Scientific Research Publishing

Journal of Computer and Communications, 2022, 10, 37-50
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

A Review: Consensus Algorithms on Blockchain

Jannah Yusoff[1], Zarina Mohamad[1], Mohd Anuar[2]

[1]Faculty of Informatics & Computing, University Sultan Zainal Abidin, Terengganu, Malaysia
[2]iExploTech, Cyberview CoPlace 1, Cyberjaya, Malaysia
Email: zarina@unisza.edu.my, sl3751@putra.unisza.edu.my, anuarls@hotmail.com

How to cite this paper: Yusoff, J., Mohamad, Z. and Anuar, M. (2022) A Review: Consensus Algorithms on Blockchain. *Journal of Computer and Communications*, **10**, 37-50.
https://doi.org/10.4236/jcc.2022.109003

Received: May 18, 2022
Accepted: September 10, 2022
Published: September 13, 2022

Abstract

Blockchain is a distributed public ledger that keeps track of all transactions that have ever taken place in the system. As a distributed ledger, a consensus mechanism is required to ensure all the transaction functions properly. In order to reach a consensus, it is critical to emphasize the importance of performance and efficiency. The use of the right consensus algorithm will significantly improve the efficiency of a blockchain application. This paper reviewed several types of consensus algorithms used in blockchain and discusses the idea of a new consensus algorithm that can improve the performance of consortium blockchain.

Keywords

Consensus Algorithms, Consortium Blockchain, Practical Byzantine Fault Tolerance (PBFT), Performance

1. Introduction

Blockchain was first introduced to the world as the underlying technology of the Bitcoin system in 2008 through "Bitcoin: A peer-to-peer electronic cash system" by Satoshi Nakamoto [1]. Before the blockchain was introduced, traditional transactions required a centralized trusted institution. The trusted institution is solely responsible for the confirmation and records of the transactions, which can lead to many problems with transaction cost, security and efficiency. Blockchain is a decentralized distributed ledger that provides the free transfer of end-to-end digital assets [2] without the involvement of a central authority or third party. Decentralization, stability, security and non-tampering are all characteristics of the blockchain that make it a distributed network protocol, enabling a trust relationship between different participants who do not know each other [2]. Hence, blockchain can greatly save costs and improve efficiency.

The structure of the blockchain is illustrated in (**Figure 1**). As new sets of

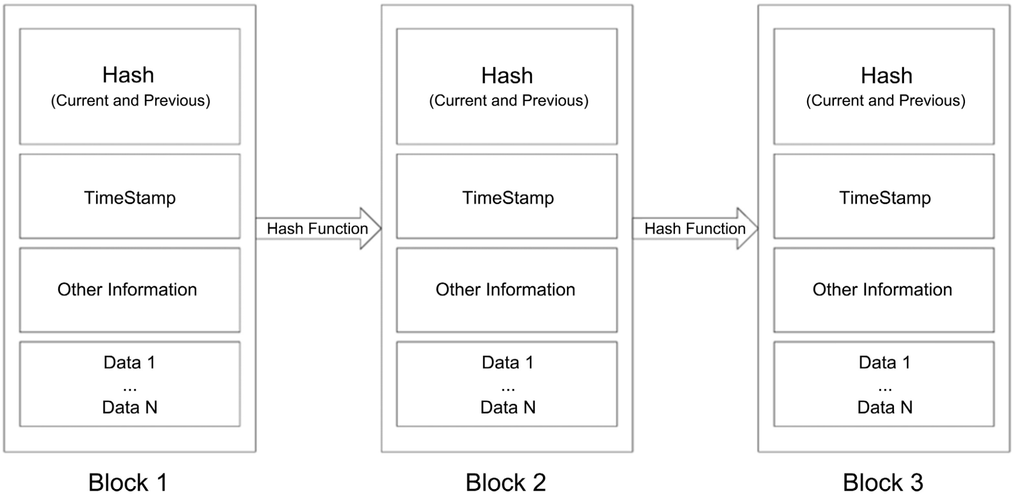

Figure 1. Blockchain structure [4].

"blocks" are added to the ledger, it continues to grow. Each block contains information on several transactions, a timestamp and a link to the preceding block, forming a continuous chain. The continuous chain is protected by hashing the previous block and then embedding the hash value into the current block. This enables a trust chain of block or temper resistant property since the genesis block. The ledger is not administered by a single entity; rather, each user on the network receives a copy of the entire ledger. Old blocks are maintained indefinitely and new blocks are added to the ledger in an irreversible manner, making it practically impossible to tamper with data by fabricating documents, transactions and other data.

A general classification divides blockchain into 3 categories, including public blockchain, private blockchain and consortium blockchain [3]:

1) Public blockchains are a type of decentralized permissionless blockchains in which all network members have access to information and can participate in its acceptance. Bitcoin and Ethereum are examples of public blockchains. This type of blockchain is secure because of its consensus mechanism, which ensures agreement among all peers. These consensus algorithms include Proof of Work (PoW), Proof of Stake (PoS), Delegated Proof of Stake (DPoS), etc.

