2024, 21(2 Special Issue): 0633-0642
https://doi.org/10.21123/bsj.2024.9735
P-ISSN: 2078-8665 - E-ISSN: 2411-7986

Baghdad Science Journal

Optimizing Blockchain Consensus: Incorporating Trust Value in the Practical Byzantine Fault Tolerance Algorithm with Boneh-Lynn-Shacham Aggregate Signature

***Dayong Zhang*[1], *Nur Haliza Abdul Wahab**[1], *Adi Wira Mohd Zin*[2]**

[1]Faculty of Computing, Universiti Teknologi Malaysia, Johor, Malaysia.
[2]Faculty of Business and Finance, Universiti Tunku Abdul Rahman, Perak, Malaysia.
*Corresponding Author.

PARS2023: Postgraduate Annual Research Seminars 2023.

Received 14/09/2023, Revised 10/02/2024, Accepted 12/02/2024, Published 25/02/2024

Abstract

The fundamental component of Blockchain technology, the consensus algorithm, is employed to guarantee data consistency among Blockchain nodes. Because it is resistant to Byzantine errors, consortium chains frequently use the Practical Byzantine Fault Tolerance (PBFT) consensus mechanism. Nevertheless, the current PBFT still has issues with high node communication complexity and random master node selection. This study proposes the IBFT consensus mechanism, which is based on the aggregate signature of Boneh-Lynn-Shacham (BLS) and node trust value. Multi-level indications in IBFT are made to determine each node's trust value. A few very reliable nodes are chosen to serve as consensus nodes. Whichever node has the highest trust value is selected as the master node. Afterwards, BLS aggregated signatures are used to optimize the consensus flow of PBFT. As a result, information sent between nodes is kept secure and node communication complexity is decreased. The simulation experiment results show that when compared to the PBFT, the IBFT consensus approach improves transaction throughputs by 61% and lowers latency by 13%.

Keywords: Blockchain, Boneh Lynn Shacham (BLS), Consensus Algorithm, Practical Byzantine Fault Tolerance (PBFT), Trust Value.

Introduction

Blockchain technology has its origins in Bitcoin. Satoshi published the Bitcoin white paper in 2008, which proposed a peer-to-peer electronic cash system that is completely independent of a central institution[1,2]. Characterized by anonymity, decentralization and tamper-resistance[3,4,5], Blockchain's appeal has spanned across various fields. Consensus algorithms are the core technology of Blockchain[6,7], ensuring data consistency among nodes in a distributed system. By allowing no more than one-third of the network's nodes to be malicious and assuring ongoing functionality even in hostile situations, the PBFT method has emerged as a particularly robust and effective consensus algorithm[8,9].

However, the PBFT uses a three-stage transmission agreement to achieve data consistency. The PBFT's communication complexity is $O(n^2)$ [10], which makes it unsuitable for massive node networks. Improvements to the PBFT algorithm are now well-grounded in research. For example, some nodes are chosen as consensus nodes via Speculative Byzantine Fault Tolerance (SBFT)[11]. The DBFT consensus algorithm is what the NEO project[12] suggests, and it allows nodes to pick some of them as consensus nodes to create new blocks by voting based on how many tokens they own. The disadvantage of the first two ways of improvement mentioned above is that consensus node selection criteria are very straightforward and vulnerable to assault. Tong et al. presented the TrustGPBFT consensus algorithm[13]. To reduce the system's latency and increase scalability, TrustGPBFT combines PBFT with the PeerTrust trust calculation model and selects nodes to engage in the consensus process. But the PBFT's time complexity is still $O(n^2)$.

In this context, the IBFT proposed in this study first figure out the trust value of every node through multi-level indicators and elects' part of nodes to engage in the consensus process. The following BLS aggregation signature procedure is carried out by the node with the highest value of trust, which is chosen to be master node, thereby boosting the system's throughput while lowering the system's latency and communication complexity.

Materials and Methods

In this fundamental study, mixed methods will be used for the methodology approaches, applying both quantitative and qualitative methods. This study will be divided into three phases: (a)construct multi-level indicators to measure the trust value of each node; (b)select the master node among nodes with higher values; (c)modify the consensus protocol of PBFT. The master node converts the information interaction process between nodes into the BLS signature process, reducing the communication complexity between nodes from $O(n^2)$ to $O(n)$.

