Journal of Basrah Researches (Sciences) **50**(2), 267 (2024)
DOI: https://doi.org/10.56714/bjrs.50.2.23

Assessment of Consensus Algorithms for Blockchain Technology to Enhance Decentralized Applications

Saja Adnan Amenr

Department of Computer and Communications Engineering, Faculty of Engineering & Computer Science, American University of Science & Technology, Beirut, Lebanon.

ARTICLE INFO

Received 2 November 2024
Revised 10 December 2024
Accepted 16 December 2024
Published 31 December 2024

Keywords:

Blockchain, Decentralized Applications, Proof of Work, Proof of Stake, Delegated Proof of Stake.

Citation: S. A. Amenr, J. Basrah Res. (Sci.) **50**(2), 267 (2024). DOI:https://doi.org/10.56714/bjrs.50.2.23

ABSTRACT

The accelerating pace of adoption of decentralized applications requires the use of efficient, high-performance blockchain infrastructures. These blockchains are supported by consensus algorithms that are critical determinants of scalability, transaction speed, costs, and security. Developers are still unaware of the most useful options because there is a large gap in information regarding the comprehensive evaluation of these algorithms in practice. This study aims to test and compare the performance of the three most popularly used blockchain consensus protocols - Proof of Work (PoW), Proof of Stake (PoS) and Delegated Proof of Stake (DPoS) with the aim to improve the application of decentralized systems. By deploying smart contracts on real blockchain test networks (Callisto for PoW, Sepolia for PoS, and Tron Nile for DPoS), the research explores key performance metrics such as block time, deploy gas fee, block gas limit, and block size. The experiments utilize tools like Remix IDE and Tron-IDE, highlighting the practical implications of consensus algorithms under varying real-world conditions, including network congestion and transaction volatility. Tron Nile excels in speed and capacity but incurs high gas fees, Sepolia balances performance with moderate costs, and Callisto emphasizes cost efficiency at the expense of speed and scalability. Insights derived from this study provide valuable guidelines for developers to choose suitable consensus mechanisms based on the specific requirements of decentralized applications.

1. Introduction

One of the 21st century's most exciting new technologies is blockchain technology. It has many benefits, including traceability, non-tampering, non-counterfeiting, and decentralization [1]. Important data pertaining to data security, anti-counterfeiting, and other realistic situations can be stored on it with great success [2]. Blockchain technology began in 2009 when the genesis block was added to Bitcoin and thus a new scientific system and advanced distributed technology was born. This technology has become a game-changer for a number of domains since it provides decentralized, secure, and unchangeable data storage and transaction processing. Blockchain networks' efficacy and scalability, however, are heavily reliant on their capacity to bring disparate nodes to consensus [3].

*Corresponding author email : Sajaadnan2018@gmail.com

ISSN: 1817-2695 (Print); 2411-524X (Online)
Online at: https://jou.jobrs.edu.iq

In the absence of a strong consensus process, blockchains are vulnerable to malicious assaults, ineffective, and unable to maintain confidence in a decentralized setting.

Consensus algorithms are the foundation of blockchain technology because they allow nodes to validate transactions and agree on the network's current state without the need for a central authority. These techniques ensure that the blockchain is able to function correctly even in the presence of malicious users [4]. As the requirements of different types of systems are different, a number of consensus approaches have been proposed, each having its own merits and demerits. It is further assumed that certain technologies do provide mechanisms for securing blockchain against malevolent nodes, and such technologies are called Byzantine fault tolerant technologies [5]. The majority of early consensus algorithms, including the VR (Viewstamped Replication) consensus method put forth by Brian M. Oki et al. in 1988 [6], are generally categorized as non-Byzantine because these methodologies did not factor in the existence of malevolent nodes. Non-Byzantine algorithms have some weaknesses which can only be applied to the uses of blockchains which consists of small number of nodes and nodes considered to be trustable. Satoshi Nakamoto, in 2008, made public the notion of 'Bitcoin'. He proposed the architecture of a system to implement the Bitcoin systems which uses Markus Jakobsson's Proof of Work (POW) [7] consensus algorithm with the threat of malicious nodes existing on a blockchain. It is since then that Byzantine algorithms have been advancing fast.

