and Telecommunications, Xi'an, China

ABSTRACT

The practical Byzantine fault-tolerant algorithm does not add nodes dynamically. It is limited in practical application. In order to add nodes dynamically, Dynamic Practical Byzantine Fault Tolerance Algorithm (DPBFT) was proposed. Firstly, a new node sends request information to other nodes in the network. The nodes in the network decide their identities and requests. Then the nodes in the network reverse connect to the new node and send block information of the current network, the new node updates information. Finally, the new node participates in the next round of consensus, changes the view and selects the master node. This paper abstracts the decision of nodes into the undirected connected graph. The final consistency of the graph is used to prove that the proposed algorithm can adapt to the network dynamically. Compared with the PBFT algorithm, DPBFT has better fault tolerance and lower network bandwidth.

KEYWORDS

Practical Byzantine Fault Tolerance, Blockchain, Consensus Algorithm, Consistency Analysis

1. INTRODUCTION

In many new Internet applications, blockchain [1, 2] is becoming more and more important. Blockchain is a technical solution to maintain a reliable distributed database. Consensus mechanism is the core of blockchain, which solves the problem of how to reach consensus in a completely free and open network without trust.

Blockchain is a decentralized distributed ledger system. It is a network composed of multiple hosts through asynchronous communication. It is necessary to solve the problem that how to reach a consensus on a certain transaction between distrustful individuals after decentralization, so as to ensure the effective operation of the whole system. In the absence of centralization, state replication between hosts is required to reach a consistent state consensus. To ensure the data consistency of each node is a key issue. The consensus algorithm is a mechanism to copy state between unreliable hosts when multiple hosts form a network cluster through asynchronous communication. It is the core of blockchain.

In the development of blockchain, scholars have proposed a variety of consensus mechanisms including Byzantine fault-tolerant algorithm. They pay more attention to resource consumption, security, and consistent time. The advantages and disadvantages of consensus mechanism directly affect the security and performance of the blockchain system. With the application of blockchain technology in various fields, it is particularly important to study consensus algorithm [3].

The consensus algorithm is used to solve the Byzantine General problem [4]. The Byzantine General problem is all nodes achieved consistency in untrusted networks.

This paper is organized as follows. The detailed literature survey is presented in section 2. Section 3 deals with the algorithm related details. Section 4 proved the correctness of the algorithm. Section 5 introduces the experimental results and performance comparison. The general conclusion and the scope for future work are given in the last part.

2. RELATED WORKS

In 1990, Leslie Lamport published the paper "the part-time partnership" and proposed the Paxos algorithm [5]. Paxos achieved the mechanism of the extreme consistency of distributed systems [6]. This mechanism has been widely used in chubby and zookeeper distributed systems. However, Paxos algorithm [7] does not consider some optimization mechanisms. And there are not too many implementation details in Paxos, which is hard to understand.

Proof of Work (POW) algorithm is mainly used in the bitcoin generation algorithm [8]. It uses hash operation to get a value. The value can be offset to resist DDoS attacks. However, it is not suitable for large block generation time.

Castro et al. improved the BFT algorithm and proposed a practical Byzantine Fault Tolerance (PBFT) [9], which reduced the complexity of the algorithm from exponential to polynomial level. The application becomes feasible in the practical system. PBFT is an algorithm based on state machine replication. It can ensure the system safe and reach a distributed consensus without exceeding the error node's limits. However, the algorithm uses C/S architecture. And it cannot adapt to P2P network. It cannot feel the changes in the number of nodes in the network dynamically.

NEO blockchain [10] mixed the Delegated Proof of Stake (DPoS) [11] and PBFT. They proposed Delegated Byzantine Fault Tolerant (DBFT) [12] through applying the DPoS authorization mechanism to PBFT. This algorithm decides the bookkeeper through voting. The block is validated and generated by the agent. In this way, it reduces the number of nodes in the consensus process and solves the inherent scalability problem of the PBFT algorithm. The disadvantages of DBFT do not be ignored. On the one hand, it is reflected in a lower fault tolerance rate. When 1/3 or more of the super nodes are malicious or downtime, the system does not provide services. On the other hand, the number of super nodes is too small. The entire system is too centralized.