Phase a

The trust value is a comprehensive quantitative evaluation of the node's integrity level, communication capabilities and node stability. Node trust can be expressed as a discrete or continuous trust interval. For example, {-2, -1, 0, 1, 2} is used to represent the node's complete untrustworthiness, basic untrustworthiness, uncertainty, basic trust, and complete trust. Or using the interval [0, 1] quantify the trust degree of each node. When it is 0, this node is completely untrustworthy. When the trust degree of a node is 1, the node is completely trustworthy. Discrete trust levels cannot quantitatively reflect the changing trend of node trust over time, and the level division is too simple. Therefore, this paper uses the trust interval [0, 1] to represent the trust degree of the consensus node. The trust degree of the node is expressed as T.

The node trust value evaluation index system should be characterized by multi-attribute, omni-directional and multi-granularity. By observing the behavioral attributes of consensus nodes, this paper quantitatively evaluates node trust from the three aspects of node security, stability and availability, and designs three first-level evaluation metrics and 11 second-level evaluation metrics of blockchain node trust. In Table 1, it displays specific description of the indicators.

Table 1. System of consensus node trust value evaluation

Level 1 indicators	Level 2 indicators	Interpretation of indicators
Safety	Data consistency ratio	Percentage of data consistency of data forwarded by nodes compared to data forwarded by other nodes
	Node Connection Ratio	Actual node connections as a percentage of the total desired node connections

	Proportion of identity fraud	Proportion of messages sent under false pretenses to other nodes in relation to overall communications
	Whether to provide invalid blocks	Whether an invalid block was created
Stability	network latency	Delay in establishing links with other nodes
	Node online hours	Runtime of the node in the system
	Node offline hours	The duration that the node cut off communication with the system
Usability	Ratio of available memory	The ratio of available memory to total memory
	CPU utilization	Reflects the current load of the CPU
	Percentage of available storage capacity on disk	The ratio of disk remaining capacity to total capacity

The three first-level indicators are a summary of the second-level indicators and reflect three aspects of node credit evaluation. Security and stability are trust characteristics that describe honest collaboration and non-fraud of nodes. Availability describes the trust characteristics of node capabilities from the aspects of the node's own storage capacity and processing capabilities. The capability trust characteristic is the basis and premise of trust in honest collaboration. The trust degree obtained in this article comprehensively considers the node's honest collaboration trust and capability trust.

There are 11 secondary indicators, which are specific indicator information that reflects the running status of the node. Secondary indicators are divided into discrete indicators and percentage indicators. For example, the values of indicators such as data consistency ratio, node connection ratio, and identity fraud ratio can be directly expressed by percentages. The value of the indicator is in the [0, 1] interval and can be used directly. However, the values of indicators such as network delay, node online time, and node offline time are specific integers, and they need to be quantified uniformly. Different attribute values are assigned different values according to their importance in the consensus process. The attribute value quantification can be seen from Table 2.

Table 2. Discrete data scorecard

Whether to provide invalid blocks	**Network latency**	**Node online hours**	**Node offline hours**
Range / score	**Range / score**	**Range / score**	**Range / score**
Yes / 1	(0, 30) / 12	(72, +oo) / 11	(0, 0.5) / 11
No / 0	(31, 50) / 8	(25, 71) / 8	(0.6, 2) / 8
	(51, 80) / 6	(13, 24) / 4	(3, 24) / 4
	(81, 100) / 4	(0, 12) / 1	(35, +oo) / 1
	(101, +oo) / 2		

After the secondary indicators are quantified according to Table 2, they are scaled proportionally according to the min-max normalization method to map the quantified values to the [0,1] interval. The specific method is as follows: set $v_{min}\ and\ v_{max}$ as the minimum and maximum values of an indicator, respectively, and v as the specific value of the second-level metric, which is the converted value. The data are normalized according to equation: $value = (v - v_{min})/(v_{max} - v_{\min})$.

Take the node online hours as an example, when the node online hours are 50h, it is recorded as 8, the maximum value is 11, and the minimum value is 1. According to the equation, the final value of 0.7 is obtained to represent the node online hours.

Combining qualitative and quantitative methodologies, the analytical hierarchy process is a method for making decisions based on several factors. When applying the analytic hierarchy process method, the problem to be solved should be broken down into relevant elements, and then divided into objectives, criteria, options, etc., and quantified by certain numerical values.