Particularly well known are the Proof of Stake (PoS) and Delegated Proof of Stake (DPoS) algorithms, as well as the superfast and energyefficient Ripple consensus algorithm. Other mentionable algorithms are The Proof of Activity (POA), which is a hybrid of PoW and PoS, and the Proof of Stake Velocity (PoSV), which promotes staking and sending transactions. To attain mining rights, participants in Proof of Burn (POB) have to pay for it by "burning" money while HoneyBadger BFT provides strong robustness to fault in the extreme case of adversarial setting [8]. Also further designed to improve scalability, security and robustness in decentralized systems are a host of algorithms such as VBFT, Snowflake to Avalanche, and the 99% Fault Tolerant Consensus algorithm which are suitable for different demands in various blockchain applications.

Decentralized applications on the basis of blockchain are innovative software applications which guarantee safety and trustworthy information and communication without central control [9]. Irrespective of its prospects of adoption, factors like scalability, high cost of transactions and low throughput stifle its widespread use. In order to guarantee smooth user experiences, these apps need to have their consensus methods enhanced, network scalability increased, and latency decreased.

Choosing a consensus algorithm is becoming increasingly important for developers, businesses, and consumers as the blockchain industry develops. However, the large number of consensus algorithms currently available, each with amazing features, makes it difficult to choose the optimal one for a particular decentralized application. The issue pertains to the lack of a comprehensive understanding and systematic assessment of various blockchain consensus algorithms within the context of decentralized applications. The absence of those evaluation studies hinders stakeholders' ability to make informed decisions, which could lead to differences between the decentralized utility's desires and the consensus solution that is selected. To effectively design and implement decentralized apps in a variety of scenarios, it is imperative that this gap be filled.

Recently, several papers have been presented that have analyzed different blockchain consensus mechanisms. In [10] the algorithms were compared based on attributes such as security, scalability, and compatibility with IoT, while in [11] the high security of PoW but inefficiency was highlighted. The study [12] discusses decision tree models for selecting scalable algorithms such as DPoS, Tendermint, and Ouroboros. In [13], PoS algorithms are described and optimization methods and product selection optimizations are demonstrated. In [14], several algorithms were evaluated based on energy efficiency, fault tolerance and attack resistance. The study [15] examines the uses and effectiveness of consensus procedures including PoW, PoS, and PBFT. The hybrid PoW-PoS technique and smart contract transparency are examined in [16]. A basic overview of algorithms is given in [17], with a focus on their history and characteristics. In contrast to the proof-of-work implementations of Ethereum and Bitcoin, Byzantine consensus in blockchain was examined in [18]. In [19], BFT algorithms were studied, analyzing message complexity, latency, and strengths. In [20], five blockchain audit consensus protocols were evaluated, with Clique identified as the most efficient.

