Journal of Information Systems Engineering and Management
2025, 10(14s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

A Novel Improved Gossip Algorithm for Blockchain Network

*1Ravindra Janardan Lawande, 2Sudhir Bapurao Lande

*1*Department of Electronics and Telecommunication Engineering, College of Engineering Malegaon (Bk), Savitribai Phule Pune University, Malegaon Budruk, Maharashtra, India*

2*Professor, Department of Electronics and Telecommunication Engineering, Vidya Pratisthan's Kamalnayan Bajaj Institute of Engineering and Technology,*
Baramati, Maharashtra, India
**rjlawande25@gmail.com*

ARTICLE INFO

Received: 18 Nov 2024

Revised: 05 Jan 2025

Accepted: 18 Jan 2025

ABSTRACT

In blockchain (BC) systems, increasing throughput is a key concern. A number of enhancements and a novel gossip algorithm are proposed to address this problem, but they are inherently constrained by the underlying peer-to-peer (P2P) network's message synchronization latency. There are still some issues with the gossip algorithms that are in use today, including low throughput, high latency, energy waste, and increased network communication requirements. In order to lower delivery latency and increase transaction throughput for BC systems, a highly efficient BC gossip algorithm is essential. The distributed ledger, known as the BC, is used to centrally store all activities and transactions. Public blockchains such as Bitcoin highlight decentralization, which raises communication costs and requires high throughput. The purpose of the partial decentralization is to boost consortium BC's effectiveness. Although the high communication overhead slows down scalability, the consensus techniques used in consortium BC have addressed a number of limitations. Therefore, the joint-graph based delegated practical byzantine fault tolerance (JG-DPBFT) consensus mechanism is used for the hyperledger fabric BC technology. The improved gossip algorithm (IGA) is utilized to complete information communication and consensus authentication. The JG-DPBFT protocol is implemented using NS-3, and the performance is evaluated by comparing it with existing methods. As a result, the experimental outcomes proved that the JG-DPBFT has attained better throughput and latency than the prevailing methods.

Keywords: Hyperledger fabric, Delegated practical byzantine fault tolerance, Enhanced Gossip algorithm, Peer-to-peer network, Hashgraph, Consensus mechanism, Blockchain.

1. Introduction

The Internet of Things (IoT) paradigm's development and emergence transformed numerous industries, including manufacturing, supply chains, smart metering, smart homes, and healthcare [1]. The network's sensors are in charge of gathering data from the linked devices, while the actuators handle the raw data processing. Gaining an understanding of productivity, system efficiency, and other performance metrics is possible through data analysis. Because of the built-in units, the majority of the network's devices have been found to have restricted battery, storage, and memory capacity [2, 3]. Owing to these restrictions, cloud computing infrastructure frequently handles data processing, storage, and security. The idea of centralization in the network is introduced by this dependency [4]. Several data centers are part of the cloud computing architecture, which enables it to execute enormous volumes of tasks at once by offering on-demand resources and services [5].

Even though the cloud can process enormous amounts of data efficiently, latency-sensitive applications may experience delays due to security and trust concerns. This causes issues and lowers productivity overall [6]. In order to tackle these problems, fog and edge computing architectures were developed, which did away with the idea of centralization and improved the performance of latency-sensitive applications [7]. These designs work well for carrying out a lot of latency-sensitive tasks, but they cannot guarantee security or trust between nodes or address the issue of malicious attacks [8]. As a result, while handling sensitive applications, industrial IoT (IIoT) designs cannot fully rely on those computing technologies [9].

The inadequacy of those architectures renders blockchain (BC) an indisputable technology that ensures the confidentiality and integrity of internet-processed data [10]. Because the participating nodes maintain this technology jointly, there is no need to trust anyone peer. Initially, the BC receives the incoming data from the nodes as transactions [11]. Subsequently, various transactions sent to BC are merged into a block, which requires validation by participating nodes [12]. The nodes run a consensus procedure to determine whether the block is legitimate before adding it to the main ledger. In addition, hashing is done, and the hash value is used to link every two blocks [13]. By preventing unauthorized changes to the data contained in blocks, BC guarantees security. Furthermore, by broadcasting the confirmed block to every network node, the worldwide consistency of the ledger is guaranteed [14].

