ISSN 2616-5775 Vol. 5, Issue 5: 17-25, DOI: 10.25236/AJCIS.2022.050503

Dynamic Transaction Confirmation Double-chain System Based on Alliance Chain

Nigang Sun, Lidong Yao*, Qiaosheng Hu

School of Computer and Artificial Intelligence, Changzhou University, Changzhou, China
**Corresponding author: yao401405972@gmail.com*

Abstract: *In recent years, the high performance, strong controllability and partial decentralization of the alliance chain make it have a wide range of application scenarios and development potential. However, restricted by the consensus algorithm, the alliance chain cannot fully meet the needs of practical applications at present. Aiming at the poor stability of the Byzantine Fault Tolerance (PBFT) consensus algorithm, this paper proposes a fast consensus algorithm, which reduces the message size by optimizing the transaction verification process and dynamically adjusting the number of verification nodes. Combined with the reward and punishment system of points, a dynamic transaction confirmation double-chain system is designed. Compared with the alliance chain system implemented by the PBFT algorithm, the system has stronger stability in the case of multiple nodes, and the TPS is increased by at least 21% under the same number of nodes. It is suitable for some nodes with frequent access and poor network.*

Keywords: *Alliance chain; Consensus algorithm; Point reward and punishment system; PBFT; TPS*

1. Introduction

After Satoshi Nakamoto proposed the Bitcoin electronic cash system [1] in 2008, the blockchain technology [2], which is composed of P2P network technology, encryption technology and other technologies, has received widespread attention from the world. As an extremely important part of the blockchain, the alliance chain is widely used in scenarios such as people's livelihood, finance and commerce due to its partial decentralization and strong controllability.

As the core of alliance chain technology, consensus mechanism is a research hotspot in recent years[3~7]. The PBFT algorithm [8] proposed by Castro et al. in 1999 is the most widely used consensus algorithm in the alliance chain at present, but due to the high communication complexity, a large amount of communication cost under the large node scale leads to performance degradation. In response to this problem, the DBFT algorithm [9] proposed in the Onchain white paper in 2016 reached a consensus through proxy voting, thereby reducing network communication consumption, improving scalability, and improving transaction processing speed. 1/3 of the total number of accounting nodes stop working or do evil, the system will be paralyzed or forked. In order to improve the security and fairness of the system, the VBFT algorithm [10] uses a verifiable random function to randomly select consensus nodes among all nodes to form a consensus network. The Elastico sharding protocol [11] proposed in 2016 is to allocate all transactions to different committees according to their hash values, and then submit them to the final committee after reaching a consensus within the different committees. The final committee packs blocks and broadcasts them, thereby reducing the number of verification nodes The number of transactions processed, thereby improving consensus efficiency. The Scalable BFT algorithm [12] proposed by Liu Jian et al. in 2018 improves system performance through threshold signature, BLS signature encryption and collector design. In 2019, Xu Zhi et al. proposed an improved PBFT efficient consensus mechanism based on credit [13], which reduces multiple interactions in the process of reaching consensus and reduces network overhead. A consensus mechanism suitable for alliance chains proposed by Cao Zhao-Lei [14] separates the verification function and the packaging function to improve the system efficiency. There are also some studies that use the pluggable consensus algorithm and the integration of consensus algorithms to improve the flexibility and stability of the system, and there are related studies that classify and test various consensus mechanisms currently applied to the alliance chain [15~18].

Although the above improved algorithms have improved the consensus efficiency to varying degrees, they still have the following defects:

ISSN 2616-5775 Vol. 5, Issue 5: 17-25, DOI: 10.25236/AJCIS.2022.050503

1) While improving consensus efficiency, system security is inevitably reduced.

2) Some algorithms have system architecture incompatibility problems, and there are fewer consensus algorithms to choose from in actual projects.