Compiling consensus algorithms into a systematic study framework was the main motivation for the research [21]. However, the perfect consensus algorithm is still elusive since almost all algorithms have significant drawbacks in one way or another regarding their security and performance. Until the consensus algorithm finds the right balance between these critical factors, we may not see widespread adoption as many cryptocurrencies enthusiast's hope. In [22], a decentralized consensus algorithm based on blockchain technology is designed, implemented and analyzed to solve the problem of optimal energy flow. Successive iterations of the solution to the energy flow problem are securely stored on the blockchain, eliminating the need for a central operating authority, while allowing network nodes to validate and improve the solution. Simulation experiments were conducted on the 39-bus New England Transmission System, and the IEEE-57 and IEEE-118 standard systems. Recently, there has been interest in applying Blockchain technology to the Internet of Things (IoT). Specifically, consensus mechanisms have been modified to be less resource-intensive, and more suitable for deployment in IoT, with consensus mechanisms such as Credit-Based Proof of Work (CBPoW) and Proof of Supply Chain Sharing (PoSCS). In [23], the suitability of permissionless Blockchains for IoT and the trade-offs required, especially for resource-constrained IoT devices, were examined. In [24], current challenges such as centralization, fair token distribution, scalability, and sustainability are highlighted. The energy consumption of blockchain networks has raised concerns about their environmental impact. Interoperability between different blockchains and security in specific environments, such as the Internet of Things, are areas that still require significant research attention. Understanding and improving these algorithms is critical to unlocking the full potential of blockchain technology in a variety of applications and industry sectors. In [25], a comprehensive review of prominent consensus mechanisms is presented to clarify their operating principles. Through a systematic comparison based on specific parameters, transaction processes across different blockchains were analyzed to understand their suitability for diverse applications. The results reveal significant trade-offs between the chosen algorithms, highlighting how efficiency is balanced with security and decentralization.

Existing research highlights important trade-offs between these algorithms, and reveals shortcomings in several features. Recent efforts to adapt consensus mechanisms in resource-constrained environments, such as the Internet of Things, demonstrate progress but underscore the lack of systematic and comprehensive evaluations. This fragmented understanding requires a systematic approach to analyzing consensus algorithms, and addressing critical gaps to enable their broader adoption and application in various fields. There is no single algorithm that achieves an optimal balance across important parameters, so there is an urgent need to understand and improve consensus algorithms, focusing on the diverse requirements of decentralized systems on the basis of which the most appropriate algorithm is chosen.

This study's main objective is to assess and contrast the main blockchain consensus algorithms in order to ascertain how well they work in decentralized applications. This entails evaluating its performance in relation to crucial metrics including network efficiency, scalability, transaction fees, and block time. The study intends to demonstrate the advantages and disadvantages of each algorithm in real-world use cases and offer insights into how these consensus mechanisms affect the security, decentralization, and operational effectiveness of decentralized applications through experiments on actual blockchain test networks.

2. Materials and Methods

The research uses a comparative approach to evaluate the performance of three widely used blockchain consensus algorithms. Experiments are performed on real test networks and key performance parameters are evaluated to determine the efficiency of each algorithm. Remix IDE is used to simulate and validate blockchain operations. Data from these test networks is analyzed to determine the strengths, weaknesses, and trade-offs associated with each consensus method. Figure 1 shows the proposed methodology for testing three consensus algorithms related to scalability, security, and efficiency in blockchain. The Remix IDE will be used as a development environment to simulate blockchain operations and smart contract deployment and real test networks will be configured to provide realistic scenarios to evaluate performance. Blockchain wallets are set up to manage cryptographic keys and interact with the respective networks. Smart contracts are developed

with functions such as deposit, withdrawal and money transfer, and then deployed on the selected test networks. Key performance metrics are finally measured to evaluate the operational efficiency of each consensus mechanism.

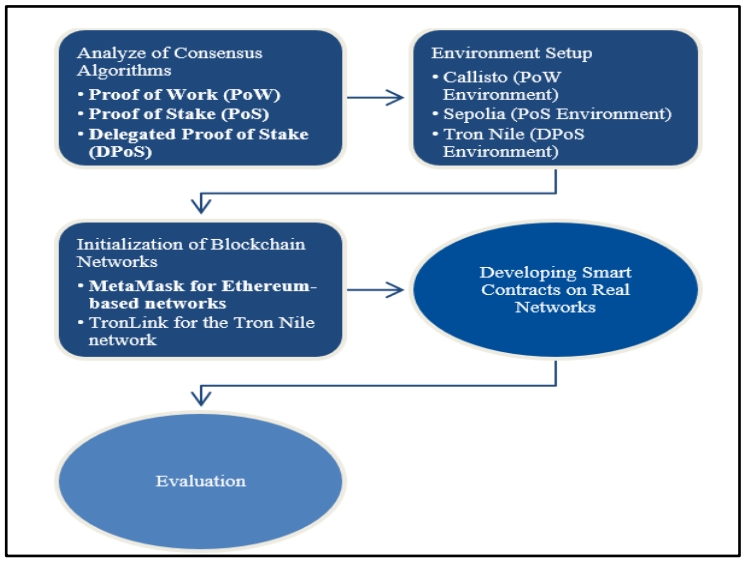

Fig. 1. Proposed methodology for evaluating consensus algorithms.