Since it is decentralized, the peer-to-peer (P2P) network paradigm is considered one of the fundamental technologies of BC [15]. As BC is continuously implemented, a number of programs have embraced semi-distributed network architectures, which combine fully distributed and centralized network structures and introduce the concept of ledger nodes [16–18]. High-speed data transfer is possible with this kind of combination. The gossip method is used by peer nodes within each ledger's channels to spread information [19, 20]. The algorithm in concern is extensively utilized in the construction of peer-to-peer (P2P) models owing to its defect tolerance, simplicity, and efficiency.

The gossip algorithm is utilized by the majority of blockchain-based initiatives, such as Hyperledger fabric and Bitcoin, to distribute block information and ensure effective data synchronization between nodes [21-24]. The original gossip algorithm has undergone a number of improvements and modifications in recent years in an attempt to increase performance, but transaction throughput still has space for growth. Therefore, in order to circumvent the aforementioned shortcomings, this work introduces an effective algorithm. The major contributions of the proposed work are presented below as follows:

- To present an enhanced, scalable, and efficient joint-graph based Delegated Practical Byzantine Fault Tolerance (JG-DPBFT) for BC.
- To present an improved gossip algorithm (IGA) for reducing the redundancy and enhancing the data synchronization efficiency.
- To introduce a block proposer and threshold based signature mechanism for assuring the validity of the generated block.
- To compare the performance of the proposed gossip algorithm with respect to latency and throughput.

The remaining structure of the paper is stated as follows: Section 2 discusses the proposed methodology. Section 3 demonstrates the result and discussion. Lastly, Section 4 exemplifies the conclusion and future scope.

2. Proposed methodology

The gossip algorithm is generally regarded as efficient and straightforward owing to its exceptional robustness and scalability. As a result, this algorithm is extensively implemented in dynamic, large-scale, non-centralized environments. A number of blockchain initiatives, such as Bitcoin and Hyperledger Fabric, employ the gossip protocol to synchronize data between nodes. However, the actual gossip algorithm employs a constant probability to select the target node at random for message transmission. As a result, messages are frequently transmitted to the same node, which reduces the effectiveness of data synchronization. In addition, this issue has the potential to generate a substantial volume of redundant data, imposing a significant strain on the network. Thus, to avoid this issue, the proposed study introduced a new Improved Gossip Algorithm (IGA) in the Joint-Graph Based Delegated Practical Byzantine Fault Tolerance (JG-DPBFT) method. The IGA is executed via the transmission control protocol (TCP), wherein each node arbitrarily selects a small set of peers to which it forwards gossip messages. All messages must be signed by the actual transmitter in order to ensure their authenticity; the validity of the signature should be verified by the other nodes prior to message relay. In the process of developing the JG-DPBFT, a threshold-based signature mechanism and a block proposer are incorporated to ensure the validity of the generated blocks. The node with the greatest reputation weight is chosen as the proposer of the block, and signature votes are generated using a threshold-based signature scheme. In conjunction with the block proposer and signature voting, gossip enhancement is carried out.

Stage 1: Proposer selection (Block proposal)

The Algorand protocol is thought to serve as the basis for the reputation-based VRF cryptographic sortition process used to choose the block proposer. Algorand uses PoS consensus and VRF to select validators and block proposers. During the PoS process, nodes with a higher token count have a higher chance of being chosen. However, since each node's identity is known, PoS consensus is not required in the consortium blockchain. As a result, a reputation model that assigns a weight to each node based on their reputation score has offered to swap Algorand's stock scheme. The cryptographic sorting mechanism known as VRF is used to select the block proposer among the nodes based on reputation weight.

The common random seeds have been resembled:

$$SEED_s = I(SEED_{s-1} \parallel \varsigma_{s-1}) \tag{1}$$

The seed of round s has computed through the seed $SEED_{s-1}$ of round $s-1$ as well as the threshold signature ς_{s-1}.

The proposal information encompasses the non-tampering and non-interactive process of creating random seeds. Until the end of the round s, if the group signature ς_{s-1} is not unveiled, it has been predicted in advance, even $SEED_{s-1}$ is reflected as a known parameter. An adversary should not govern the seed for breaking the cryptographic sortition even though a single node finishes consensus before the others and is considered as the first recovery ς_{s-1}. This is due to the inability of the node to adjust the recovery outcome.