Gueta and Guy proposed the Simplified Byzantine fault-tolerant algorithm (SBFT) [13]. In SBFT, a designated block collector collects and broadcasts transaction information. It batches the information into a new block transaction periodically. The generator provides consensus. Although the communication is reduced, the block verification by the collector has a very high centralization trend.

After analysis of the existing consensus algorithms, each algorithm has its advantages and disadvantages. The original PBFT algorithm cannot add nodes flexibly and dynamically. When the number of nodes in the system increasing, the original algorithm still runs according to the previously fixed number of nodes. There is no suitable admission mechanism to deal with the increasing of the node. It wastes resources. In this paper, an improved PBFT algorithm was proposed. It can realize the node join the network dynamically and participate in consensus. The system is decentralized and improves fault tolerance.

3. DPBFT

PBFT is a distributed protocol, it uses the C/S response mode. It is not suitable for the Peer-to-Peer [14] network of the blockchain [8]. The protocol is a closed-loop operation and cannot add nodes dynamically. To solve this problem, DPBFT was proposed. Figure 1 shows the process of the consistency protocol in DPBFT.

There are five stages in the algorithm, Addrequest, AgreeResponse, Recovery, JoinAndUpdate and Finish. In the Recovery phase, the nodes in the network synchronize with the new nodes. In the JoinAndUpdate phase, the new node sends its information again to prevent malicious nodes from posing as identities to enter the network. Since their information was sent in the Request phase, so this stage is to reconfirm the identity again of the node.

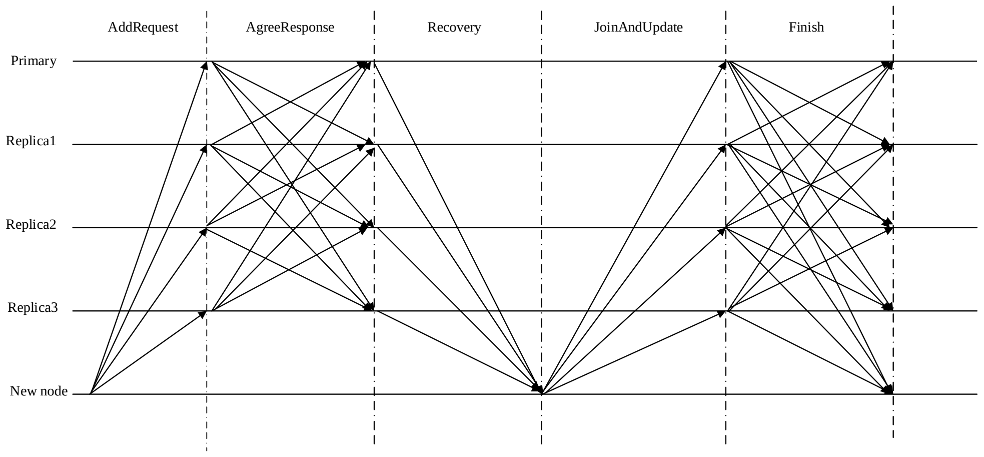


Figure 1. Flow chart of adding a node

Step 1: The new node obtains the network routing table. It sends AddRequest message to the nodes in the network. The format is <Add-Request, s, d> where s is the information of its node and d is the summary after information encryption, then broadcast the message to the network.

Step 2: The node in the system sends its decision information <<Agree-Response,s,d>,i> to the remaining nodes after receiving the request message, where i represents the node number, to ensure the newly added node of information cannot tamper. While collecting AgreeResponse messages from other nodes, receiving at least Q pieces of consent information represents that the remaining nodes allow new nodes to join the network.

Step 3: The node returns its decision information to the newly added node and connects back to the newly added node to synchronize the data. Each node sends the <<Pre-Recovery,V_m>a, i> message, where V_m represents the block message, a is the summary of the message m, and i is the number of its own node.

Step 4: The newly added node broadcasts <Join-Update, s, d> to each node when finished the synchronizing data, where s is the information of its node and d is the encrypted information of its node. Sending the information of the node again is to prevent other nodes join the network.

Step 5: The node in the network updates its routing table and recalculates the view after receiving the message of the new node. Nodes broadcast the updated message <Finish-Update, Vs, i>, where Vs is the information of view after the joined node and i represents the number of its own node. Broadcasting the message to others nodes ensures that the remaining nodes can update data correctly.