In response to the above problems, this paper proposes a dual-chain system for dynamic transaction confirmation. The verification chain is used for the preliminary verification of transactions. If the verification node doubts the transaction, the transaction will be reviewed, and the exchange will be stored in the block information and review. The information is packaged and submitted to the review chain for final confirmation. Among them, the consensus algorithm part saves communication costs by optimizing the transaction verification process. The master node adjusts the number of transaction verification nodes according to the current transaction volume,and combines the point reward and punishment system to distinguish normal nodes and malicious nodes, and increases the evil counter to increase the cost of repeated evil.

2. Dynamic Transaction Confirmation Double-chain System

2.1. Transaction Confirmation and Review Process

The dynamic transaction confirmation alliance chain system is a dual-chain system architecture: a verification chain and a review chain. Among them, the verification chain stores the transaction information and voting information successfully on the chain in the transaction confirmation stage, and the review chain stores the wrong transaction information and review information. Nodes are divided into three categories: master nodes, user nodes, and validating nodes. The master node is responsible for packing blocks and adjusting the k value of the current system (the number of nodes required for transaction confirmation). It can be increased or decreased, but the increase or decrease ratio cannot exceed 10%. The result is rounded down and released to the other validator nodes serve as master nodes in sequence, and the term is one block.

The verification chain verification transaction steps are as follows:

1) The master node broadcasts the k value to the validating nodes. If the verification node does not receive the k value within the specified time, it will broadcast no-k information to other verification nodes, and $\left(\frac{2}{3}n+1\right)$ no-k messages are received, it will replace the next master node.

2) The user broadcasts the transaction information to the verification node, and the transaction is stored in the transaction pool. If the verification node verifies the transaction is correct, it will put the transaction into the verified transaction set and broadcast the confirmation information. When the master node receives k confirmation messages, it considers that the transaction is correct, and puts it into the packaged transaction queue. After t time, the transaction in the queue is packaged into a block, and the block information is sent to other verification nodes. Block information includes transaction information, version information, k value, voting information, etc.

3) After the verification node receives the block information, it traverses and compares its own transaction pool and the verified transaction set. If the transaction information in the block is in the verified transaction set, it will be skipped. If the transaction has not been verified by itself in the transaction pool, the verification will be taken out. If the transaction information is incorrect, the master node will be challenged and the application will enter the review stage. If the transaction is correct, update its own blockchain.

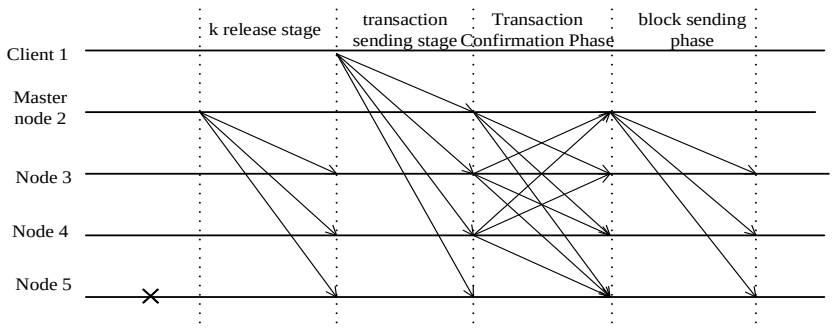


Figure 1: Dynamic transaction confirmation flow chart

ISSN 2616-5775 Vol. 5, Issue 5: 17-25, DOI: 10.25236/AJCIS.2022.050503

The dynamic transaction confirmation process is shown in Figure 1.

The censorship chain censors the questioned transaction steps as follows:

1) The verification node receives the block information sent by the master node in the block sending stage, and compares it with the verified transaction set through the local transaction pool.

2) If the master node does not process the challenge information and the number of challenge information received from other verification nodes exceeds 2n/3, the next master node will be replaced according to the system preset.

3) If the master node receives the challenge information raised by at least one verification node, it will enter the review stage.

4) The master node broadcasts the questioned block information and the position of the questioned transaction to the whole network, which requires the entire network nodes to conduct PBFT consensus. If more than 2n/3 verification node confirms the transaction error, the block information and review information will be packaged into a new Block commit review chain.