After constructing the node trust evaluation index system, the weights of each index need to be calculated. The discriminant matrix is constructed by comparing the importance of each attribute of the same level about an index attribute in the previous level two by two. To prevent the difficulty in making decisions brought on by too many discriminatory levels, the 9-percentile ratio is adopted to determine the relative advantages and disadvantages of each evaluation index, and the discriminant matrix of the first-level and second-level indexes is constructed sequentially. In this paper, multiple discriminant matrices are given by several experienced experts, and the final discriminant matrix is aggregated by adding expert weights through the geometric mean method to reduce the subjectivity of expert evaluation. The expert weights can be obtained by two-by-two judgment from technical title, practice time, experience, etc. using hierarchical analysis. Fig. 1 shows the comparison matrix constructed for the primary indicator and the three secondary indicators.

$$\begin{bmatrix} 1 & 5 & 9 \\ 1/5 & 1 & 3 \\ 1/9 & 1/3 & 1 \end{bmatrix} \begin{bmatrix} 1 & 3 & 1/3 & 5 \\ 1/3 & 1 & 1/3 & 1 \\ 3 & 3 & 1 & 3 \\ 1/5 & 1 & 1/3 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1/5 & 1/7 \\ 5 & 1 & 1/2 \\ 7 & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1/5 & 1/7 \\ 5 & 1 & 1/2 \\ 7 & 2 & 1 \end{bmatrix}$$

Figure 1. Comparison matrix for primary and secondary indicators

After passing the consistency test, the eigenvectors represent the weights of metrics. As shown in Table 3, the final comprehensive weights of the second-level metrics are obtained by combining the weights of the first-level and second-level metrics.

Table 3. The weights of the node indicators

Level 1 indicators	Weight	Level 2 indicators	Weight
Safety	0.7514	Data consistency ratio	0.2311
		Node Connection Ratio	0.0839
		Proportion of identity fraud	0.3598
		Whether to provide invalid blocks	0.0766
Stability	0.1782	network latency	0.0134
		Node online hours	0.0594
		Node offline hours	0.1054
usability	0.0704	Ratio of available memory	0.0433
		CPU utilization	0.0088
		Percentage of available storage capacity on disk	0.0183

Phase b

After getting the values and weights of the metrics, the trust value of the node is determined. Rank consensus nodes in accord with their trust values. Node are classified according to the rank of the trust values.

From the perspective of the system as a whole, when the client sends a request, if f nodes become invalid nodes due to network interruption, then the client receives only $n - f$ replies in the worst case. However, among the $n-f$ replies, due to the asynchronous reply, the client may receive less than n - f replies. Some nodes are included in the reply because the reply speed is too slow. In the worst-case scenario, each node receives f fewer replies. These replies are due to network delays and not Byzantine errors. Among the $n - f$ replies received by the client, the existence of f Byzantine nodes is the worst-case scene. To identify the reply results of normal nodes, the number of normal nodes must be at least one more than the number of Byzantine nodes, that is, there are $f + 1$ normal nodes, satisfying $n-f \geq f+1+f$, that is, $n \geq 3f+1$. Each consensus process of PBFT takes place in a view, which contains a master node and the rest are replica nodes. The master node acts as the initiator of consensus and runs the consensus protocol with other replica nodes. If the master node in the view fails, the view change process needs to be initiated, and the master node of the next view is responsible for initiating the consensus on the messages that did not complete the consensus in the previous round and continues to run.

In this paper, consensus nodes are categorized into 3 types: honest nodes, normal nodes and malicious nodes. The node with the greatest trust value is the master node.

The top one-third of nodes with greater trust values are honest nodes. This type of node provides reliable communication services with high performance while ensuring the security of transmitted data when communicating with other nodes. In the view change, the honest node is to be preferred to become the master node to forward the transaction requests received from the client for the other nodes. If the honest node performs poorly in the consensus process and the trust value ranking decreases, it may be downgraded to a normal node or even a malicious node.

The bottom one-third nodes with the lower trust values are malicious nodes. This type of node tends to forward wrong messages or does not provide communication services to other nodes during the consensus process. In addition to consuming the bandwidth resources of the system, malicious nodes also interfere with the achievement of consensus. Therefore, malicious nodes need to be isolated from the set of consensus nodes before each consensus. A slow recovery strategy of trust level is adopted for malicious nodes. When the trust level is higher than the $2f+1$th node at the time, it is upgraded to a normal node, which can join the set of consensus nodes again. The recovery formula is: $Trust = T + \Delta t * k/t$. T is current trust value of the node, Δt is the total time elapsed, t is the time required for a consensus to be reached, which can also be referred to as a time frame, and k denotes the rate of trust restoration, which can be determined depending on the specific business.

The remaining nodes are normal nodes. A normal node can be upgraded to become an honest node by performing better in the network, or it can be downgraded to a malicious node due to reduced trust. Normal nodes can participate in the consensus of the network along with honest nodes, but try not to select the normal node as a master node if there are many honest nodes in the consortium chain. Only honest and normal nodes can engage in the consensus process.

Phase c

The problem with the high computational overhead of PBFT is the transmission complexity of the network[14,15], as n nodes need to broadcast to the remaining n-1 nodes[16], the entire transmission complexity is $O(n^2)$. It is a constraint on the efficiency.