2) Private blockchains are permissioned blockchains in which information is exclusively available to a certain group and its change acceptance is limited to that group, e.g., a blockchain-based payroll system. This is a centralized blockchain, which means that a central authority decides who can read, write or participate in the blockchain. Hence, a single authority defines the consensus mechanism in private blockchains. These consensus algorithms include Practical Byzantine Fault Tolerance (PBFT), Proof of Authority (PoA), Proof of Elapsed Time (PoET), etc.

3) Consortium blockchains are also known as federated blockchains in which multiple organizations manage the platforms, rather than just one. It is between those of public and private blockchains, combining elements from both. Multiple organizations can make decisions compared to private blockchains which are decisions made by central authority only. These consensus algorithms include Practical Byzantine Fault Tolerance (PBFT), Proof of Vote (PoV), Proof of Trust (PoT), etc.

These 3 types of blockchain differ in terms of how nodes come to an agreement and the consensus algorithm used. The selection of the appropriate consensus algorithm is very important in determining the performance of these types of blockchains. Thus, this paper focuses on studying and evaluating the existing common consensus algorithms used in blockchain by providing the flow, advantages and disadvantages of those consensus algorithms and discussing the idea of the new consensus algorithm for the consortium blockchain.

The rest of the paper is organized as follows. This paper introduces the existing common consensus algorithms, how their work, pros and cons in chapter 2. This paper also proposes a new consensus that can be used in consortium blockchain in chapter 3. Finally, this paper makes a conclusion in chapter 4.

2. Consensus Algorithms

Consensus algorithms originated from the famous Byzantine general problems [2], which was first presented in the paper "The Byzantine generals problem" by Lamport in 1982 [5]. The Byzantine general problem can be described as follows. Byzantine is the capital of the ancient eastern Roman Empire. To resist foreign enemies, a general and his troops are each stationed on several fiefs (estates of land) in Byzantine. Each general can give 2 orders: whether to attack or retreat when facing enemies. A war can be won with the fewest possible casualties only when all honest generals agree on an attack or withdrawal order. However, Byzantine is so large that these generals are unable to discuss the order together because they must guard their own fiefs. Therefore, the general's commands are sent by messengers. The generals make their last decisions on whether to attack or retreat by giving orders to the other generals and collecting orders from them. In this case, there are 2 possibilities. Either some of these generals or the messengers are traitors. If the generals are traitors, they may send the wrong orders or different orders to different generals. If the messengers are traitors, they could intentionally sabotage the mission by delivering the wrong information. As a result, this would ultimately undermine the overall decision of the honest generals. It is concluded that the Byzantine generals' problem can be defined as the problem of getting honest generals to reach a consensus in presence of several traitors.

In the early days of blockchain systems, blockchain consensus algorithms such as Proof of Work (PoW), Proof of Stake (PoS) and Practical Byzantine Fault Tolerance (PBFT) were used. Recently, a huge number of new blockchain consensus algorithms have come out. According to some surveys [2] such as [6] [7] [8] [9], these new consensus algorithms can be broken down into 3 categories:

1) Variants of the original consensus algorithms, e.g., Bitcoin-NG [10] and

Algorand [11], which are the improvements of PoW and PBFT, respectively;

2) Combinations of the original consensus algorithms, e.g., Delegated BFT (DBFT), which is the combination of PoS and PBFT, and e.g., Proof of Activity (PoA) [12], which is the combination of PoW and PoS;

3) DAG-based consensus algorithms, e.g., ByteBall [13] and Hashgraph [14].

The consensus mechanism is the blockchain's cornerstone and a key assurance of the blockchain system's security. It is used to deal with the problem of ensuring data consistency in a distributed system with the presence of several failure nodes. Crash fault nodes and Byzantine fault nodes are 2 types of failure nodes. Crash fault nodes fail only by halting; that is, they can only stop working and have no other malicious behaviors present [15]. Messages can only be delayed or lost in this situation. Byzantine fault nodes, on the other hand, behave abnormally. They can send incorrect messages to other nodes or send different messages to different nodes to sabotage the consensus process.

In the field of traditional distributed systems, consensus algorithms have been explored for many years. How these consensus algorithms work, and their advantages and disadvantages are explained in the next section.

2.1. Proof of Work (PoW)

Proof of Work is the first Blockchain algorithm introduced in the blockchain network [16]. Cynthia Dwork and Moni Naor [17] proposed the idea of Proof of Work (PoW) and later Satoshi Nakamoto applied it in the Bitcoin paper in 2008. A PoW algorithm works by requiring nodes on the network to solve a mathematical problem in order to create the next block and verify the legitimacy of transactions on the network. PoW algorithm flow is illustrated in (**Figure 2**). This mathematical solving is done through the Hash function. Hash is a random and complex mathematical formula that is used for confirmation of the transactions stored in blocks [18]. All the nodes compete to be the first to find a solution via brute force, which requires a huge number of attempts. Whoever is the first to find the solution, can have the right to create a new block and once it is verified, the block will be added to the platform. These nodes that participate in the computation are called miners and the process of solving the problem is called mining [19].