Step 6: The updated master node starts a new round of consensus after adding the new node.

4. ALGORITHM ANALYSIS AND PROOF

4.1. Algorithm analysis

The nodes in the network make a judgment in the second stage of algorithm when the new node joining. Recovery and update data in the other stages. This algorithm defaults that there are four nodes and one Byzantine node in next analysis. The node in the network make a decision when the newly added node sending the request message, there are two results: agree or disagree. The decision analysis diagram is shown in Figure 2.

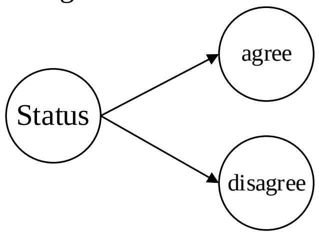


Figure 2. Node decision analysis diagram

The nodes in the network reach consistency means making a same decision. It means that there is only one possibility in the end. State consistency as shown in Figure 3.

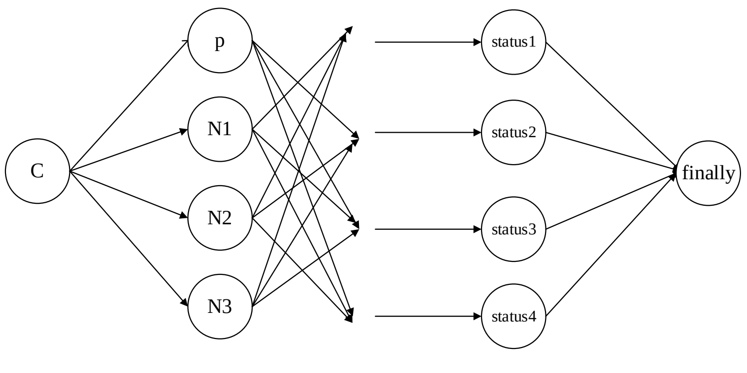


Figure 3. State consistency

The algorithm has abstracted a model. The specific model is shown in Figure 4.

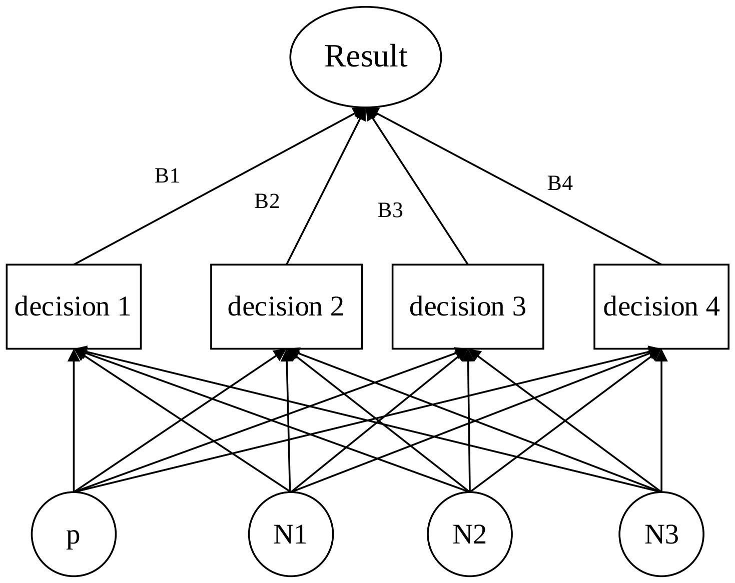


Figure 4. Algorithm model diagram

Construct a judgment matrix of the criterion layer. N3 is a Byzantine node. B1 indicates that after making its own decision, it can make a judgment when collecting the subsequent message of at least two nodes that have made decision who same as itself. The importance of B2's decision is recorded as 8. After receiving the B2 as the decision side, the importance of B3 is recorded as 5. This moment, enough replies have been collected. Therefore, the weight of B4's reply is not so important, it is recorded as 3. The judgment matrix is shown in Table 1.

Table 1. Judgment matrix of criterion layer.

A	**B1**	**B2**	**B3**	**B4**
B1	1	8	5	3
B2	1/8	1	1/2	1/6
B3	1/5	2	1	1/3
B4	1/3	6	3	1

By normalized the judgment matrix, the maximum eigenvalue λ is 4.073, then $CI = \frac{\lambda\text{-n}}{n-1} = 0.024$, where n is the dimension of the matrix.