Boneh et al[17,18] proposed a BLS signature scheme in 2004 and updated it in 2018. Given a set of signatures, a BLS aggregated signature can be generated and allows verification of the authenticity of the signatures. BLS aggregated signature is composed of 4 parts: initialization, key generation, aggregated signature, and signature verification. In the second phase of the project, each node instead sends the information uniformly to the master node, who verifies it and then sends it to the nodes, thus reducing the complexity. Fig. 2 shows the complete process of applying BLS signatures to the IBFT algorithm.

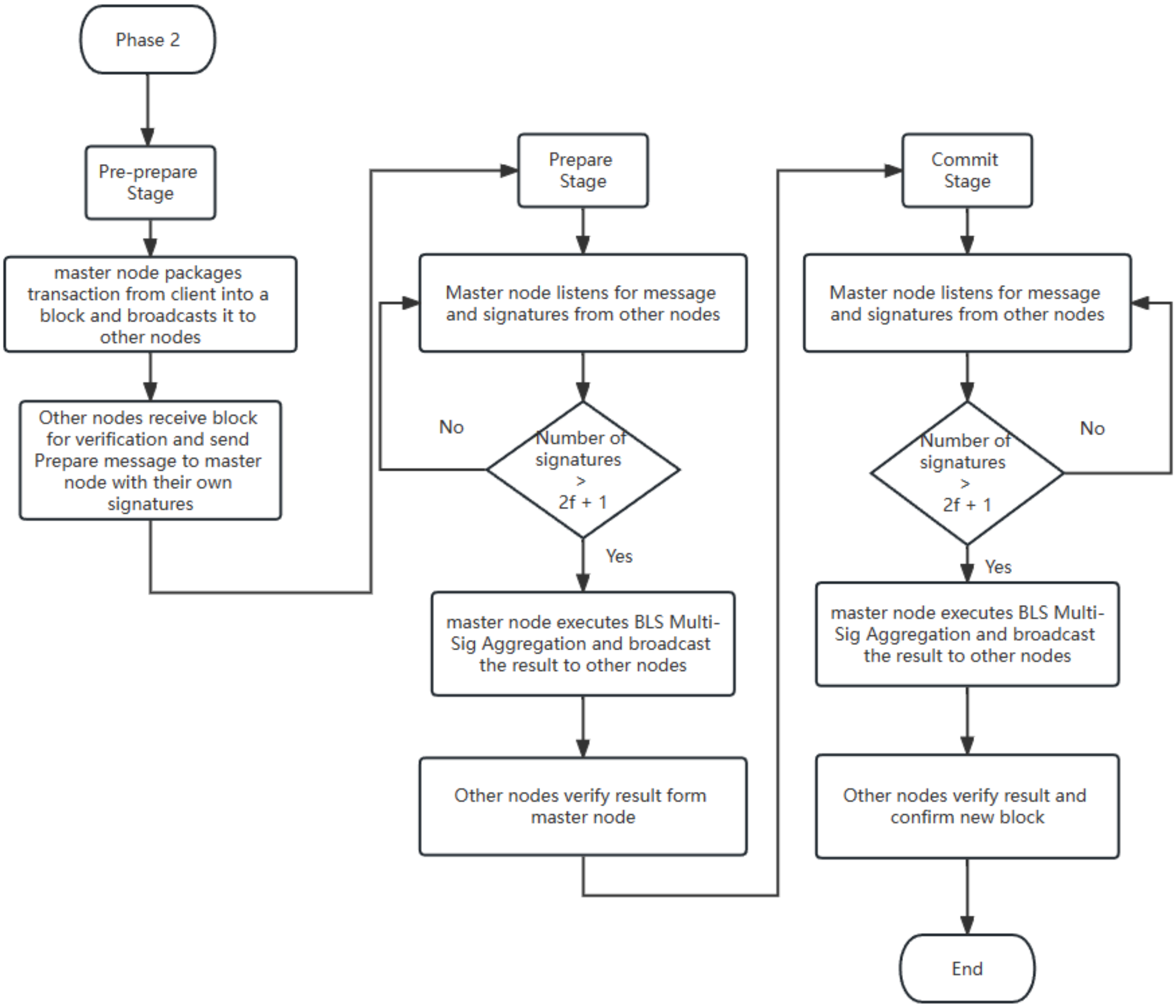
Phase 2
Pre-prepare Stage
master node packages transaction from client into a block and broadcasts it to other nodes
Other nodes receive block for verification and send Prepare message to master node with their own signatures
Prepare Stage
Master node listens for message and signatures from other nodes
Number of signatures > 2f + 1
No
Yes
master node executes BLS Multi-Sig Aggregation and broadcast the result to other nodes
Other nodes verify result form master node
Commit Stage
Master node listens for message and signatures from other nodes
Number of signatures > 2f + 1
No
Yes
master node executes BLS Multi-Sig Aggregation and broadcast the result to other nodes
Other nodes verify result and confirm new block
End

Figure 2. The overall process of applying BLS to PBFT

As shown in Fig. 2, the overall process is still divided into three stages[19]: pre-preparation, preparation, and submission. The detailed steps for each process are listed below.