The advantage of the Proof of Work algorithm is its high security and a significant degree of decentralization. However, its main disadvantage is greater energy and resource consumption. Miners need a lot of processing power to figure out the solution to the difficult mathematical problem in terms of hashing billion nonces or more. It leads to a waste of precious resources (money, energy, space, hardware). Moreover, it is time-consuming. Miners must examine a large number of nonce values to find the right solution to the problem that must be solved to mine the block. Furthermore, to solve this problem, it will take some time due to the complexity of solving the hash function. Therefore, this algorithm is not suitable for a big and fast-growing network that requires huge numbers

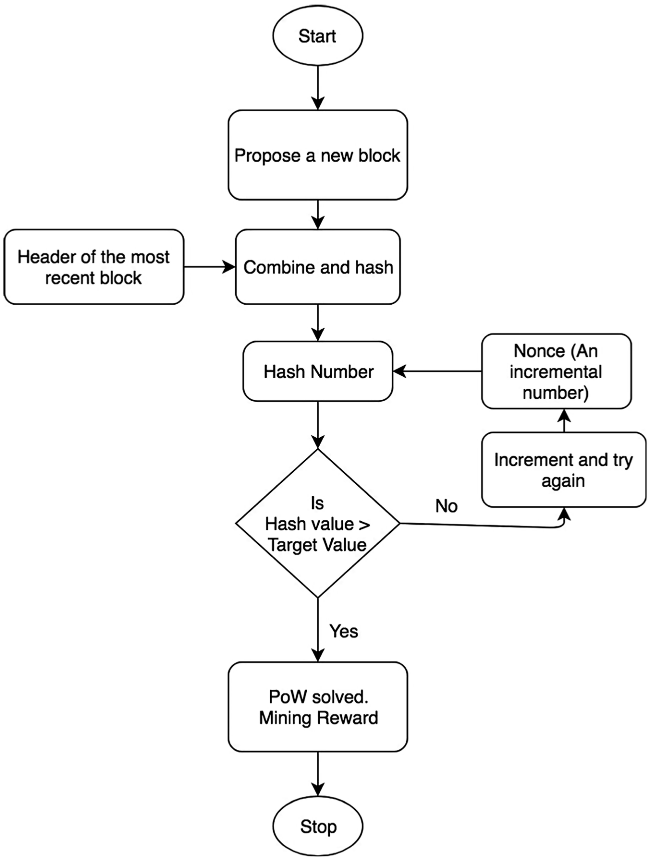
Start
Propose a new block
Header of the most recent block
Combine and hash
Hash Number
Nonce (An incremental number)
Increment and try again
Is Hash value > Target Value
No
Yes
PoW solved. Mining Reward
Stop

Figure 2. PoW algorithm flow [22].

of transactions per second [20]. Tschorsch *et al.* [21] state that low throughput, high latency and inefficiency are the drawbacks of the PoW algorithm.

2.2. Proof of Stake (PoS)

To solve the large computing power consumption problem of PoW, researchers proposed the Proof of Stake (PoS) consensus algorithm. The process of PoS is different compared to PoW because the users of the PoS protocol do not require to solve the mathematical problem to achieve consensus. Users, on the other hand, only need to use cryptocurrency as a stake to achieve consensus. There are 2 ways to participate in staking. PoS algorithm flow is illustrated in (**Figure 3**). First, the user can loan their coins to other users that will participate in the pool and then share the profit with them. However, the user will need to find a reliable person to stake with. Another method is to join the pool. Everyone that participates in that specific pool will divide the profit based on the stake amount. The creator of a new block is chosen from a pool of users that have staked a certain amount of cryptocurrency and no users can predict its turn in advance. Islahuddin *et al.* [23] state that the amount of stake a person has in the system determines the mining. If a miner has more stakes in the blockchain, the chances of mining are more. For instance, if the stake in the given cryptocurrency is at 1%, the users can mint up to 1% of the transactions [24].

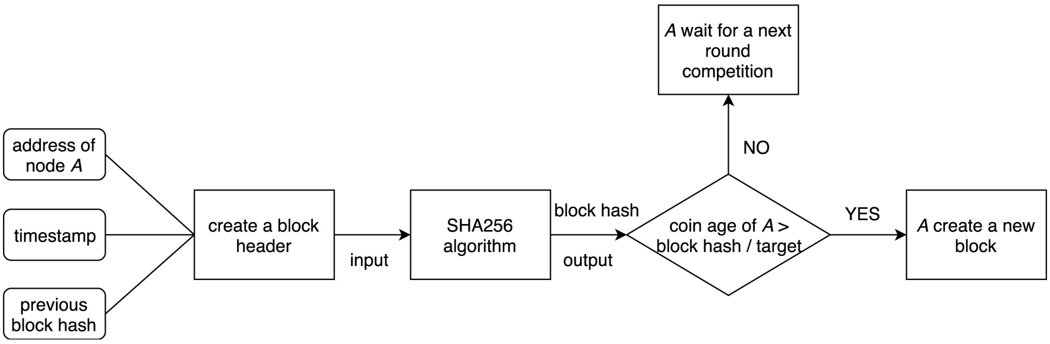

Figure 3. PoS algorithm flow [26].