5) Change the master node to enter the next consensus cycle.

2.2. Point Pool and Reward and Punishment System

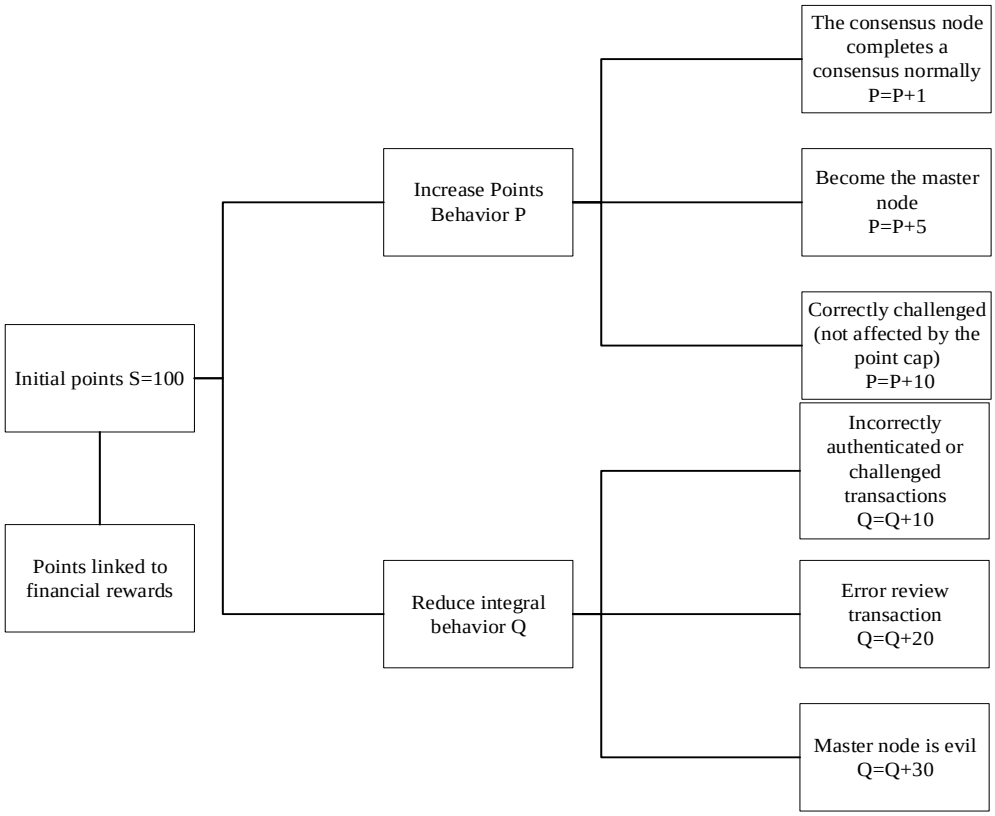


Figure 2: Dynamic transaction confirmation points rules

Because the master node has the function of packing blocks and adjusting the k value in this system, the authority of the master node is greater than that of the ordinary verification node. In the PBFT consensus, the consistency of the blockchain system is mainly ensured through two state synchronizations, while the dynamic transaction confirmation mechanism only performs one state synchronization process in order to reduce the number of communications and transaction delays, so the reward and punishment system must be used in addition to the confirmation mechanism. Force all nodes in the entire network to jointly maintain system security, and use the form of a point pool to distinguish between normal nodes and malicious nodes. Since the nodes in the consortium chain can have a one-to-one mapping relationship with entities, it is possible to use economic games to motivate all nodes to jointly maintain system security by linking points with personal economic rewards or punishments, and to restrict nodes from doing evil, increase the cost of doing evil, and the point system as shown in figure

ISSN 2616-5775 Vol. 5, Issue 5: 17-25, DOI: 10.25236/AJCIS.2022.050503

2.