Pre-prepare Stage: The client transactions are collected by the master node, which then groups them into a new block, then delivers it to other nodes. The message is received by the other nodes' who then confirm the master node's identification. If they verify it, they send a prepare message with their signature.

Prepare Stage: After the master node get prepare information and signatures from more than $2f + 1$ nodes[20], BLS technology combines the signatures into 1 signature and sends the outcome to the remaining nodes. The remaining nodes receive the information, verify it, and then enter the Commit Stage.

Commit Stage: The remaining nodes send commit messages and signatures to the master node, which receives commit messages and signatures from more than $2f + 1$ nodes and then broadcasts the listening result to the remaining nodes. After the remaining nodes receive it, the new block is confirmed.

In summary, the use of BLS Multi-Sig for signature aggregation reduces the number of node communications, decreases the communication complexity of PBFT from $O(n^2)$ to $O(n)$, thus reduces the computational overhead of the PBFT. In the Fig. 3, it shows the node intercommunication process of IBFT.

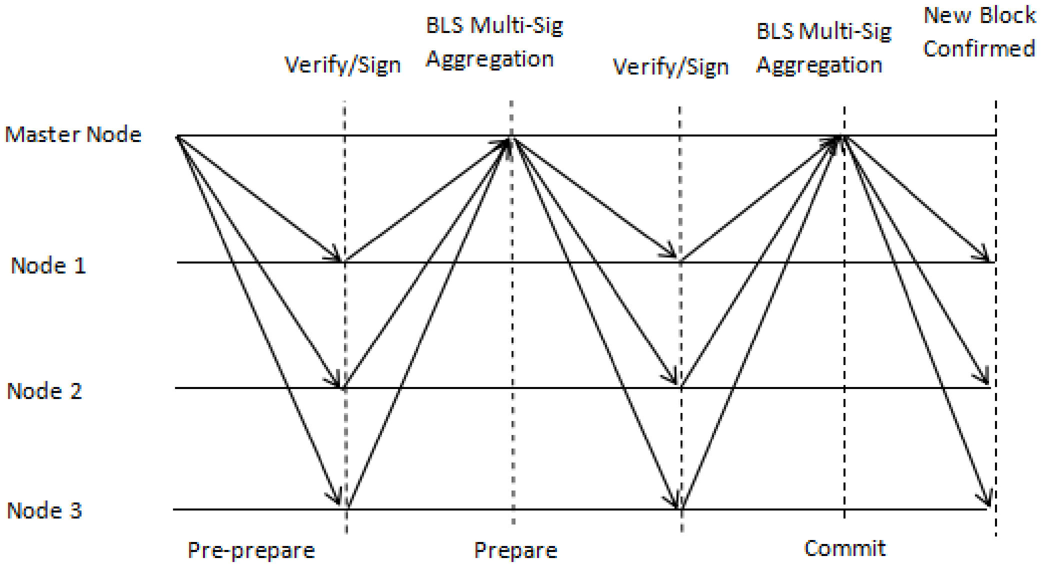

Figure 3. Node intercommunication process of IBFT

Results and Discussion

The IBFT algorithm is tested in terms of consensus throughput and consensus latency. The server runs Windows 10 and has an Intel i7 3GHz processor and 16GB of RAM. The Go programming language is used to create the IBFT and PBFT algorithms. Threads are used to listen to various ports in place of nodes, and several threads are opened to mimic the communication process of consensus nodes. The effectiveness of IBFT algorithm is verified by analyzing the throughput and transaction delay comparison with PBFT algorithm.

The number of legitimate transactions that the blockchain system submits in a certain amount of time is known as transaction throughput, which is one of the main indicators of blockchain performance testing. When the client sends 200 transactions, the experiment adopts the node number n is 4, 7, 10 respectively. The quantity of malicious nodes f is 1, 2, 3 respectively, which satisfies the inequality: $n \geq 3f + 1$. The throughput is determined by the equation: $TPS = Transactions/t$. TPS represents throughput. Transactions are the total number of transactions the system processed in a certain period of time, while the new block is being produced, and t is the generation time of the new block.