Because the number of proposers for each block is unpredictable, there may be proposers for more than one block. Large information communication expenses could arise when these proposers disseminate the proposed blocks. To reduce the cost, the distinct block proposals are prioritized for hashing the hash result of VRFs concatenated with the sub-R index (reputable unit). The node with the highest priority (lowest hash value) in this round is the one making the block proposal.

Block proposal: The blocks proposed by the block proposer q_j are categorized into two types: (a) q_j formerly proposed block C_{ped}, which has not been committed owing to some reasons or timeout; (b) q_j packs a new block with the information of the proposal. The proposer q_j offers a proposal certificate d for the proposed block C for proving its validity of the proposed block. When the chosen block C is the previously submitted block C_{ped} and it is unsuccessful, the proposer q_j must offer the group signature ς_q that has been initiates during the prepare stage, which is recognized as the P-certificate. The proposal certificate will be the D-certificate of the final block uploaded to the blockchain while the proposed block is new.

To recover, the proposal certificate desires to receive at least $2g+1$ signatures, which are threshold signatures. The verification process has been carried out using the system's public key. Thereby, it is definitive, unique, and cannot be forged. Eventually, the proposer generates the PR message (which means $< I(C), i, d, \pi >$) and uses gossip to send it to each node. Other nodes can check the certificates, signatures, and proposed blocks to verify the legitimacy of the PR data. Additionally, it is necessary to verify the block height in relation to the locally committed blockchain. When some of the committed blocks are missing, a node must do an asynchronous recovery. Before restoration, blocks cannot be offered; instead, signature votes are the only method used.

Stage 2: Signature voting

Broadcasting the signature data and essentially committing the block are the goals of signature voting. The BLS threshold signature model is used as the voting mechanism. The gossip communication technique was used in the initial phase to spread all messages. Each node q_j is represented as a finite state machine that shows the initial local state and the votes with respect to the current local state and the received proposals. Every node q_j should verify the validity of every block proposal and examine the block with maximum priority. Although the most recent committed block's state information is on the local state chain with the proposed certificate, the initial node state has been assessed.

Timeout nodes will sign and transfer both invalid proposal blocks and empty blocks when the proposer passes an invalid block, proving that the current block proposer is misbehaving. The node q_j initiates to work on the information on the received proposal after initialization. The BLS signature is performed by node q_j on block C and appends it to the Q message after coming to prepare phase. Next, the block C is attached to every assembled signature as well as its own signatures and gossip together. Node q_j recovers the single group signature ς_q, transfers, and updates the status information for proceeding to the commit phase once it receives more than $2g+1$ verified Q messages. ς_q serves as evidence that the preparation stage is completed. After receiving ς_q, other nodes confirm it by utilizing the threshold signature's single-message confirmation characteristic based on the public key of the system and end the preparation stage. Similar to prepare stage, the node q_j carries out operations during the commit stage with the exemption that block finalization is verified through the recovered group signature ς_d. Node state information is appended to the local state chain. In the event that a timeout prevents the block from being committed, the node will sign an empty block. The block will then be completed, and the next round will begin when other nodes with timeout capabilities sign similar empty blocks. The empty block clearly shows no commitment to the blockchain.

2.1 Improved gossip network communication

When it comes to peer-to-peer networks, gossip is a common algorithm. Being a powerful tool in distributed network systems, it also has the benefits of being simple, efficient, and fault-tolerant [39]. Multiple blockchain projects have made use of the gossip protocol to synchronize data between nodes. These projects include Hyperledger Fabric and Bitcoin [40]. The proof that every node has an equal and maximum probability of receiving every message is provided by the gossip protocol.

In a small network, each node engages in gossip protocol-based, random interactions with its neighbors while tending to the sharing of the same information among all network nodes. Every node participating in the gossip protocol has the ability to store data about its neighbors, and there are three channels of communication available between two nodes:

Push: Node A selects neighbor B at random as a communicatee and transfers information to B. Next, the local data is updated by Node B, conferring to the information it has received. In this circumstance, new information is acquired by node A.

Pull: Node A picks neighbor B at random as the communicatee and forwards a request to update passive information. After that, Node B will send the missing information to Node A. Node A will then make use of the information it has received in order to update the local data. Under these circumstances, node B obtains new information.

Push-Pull: This communication style is comparable to the Pull style from the initial phase. In the subsequent phase, node A relays the data that node B is missing. Next, Node B will update the local data based on the information it has received. A cycle is defined as the total amount of time that two nodes need to use the aforementioned modes in order to share similar information.