In order to prevent a node from becoming a normal node again through a series of extra points behaviors after doing evil, thus causing the problem of low cost of doing evil, the concept of a counter for the number of times of node evil is introduced into the punishment mechanism, in which the point S and the increase point behavior P, the reduction point behavior Q And the relationship between the number of times of evil is shown in Equation 1.

$$S = S + P - QT \tag{1}$$

Therefore, the loss of points caused by repeated acts of evil will become larger and larger until the point S is lower than 0 for one act of evil, thus being kicked out of the consensus node set.

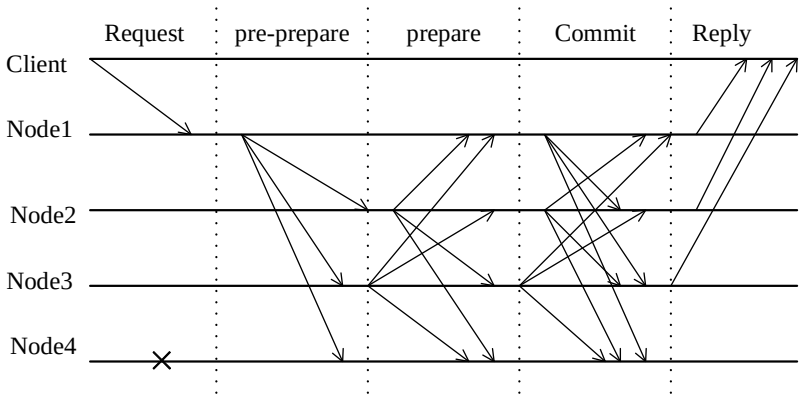
2.3. The Role of Dynamically Adjusting the K Value

In the transaction confirmation stage, transactions confirmed by k consensus nodes will be packaged into blocks and put on the chain, so the value of k directly affects the delay and security of transaction confirmation. It is easy to know that when the value of k is larger, it means that the transaction needs to reach a higher degree of consensus of the entire network, and it has stronger security, but if some nodes are offline or under poor network conditions, the transaction cannot be received. When verification or confirmation information cannot be sent, the entire network does not reach the number of k nodes required for confirmation, and the transaction will be put on hold to cause system congestion. At this time, the delay of the transaction will increase sharply, and even a consensus cannot be reached in the end. Conversely, the smaller the k value, the smaller the number of malicious nodes that need to make the transaction wrongly confirmed and put on the chain, and the greater the probability of doing evil. Even if the wrong transaction is questioned and censored in the block, it will affect the system throughput, increase network resources and storage overhead, and thus affect the stability of the entire system. Therefore, a k value that can be changed flexibly is needed to balance the performance, security and stability of the system, so that the system has better flexibility and is suitable for various application scenarios, especially those with variable transaction volumes.

Since the value of k affects the performance, security and stability of the blockchain system, and the value of k is adjusted by the master node, the authority of the master node is greater than that of ordinary verification nodes, and the reward and punishment system does not do anything for the master node to maliciously adjust the k value. Therefore, in order to conform to the idea of decentralization and reduce the probability of the master node doing evil, the dynamic transaction confirmation mechanism design master node can only propose to reduce or increase the k value, and the range of each reduction or increase shall not be higher than 10%, thereby limiting the permissions of the master node. Because of the fast block generation speed of the alliance chain, the k value can be reduced to 0.53 times or increased to 1.77 times of the original in 12s (6 block time). Even if the malicious master node maliciously adjusts the k value in the reverse direction, multiple consecutive malicious master nodes are required to ensure the impact on the system. Therefore, dynamically adjusting the k value not only ensures the flexibility of the system, but also improves its security.

3. Experimental Design

3.1. System Performance



ISSN 2616-5775 Vol. 5, Issue 5: 17-25, DOI: 10.25236/AJCIS.2022.050503

Figure 3: PBFT consensus flow chart

It is assumed that the time for a node to process a message is fixed as a, and the time for information transmission is fixed and t. The time to complete a message confirmation in PBFT consensus is set to T_2. The PBFT process is shown in Figure 3.