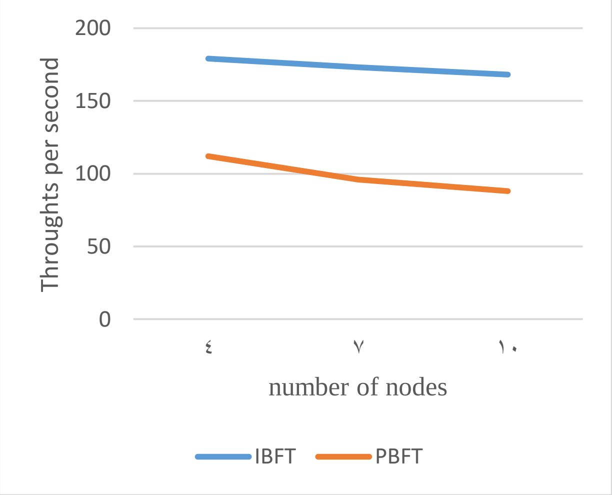

Figure 4. Comparison of transaction throughput of BFT and PBFT consensus algorithms

Fig. 4 shows that compared with PBFT consensus algorithm, IBFT consensus algorithm's data throughput is weaker affected by the increase of node number, and can maintain a more stable data throughput and higher. Under the same number of transaction requests, the enhanced IBFT's throughput is 61% more than the PBFT's. The latency is an important index of the consensus algorithm, the calculation formula is $T_d = T_c - T_p$. T_c is the confirmation time of the block added to the ledger; T_p is the start time of the client to send a transaction request. Both T_c and T_p are recorded by timestamps. In this experiment, T_c and T_p are recorded for 200 times, and T_d is calculated and averaged to get the statistical results. The latency test of PBFT and IBFT is conducted under the number of nodes 4, 7 and 10 respectively, and the latency comparison graph in Fig. 5 is obtained.

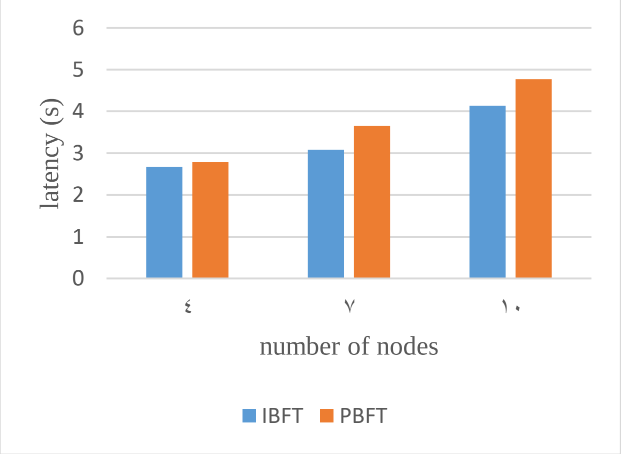

Figure 5. Latency Comparison of IBFT and PBFT Consensus Algorithms

Fig. 5 illustrates how the IBFT and PBFT consensus methods eventually get more and more delayed as the number of nodes rises. IBFT typically executes the same transaction request with a 13% lower latency than PBFT with the same number of nodes. This is because the master node is used as an intermediary for message transmission in IBFT, changing the n-to-n communication method in PBFT into an n-to-1 communication method. Experiments show that the IBFT consensus mechanism can effectively reduce the communication complexity by using this method.

Conclusion

This paper analyzes PBFT and proposes the IBFT to help solve the PBFT's current issues such as simple master node selection rules, high latency and high communication complexity. With the master node acting as the coordinator of information sending and receiving, the time complexity of PBFT can be lowered to $O(n)$. This approach selects the master node according to the node's trust value, which can assure the randomness and security of the master node selection. Combining BLS signatures in the consistency agreement ensures that the master node will not do evil during the agreement process. Multi-node simulation experiments show that the IBFT consensus algorithm improves transaction throughput and reduces latency compared to the PBFT consensus algorithm, and has certain practical value. In order to boost the consensus algorithm's efficiency even more, additional study will be conducted on the issue of high communication complexity brought on by the high complexity of the view conversion protocol in the PBFT consensus algorithm.

Acknowledgment

This research was supported by the Ministry of Higher Education (MOHE) through Fundamental Research Grant Scheme (FRGS/1/2021/ICT10/UTM/02/3). We also want to thank the Government of Malaysia which provides the MyBrain15 program for sponsoring this work under the self-fund research grant and L0022 from the Ministry of Science, Technology and Innovation (MOSTI). This research was supported by a UTM Fundamental Research Grant Q.J130000.3828.23H38.