Gossip improvement in phase I: During the block proposal phase, a predetermined message is broadcast by the block proposer. The list of global historical nodes is expanded in order to document the historical nodes that receive information. The possibility of selecting duplicate nodes is diminished as a result of the dissemination of ongoing updates to historical node inventories; this, in turn, tends to reduce the message redundancy of the network. In order to determine the list of new nodes, each node generates a difference set consisting of all consensus-participating nodes and the list of prior nodes. In order to guarantee that a single block message is transmitted to the majority of nodes, the target node's broadcast block is subsequently selected at random from a new list. The node can proceed to the next phase if the block proposal stage ends at a fixed interval $U1$ or if the difference set is empty. Here, Tv defines the block information that is required to be synchronized. The array of H historic nodes is deliberated as MI. Consider, p be the number of peer nodes, l represents the number of targets sending nodes, and the number of nodes in MI is specified as a variable, and it has been expressed mathematically as follows:

$$H(j) = l(j-1)+1 \tag{2}$$

where, j denotes the number of the current network's gossip period. Further, the number of nodes chosen by every node while sending a message is fixed to l. Every new node that accepts the message has been propagating the message again to l new nodes. Thereby, the number of new nodes that accept messages in a period j is deliberated as Y:

$$Y = l^j \tag{3}$$

Based on the above expression, the total number of nodes that accomplished data synchronization is obtained at the completion of the period j. It can be expressed using the function $G(j)$.

$$G(j) = l^0 + l^1 + l^2 + ... + l^{j-1} + l^j = \sum_{j=0}^{P} l^j \tag{4}$$

If $G(j) = p+1$, the required period j is expressed as:

$$j = \log_l \left(p+1-\frac{p}{l} \right) \tag{5}$$

Additionally, the overall improved process in stage I is scientifically described as follows:

- A fixed message is broadcasted by the block proposer to every node in P.
- As nodes receive the information, the list of global historical nodes I is continuously updated.
- Take the difference set between the historical node list (I) and every node participating in consensus (C) to choose a new node list N_k by each node k.
- Then, the target node is chosen randomly to broadcast the block from a new list N_k. This ensures that the majority of nodes have exclusively received a singular block message.
- An empty difference set signifies that all nodes have obtained the information; alternatively, the block proposal stage concludes, and the node may proceed to the subsequent stage after a predetermined time interval elapses.

Gossip improvement in phase II: A node does not consistently transmit a fixed quantity of signature data throughout the signature voting phase. Consequently, every node appends a list of local history nodes and transmits a diverse array of messages to a variety of target nodes. The node is entirely obligated to transmit the recently acquired signature information to a distinct target node if it is possible to access the desired target node from the local historical node list. Conversely, each signature message is acquired by the target node, which is also appended to the group of local history nodes. This process has been mathematically described as follows:

- Let U_k be the set of target nodes that a node k can possibly forward the information of signature for each node in the current round.
- Consider R_k represents the set of nodes in N_k (set of nodes in the list of local history node of node k) that are also in U_k. These are considered as the queried nodes in the local history node list of the node k.
- The newly received signature information T_k is only send from the node k to the node in R_k, if R_k is not empty.
- When R_k is empty, the node k forwards every signature message T_k to target nodes in U_k as well as includes these nodes to N_k.

As a result, it is stated that the goal of this strategy is to reduce unnecessary communication by sending information only to nodes that have never received it before, as shown by the local history node list.

3. Results and discussion

The outcomes of the proposed JG-DPBFT model are discussed in this subdivision in order to reach a consensus on whether it is useful for BC technology on hyperledger fabrics. A simulation running on an Intel Core i5-4570S CPU with 8GB RAM on the NS3 platform validates the JG-DPBFT model. A series of studies were used to confirm the protocol's viability, and the overall performance of the JG-DPBFT procedure was assessed. The delay resulting from the block size is included to simulate the validation of the transaction.

3.1 Performance evaluation with existing methods

A performance comparison between JG-DPBFT and other related protocols is offered in order to comprehensively illustrate the impact of the protocol on solutions. Next, JG-DPBFT's overall performance is shown on a test platform with 200 nodes, and every parameter's impact on the protocol and optimization outcome is thoroughly examined. The average throughput for finishing 100 kb of data without failure is evaluated using both traditional protocols and the proposed JG-DPBFT, as shown in Figure 1.