2.1. Analyze of Consensus Algorithms

Distributed consensus algorithms, which guarantee agreement among network participants despite any errors or conflicts, are essential to the integrity and operation of blockchain systems. In a decentralized setting, these algorithms allow the system to produce a final, consistent version of the truth, which is essential for preserving dependability and confidence. Figure 2 illustrates how a blockchain network's consensus mechanism operates. A peer-to-peer (P2P) network is used to broadcast the transaction after it has been initiated. A consensus procedure is used by the network of nodes to verify the transaction. A new block is added to the blockchain when the transaction has been verified. The procedure is finally finished when the transaction is validated and included to the block.

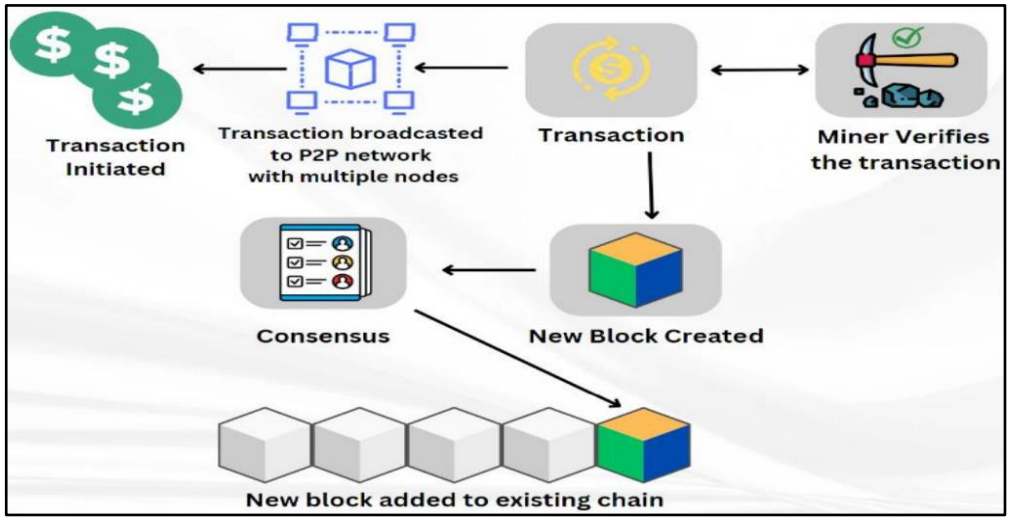

Fig. 2. Consensus mechanism in blockchain.

The three main consensus mechanisms—Proof of Work (PoW), Proof of Stake (PoS), and Delegated Proof of Stake (DPoS)—each address distinct issues and trade-offs in terms of scalability, efficiency, and security.

2.1.1. Proof of Work (PoW)

Blockchains harvest the benefits of the security and legitimacy corroboration provided by a specific consensus mechanism termed a Proof of Work (PoW) algorithm [26]. The main aim 'target' of the said technology of blockchain is to reach a conclusion among the participants (nodes or miners) within trusted and distributed environment with respect to a single version of the blockchain at a specific moment in time". An overview of the PoW algorithm's working principles can be found in Fig. 3. The algorithm stipulates that miners are required to solve various forms of computational puzzles by means of searching for a nonce which is less than or equal to a predefined target, which guarantees that the transaction is legitimate. The process of mining involves a hash of the block data being formed with the rate of forming these hashes being controlled to create a time interval between hashes. Once a block has been validated through the process of hashing, the block is then linked and added in chains forming a blockchain. This practice enables security and decentralization of the network. All in all, PoW guarantees the existence of a dependable approach for preserving the integrity of all operations performed on the blockchain.