Du *et al.* [25] addressed that PoS encourages the coins holders to increase the holding time. The blockchain is no longer fully dependent on Proof of Work thanks to the concept of coinage. That effectively solves the resource-wasting problem in PoW. With the rising value in the blockchain, the security of the blockchain utilizing PoS improves. The attackers need to accumulate a large number of coins and keep them for a long time to attack the blockchain. This also greatly increases the difficulty of attack. Xiang *et al.* [2] said that, although this method reduces the waste of computing power, it may have the risk of monopoly, which leads to the centralization trend of the system while allowing malicious attackers to have a clear target to attack, risking the security.

2.3. Delegated Proof of Stake (DPoS)

This algorithm was introduced by Daniel Larimer [27]. It is proposed to improve PoS security by relying on stakeholders voted to pick block producers or witnesses. According to Qianwen Wang *et al.* [28], DPoS similar to the board vote, allows the holder to cast a certain number of nodes and proxy them for verification and accounting. In the blockchain with DPoS, each node can choose the witnesses based on its stake. DPoS algorithm flow is illustrated in (**Figure 4**). The top N witnesses who participated in the campaign and received the most votes have the accounting right across the entire network. The number of N of witnesses is defined such that at least 50% of voting stakeholders believe there is sufficient decentralization. The elected witnesses are rewarded for creating new blocks one by one as directed. The witnesses need to ensure adequate online time. If a witness is unable to create the block for which they were assigned, the activity of the block will be transferred to the next block and stakeholders will vote for a new witness to take its place. DPoS makes the most use of the stakeholders' votes to reach a consensus fairly and democratically [26].

The DPoS consensus mechanism is simple and efficient since it does not require mining or complete node verification. Instead, it is validated by a limited number of witness nodes. It is also power-saving compared to PoW and PoS. However, this limitation on the number of witness nodes would make the network more centralized [18]. Moreover, due to the mechanism that each witness node takes turns generating blocks, the identity of the witness is already known

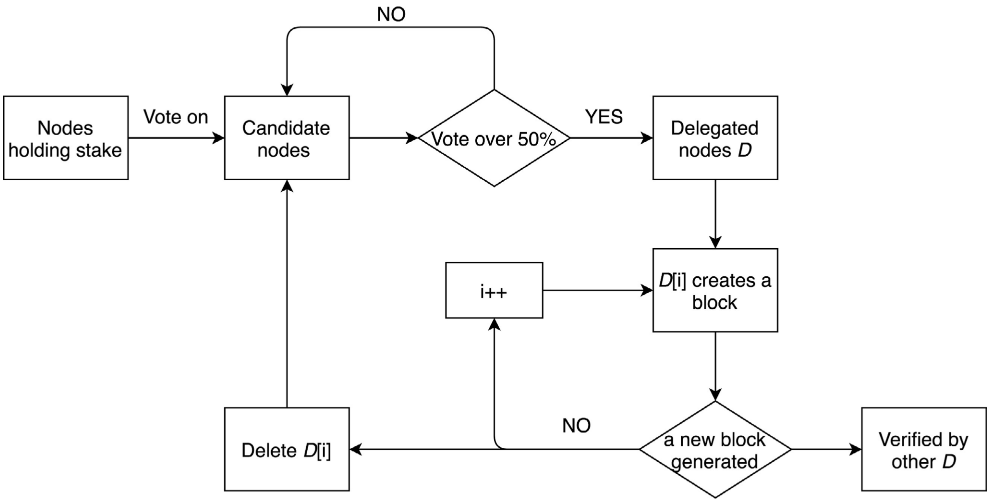

Figure 4. DPoS algorithm flow [26].

and always constant, which would make the blockchain system more vulnerable to collusion attacks [29].

2.4. Practical Byzantine Fault Tolerance (PBFT)

PBFT was proposed by Castro and Liskov in a paper published in 1999 to solve the problem of General Byzantine originally [5]. The focus of the PBFT algorithm is to provide practical Byzantine state machine replication for tolerating the Byzantine fault [20] and making the algorithm workable in practical system applications [30]. The maximum number of malicious nodes, f cannot exceed 1/3 of the total number of nodes, n hence, $n > 3f + 1$ is required by PBFT. The process of PBFT algorithm is divided into 3 phases: pre-prepare, prepare and commit [31]. PBFT algorithm flow is illustrated in (**Figure 5**). The pre-prepare and prepare phase are used to completely organized requests sent in the same view even when the primary, which proposes the ordering of requests, is faulty. The prepare and commit phase are used to verify that requests are completely sorted across all views before they are committed. In each phase, a node advances to the next phase if it receives votes from more than two-thirds of all nodes. Under the premise of ensuring activity and safety, the PBFT algorithm provides a fault tolerance of $(n - 1)/3$.