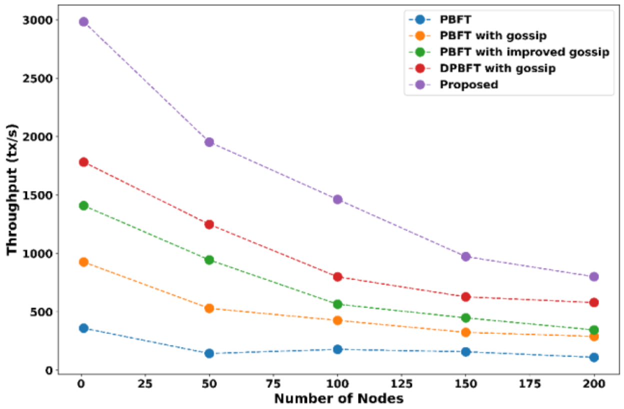

Figure 1: Performance evaluation in terms of throughput for completing 100 KB data

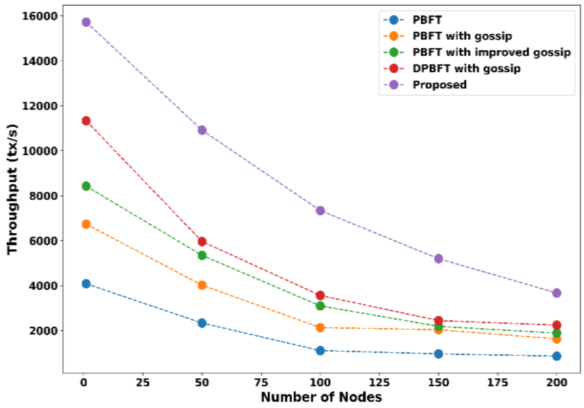

Figure 2: Performance evaluation in terms of throughput for completing 1 MB of data

The average throughput for finishing 1 MB of data without failure is evaluated using both traditional protocols and the proposed JG-DPBFT, as shown in Figure 2. Analyzing each set of data demonstrates that, for a given block size, the performance of each protocol is inversely related to the size of the network. This problem has been linked to the network's attempt to maximize the number of nodes and messages it can send, which ultimately causes transmission latency to increase and performance to decrease.

Figure 3 illustrates that the proposed JG-DPBFT and traditional protocols compare in terms of average latency for successfully completing 100 KB of data. Latency is defined as the interval of time between sending a request to a user and its completion. Both message latency and encryption overhead are the main factors influencing lag. The average delay performance for a range of nodes in successfully completing 1 MB of data is shown in Figure 4.