In the dynamic transaction confirmation consensus algorithm, the time to complete a consensus is set to T_1. According to the dynamic transaction confirmation consensus flow chart, the algorithm complexity of reaching a consensus is calculated as shown in Equation 2.

$$T_1 = 4t + (k+2)a \tag{2}$$

According to the PBFT consensus flow chart, the time to complete a consensus is shown in Equation 3.

$$T_2 = 5t + (\frac{4}{3}n + 3)a \tag{3}$$

From (3)-(2), formula (4) is obtained.

$$T_2 - T_1 = t + \left(\frac{4}{3}n + 1 - k\right)a \tag{4}$$

Therefore, the performance efficiency of the dynamic transaction confirmation consensus algorithm is better than that of the PBFT algorithm. The specific value is determined by the message transmission time t, the number of nodes n, the value of k, and the message processing time of a.

In addition, if the value of k is set to k_1, k_2, then for different k values, the time difference for completing a message confirmation, that is, the reduced transaction confirmation delay is shown in Equation 5.

$$\Delta T = \left|k_2 - k_1\right| a \tag{5}$$

So, changing the value of k in the dynamic transaction confirmation mechanism can indeed affect transaction latency, but since the value of a is usually measured in milliseconds, the impact on system throughput is limited.

3.2. Stability

For the dynamic transaction confirmation mechanism, the number of messages that need to be sent to complete one transaction information confirmation is shown in Equation 6.

$$S_1 = (n-1)^2 + 2n - 3 \ \Delta T = \left|k_2 - k_1\right| a \tag{6}$$

The number of messages that the PBFT algorithm needs to send to complete a transaction confirmation is shown in Equation 7.

$$S_2 = 2(n-1)^2 + 2n - 2 \tag{7}$$

It can be seen from the above that the number of messages required by the PBFT algorithm to complete a consensus is about twice that of the dynamic transaction confirmation mechanism. Under the same computer and network conditions, although the dynamic transaction confirmation algorithm and the PBFT algorithm are both affected by the network scale (the number of consensus nodes), the dynamic transaction confirmation algorithm needs to send fewer messages to complete a consensus and occupies more computer resources. It has better stability, and can also reduce the consumption of network resources and storage space. In the case of correct transactions and successful consensus, the value of k does not affect its stability. However, if some verification nodes are incorrectly verified, the system will repeatedly question the transaction due to normal nodes, causing the transaction to go on the chain after "secondary verification", which will reduce the system transaction throughput until the malicious node is proposed to verify the node. The set returns to normal after that, so the value of k affects system

ISSN 2616-5775 Vol. 5, Issue 5: 17-25, DOI: 10.25236/AJCIS.2022.050503

stability when there are malicious nodes or erroneous transactions.

3.3. Safety

From the three parts of transaction verification, transaction questioning, and transaction review, the possible malicious ways of nodes are analyzed. First of all, for the master node, the master node is responsible for four functions: adjusting the k value, counting the number of confirmed nodes, packaging and publishing blocks, and counting the number of questioned nodes. If the master node publishes a different k value to each verification node, or does not publish the k value to some nodes, the node that has not received the k value thinks that the state is inconsistent due to the replacement of the master node. The master node is required to record the k value and voting information (confirmation node address) in the block. If the block information is found to be inconsistent, it will be deemed that the master node has done evil and the points will be deducted. The questioning information is broadcast to all nodes. If the master node does not process it, the next master node will also process it. Second, for the verification node, the verification node is responsible for the two responsibilities of verifying the transaction and questioning the transaction. Incorrectly verifying transactions will be regarded as reducing points and deducting points, and the verification nodes will be incentivized to check every transaction on the blockchain by rewarding points. As for the instability of the system caused by the reverse adjustment of the k value by the master node, the impact on the blockchain is limited, and it requires a huge cost of doing evil. From this, it can be preliminarily concluded that the system is safe and practical.

