

Dual Vote Confirmation based Consensus Design for Blockchain integrated IoT

Zheng Qiu, Jianjun Hao, Yijun Guo, Yi Zhang

Beijing Key Laboratory of Network System Architecture and Convergence

Beijing University of Posts and Telecommunications

Beijing, China

qz.doris@bupt.edu.cn, jjhao@bupt.edu.cn, guoyijun@bupt.edu.cn, buptzy@bupt.edu.cn

Abstract—Blockchain technology is adopted by many research to provide underlying security protection for the booming Internet of Things (IoT) industry. Nonetheless, due to some aspects of IoT, such as resource-constrained devices, heterogeneous mass devices, and real-time transactions, the traditional blockchain consensus algorithm is not suitable for the scene of blockchain-IoT integration. This paper proposes a Dual Vote Confirmation based Consensus (DVCC) mechanism for the Blockchain-IoT integration (BlockIoT). First, we adopt a role division approach to solve the problem of limited resources of devices in IoT. Second, multi-dimensional credit score, dual confirmation mechanism, classified random selection mechanism, and credit score based incentivization are designed to improve the security and fairness of the consensus. Third, a voting mechanism is proposed to improve the delay performance and avoid energy waste. Finally, the experimental results show that DVCC gains better delay and throughput performance on a large scale while guaranteeing the security and fairness of the consensus.

Index Terms—blockchain, Internet of Things, consensus, credit score, dual vote confirmation

I. INTRODUCTION

With the rapid development of electronics and communication technology, the Internet of Things (IoT) industry is booming all over the world. It was predicted that by 2020, there would be 26 billion IoT devices in the world [1]. In the next few years, the machine-to-machine connection would increase from 780 million in 2016 to 3.3 billion in 2021 [2].

IoT is a revolutionary technology and can be applied in various areas such as food traceability, smart transportation, smart industry and smart healthcare, etc [3]. In these areas, IoT devices generate sensitive information which needs to be securely transmitted and stored in IoT networks. To address the security and privacy issues in IoT networks, Blockchain is proposed to integrate with IoT since blockchain technology can provide an immutable and traceable form of data that would be used across a peer-to-peer network [4]. Recently, there are lots of studies on strengthening IoT security by using blockchain technology. One of them is research conducted by Dinan Fakhri, where blockchain is used for making secure communication among IoT devices [5]. Besides, there are research [6] [7] focusing on making access management in IoT via blockchain technology. Sotirios Brotsis [8] proposed a blockchain-based solution to deal with the preservation of digital forensic evidence collected from IoT networks. In [9]

was proposed blockchain as a solution for IoT security issues, involving secure communication, authentication of users, discovering legitimate IoT devices at large scale, and secure configuration for IoT devices, etc.

However, owing to mass resource-constrained devices and numerous real-time transactions in IoT networks, there will be certain challenges in efficiency and fairness when applying blockchain technology to the IoT domain. The well-known proof-of-work (PoW) consensus, which is widely used in many platforms such as Bitcoin and Ethereum, is not suitable for IoT since it needs resource-intensive devices and has low transaction throughputs. Therefore, Deepak Puthal proposed proof-of-authentication (PoAH) [10] running in resource-constrained distributed systems, but it could not guarantee the security if the trusted nodes are attacked. Proof of Understanding (PoU) was proposed in Babelchain [11] to adapt PoW for IoT applications. Instead of solving hash puzzles, mining efforts are concentrated on useful computation, such as tackling a critical problem in IoT communications. Another classic type of consensus algorithm, the Practical Byzantine Fault Tolerance (PBFT), is not proper for the practical IoT applications either since the control of the blockchain is concentrated on a few nodes. Furthermore, the performance of PBFT would degrade dramatically as the number of nodes participating in the consensus increases.

Regarding those problems above, this paper proposes a dual vote confirmation based consensus (DVCC) mechanism for the blockchain integrated with IoT(BlockIoT). To improve the efficiency and security of the whole system, devices in different layers of IoT are assigned to play different roles in the blockchain. As the credit score mechanism in DVCC, a node's credit score determines the voting rights in consensus and the probability of being selected as the miner, thus enhancing the security of the consensus process. Additionally, the voting mechanism can help reduce energy waste and improve efficiency, and the dual confirmation mechanism could reduce the security risks because the consensus need confirmation from both ordinary participants and IoT service providers. Also, by classifying the nodes and selecting the verifiers from each class randomly, DVCC guarantees the fairness and efficiency on a large scale.