According to the size of n, look up the corresponding average random consistency index RI. The table of RI values is shown in Table 2.

Table 2. RI value table.

n	1	2	3	4	5	6	7	8	9
RI	0	0	0.58	0.90	1.12	1.24	1.32	1.41	1.45

$CR = \frac{CI}{RI} = 0.027 < 0.1$. Therefore, consistency is considered acceptable.

Assuming that there are f Byzantine nodes and N summary points in the network. So the number of non-Byzantine nodes is N-f, and the probability of N nodes receiving the error information of the Byzantine nodes is the same. If it is set to p_1, as shown in Eq. 1.

$$p_1 = \frac{f}{N} \tag{1}$$

The probability that each node receiving a Byzantine node message is p_2, as shown in Eq. 2.

$$p_2 = C_N^f * (\frac{f}{N})^f * (1 - \frac{f}{N})^{N-f} \tag{2}$$

Through Eq. 2, the probability of the Byzantine node influences other nodes to make decisions in the network that can be calculated.

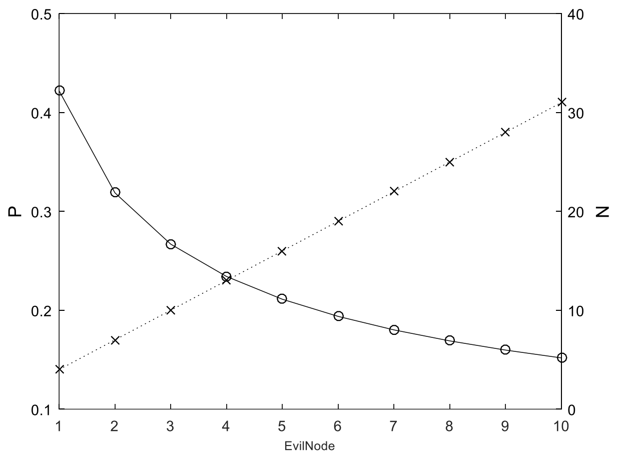


Figure 5. Byzantine nodes influence the probability of network nodes making decisions

Figure5 shows that the number of nodes and Byzantine nodes increases, the probability ofnodes in the network receiving malicious node communication gradually decreases. For newly added node to join the network, the nodes in the network will make correct decisions and not be affected by Byzantine node interference.

If there have a newly added node and n-1 network nodes in the network, two full-node broadcasts and three single-node broadcasts are required for the admission of the new node. From Figure 1 known, the newly added node needs to broadcast the request information in the Addrequest stage. For other nodes in the network, the number of communications is n-1. In the AgreeResponse stage, each node needs to broadcast decision information to other nodes after making its own decision. So the total number of communications is $(n-1)*(n-2)$. The nodes in the networkneed to be reverse connected to the new node for data synchronization, so the number of communications is n-1. The final update stage requires the nodes in the network to communicate with each other, the number of communications is $(n-1)*(n-2)$. Therefore, the total number of communications is as follows Eq. 3.

$$2n^2 - 3n + 1 \tag{3}$$

4.2. Algorithm proof

This paper abstracts the communication of nodes into the undirected connected graph. To prove the newly added node joined network means prove the consistency of the connected graph. Now the communication between nodes is defined as a communication graph with node set G (node G) and edge set (edge G) so that the directed edges of the two-node communication appear in pairs. And edge (u, v) $\in$ edges (G), (v, u) belongs to edges (G). Analyse the communication in each direction by abstracting a pair of directed edges into an undirected edge.

Graph G is an ordered pair consisting of vertex set V and edge set E G=(V, E). Graphs G is a simple undirected graph with $E \subseteq \{uv|u,v \in V, u \neq v\}$ .The number of vertices of graph G is finite. The vertex set can be assumed is $V=\{v_1,v_2,...v_n\}$, the edge set is $E=\{e_1,e_2,...e_m\}$ .Figure 6 shows that the node communication in the system is abstracted as an undirected graph.