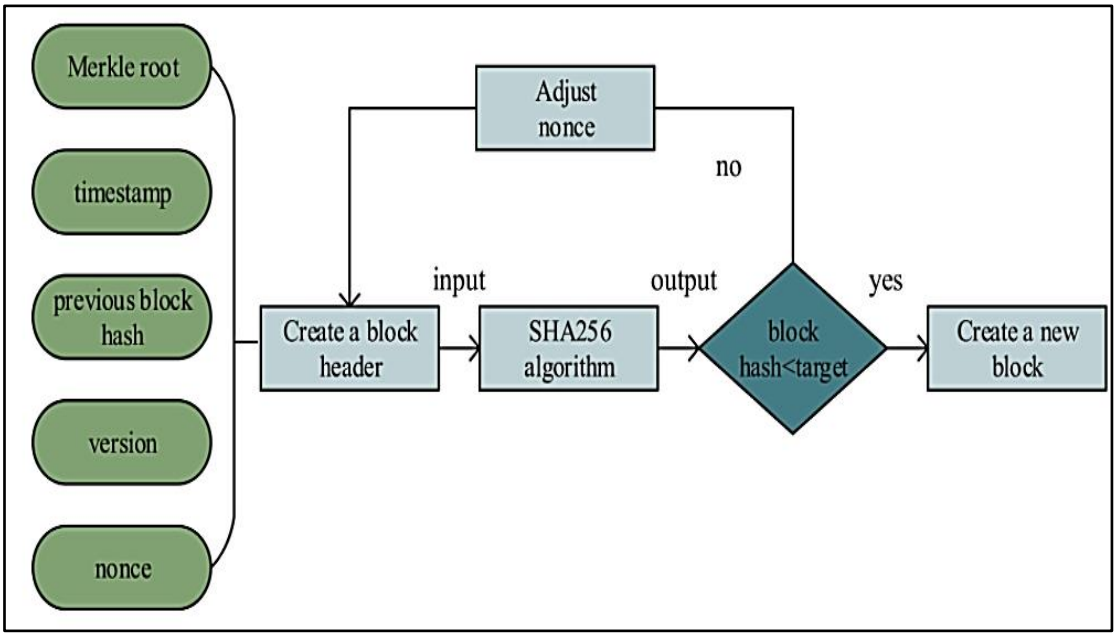

Fig. 3. PoW working principle.

2.1.2. Proof of Stake (PoS)

PoS offers a revolutionary approach to reaching consensus in decentralized networks, aiming to address energy-use constraints [27]. Figure 4 shows the PoS method's working principle. It works by granting users varying degrees of power over the consensus process according to how much cryptocurrency they stake. In order to boost their chances of being selected to build a new block, validators lock some of their assets as collateral. Fair participation is encouraged by the selection method, which is based on stake-weighted randomness rather than processing power. Because they risk losing their staked assets if they act maliciously, validators are incentivized to perform honorably. The new block is appended to the blockchain as soon as the majority agrees. PoS offers a more ecologically friendly and energy-efficient substitute for PoW. By balancing participant interests with network integrity, it guarantees decentralization, security, and trust.