The remaining of this paper is organized as follows: Section II describes the overall system model. In section III intro-

duces the consensus mechanism (DVCC). The performance of DVCC and experimental results are explicitly discussed in section IV; Section V concludes the paper.

II. SYSTEM MODEL

Our system model integrates the blockchain technology into the Internet of Things, namely BlockIoT. In consideration of differences in needs and capabilities, IoT devices in different layers are assigned different roles. BlockIoT architecture is depicted on the left side of Fig. 1.

A. Sensing Layer

Proposer: Lack of computing and storage resources, IoT devices in sensing layer such as cameras, traffic lights, refrigerators, etc., can become the proposers, which do not participate in consensus but can send transactions, share information and exchange energy, etc. They can join or leave the network anytime.

B. Network Layer

Devices in the network layer are divided into three types.

a) Bookkeeper: Resource-intensive devices in the network layer, such as a gateway device, a base station, and so on, can become bookkeepers by applying and registering. Bookkeepers participate in the maintenance of the blockchain, the distribution of the block and transaction in the network.

b) Verifier: Responsible for verifying the transactions and voting for the block, verifiers are randomly selected from classified bookkeepers.

c) Miner: The miner, the node with the highest score among the verifiers, is responsible for packing multiple transactions and generating the block.

C. Middleware Layer

a) Members: In the middleware layer, there are servers controlled by government regulatory institutions and some IoT giants, involving IoT equipment vendors, IoT service providers, etc. They are members who ally to guarantee the security of the blockchain together. Members can verify and vote on the block. Specifically, they maintain a consortium blockchain on which a smart contract will be established to act as the Election Committee and Score Manager.

b) Election Committee: The Election Committee is responsible for selecting verifiers and the miner at the end of each consensus process.

c) Score Manager: The Score Manager scores all bookkeepers. The scoring criteria are negotiated by members and approved by all participating nodes.

Besides, there are two essential managerial roles in the consensus process.

d) Identity Manager: The Identity Manager manages the identity of all the bookkeepers in the network. To become a bookkeeper, the proposer needs to submit an application and obtain the real-name authentication of the Identity Manager.

e) Classifier: The Classifier divides the bookkeepers into N categories according to multiple factors such as the geographical distribution, the business type, the device contribution and the transaction relationship, etc. N is determined by members.

D. Application layer

The IoT applications are based on cloud computing, service support, and the blockchain, etc., including intelligent transportation, smart industry, food traceability and so on.

III. CONSENSUS MECHANISM

The DVCC mechanism initially assesses the credibility of the bookkeepers from multiple perspectives. Then, in each consensus process, all verifiers and members will vote for the block. Only when both members and verifiers reach an agreement on the block, can it be confirmed, which ensures the security of the consensus process. At the end of each consensus process, the Election Committee will select verifiers and the miner for the next round of consensus. Finally, an incentive mechanism based on the consensus contribution and credit score is proposed to reward all participating nodes. The proposed mechanisms are shown on the right side of Fig. 1.

A. Multi-dimensional Credit Score

The credit score expresses the credibility of a node. The higher the score, the higher the reliability and credibility of the node. The following 7 aspects determine the total credit score of a bookkeeper:

a) Endorsement of alliance members: Members can endorse a bookkeeper by calling the endorsement function of the Score Manager. After the block containing the endorsement transaction is confirmed, the Score Manager will automatically increase the bookkeeper's credit score.

b) Evaluation of the trading importance: The PageRank algorithm [12], which was applied in evaluating the relevance and importance of web pages, is used to assess the trading importance of nodes on the blockchain. The more important the node is in the trading network, the higher the score.

c) Node rating: After completing the transaction, two trading parties can score each other subsequently based on the performance and submit the score to the Score Manager.

d) Behavior in consensus: If a node behaves honestly in the consensus process, such as generating or verifying a valid block, it can receive a bonus score, and vice versa.

e) The amount of deposit: This part of the score is determined by the amount of deposit paid by bookkeepers.

f) Online time: Online time is an important factor to be considered in evaluating the credibility of a node. The longer the online time, the higher the score in this part.

g) Equipment security performance evaluation: Bookkeepers' security performance will be evaluated in combination with its CPU power, memory and storage capacity.