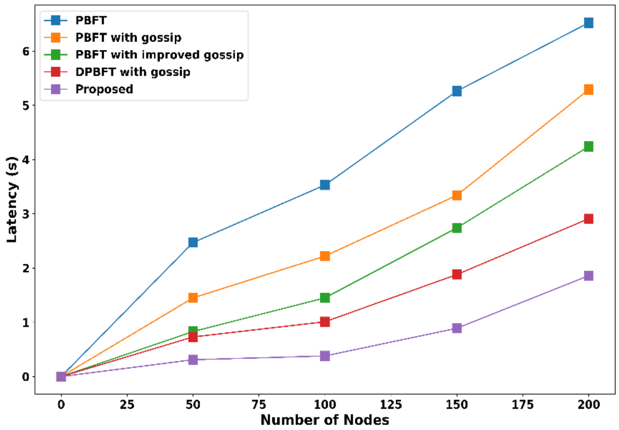

Figure 3: Performance evaluation in terms of latency for completing 100 KB data

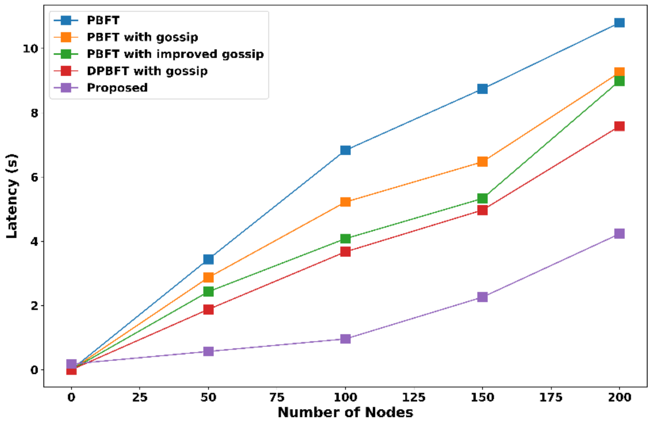

Figure 4: Performance evaluation in terms of latency for completing 1 MB of data

3.2 Comparison with state-of-the-art methods

In this section, the proposed JG-DPBFT is compared against other state-of-the-art consensus algorithms like PBFT, DPoS, PoW, and PBFT-derived protocols, such as DBFT, DRBFT, and DPoA+PBFT. Table 6 exhibits the comparison outcomes with state-of-the-art consensus algorithms. It is significant to note that the network composed of large-scale nodes takes efficiency into consideration. DPoS and PoW find it challenging to withstand Byzantine faults. If more nodes are involved, PBFT is incompetent. DPoA+PBFT and DBFT are intended to decrease system activity since voting mechanisms alone cannot guarantee adequate fairness. Additionally, nodes waste more resources in DPoA+PBFT and PoW since they are challenging accounting rights. The proposed JG-DPBFT is more practical than other algorithms.

Table 1: Comparison with state-of-the-art methods

Consensus protocol	**Activity of system**	**Fault tolerance**	**Efficiency**	**Resource waste**
DPoS	Medium	No	High	Low
PoW	Low	No	Medium	High
DBFT	Medium	Yes	High	Low
PBFT	High	Yes	Low	Low
DRBFT	High	Yes	High	Low
DPoA+PBFT	Low	Yes	High	High
Proposed	**High**	**Yes**	**High**	**Low**

mixed throughput, and guaranteed scalable and secure consensus in decentralized systems.

4. Conclusion

The main contribution of this work is a novel JG-DPBFT for the hyperledger fabric BC technology, which helps to maximize performance and solve various issues. The JG-DPBFT offered an innovative IGA for communicating efficient information and consensus verification, thereby improving security and scalability. The JG-DPBFT protocol outperforms other protocols, including PBFT variations and DPBFT with gossip, based on comparison analysis. While varying the number of nodes to 200, the proposed JG-DPBFT protocol has achieved minimum latency of 1.86 sec, 2.23 sec, and 3.23 sec for 100kB, 1MB, and 2MB of data. Similarly, the proposed JG-DPBFT has attained higher throughput of 801.15 tx/s, 3672.76 tx/s, and 3588.09 tx/s for 100kB, 1MB and 2MB. In addition, while varying the block size to 5000 KB, the proposed JG-DPBFT has achieved better throughput and latency of 5681.46 tx/s and 2.89 sec, respectively. However, there are concerns about the possibility of individuals with high reputation values colluding to disrupt the network. This requires the implementation of incentive structures to encourage active involvement and discourage malevolent behavior. To summarize, although the JG-DPBFT protocol offers notable improvements in consensus protocol design, further study is necessary to address the remaining obstacles and guarantee its resilience and scalability in real-world blockchain applications. In the future, the proposed study can also be expanded by taking into account numerous users and assessing performance.

References

[1] Haddock J, Jarman B, Yap C. Paving the way for consensus: Convergence of block gossip algorithms. *IEEE Transactions on Information Theory*. 2022; 68(11):7515-27.

[2] Hao W, Zeng J, Dai X, Xiao J, Hua Q, Chen H, Li KC, Jin H. BlockP2P: Enabling fast blockchain broadcast with scalable peer-to-peer network topology. In *Green, Pervasive, and Cloud Computing: 14th International Conference, GPC 2019, Uberlândia, Brazil, May 26–28, 2019, Proceedings 14* 2019; 223-237. Springer International Publishing.

[3] Huang J, Tan L, Mao S, Yu K. Blockchain network propagation mechanism based on P4P architecture. *Security and Communication Networks*. 2021; 2021:1-2.

[4] Antwi R, Gadze JD, Tchao ET, Sikora A, Nunoo-Mensah H, Agbemenu AS, Obour Agyekum KO, Agyemang JO, Welte D, Keelson E. A survey on network optimization techniques for blockchain systems. *Algorithms*. 2022; 15(6):193.

[5] Shaleva A, Korkhov V. Evaluation of the Neo P2P Blockchain Network Protocol Efficiency. *In Computational Science and Its Applications–ICCSA 2021: 21st International Conference, Cagliari, Italy, September 13–16, 2021, Proceedings, Part IV 21* 2021; 56-71. Springer International Publishing.