In addition, the value of k determines the number of confirmations required for the transaction to be on the chain. Obviously, when the value of k is larger, the more confirmations are required, the more malicious nodes are required to complete an error on the chain, and the greater the cost of doing evil. big. Therefore, the value of k affects the security of the system.

4. Experimental Design

4.1. Lab Environment

The simulation system is written in C++ and uses multi-threading to simulate clients and nodes. In this test, one client is used to continuously send transactions, multiple consensus nodes and one master node. This system has three parts: transaction module, consensus module and review module. The performance, security and stability of the system will be evaluated through system throughput, transaction delay, review rate and the number of nodes that can be carried. The detailed configuration is shown in Table 1.

Table 1: Software and hardware environment configuration

Software and hardware environment	Configure
CPU	2.6 GHz Intel Core i7-9750H
RAM	16GB 2667 MHz DDR4
System	Mac OS

4.2. Purpose

1) Test the performance of the dynamic transaction confirmation alliance chain system in performance efficiency, fault tolerance, security and network scale and compare it with the system designed based on the PBFT algorithm

2) Test the effect of k value on performance efficiency, fault tolerance, security and network scale

4.3. Experimental Results

4.3.1. The Effect of K Value on Performance Efficiency

Based on the dynamic transaction confirmation system, the consensus process is modified to PBFT consensus, and the control variables ensure that the two are the same in other aspects, without affecting the experimental data. The final measured data is shown in Figure 4.

ISSN 2616-5775 Vol. 5, Issue 5: 17-25, DOI: 10.25236/AJCIS.2022.050503

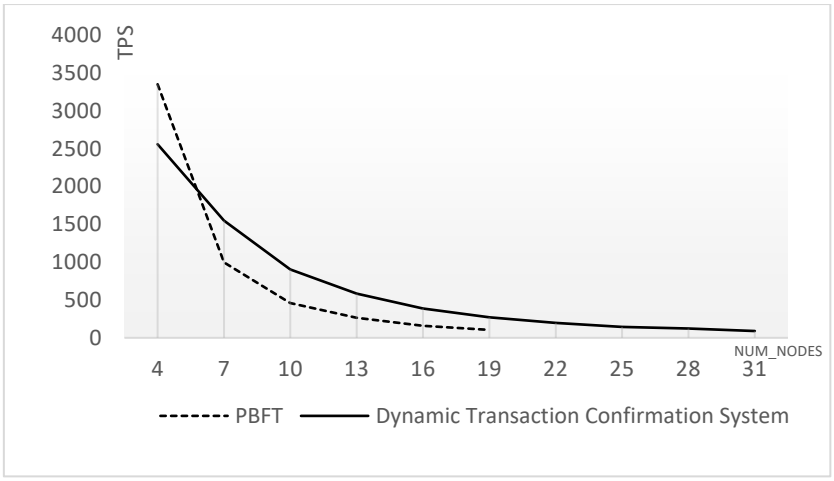


Figure 4: Dynamic transaction confirmation system, number of PBFT nodes and changes in TPS

As can be seen from the above figure, when the number of nodes is 4, the TPS of the dynamic transaction confirmation system (k=2) is smaller than that of the PBFT algorithm, because the number of nodes required to be confirmed and submitted by the PBFT algorithm is also 2, and the dynamic transaction The structure of the validation system is more complex. In addition, when the number of nodes is small, the resources of both parties are not much, and the dynamic transaction confirmation system is not dominant in comparison. In the case of an increase in the number of nodes, the dynamic transaction confirmation system, which occupies less system resources and has fewer message transmission times, can tolerate more nodes than the PBFT algorithm whether it is TPS or a single computer. Therefore, the dynamic transaction confirmation system does have better performance and stability.

Next, test whether the value of k will affect the TPS of the system. Set up a client to send the correct transaction, simulate 13 consensus nodes in a multi-threaded manner to receive and verify the transaction, adjust the size of the k value, test the change of the system throughput, take the average of multiple measurements, and measure the k value and the system throughput relationship is shown in Figure 5.