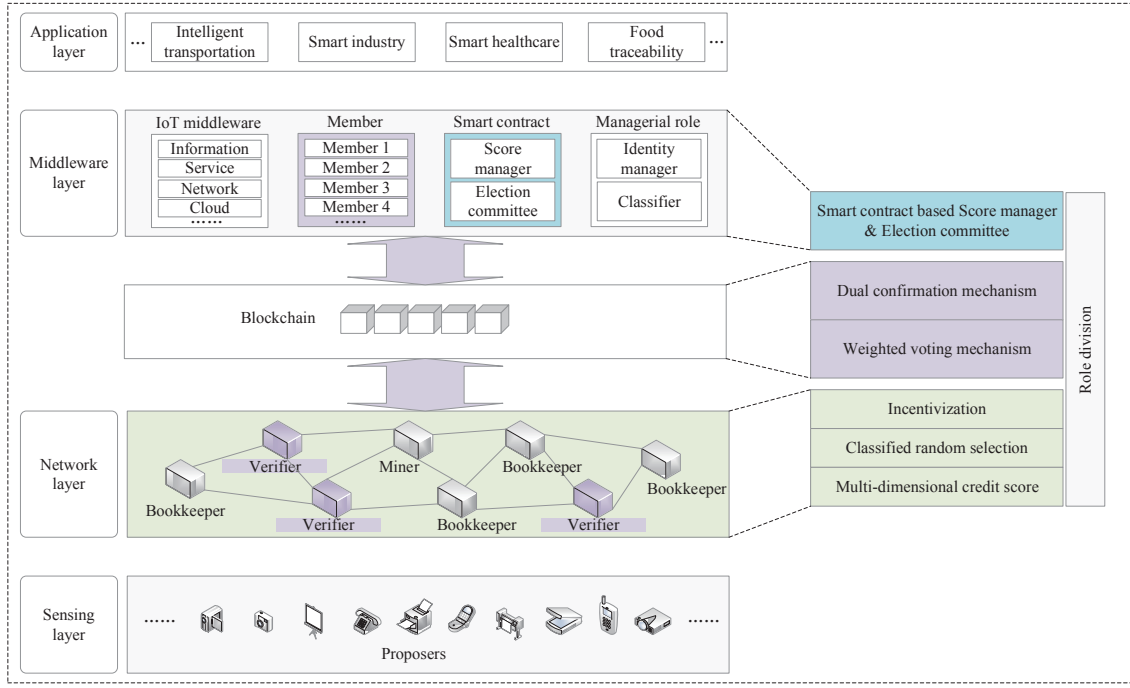


Fig. 1. BlockIoT Architecture

Suppose the score of each part is $\{x_1, x_2, \dots, x_7\}$ with coefficients $\{\alpha_1, \alpha_2, \dots, \alpha_7\}$, then the total score can be calculated as follows:

$$X = \sum_{i=1}^7 \alpha_i x_i \quad (1)$$

where $x_i \in [0, 100]$, $\alpha \in [0, 1]$, $\sum_{i=1}^7 \alpha_i = 1$. The credit score based on multi-dimensional evaluation reflects the trustworthiness of the node and improves the security of the system by affecting the voting and selection process in the consensus.

B. Consensus process

The consensus process proceeds in rounds. Algorithm 1 outlines the block generation process. In this process, there are two proposed mechanisms which improve the security and efficiency of the consensus process:

a) *Dual Confirmation Mechanism*: For the final confirmation of the block, a dual confirmation mechanism is designed: more than $2/3$ votes from both verifiers and members are needed. Verifiers, ordinary nodes in the network layer, and members, the government regulators and some IoT giants are less likely to be attacked simultaneously or commit crimes together. Hence, the dual confirmation mechanism can help guarantee the correctness and legitimacy of the generated block.

b) *Voting Mechanism*: For members, their status is equal, so they have the same voting rights in the consensus process. For verifiers, two voting mechanisms are proposed, which can be adjusted according to the application requirements. One is called the Simple Voting mechanism, in which each verifier

Algorithm 1: Procedure of Block Generation

Input: $maxTimeout$,
 Verifiers($\{V_i : i \in N\}$), Members($\{M_j : j \in M\}$).
 Private (PrK) and Public key(PuK)