In [32], the authors said, energy efficiency and high throughput are considered as its advantages and some points such as few or no parameter available for being scalable and possible delays as the network should wait for all nodes' votes are noted as its disadvantages.

2.5. Proof of Luck (PoL)

The goal of the development of this protocol is to overcome the issues available in the previously available consensus algorithms such as slowness, energy usage and time consumption. This algorithm uses the trusted execution environment (TEE) platform [33] to generate a trusted random number for choosing a leader.

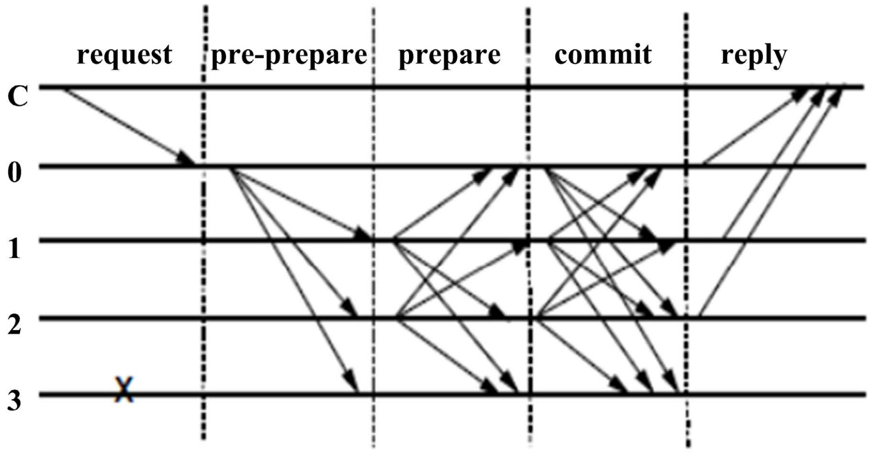

Figure 5. PBFT algorithm flow [25].

It is secure due to the TEE, which prevents the attacker from manipulating the blockchain without controlling a majority of CPUs and disrupting the TEE platform. PoL algorithm consists of 2 functions: PoLRound and PoLMine. At the start of each round, the participant calls the PoLRound function and passes the most recent block to a particular chain. When the ROUND_TIME expires, the participant calls the PoLMine function to create a new block and will be connected with the previous block by passing the new block's header. The PoLMine function generates a random value of [0, 1] from the uniform distribution, that is used to identify the winning block among all mining blocks of the participants in this round.

This algorithm has high immunity against the double-spending attack because the attacker must be very lucky to perform such an attack [34]. However, the main problem of this protocol is its reliance on Intel, which conflicts with the blockchain principle of decentralization. Other than that, another problem with this algorithm is that if the node's clock is not synchronized with the network clock, it may lose its chance of being lucky.

2.6. Proof of Burn (PoB)

The concept of Proof of Burn (PoB) idealized by Iain Stewart [35] is an alternative consensus algorithm that attempts to address the PoW system's high energy consumption issue. It is often called a PoW system without energy waste. In this algorithm, miners burn coins by sending them to a non-spending address. Burning here means that a user is required to send some cryptocurrency to an "eater address" to receive coins, tokens or mining privileges on the network. The Eater Addresses are unable to use these coins for any purpose. The burned coins are recorded in a ledger, making them truly unspendable. It is an expensive activity for the user but this activity consumes no resources and energy. An Eater Address is randomly generated and does not have any private key. As a result, no user will ever be able to spend the coins stored on these addresses.

This algorithm is creating more stability as the user who would risk a shortterm loss and invests in this way will stay in the network for a longer period to gain profits. Furthermore, this enhances decentralization and results in a better-distributed network. However, PoB algorithm is not proven to work on larger scales. It will need to be tested further to ensure its effectiveness and security.

2.7. Proof of Authority (PoA)

Wood *et al.* proposed the Proof of Authority (PoA) algorithm [36] which is essentially an optimized equity proof model. The difference between PoS and PoA consensus is that the latter, leverage user's identity instead of user's digital assets. This means that, it is based on the reputation of trusted parties in a blockchain network [37]. The network relies on a number of pre-approved validators known as "authorities" to validate transactions and create new blocks. As the PoA algorithm stakes identity, users who want to be "authorities" must willingly disclose their identities. To be regarded as trustworthy, validators must follow a set of norms. One of these requires them to register with the same identity they use on the platform in the public notary database. More rules must be followed in order for the network to function. Becoming a validator is not easy. Candidates must go through a screening process in which they must show their long-term commitment to the network. They should also be willing to invest money and risk their reputation during the selection. At last, the method for selecting authorities should follow established guidelines to ensure that all candidates have an equal opportunity of achieving the coveted post. Validators receive power and rewards in exchange for identifying themselves and proving who they are with government-issued documents.