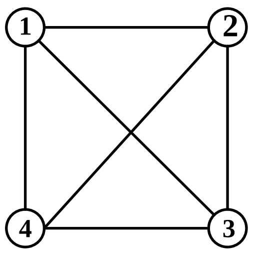


Figure 6. System node communication diagram

Figure 6 is a connected graph, so there are Definition 1 Let G=(V,E) be a connected graph, arbitrary vertex $v \in V(G)$,let $\varepsilon(v) = \max\{d(u,v) \mid u \in V(G)\}$, call $\varepsilon(v)$ the eccentricity of vertex v, and $R(G) = \min\{\varepsilon(v) \mid v \in V(G)\}$, refer to $R(G)$ as the radius of graph G. Therefore, the diameter of graph G is defined as $D(G) = \max\{\varepsilon(v) \mid v \in V(G)\}$, and $R(G) \leq D(G) \leq 2R(G)$. As shown in Figure 7, the radius of the graph is 1.

The problem is studied in an interactive network. A linear combination of the storage state of a node in a network and synchronize with other node states. If $s_i^{(t)}$ represents at time t the state of node i, the $N_i \in V$ represents the set of all nodes that can be communicated with. The method of updating node state is expressed as $s_i^{(t+1)} = a_{ii}^{(t+1)} s_i^{(t)} + \sum_{j \in N_i} a_{ij}^{(t+1)} s_j^{(t)}$,where $a_{ij}^{(t)}$ represents the probability of node reply. The status update can be expressed as $X^{(t+1)} = A^{(t+1)} X^{(t)}$. If each node's state has reached consistency, it means the entire network has reached consistency.

If the undirected graph G satisfies the consistency, it indicates that its state has reached consistency. In this problem, it means that the nodes in the network have made a consistent decision to the new node.

Theorem 1 Let the radius of graph G be r, then there are r matrices $A_1,\ A_2,\ \dots, A_r \in M(G)$ make

$$A_r \dots A_2 A_1 = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & \cdots & 0 \end{pmatrix}$$

, where M(G) represents the value of all matrix elements in the set, corresponding to the correct and error messages sent by the network nodes.

Prove Let the distance from vertex 1 to other vertices <=r. According to the distance from 1, we can divide the vertex set [n] into r+1 subset. Mark them with $a_0, a_1, \dots, a_r$. Vertex V_i means the distance from 1 is I , and then marked the vertices $V_0, V_1, \dots, V_r$ sequentially.

Use Mathematical Induction can prove the existence of $A_k, \dots, A_2, A_1 \in M(G)(k \le r)$ makes

$$A_k, \dots, A_2, A_1 = \begin{pmatrix} 1_{s\times 1} & 0_{s\times(n-1)} \\ 0_{(n-s)\times 1} & 0_{(n-s)\times(n-1)} \end{pmatrix}, \text{ where } s = 1+|a_1|+|a_2|+\dots+|a_k|.$$

(1) When r=1, $A_1 = (1_{n\times 1} \quad 0_{n\times(n\times 1)}) \in M(G)$.

(2) When r=2, $A_1 = \begin{pmatrix} 1_{k\times 1} & 0_{k\times(n-1)} \\ 0_{(n-k)\times 1} & 0_{(n-k)^2} \end{pmatrix} \in M(G)$.

Each row of the submatrix formed by A2 has one element of 1, the rest of the elements of 0, and $A_2 A_1 = (1_{n\times 1} \quad 0_{n\times(n-1)})$.

(3) Suppose there are k matrices such that $A_k, \dots, A_2, A_1 = \begin{pmatrix} 1_{s\times 1} & 0_{s\times(n-1)} \\ 0_{(n-s)\times 1} & 0_{(n-s)\times(n-1)} \end{pmatrix}$, where $s = 1 + |a_1| + |a_2| + \dots + |a_k|$. For every vertex j in a_{k+1}, there is always a vertex s adjacent to j in a_k. Let the element A_{k+1} of $a_{js} = 1$ and the other elements are defined as 0. We can get$A_{K+1}A_k, \dots, A_2, A_1 = \begin{pmatrix} 1_{s\times 1} & 0_{s\times(n-1)} \\ 0_{(n-s)\times 1} & 0_{(n-s)\times(n-1)} \end{pmatrix}$, where $t = 1 + |a_1| + |a_2| + \dots + |a_k|$.So the Theorem 1 is true by Mathematical Induction.