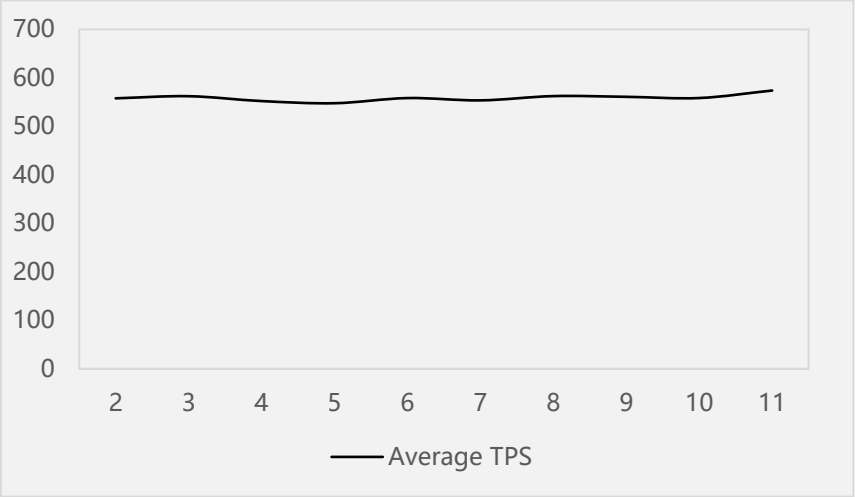


Figure 5: The relationship between TPS and k value under the same number of nodes

As can be seen from the above figure, in the case of the same number of nodes and no erroneous transactions, the size of the k value does not affect the system throughput.

4.3.2. Influence of K Value on Security and Stability

Simulate a dynamic transaction confirmation blockchain system with 13 consensus nodes. In order to facilitate the experimental test, it is assumed that there are 6 malicious nodes, and the malicious method is to incorrectly confirm the transaction so that the number of confirmations reaches the value of k, and the malicious node continuously incorrectly verifies the transaction until it is kicked out of the verification node set. The TPS of the system to complete the consensus of each block under different k

ISSN 2616-5775 Vol. 5, Issue 5: 17-25, DOI: 10.25236/AJCIS.2022.050503

values is measured, as shown in Figure 6.

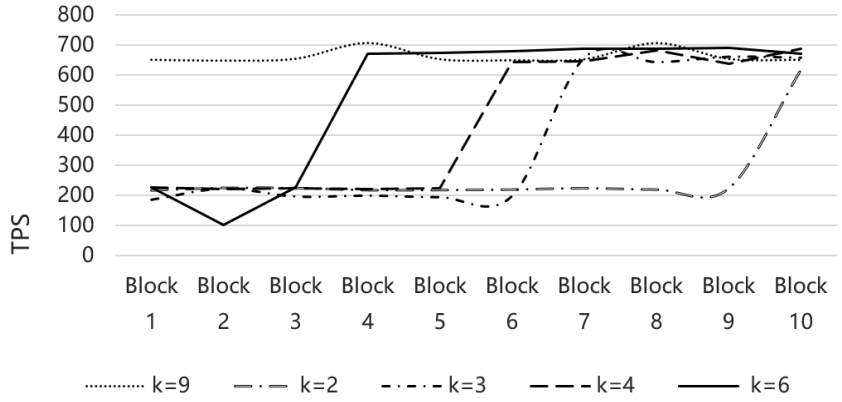


Figure 6: TPS diagram of the system under different k values

As can be seen from the above figure, if multiple malicious nodes mistakenly confirm the transaction and make it enter the review stage, the TPS of the system will drop to 1/3 of the original, and when some malicious nodes are deducted from the consensus node set due to evil deductions, When the number of remaining malicious nodes is less than k, TPS returns to normal. The larger the value of k, the fewer blocks are affected. If the value of k is greater than the sum of malicious nodes in the network, the system will not be affected by it.