PoA algorithm does not require high-performance hardware because the nodes do not need to use computational resources to solve complex mathematical problems compared to the PoW algorithm. Moreover, this algorithm allows authorities to verify transactions more quickly. The blockchain registers a higher transaction rate than PoW and PoS because blocks are generated in a sequence by authorized network nodes at a predetermined time interval. However, the identities of PoA validators are visible to anyone. This could potentially lead to third-party manipulation.

2.8. Use Cases for Public, Private and Consortium Blockchain

In a permissionless or public blockchain, anyone can access and create data [38]. Smart contracts and nodes can be published and run by anybody. They do not require approval to join the blockchain and communicate with other users. The consensus algorithms in this blockchain are often used in cryptocurrency and document validation.

In a private blockchain, a single organization will have the authority over the network. It can define as a blockchain that operated in a restricted environment, such as a closed network. The consensus algorithms in this blockchain are often used in asset ownership and supply chains [39].

In a permissioned or consortium blockchain, not everyone can join the blockchain. To join the network, each member will need specific permission from the network administrator or owner. This is important for businesses, banks and other institutions that wish to follow the rules and have complete control over their data. All of the participating nodes in a consortium blockchain are known and chosen. The consensus algorithms in this blockchain are often used in digital assets-backed platforms, trade, finance and supply chain industries [40].

3. Discussion

Despite the fact that there are many consensus algorithms introduced and used in the blockchain, the existing consensus algorithms are mostly concerned with the public blockchain while the consortium blockchain is given the least attention [41]. This can be supported by [42] and [43] that said, at present, most of the researches on consensus algorithms are focused on the public blockchain and there are few consensus algorithms that have been developed for consortium blockchain.

Consortium blockchain is a permissioned blockchain, in which the primary nodes are pre-specified by the participants which are composed of many parties. Thus, whoever wants to access to the ledger needs to be a member of any organization. Consortium blockchain also consists of known and trusted users. So, this paper suggested an optimized PBFT algorithm to meet the needs of the consortium blockchain. Generally, PBFT algorithm is widely used in a consortium blockchain because PBFT greatly improves the performance of the blockchain consensus. PBFT algorithm is suitable for a high-performance network with a small number of nodes [26]. However, when there are a large number of nodes involved, the communication overhead occurs. The number of messages exchanged and processed in the network increase dramatically as the number of nodes increase. Therefore, to overcome the shortcomings of PBFT, the clustering method will be added in the phase of PBFT algorithm to effectively reduce the communication overhead in PBFT consensus algorithm.

In the current algorithm which is PBFT algorithm, all nodes need to communicate with each other to achieve a consensus. This leads to high communication with a larger number of nodes which resulted in higher communication overhead. Therefore, by implementing the clustering method by grouping the nodes sequentially, the nodes only need to communicate with their group member instead of communicating will all the nodes in the network. The nodes will be grouped into a few groups, sequentially, which is the first 3 nodes will be grouped in group 1 and followed by group 2, group 3 and so on. The nodes will communicate and achieve the consensus in their group only and the result for the communication overhead will be reduced. The concept to achieve the majority node, which is two-thirds in this proposed algorithm is still the same as with PBFT algorithm.

4. Summary

This paper has summarized existing common consensus algorithms used in the blockchain. By describing how the algorithms work, the advantages and disadvantages of the 7 consensus algorithms of PoW, PoS, DPoS, PBFT, PoL, PoB and PoA are expounded. We also suggested the enhanced consensus algorithm based on PBFT for consortium blockchain. This proposed algorithm can reduce communication overhead and increase the performance of the consortium blockchain. However, the cons of this proposed algorithm are still available but we still can try to improve it because the consensus algorithm that is specially designed for each scenario is still very rare. What can be done to improve the performance of the blockchain in other scenarios? We still need to do further research work to comprehend the implication of the existing algorithms and proposed upgraded solutions.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Nakamoto, S. (2008) Bitcoin: A Peer-to-Peer Electronic Cash System. DecentralizedBusiness Review, 21260.

[2] Fu, X., Wang, H. and Shi, P. (2021) A Survey of Blockchain Consensus Algorithms: Mechanism, Design and Applications. *Science China Information Sciences*, **64**, Article ID: 121101. https://doi.org/10.1007/s11432-019-2790-1

[3] Korpela, K., Hallikas, J. and Dahlberg, T. (2017) Digital Supply Chain Transformation toward Blockchain Integration. *Hawaii International Conference on System Sciences*, Hawaii, 4-7 January 2017, 4182-4191. https://doi.org/10.24251/HICSS.2017.506

[4] Prashanth Joshi, A., Han, M. and Wang, Y. (2018) A Survey on Security and Privacy Issues of Blockchain Technology. *Mathematical Foundations of Computing*, **1**, 121-147. https://doi.org/10.3934/mfc.2018007

[5] Lamport, L., Shostak, R. and Pease, M. (2008) The Byzantine Generals Problem. *Dr. Dobb's Journal*, **33**, 30-36.