[6] Hu W, Hu Y, Yao W, Li H. A blockchain-based byzantine consensus algorithm for information authentication of the internet of vehicles. *IEEE Access*. 2019; 7:139703-11.

[7] Hao W, Zeng J, Dai X, Xiao J, Hua QS, Chen H, Li KC, Jin H. Towards a trust-enhanced blockchain P2P topology for enabling fast and reliable broadcast. *IEEE Transactions on Network and Service Management*. 2020; 17(2):904-17.

[8] Wenbo M, Wenxiang W. GoUncle: A Blockchain Of, By, For Modest Computers. *Cryptology ePrint Archive*. 2021.

[9] Kuo TT, Kim J, Gabriel RA. Privacy-preserving model learning on a blockchain network-of-networks. *Journal of the American Medical Informatics Association*. 2020; 27(3):343-54.

[10] Danos V, Lelarge M. Improving blockchain consensus protocol.

[11] PAREKH N, SADANAND P, JAIN S. Blockchain technology. *Journal of Emerging Technology and Innovative Research (JETIR)*. 2017; 4(0):177-9.

[12] Wu S, Wei Y, Gao Y, Zhang W. Probabilistic quantization of unbiased broadcast gossip algorithms for consensus in distributed networks. *Wireless Networks*. 2019:1-7.

[13] Akhtar Z. From blockchain to hashgraph: distributed ledger technologies in the wild. In *2019 International Conference on Electrical, Electronics and Computer Engineering (UPCON)* 2019; 1-6. IEEE.

[14] Khullar K, Malhotra Y, Kumar A. Decentralized and secure communication architecture for fanets using blockchain. *Procedia Computer Science*. 2020; 173:158-70.

[15] Mišić J, Mišić VB, Chang X, Motlagh SG, Ali MZ. Modeling of bitcoin's blockchain delivery network. *IEEE Transactions on Network Science and Engineering*. 2019; 7(3):1368-81.

[16] Thai PD, Doan M, Liu W, Liu T, Li S, Zhou HS, Dinh TN. Blockchain Peer-to-Peer Network: Performance and Security. In *Handbook on Blockchain* 2022; 55-83. Cham: Springer International Publishing.

[17] Wang Y, Li Y, Jiao W, Wang G, Zhao J, Qiang Y, Li K. An efficient, secured, and infinitely scalable consensus mechanism for peer-to-peer energy trading blockchain. *IEEE Transactions on Industry Applications*. 2023.

[18] Yadav AK, Singh K, Amin AH, Almutairi L, Alsenani TR, Ahmadian A. A comparative study on consensus mechanism with security threats and future scopes: Blockchain. *Computer Communications*. 2023; 201:102-15.

[19] Ahmed M, Akhter AS, Rashid AB, Pathan AS. A dependable and secure consensus algorithm for blockchain assisted microservice architecture. *Computers and Electrical Engineering*. 2023; 109:108762.

[20] Jain S, Doriya R. Performance evaluation of hyper-ledger fabric-based consensus mechanism on multi-robot path planning. *Multimedia Tools and Applications*. 2023:1-5.

[21] Ahmad K, Ricci LE, Baiardi F, Arsheen S. Hyperledger Fabric Enabled Vaccine Intelligent Network to Implement Immunization Program. In *2023 IEEE 12th International Conference on Communication Systems and Network Technologies (CSNT)* 2023; 822-828. IEEE.

[22] Onifade M, Adebisi JA, Zvarivadza T. Recent advances in blockchain technology: prospects, applications and constraints in the minerals industry. *International Journal of Mining, Reclamation and Environment*. 2024:1-37.

[23] Malik R, Raza H, Saleem M. Towards A Blockchain Enabled Integrated Library Management System Using Hyperledger Fabric: Using Hyperledger Fabric. *International Journal of Computational and Innovative Sciences*. 2022; 1(3):17-24.

[24] Moses, M. B., Nithya, S. E. & Parameswari, M. (2022). Internet of Things and Geographical Information System based Monitoring and Mapping of Real Time Water Quality System. International Journal of Environmental Sciences, 8(1), 27-36. https://www.theaspd.com/resources/3.%20Water%20Quality%20Monitoring%20Paper.pdf