Authors’ Declaration

- Conflicts of Interest: None.
- We hereby confirm that all the Figures and Tables in the manuscript are ours. Furthermore,

any Figures and images, that are not ours, have been included with the necessary permission for re-publication, which is attached to the manuscript.

- Ethical Clearance: The project was approved by the local ethical committee in University of Universiti Teknologi Malaysia.

Authors' Contribution Statement

D. Z. undertook multiple pivotal roles, including designing the research methods, gathering and analyzing data, interpreting the findings, and writing the first version of the manuscript.

Simultaneously, N. H. A.W. was responsible for the conception of the research at its outset, offering valuable insights into the direction and focus of the study. She also participated in revising and proofreading the manuscript, ensuring the text's quality and accuracy.

A. W. M. Z. was responsible to edit and improve the paper, checking the grammar and proper paper organization.

References

1. Ghani RF, Al-Karkhi AA, Mahdi SM. Proposed Framework for Official Document Sharing and Verification in E-government Environment Based on Blockchain Technology. Baghdad Sci. J. . 2022 Dec 5;19(6 (Suppl.)):1592-. https://doi.org/10.21123/bsj.2022.7513.
2. Dayong Z, Wahab NH, Kadir KA, Aldhaqm A, Nasir HM, Wong KY. Research on Blockchain: Privacy Protection of Cryptography Blockchain-Based Applications. In2023 3rd International Conference on Emerging Smart Technologies and Applications (eSmarTA). 2023 Oct 10; (pp. 1-6). IEEE. https://doi.org/ 10.1109/eSmarTA59349.2023.10293507.
3. Oyinloye DP, Teh JS, Jamil N, Alawida M. Blockchain consensus: An overview of alternative protocols. Symmetry. 2021 Jul 27;13(8):1363. https://doi.org/10.3390/sym13081363.
4. Badr AM, Fourati LC, Ayed S. A Novel System for Confidential Medical Data Storage Using Chaskey Encryption and Blockchain Technology. Baghdad Sci. J. . 2023 Dec 5;20(6 (Suppl.)):2651-. https://doi.org/10.21123/bsj.2023.9203.
5. Kadir KA, Wahab NH, Soh NZ, Teoh XG, Luk SK, Zin AW. OWNTRAD: A Blockchain-Based Decentralized Application for Vintage E-commerce Marketplaces. In2023 IEEE Symposium on Wireless Technology & Applications (ISWTA). 2023 Aug 15; (pp. 12-17). IEEE. https://doi.org/10.1109/ISWTA58588.2023.10250080 .
6. Xiong H, Chen M, Wu C, Zhao Y, Yi W. Research on progress of blockchain consensus algorithm: A review on recent progress of blockchain consensus algorithms. Future Internet. 2022 Jan 30;14(2):47. https://doi.org/10.3390/fi14020047.
7. Azli AM, Wahab NH, Zhang D, Kadir KA, Sunar N. Implementing Blockchain Technology for Accreditation and Degree Verification. InAsia Simulation Conference 2023 Oct 13 (pp. 81-95). Singapore: Springer Nature Singapore. https://doi.org/10.1007/978-981-99-7240-1_7.
8. Xie M, Liu J, Chen S, Lin M. A survey on blockchain consensus mechanism: research overview, current advances and future directions. Int. J. Intell. Comput. Cybern. . 2023 May 15;16(2):314-40. https://doi.org/10.1108/IJICC-05-2022-0126.
9. Li Y, Qiao L, Lv Z. An optimized byzantine fault tolerance algorithm for consortium blockchain. Peer Peer Netw Appl. 2021 Sep;14:2826-39. https://doi.org/10.1007/s12083-021-01103-8.
10. Suliyanti WN, Sari RF. Blockchain-Based Double-Layer Byzantine Fault Tolerance for Scalability Enhancement for Building Information Modeling Information Exchange. BDCC. 2023 May 9;7(2):90. https://doi.org/10.3390/bdcc7020090.
11. Gao N, Huo R, Wang S, Liu J, Huang T, Liu Y. SBFT: A BFT consensus mechanism based on DQN algorithm for industrial Internet of Thing. China Commun. . 2023 Oct;20(10):185-99. https://doi.org/10.23919/JCC.fa.2021-0080.202310.
12. Zhang J, Rong Y, Cao J, Rong C, Bian J, Wu W. DBFT: A Byzantine fault tolerance protocol with graceful performance degradation. IEEE Trans Dependable Secure Comput . . 2021 Jul 8;19(5):3387-400. https://doi.org/10.1109/TDSC.2021.3095544.
13. Tong W, Dong X, Zheng J. Trust-pbft: A peertrust-based practical byzantine consensus algorithm. In2019 International Conference on Networking and Network Applications (NaNA). 2019 Oct 10; (pp.