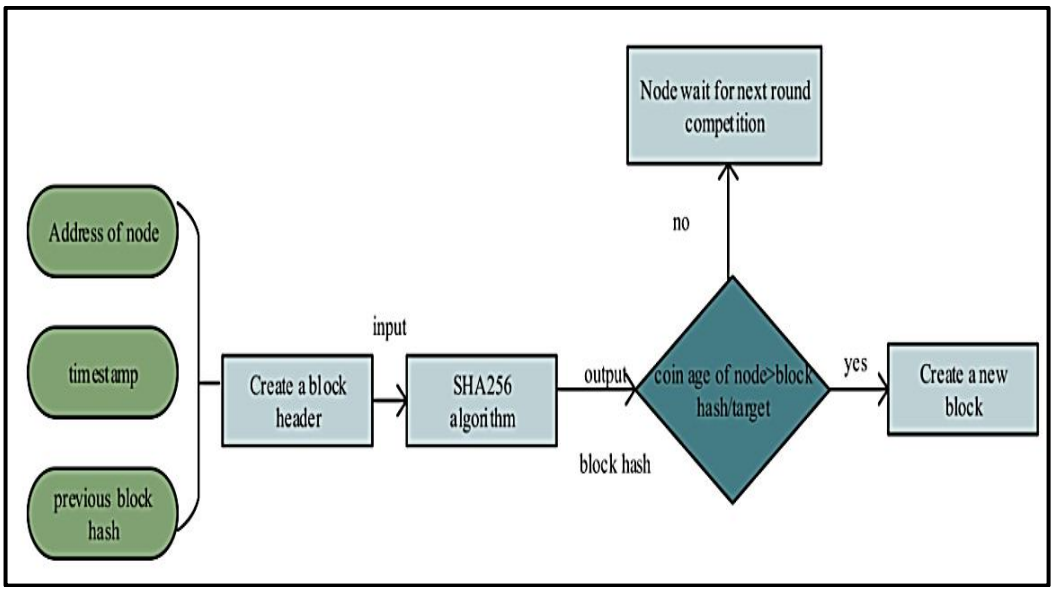

Fig. 4. PoS working principle.

2.1.3. Delegated Proof of Stake (DPoS)

DPoS reduces centralization and scalability problems that are frequently present in other consensus methods [28]. The DPoS algorithm's working principle is depicted in Fig. 5. It expands on PoS by implementing a delegation mechanism in which token owners select a select few dependable delegates. The order and frequency of these delegates' block creation and transaction verification are usually set by the protocol. DPoS balances efficiency and decentralization by lowering the number of active validators. The system uses a deterministic algorithm or randomization to decide the order in which blocks are created in order to avoid power concentration. While preserving network scalability, this structure guarantees equity. In addition to encouraging a decentralized decision-making process, the delegation process increases the effectiveness of consensus.

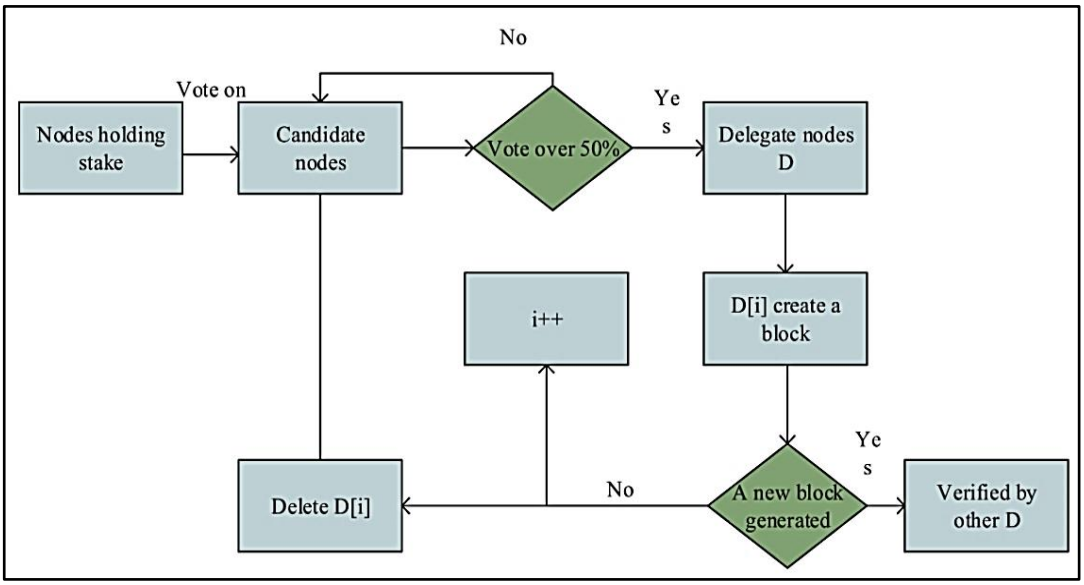

Fig. 5. DPoS working principle.