Theorem 2 Let the radius of G be r, and the vertex $n \in V(G)$such that the distance from any vertex to n $\le$ r. There are 2r matrices $A_1,\ A_2, \dots A_r, A_1^T, A_2^T, \dots, A_r^T \in M(G)$, such that $A_r \dots A_2 A_1 A_1^T A_2^T \dots A_r^T = 1_{n\times n}$, where $1_{n\times n}$ represents an $n \times n$ order matrix where all elements are 1.

Prove According to Theorem 1, the existence of $A_1, A_2, \dots, A_r \in M(G)$ makes $A_r, \dots, A_2 A_1 = (1_{n\times 1}, 0_{n\times(n-1)})$. We can know that $A_1^T A_2^T \dots A_r^T = \begin{pmatrix} 1_{1\times n} \\ 0_{(n-1)\times n} \end{pmatrix}$. Therefore

$$A_r \dots A_2 A_1 A_1^T A_2^T \dots A_r^T = (1_{n\times 1}, 0_{n\times(n-1)}) \begin{pmatrix} 1_{1\times n} \\ 0_{(n-1)\times n} \end{pmatrix} = 1_{n\times n}$$

.Means that N satisfies certain consistency.

It can be seen from Theorem 1 that nodes send correct messages and error messages satisfy the 0-1 matrix and the matrix exists. From Theorem 2, the radius of Figure 7 is 1, which satisfies the deterministic consistency. It means the nodes in the network join in the request of the new node Consistency will be reached. It can be added dynamically.

5. Algorithm Comparison

5.1. Analysis of communication times

The total number of communications of the PBFT is $2n^2 - n - 1$. The number of communications of DPBFT is $2n^2 - 3n+1$ from Eq. 3. From Figure 7, it can be seen that the DPBFT can reduce the communication bandwidth effectively during the consensus process.

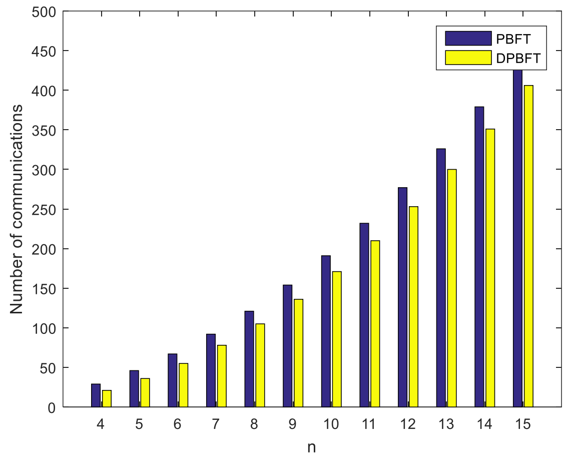


Figure 7. Comparison of communication bandwidth

When the network has the same number of nodes and the same size of block, with the increase of the number of nodes, the improved algorithm needs fewer communication times than the original algorithm, has lower bandwidth and less resource consumption than PBFT.

5.2. Fault tolerance analysis

In PBFT, the network needs to be restarted to add the new node. The node in the network needs to update when adding a new node. In DPBFT, at least f+1 correct node reach consensus to complete the new node. The number of nodes to reach consensus f_1 is N in PBFT. The number of nodes to reach consensus f_2 is f+1 in DPBFT. When the two algorithms have the same number of nodes, $f_1 - f_2 = \frac{2N-2}{3}$, where N is always greater than 0 and the minimum is 4. We can know $f_1 > f_2$. It means that the original algorithm is uncontrollable.