- 1 The miner i collects some transactions into a block
- 2 $(S_{PrK})(Block)$ is broadcast to verifiers and members
- 3 Verifiers and members verify the block:
- 4 **if** *block is valid* **then**
- 5 $Vote = S_{PrK}(Hash(Block))$
- 6 $Vote$ is sent back to miner
- 7 **end**
- 8 Miner collect Votes:
- 9 **while** $Time \leq maxTimeout$ **do**
- 10 **if** $(\#of(Votes \in V_i) \geq \lceil \frac{2N}{3} \rceil) \& (\#of(Votes \in M_j) \geq \lceil \frac{2M}{3} \rceil)$ **then**
- 11 $(S_{PrK})(Block + \sum Votes)$ is broadcast
- 12 **break**
- 13 **end**
- 14 **end**
- 15 **if** *time – out occurred and consensus not achieved* **then**
- 16 drop the block
- 17 $(S_{PrK})(failure\ message)$ is broadcast
- 18 **end**
- 19 All nodes in network:
- 20 **if** *valid block received* **then**
- 21 delete the transactions of the block
- 22 add the block into the chain
- 23 **end**
- 24 The Election Committee selects miner and verifiers for next round of consensus
- 25 Goto(Step 1) for next consensus.

has the same voting rights. One node, one vote. The other is the Weighted Voting mechanism based on the credit score. The voting rights are proportional to the credit score. The higher the score, the greater the voting rights.

Since many nodes in IoT networks are willing to participate in blockchain consensus, the voting mechanism ensures consensus efficiency while allowing more nodes to participate in the consensus, and avoids the massive energy waste of PoW. Particularly, the Weighted Voting mechanism enhances security by giving more votes to the nodes with higher scores, which are more reliable.

C. Selection of Verifiers and Miner

a) *Verifiers: classified random selection:* As bookkeepers are divided into N categories, verifiers will be randomly selected from each category. The selection algorithm is as follows:

Suppose N is the number of verifiers, and the public key of the verifiers in consensus epoch t is $PuK_t[i], i \in N$, then the random seed $S_i(i \in N)$ for epoch $(t+1)$ can be specified as follows:

$$S_i = \begin{cases} (\bigoplus_{j=1}^K Vote_j) \oplus T \oplus PuK_t[i] & \text{consensus succeeds} \\ T \oplus PuK_t[i] & \text{consensus fails} \end{cases} \quad (2)$$

where $Votes(\{Vote_j : j \in K\})$ are the verifiers' confirmation signatures for the block in epoch t and T represents the current time. Ultimately, the random seeds $S_i(i \in N)$ can be used to select verifiers of the next consensus epoch $(t+1)$ randomly.

The classified random selection can improve consensus security and fairness in IoT scenarios. First, there are numerous nodes in IoT networks, varying in the geographical distribution, the business type, the device resource, etc. Verifiers selected from different categories can better represent community consensus and are less likely to collude. Second, since the value of $Votes$ and T in the random selection process are unpredictable, the selected verifiers will also be unpredictable. Therefore, the random selection helps reduce the probability of verifiers being attacked and improve security. Third, the random selection ensures that each node in IoT networks has the same opportunity to participate in the consensus, which is essential in some IoT applications, e.g. food traceability.

b) *Miner: credit score based selection:* Among verifiers, the node with the highest score will be appointed as the miner. Since the credit score represents the node reliability, the selected miner is the least likely to be evil, and the generated block is the most reliable.

D. Incentivization

To regulate the behaviors of nodes and encourage more nodes to participate in the consensus, it is necessary to build an incentive mechanism and offer some rewards to honesty nodes.

a) *Money incentive:* The transaction fees will be allocated to nodes participating in the consensus according to the contribution of each node to motivate nodes to behave honestly in the consensus. For the sake of the fairness of commission distribution, we induce the power index concept in game theory and compute the Shapley value which indicates the contribution of each node in the voting process. Details about the Shapley value can be explained in [13].

b) *Score incentive:* To ensure fairness, each verifier will gain one point as long as it contributes to the consensus, regardless of its score. However, the derating mechanism is different. It is obvious that in the Weighted Voting mechanism, nodes with a higher score will have more power in the consensus, thus breaking the consensus more easily. To address this issue, a more severe punishment policy for nodes with higher scores is put forward. The higher the node's score, the more points it reduces for evil.