The benefits and drawbacks of each algorithm are compared in Table 1 according to resource consumption, level of centralization, throughput, and transaction confirmation time.

Table 1. Comparing the benefits and drawbacks of the three consensus algorithms.

Protocols	Benefits	Drawbacks
PoW	- Ensures security and stability - Offers a high level of node flexibility - Provides a high level of decentralization - Utilizes an open node network	- Limited scalability - Poor performance - Resulting in wastage of hardware resources
PoS	- High degree of decentralization - Open node system	- Elaborate implementation procedure - Unauthorized access or intrusion into a secure system
DPoS	- High performance - Finality	- Level of decentralization is low - Closed node system

2.2. Environment Setup

The transaction throughput of the three technologies will be examined and compared by implementing them on the real network. Real networks provide valuable insights into the performance of consensus algorithms in real-world scenarios, which contain network congestion, volatility of transaction amounts, and real consumer interactions. The real networks, shown in Fig. 6, will be used to evaluate the block size and execution time using the Remix IDE.

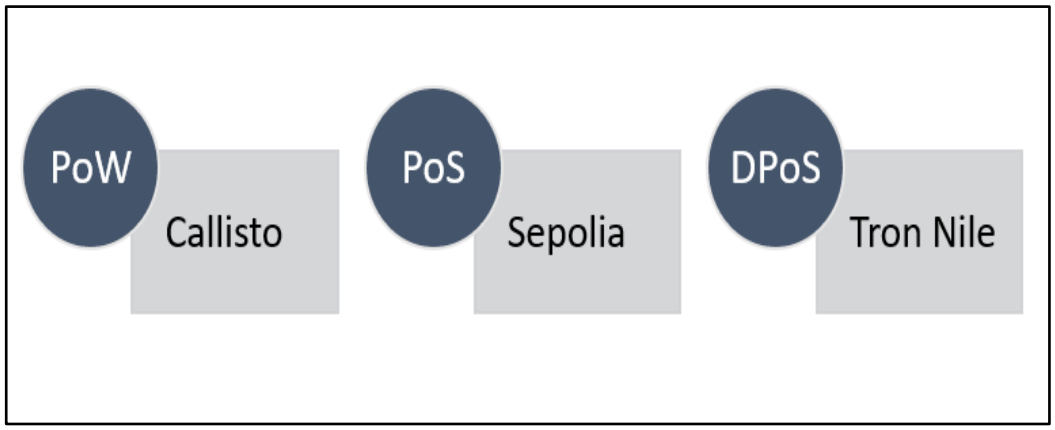

Fig. 6. Proposed real network for each algorithm.

Callisto serves as the PoW enforcement environment to create a robust, distributed network in which miners donate computational resources to authenticate transactions with maximum security. At Sepolia the selection of validators is determined based on the amount of stake they have which allows the performance of PoS to be verified in a real environment. In the Tron Nile network, delegates are selected to validate transactions and create blocks through a voting process, providing exposure to DPoS dynamics in a real-world environment using read block length and execution speed.

2.3. Initialization of Blockchain Networks

The Remix IDE is a free online application that provides a comprehensive development environment for creating intelligent contracts made especially for Ethereum. Code modification, debugging, testing, and deployment of smart contracts on the Ethereum network are among its

features. The IDE was run via the Remix IDE website, and each network's suitable environment—including a dedicated Ethereum node—was chosen. Tron-IDE, a new IDE created especially to optimize the Tron blockchain, was also utilized.