[6] Yuan, Y. and Wang, F.Y. (2018) Blockchain and Cryptocurrencies: Model, Techniques, and Applications. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, **48**, 1421-1428. https://doi.org/10.1109/TSMC.2018.2854904

[7] Bano, S., Sonnino, Al., Al-Bassam, M., Azouvi, S., McCorry, P., Meiklejohn, S. and Danezis, G. (2017) Consensus in the Age of Blockchains. http://arxiv.org/abs/1711.03936

[8] Bach, M., Mihaljevic, L.M. and Zagar, B. (2018) Comparative Analysis of Blockchain Consensus Algorithms. 2018 41*st International Convention on Information and Communication Technology*, *Electronics and Microelectronics*, Opatija, 21-25 May 2018, 1545-1550. https://doi.org/10.23919/MIPRO.2018.8400278

[9] Wang, W., Hoang, D.T., Hu, P., Xiong, Z., Niyato, D., Wang, P., *et al.* (2019) A Survey on Consensus Mechanisms and Mining Strategy Management in Blockchain Networks. *IEEE Access*, **7**, 22328-22370. https://doi.org/10.1109/ACCESS.2019.2896108

[10] Eyal, I., Gencer, A.E., Sirer, E.G. and Van Renesse, R. (2016) Bitcoin-NG: A Scalable

Blockchain Protocol. 13*th USENIX Symposium on Networked Systems Design and Implementation*, Santa Clara, 16-18 March 2016, 45-59.

[11] Gilad, Y., Hemo, R., Micali, S., Vlachos, G. and Zeldovich, N. (2017) Algorand: Scaling Byzantine Agreements for Cryptocurrencies. *Proceedings of the* 26*th Symposium on Operating Systems Principles*, Shanghai, 28-31 October 2017, 51-68. https://doi.org/10.1145/3132747.3132757

[12] Zhang, Z.W. (2019) A Byzantine Fault-Tolerant Algorithm for Blockchains.

[13] Churyumov, A. (2018) Byteball. White Paper. 1-49. https://byteball.org/

[14] Baird, L. (2016) The Swirlds Hashgraph Consensus Algorithm: Fair, Fast, Byzantine Fault Tolerance. 1-28. http://www.swirlds.com/downloads/SWIRLDS-TR-2016-01.pdf

[15] Perry, K.J. and Toueg, S. (1986) Distributed Agreement in the Presence of Processor and Communication Faults. *IEEE Transactions on Software Engineering*, **SE-12**, 477-482. https://doi.org/10.1109/TSE.1986.6312888

[16] Anwar, H. (2018) Consensus Algorithms: The Root of Blockchain Technology. https://101blockchains.com/consensus-algorithms-blockchain/

[17] Hooda, P. (2022) Proof of Work (PoW) Consensus. https://www.geeksforgeeks.org/proof-of-work-pow-consensus/

[18] Salimitari, M. and Chatterjee, M. (2018) A Survey on Consensus Protocols in Blockchain for IoT Networks. http://arxiv.org/abs/1809.05613

[19] 101 Blockchains (2019) Know Everything About Blockchain Proof of Work (PoW). https://101blockchains.com/blockchain-proof-of-work/

[20] Alsunaidi, S.J. and Alhaidari, F.A. (2019) A Survey of Consensus Algorithms for Blockchain Technology. 2019 *International Conference on Computer and Information Sciences*, Sakaka, 3-4 April 2019, 1-6. https://doi.org/10.1109/ICCISci.2019.8716424

[21] Tschorsch, F. and Scheuermann, B. (2016) Bitcoin and Beyond: A Technical Survey on Decentralized Digital Currencies. *IEEE Communications Surveys & Tutorials*, **18**, 2084-2123. https://doi.org/10.1109/COMST.2016.2535718

[22] Ghimire, S. and Selvaraj, H. (2019) A Survey on Bitcoin Cryptocurrency and Its Mining. 2018 26*th International Conference on Systems Engineering*, Sydney, 18-20 December 2018, 1-6. https://doi.org/10.1109/ICSENG.2018.8638208

[23] Eigelshoven, F., Ullrich, A. and Bender, B. (2020) Public Blockchain—A Systematic Literature Review on the Sustainability of Consensus Algorithms. 2020 28*th European Conference on Information Systems*, Marrakesh, 15-17 June 2020, 1-17.

[24] Jain, A., Arora, S., Shukla, Y., Patil, T.B. and Sawant-patil, S.T. (2018) Proof of Stake with Casper the Friendly Finality Gadget Protocol for Fair Validation Consensus in Ethereum. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, **3**, 291-298.