IV. PERFORMANCE EVALUATION

The performance of DVCC is analyzed and evaluated by a simple blockchain prototype system. The whole system is written in Java language, and multiple threads are created to simulate consensus nodes. The security performance, consensus delay and throughput capacity of DVCC are evaluated.

A. Security

a) *Dual confirmation mechanism:* We introduce the Nash equilibrium theory in game theory to illustrate the superiority of the dual confirmation mechanism. In this model, all members are regarded as one party, so are all verifiers. Meanwhile, it is assumed that two sides make decisions independently in each round of consensus since there is little chance for them to collude. In the consensus process, each side can choose to be honest, meaning that more than two-thirds of the nodes do the right validation, or do evil. Only when both sides are honest, can the right consensus be reached. The benefits of all strategy profiles for the two sides are listed in Table 1.

TABLE I
STRATEGY PROFILES

(members, verifiers)		members	
		honest	evil
verifiers	honest	(1,1)	(-4,2)
	evil	(2,-4)	(1.5,1.5)

When both sides are honest, they get equal rewards, let's say 1. However, if one is honest and the other does evil, the honest one will report the evil so as to get a higher reward of 2, and the guilty side will be severely punished with 4. On the contrary, when both sides do evil, they will get more reward than honesty with 1.5, since they can manipulate the blockchain to meet their interests.

This is a typical Nash equilibrium problem. In order to maximize their expected returns, players will choose the Nash equilibrium strategy profile (1, 1), that is, both sides choose to be honest. This also shows that the dual confirmation

mechanism can help improve the security of the consensus. Meanwhile, since the verifiers will be reselected in each consensus process, there is no chance for the two sides to collude through multiple rounds of consensus and reach the Pareto optimality (1.5, 1.5).

b) Classified Randomized Selection: Since members are consist of government departments and giant enterprises in the IoT industry, their equipment is supposed to be reliable and the evil probability is small enough to be ignored. For verifiers, suppose that there are N verifiers, and the probability that any node will do evil is P_e . Then the probability that less than $\frac{N}{3}$ verifiers do evil, also the probability of consensus success is represented by P :

$$P = \sum_{i=0}^{\lfloor \frac{N}{3} \rfloor} \binom{N}{i} P_e^i (1 - P_e)^{N-i} \quad (3)$$

Drawn the theoretical curve of P for different values of P_e and number of verifiers N , as shown in Fig. 2. As can be seen from Fig. 2, when P_e is less than 33%, P increases with N . P quickly approaches 1 as P_e decreases. Even if P_e is high, say 20%, the success probability P is extremely high, and as the number of verifiers N increases, P increases and gets closer to 1.

The classified random selection improves the consensus security by effectively reducing the evil probability, also the faulty ratio of bookkeepers, P_e . On the one hand, since verifiers and the miner are selected randomly, they are unpredictable and not easily attacked. On the other hand, each verifier is selected from different classes, meaning that they have a wider representation and a smaller probability of collusion [14], that is, P_e will be small.

We evaluate the probability of the consensus success with different ratio of faulty nodes and number of verifiers. During each experiment, we count 100 consensus process, and the number of the block which is successfully added into the blockchain represents the success probability. Fig. 2 illustrates the change of success probability in the Simple Voting mechanism and compares it with the theoretical curves. The simulation curve is a fitting of several scatter points, which is basically consistent with the theoretical curve. The difference is caused by experimental fluctuation.

c) Weighted Voting mechanism: Fig. 3 illustrates the comparison of the Simple Voting mechanism and the Weighted Voting mechanism in the success probability with the number of verifiers is 49 and the faulty ratio is 0.3. The results show that the Weighted Voting mechanism guarantees a higher success probability than the Simple Voting mechanism since nodes with higher scores which are less likely to be malicious, will have more power than lower score nodes in the voting process. Particularly, the steep drop in the experiment 6 of the simple voting is a random fluctuation due to the randomized selection of verifiers.