344-349). IEEE. https://doi.org/10.1109/NaNA.2019.00066.

14. Chen Y, Li M, Zhu X, Fang K, Ren Q, Guo T, Chen X, Li C, Zou Z, Deng Y. An improved algorithm for practical byzantine fault tolerance to large-scale consortium chain. Inf. Process. Manage. . 2022 Mar 1;59(2):102884. https://doi.org/10.1016/j.ipm.2022.102884.
15. Bhardwaj R, Datta D. Consensus algorithm. Decentralised Internet of Things: A Blockchain Perspective. 2020:91-107. https://doi.org/10.1007/978-3-030-38677-1_5.
16. Xu G, Bai H, Xing J, Luo T, Xiong NN, Cheng X, Liu S, Zheng X. SG-PBFT: A secure and highly efficient distributed blockchain PBFT consensus algorithm for intelligent Internet of vehicles. JPDC. 2022 Jun 1;164:1-1. https://doi.org/10.1016/j.jpdc.2022.01.029.
17. Jalil BA, Hasan TM, Mahmood GS, Abed HN. A secure and efficient public auditing system of cloud storage based on BLS signature and automatic blocker protocol. J. King Saud Univ. - Comput. Inf. Sci. . 2022 Jul 1;34(7):4008-21. https://doi.org/10.1016/j.jksuci.2021.04.001.
18. Luo X, Zhou Z, Zhong L, Mao J, Chen C. An effective integrity verification scheme of cloud data based on BLS signature. Secur. Commun. Netw. . 2018 Nov 19;2018:1-1. https://doi.org/10.1155/2018/2615249.
19. Liu J, Li W, Karame GO, Asokan N. Scalable byzantine consensus via hardware-assisted secret sharing. IEEE Trans Comput. 2018 Jul 25;68(1):139-51. https://doi.org/10.1109/TC.2018.2860009.
20. Huang D, Li L, Chen B, Wang B. RBFT: A new Byzantine fault-tolerant consensus mechanism based on Raft cluster. J. Commun. 2021;42(3):209-19. https://doi.org/ 10.11959/j.issn.1000−436x.2021043 .

تحسين إجماع Blockchain: دمج قيمة الثقة في خوارزمية التسامح مع الأخطاء البيزنطية العملية مع التوقيع الإجمالي لـ Boneh-Lynn-Shacham

دايونغ تشانغ[1]، نور حليزا عبد الوهاب*[1]، عدي ويرا محمد زين[2]

[1]كلية الحاسبات، الجامعة التكنولوجية الماليزية، جوهور، ماليزيا.
[2]كلية الأعمال والمالية، جامعة تونكو عبد الرحمن، بيراك، ماليزيا.

الخلاصة

يتم استخدام المكون الأساسي لتقنية Blockchain، وهو خوارزمية الإجماع، لضمان اتساق البيانات بين عقد Blockchain. نظرًا لمقاومتها للأخطاء البيزنطية، تستخدم سلاسل الكونسورتيوم في كثير من الأحيان آلية التوافق العملي للتسامح مع الأخطاء البيزنطية (PBFT). ومع ذلك، لا يزال PBFT الحالي يواجه مشكلات تتعلق بتعقيد اتصالات العقدة العالية والاختيار العشوائي للعقدة الرئيسية. تقترح هذه الدراسة آلية إجماع IBFT، والتي تعتمد على التوقيع الكلي لـ Boneh-Lynn-Shacham (BLS) وقيمة ثقة العقدة.. يتم إجراء مؤشرات متعددة المستويات في IBFT لتحديد قيمة الثقة لكل عقدة. يتم اختيار عدد قليل من العقد الموثوقة جدًا لتكون بمثابة عقد إجماع. يتم تحديد أي عقدة لها أعلى قيمة ثقة لتكون العقدة الرئيسية. بعد ذلك، يتم استخدام التوقيعات المجمعة لـ BLS لتحسين تدفق الإجماع لـ PBFT. ونتيجة لذلك، يتم الحفاظ على أمان المعلومات المرسلة بين العقد وتقليل تعقيد اتصالات العقد. تظهر نتائج تجربة المحاكاة أنه عند مقارنتها بـ PBFT، فإن نهج إجماع IBFT يعمل على تحسين إنتاجية المعاملات بنسبة 61% ويقلل زمن الوصول بنسبة 13%.

الكلمات المفتاحية: Blockchain، بون لين شاشام (BLS)، خوارزمية الإجماع، التسامح مع الأخطاء البيزنطية العملية (PBFT)، قيمة الثقة.