[25] Du, M., Ma, X., Zhang, Z., Wang, X. and Chen, Q. (2017) A Review on Consensus Algorithm of Blockchain. 2017 *IEEE International Conference on Systems, Man, and Cybernetics*, Banff, 5-8 October 2017, 2567-2572. https://doi.org/10.1109/SMC.2017.8123011

[26] Zhang, S. and Lee, J.-H. (2020) Analysis of the Main Consensus Protocols of Blockchain. *ICT Express*, **6**, 93-97. https://doi.org/10.1016/j.icte.2019.08.001

[27] Ciberexplosion (2014) Delegated Proof-of-Stake (DPOS). https://www.geeksforgeeks.org/delegated-proof-of-stake/

[28] Wang, Q., Huang, J., Wang, S., Chen, Y., Zhang, P. and He, L. (2020) A Compara-

tive Study of Blockchain Consensus Algorithms. *Journal of Physics*: *Conference Series*, **1437**, Article ID: 012007. https://doi.org/10.1088/1742-6596/1437/1/012007

[29] Yang, F., Zhou, W., Wu, Q., Long, R., Xiong, N.N. and Zhou, M. (2019) Delegated Proof of Stake with Downgrade: A Secure and Efficient Blockchain Consensus Algorithm with Downgrade Mechanism. *IEEE Access*, **7**, 118541-118555. https://doi.org/10.1109/ACCESS.2019.2935149

[30] Zhang, C., Wang, R., Tsai, W.-T., He, J., Liu, C. and Li, Q. (2019) Actor-Based Model for Concurrent Byzantine Fault-Tolerant Algorithm. In: 2019 *International Conference on Computer*, *Network*, *Communication and Information Systems* (*CNCI* 2019), Atlantis Press, 552-558. https://doi.org/10.2991/cnci-19.2019.77

[31] Castro, M. (2001) Practical Byzantine Fault Tolerance. *Third Symposium on Operating Systems Design and Implementation*, New Orleans, 22-25 February 1999, 1-14. http://pmg.csail.mit.edu/papers/osdi99.pdf

[32] Bamakan, S.M.H., Motavali, A. and Babaei Bondarti, A. (2020) A Survey of Blockchain Consensus Algorithms Performance Evaluation Criteria. *Expert Systems with Applications*, **154**, Article ID: 113385. https://doi.org/10.1016/j.eswa.2020.113385

[33] Sabt, M., Achemlal, M. and Bouabdallah, A. (2015) Trusted Execution Environment: What It Is, and What It Is Not. 2015 *IEEE Trustcom*/*BigDataSE*/*ISPA*, Vol. 1, Helsinki, 20-22 August 2015, 57-64. https://doi.org/10.1109/Trustcom.2015.357

[34] Milutinovic, M., He, W., Wu, H. and Kanwal, M. (2016) Proof of Luck: An Efficient Blockchain Consensus Protocol. *SysTEX* 2016: 1*st Workshop on System Software for Trusted Execution*, Trento, December 2016, Article No. 2. https://doi.org/10.1145/3007788.3007790

[35] Frakenfield, J. (2021) Proof of Burn (Cryptocurrency). https://www.investopedia.com/terms/p/proof-burn-cryptocurrency.asp

[36] Wood, G. (2017) Polkadot: Vision for a Heterogeneous Multi-Chain Framework. White Paper. 1-21. https://polkadot.network/PolkaDotPaper.pdf

[37] Chawla, V. (2020) What Are The Top Blockchain Consensus Algorithms? https://analyticsindiamag.com/blockchain-consensus-algorithms/

[38] Rahul, A.R. (2020) How to Choose the Right Consensus Protocol for Permissioned Blockchain Networks. https://blog.accubits.com/consensus-protocol-for-permissioned-blockchain-networks/

[39] Parizo, C. (2021) What Are the 4 Different Types of Blockchain Technology? https://www.techtarget.com/searchcio/feature/What-are-the-4-different-types-of-blockchain-technology

[40] Liu, Z. (2021) Literature Review of Supply Chain Finance Based on Blockchain Perspective. *Open Journal of Business and Management*, **9**, 419-429. https://doi.org/10.4236/ojbm.2021.91022

[41] Khan, F.A., Abubakar, A., Mahmoud, M., Al-Khasawneh, M.A. and Alarood, A.A. (2019) Rift: A High-Performance Consensus Algorithm for Consortium Blockchain. *International Journal of Recent Technology and Engineering*, **7**, 989-997.

[42] Li, K., Li, H., Wang, H., An, H., Lu, P., Yi, P. and Zhu, F. (2020) PoV: An Efficient Voting-Based Consensus Algorithm for Consortium Blockchains. *Frontiers in Blockchain*, **3**, Article No. 11. https://doi.org/10.3389/fbloc.2020.00011

[43] Li, Y., Qiao, L. and Lv, Z. (2021) An Optimized Byzantine Fault Tolerance Algorithm for Consortium Blockchain. *Peer-to-Peer Networking and Applications*, **14**, 2826-2839. https://doi.org/10.1007/s12083-021-01103-8

Abbreviations

PBFT	Practical Byzantine Fault Tolerance
PoW	Proof of Work
PoS	Proof of Stake
DPoS	Delegated Proof of Stake
PoA	Proof of Authority
PoET	Proof of Elapsed Time
PoV	Proof of Vote
PoT	Proof of Trust
DBFT	Delegated Byzantine Fault Tolerance
PoA	Proof of Activity
PoL	Proof of Luck
PoB	Proof of Burn