B. Consensus Delay

The consensus delay is the time required for a new block from being generated to being confirmed. Suppose T_{delay}

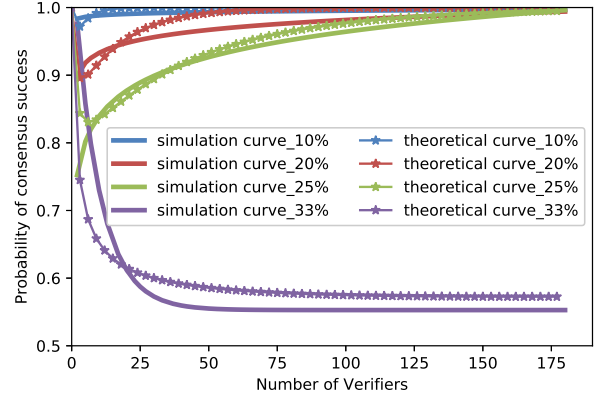


Fig. 2. Consensus success probability

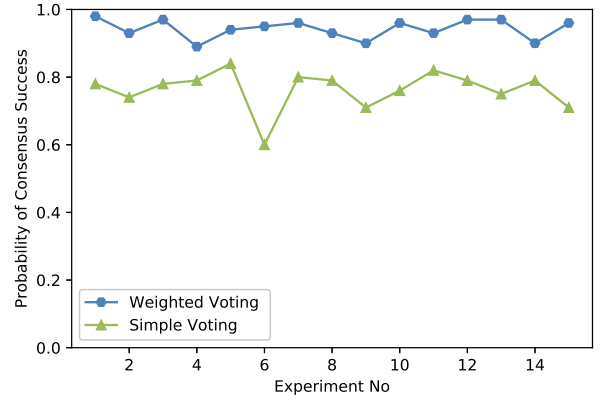


Fig. 3. Consensus success probability comparison between Simple Voting and Weighted Voting

represents the delay time, then it can be specified as follows:

$$T_{delay} = T_{consensus} + T_{broadcast} \quad (4)$$

where T_{delay} and $T_{broadcast}$ represent the time spent in block consensus and broadcast.

We evaluate the delay of DVCC and PBFT with a different number of consensus nodes, count 100 blocks and calculate the average delay. Fig. 4 illustrates the experimental results. The average delay of DVCC remains stable whereas the delay of PBFT has an upward trend with the number of consensus nodes increases, because DVCC requires only one broadcast process and one voting process, and PBFT requires multiple rounds of communication among nodes.

Fig. 5 illustrates the effects of the incentive mechanism on the delay time over time in the configuration that the number of consensus nodes and the faulty ratio is 49 and 33% respectively. In Fig. 5, each time period refers to 20 rounds of consensus. Since the delay time of single-round consensus may fluctuate greatly, this paper takes the average delay time as the result.

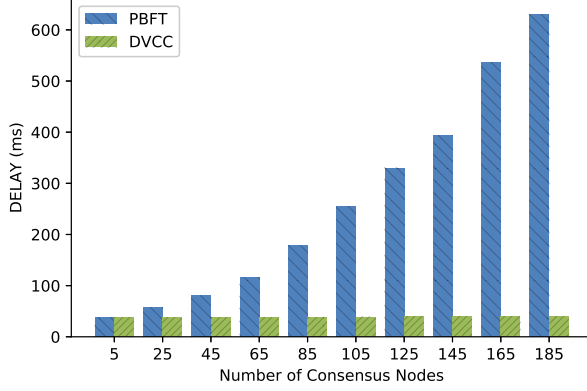


Fig. 4. Delay comparison under different number of consensus nodes

Since the verifiers in each consensus process are randomly selected, when the faulty ratio is 33%, the success of each consensus process is relatively random, and the delay time fluctuates greatly, which explains the steep drop at time period 2 of DVCC without incentives. At the overall situation: in the case of incentives, the delay time fluctuates around 800ms; without incentives, the delay time gradually decreases from 800ms to around 350ms. As time goes by, the average delay of DVCC with score incentive is gradually decreasing while the DVCC without incentives remains at the same time level.