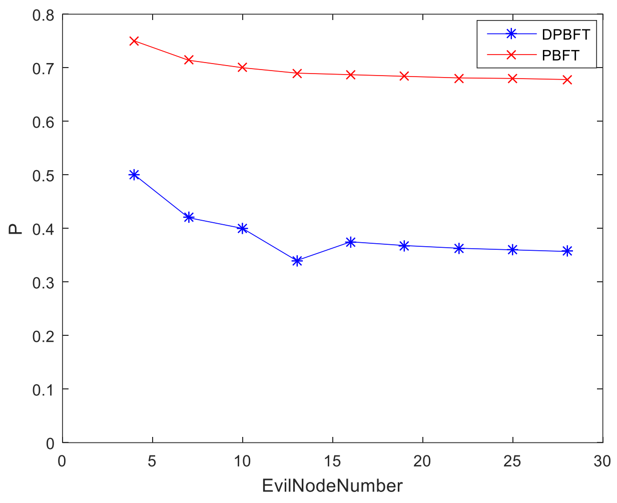


Figure 8 Comparison of node error rates

With the long-term operation of the system, the number of nodes in the network increases, and the error rate decreases. The error rate of the node in DPBFT is lower than the original algorithm from Figure 8. It means the proposed algorithm has higher fault tolerance.

6. Conclusions

In order to solve the problem that the traditional PBFT algorithm cannot sense the changes in the number of nodes in the network dynamically. The original algorithm does not adapt to the dynamic network. This paper proposed DPBFT that can add nodes dynamically. This algorithm maintains the characteristics of blockchain decentralization. The whole nodes in the network decision whether the new node participants in the network. The proposed algorithm does not need to restart the whole network. However, there are still some problems with this algorithm. The decision information of each node to communicate needs smaller delay of communication. Otherwise wrong decisions will occur. This problem needs to solve in the future.

References

[1] Pierro M D. What is the blockchain?[J]. Computing in science & engineering, 2017,19(5):92-95.
[2] Thakur S, Kulkarni V. Blockchain and Its Applications – A Detailed Survey[J].International Journal of Computer Applications, 2017, 180(3):29-35.
[3] Huang D, Ma X, Zhang S. Performance Analysis of the Raft Consensus Algorithm for Private Blockchains[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2018, pp(99):1-10.
[4] Lamport L& Shostak R, &Pease M. (1982) “The Byzantine generalsproblem” ACM Transactions on Programming Languagesre and Systems, Vol. 4, pp382-401.
[5] De Prisco R, Lampson B, Lynch N. Revisiting the Paxos algorithm[C]//International Workshop on Distributed Algorithms. Springer, Berlin, Heidelberg, 1997: 111-125.
[6] Connell A P. Prelates as Part-Time Parliamentarians: The Attendance and Participation of the LordsSpiritual in the Contemporary House of Lords[J]. Parliamentary Affairs A Journal of Representative Politics, 2017, 70(2):págs. 233-253.
[7] Lamport L, Massa M. Cheap paxos[C]//International Conference on Dependable Systems and Networks, 2004. IEEE, 2004: 307-314.
[8] NAKAMOTO S. (2009) “Bitcoin: a peer-to-peer electronic cash system”Cryptography Mailing list at http://metzdowd.com.

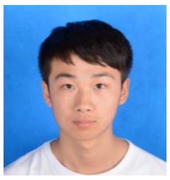
[9] Castro &M & Liskov & B, (2002) "Practical byzantine fault tolerance and proactive recovery" ACM Transactions on Computer Systems, Vol. 20, pp398-461.
[10] Elrom & Elad (2019) "NEO Blockchain and Smart Contracts" pp257-298
[11] Daniel Larimer, (2014) "Delegated Proof-of-Stake(DPOS)" White paper, http://bitshares.org/technology/delegated-proof-of-stack-comsensus/
[12] Crain,T&Gramoli,V,(2018) "DBFT:Efficient Leaderless Byzantine Consensus and its Application to Blockchains" pp1-8
[13] Gueta, Guy & Abraham, Ittai& Grossman, Shelly &Malkhi, Dahlia (2019). "SBFT: A Scalable and Decentralized Trust Infrastructure for Blockchains".
[14] Spinellis&Diomidis (2004) "A survey of peer-to-peer content distribution technologies" ACM Computing Surveys, Vol. 36, pp335-371

ACKNOWLEDGEMENTS

This research was funded by National Natural Science Foundation of China (61136002), Industrial Public Relations Project of Shaanxi Province(2014k06-36), Science and Technology Plan Project of Xi'an (CX12188(7)).

AUTHORS

Cao Xiaopeng is a Professor in Xi'an University of Posts and Telecommunications. His research interests include Natural Language Processing and Blockchain.



Shi Linkai is a M.S. candidate. His main research interests are Blockchain and Distributed application.