Therefore, when there are malicious verification nodes or wrong transactions are wrongly verified, the value of k does affect the stability of the system.

5. Conclusions

This paper mainly proposes and designs a dynamic transaction confirmation alliance chain system. This system is not only a consensus algorithm, but also ensures the normal operation of the system through consensus algorithms, review mechanisms, point pools, and reward and punishment systems. It has good performance in terms of scalability and scalability, and by dynamically adjusting the k value, it has more flexibility and a wider range of application scenarios, especially suitable for some nodes with frequent access and poor performance of network or node facilities. In the follow-up, we will further study the optimization and improvement of the master node election system and points system.

References

[1] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system [J]. Decentralized Business Review, 2008: 21260.
[2] Shen Xin, Pei Qing-qi, Liu Xue-feng. A Review of Blockchain Technology [J]. Journal of Network and Information Security, 2016, 2(11): 11-20.
[3] Lamport L. Paxos made simple [J]. ACM Sigact News, 2001, 32(4): 18-25.
[4] Ongaro D, Ousterhout J. In search of an understandable consensus algorithm [C]// 2014 {USENIX} Annual Technical Conference ({USENIX}{ATC} 14). 2014: 305-319.
[5] Bessani A, Sousa J, Alchieri E E P. State machine replication for the masses with BFT-SMART [C]// 2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks. IEEE, 2014: 355-362.
[6] Rocket T. Snowflake to avalanche: A novel metastable consensus protocol family for cryptocurrencies [J]. Available [online]. [Accessed: 4-12-2018], 2018.
[7] Aublin P L, Mokhtar S B, Quéma V. Rbft: Redundant byzantine fault tolerance [C]//2013 IEEE 33rd International Conference on Distributed Computing Systems. IEEE, 2013: 297-306.
[8] Castro M, Liskov B. Practical byzantine fault tolerance [C]// OSDI. 1999, 99(1999): 173-186.
[9] Delegated Byzantine Fault Tolerance.https://docs.neo.org/v2/docs/zh-cn/basic/technology/dbft.html. Jan. 2021
[10] Verifiable Byzantine Fault Tolerance.https://github.com/ontio/documentation/blob/master/vbft-intro/vbft-intro.md.Jan.2021
[11] Luu L, Narayanan V, Zheng C, et al. A secure sharding protocol for open blockchains [C]//

ISSN 2616-5775 Vol. 5, Issue 5: 17-25, DOI: 10.25236/AJCIS.2022.050503

Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. 2016: 17-30.
[12] Liu J, Li W, Karame G O, et al. Scalable byzantine consensus via hardware-assisted secret sharing [J]. IEEE Transactions on Computers, 2018, 68(1): 139-151.
[13] Xu Zhi-li, Feng Hua-min, Liu Biao. An Improved PBFT Efficient Consensus Mechanism Based on Credit [J]. Computer Application Research, 2019, 36(9): 2788-2791.
[14] Cao Zhao-lei. A Consensus Mechanism for Consortium Chains [J]. Space Network Security, 2019, 10(1): 1-6.
[15] Leng Ji-dong, Lv Xue-qiang, Jiang Yang, et al. Research Review on Consensus Mechanism of Consortium Chain [J]. Data Analysis and Knowledge Discovery, 2021, 5(1): 56-65.
[16] Zheng Min, Wang Hong, Liu Hong, et al. Research Review of Blockchain Consensus Algorithms [J]. Information Network Security, 2019, 19(7): 8-24.
[17] Lu Ge-hao, Xie Li-hong, Li Xi-yu. Comparative Research on Blockchain Consensus Algorithms [J]. Computer Science, 2020, 47(6A): 332-339.
[18] Jin Shi-xiong, Zhang Xiao-dan, Ge Jing-guo, et al. Research Review of Blockchain Consensus Algorithms [J]. Journal of Information Security, 2021, 6(2): 85-100.