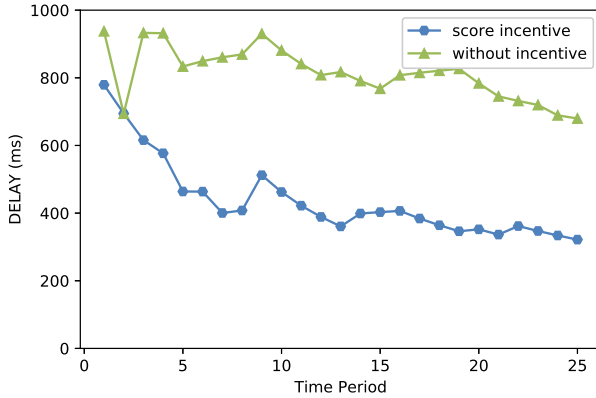


Fig. 5. The change of delay over time

C. Consensus Delay

The transaction per second (TPS) represents the throughput capacity of the system, and it can be specified as follows:

$$TPS = \frac{SumTransactions}{\Delta_t} \quad (5)$$

where Δ_t is a period of time, $SumTransactions$ is the total number of transactions confirmed and written to the block during this time.

The block time is the block packing interval. We take 9 different block time, such as 1ms, 50ms, 100ms, 500ms, 1s, 5s, 10s, 15s, etc., and test 100 times each block time, taking the average as the *TPS*. Fig. 6 illustrates the experimental results in the configuration that the number of verifiers is 16, 24 and 32. As Fig. 6 shown, *TPS* will first rise and then fall with the increase of the block time. First, when the block time is small, the block contains fewer transactions and does not take full advantage of the computing power of verifiers and the transport capacity of the network. Second, when the block time increases to a certain value, the nodes' computing capacity and network transmission capacity are just saturated and the peak of *TPS* occurs. Third, When the block time continues to increase, on the one hand, with the increased number of transactions contained in the generated block, the block volume increases, thus both the time of block packaging for the miner and the network transmission time increase; on the other hand, the increasing number of transactions makes the verifier nodes spend more time on the verification of transactions. Combined with the above two aspects, *TPS* will decrease accordingly.

Under the same hardware conditions, the transaction rate and computing power of nodes remain stable, and the change of network transmission time from 16 verifiers to 24 or 32 verifiers is negligible, thus, changing the number of verifiers will not significantly affect the block time at the peak. However, the value of *TPS* is related to the delay time of consensus. Even if the number of transactions generated under the same hardware condition is stable, the consensus delay time will increase slightly as the number of verifiers increases (as shown in Fig. 4 of the paper), thus *TPS* will decrease slightly. Thus, in Fig. 6, *TPS* peaks at around 2s of the block time, roughly 2850, and when the block time is controlled under 10s, *TPS* can be guaranteed to be above 2000, which can meet the performance requirements of most applications.

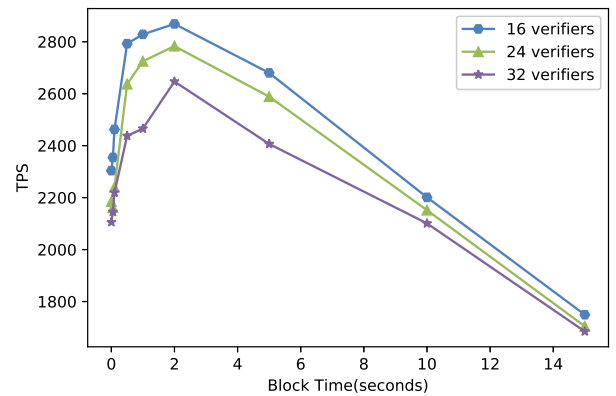


Fig. 6. TPS under different block production period

D. Fairness

The randomness of selection ensures that each bookkeeper has the same chance to become a verifier, reflecting a wide

range of community consensus and avoiding autarchy.

E. Finality

The block can always be generated since the Election Committee will reselect a miner to generate the block when consensus fails. Also, if one block obtains votes of over two-thirds members and verifiers, another block cannot get sufficient votes, so there will be no bifurcation in the blockchain.

F. Reliability

In view of the unsatisfactory network, such as packet loss and timeout, we set up a timeout mechanism for both the consensus process and the election process. In the consensus process, if the miner does not receive sufficient verified votes within the specified time frame, the consensus is judged to have failed, a failure message is broadcast, and the next round of consensus is restarted. In the selection process, if the election committee did not receive the message sent by the miner after the timeout, it is determined that the miner may be evil, the consensus fails, and the verifier and the miner are re-selected by using the current time and the public key of verifiers. Although the timeout mechanism is designed to ensure that the next round of consensus or selection can be completed in the case of poor network conditions, the re-consensus will certainly cause delay and throughput performance degradation to some extent.

It should be noted that due to timeout and randomized selection mechanism, the impact of poor network conditions is limited. In addition, in extreme cases where the overall network is in poor condition, the system can take protective measures of relying on the reliable members to complete the consensus and keep the system running.

G. Performance

a) *Energy Saving*: In the consensus process, nodes verify and vote for the block instead of spending massive energy solving a difficult math problem. Thus the DVCC is more resource-saving compared with PoW.

b) *Scalability*: Due to the division of node roles, only a small group of nodes are selected as verifiers to verify the block, and numerous bookkeepers could participate in the maintenance of the blockchain without lowering the system performance. The DVCC offers good node scalability compared with those BFT-based algorithms.

V. CONCLUSION

In this paper, we propose a dual vote confirmation based consensus (DVCC) mechanism for blockchain integrated IoT (BlockIoT). First of all, IoT devices in different layers are arranged for playing different roles in consensus according to their computing and storage capabilities in order to integrate IoT and blockchain better. Second, the proposed credit score model, dual confirmation mechanism, and classified random selection mechanism help improve security and fairness. Besides, the voting mechanism benefits to reduce energy waste and improve the efficiency of the consensus. Finally, an

incentive mechanism is developed to motivate nodes to behave honestly in consensus.

It is demonstrated from the experimental results that DVCC ensures the efficiency and security of the consensus while enhancing the fairness in the IoT scenario. The consensus delay remains low and steady with the number of nodes increases, which is suitable for IoT scenarios with many nodes involved. Moreover, the throughput capacity of the system is capable of accommodating most IoT applications.

ACKNOWLEDGMENT

This work was supported by grants from the National Key Research Program Projects (No. 2018YFB1800704).

REFERENCES

- [1] Forecast: The Internet of Things, Worldwide, 2013, Gartner, Stamford, CA, USA, Nov. 2013.
- [2] Index C V N. Global mobile data traffic forecast update, 2016–2021 white paper[J]. Cisco: San Jose, CA, USA, 2017.
- [3] Ahmad F, Ahmad Z, Kerrache C A, et al. Blockchain in Internet-of-Things: Architecture, Applications and Research Directions[C]//2019 International Conference on Computer and Information Sciences (ICCIS). IEEE, 2019: 1-6.
- [4] Underwood Sarah, "Blockchain beyond bitcoin", Commun. ACM, vol. 59, no. 11, pp. 15-17, October 2016.
- [5] D. Fakhri and K. Mutijarsa, "Secure IoT Communication using Blockchain Technology," 2018 International Symposium on Electronics and Smart Devices (ISESD), Bandung, 2018, pp. 1-6.
- [6] R. Agrawal et al., "Continuous Security in IoT Using Blockchain," 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, 2018, pp. 6423-6427.
- [7] Novo O. Blockchain meets IoT: An architecture for scalable access management in IoT[J]. IEEE Internet of Things Journal, 2018, 5(2): 1184-1195.
- [8] Brotsis S, Kolokotronis N, Limniotis K, et al. Blockchain solutions for forensic evidence preservation in IoT environments[J]. arXiv preprint arXiv:1903.10770, 2019.
- [9] Singh M, Singh A, Kim S. Blockchain: A game changer for securing IoT data[C]//2018 IEEE 4th World Forum on Internet of Things (WF-IoT). IEEE, 2018: 51-55.
- [10] Puthal D, Mohanty S P. Proof of authentication: IoT-friendly blockchains[J]. IEEE Potentials, 2018, 38(1): 26-29.
- [11] Bitcoin Fog, 2016. Available online: <http://www.the-blockchain.com/2016/05/01/babelchain-machine-communication-proof-understanding-new-paper/>. (Accessed 1 February 2018).
- [12] Page L, Brin S, Motwani R, et al. The PageRank citation ranking: Bringing order to the web[R]. Stanford InfoLab, 1999.
- [13] L. S. Shapley, "A value for N-person games" in Contributions to the Theory of Games Volume I, USA, NJ, Princeton:Princeton Univ. Press, 1950.
- [14] Schwartz D, Youngs N, Britto A. The Ripple protocol consensus algorithm[J]. Ripple Labs Inc White Paper, 2014, 5.