2.4. Developing Smart Contracts on Real Networks

In order to verify the results of consensus methods on the ability of smart contracts, each protocol is deployed on real network. Figure 7 depicts the experimental procedures carried out using the Remix IDE to reveal the complexities of contract implementation, with particular emphasis on speed and complexity.

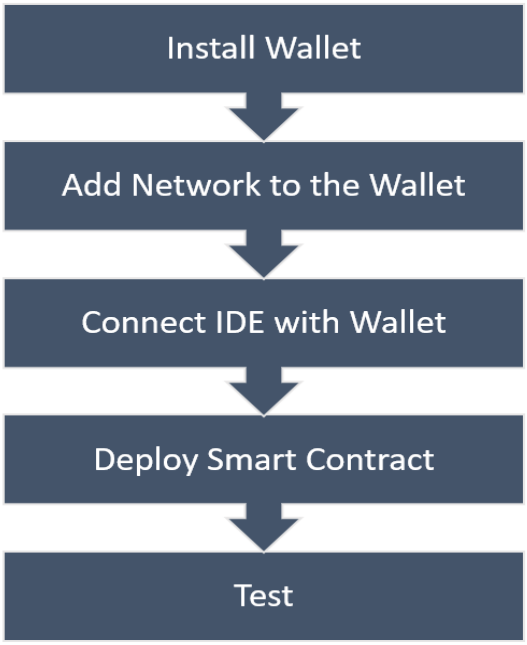

Fig. 7. Smart contract development procedures.

2.4.1. Install Wallet

Before using blockchain networks and decentralized applications, it is essential to install a wallet. A wallet is an electronic tool that lets users manage their cryptographic keys, which are necessary to access and manage their bitcoin holdings on the blockchain. A blockchain-based wallet has been selected, set up, and configured. As seen in Fig. 8, the solidity smart contract is created as a basic implementation of the Ethereum blockchain's wallet contract.

```
pragma solidity ^0.8.0;

// Define the contract
contract Mywallet {
    // The address of the contract owner
    address payable public  owner;

    // Set the creator of the contract as the owner
    constructor() {
        owner = payable(msg.sender);
    }
```

Fig. 8. Install a wallet that uses blockchain technology.

On the Ethereum network, the Solidity smart contract is an example of a basic cryptocurrency wallet. Nile network users can interact with decentralized applications built on the Tron blockchain through the TronLink wallet, a browser extension. It also provides users with a secure way to connect with the Tron ecosystem and manage TRX, the native coin of Tron. Main components and aspects included in the contract:

- Owner variable is included, which identifies the person who initially created the contract (publisher) as the initial owner with a function created to transfer ownership when needed.
- Receiver functionality to enable nodes to accept Ether, allowing users to transfer funds directly to the node.
- Deposit function to enable users to transfer Ether to the contract, provided that the contract balance and the deposited amount are positive.
- Withdrawal function to provide the owner with the ability to initiate withdrawals, ensuring that the required amount does not exceed the contract balance.
- Transfer function to facilitate the transfer of Ether to specific addresses. This function includes checks to ensure that the transfer amount is positive and within the contract balance.
- Balance inquiry function to enable users and external entities to retrieve the current balance of the contract.

The contract also contains many basic data necessary to apply the rules that ensure safe and valid transactions. These conditions include verifying the caller's ownership, ensuring that sufficient funds are available in the contract, and verifying the accuracy of the transfer amounts.

2.4.2. Add Network to the Wallet

To handle the blockchain, the network is integrated into the wallet and the MetaMask wallet is used that runs on the Ethereum network. Different blockchains have distinct networks, each with their own exclusive parameters, including the RP endpoint and chain ID.

2.4.3. Connect IDE with Wallet

The Remix IDE is linked to MetaMask to create a connection between the wallet and the integrated development environment (IDE) in order to effectively install and interact with smart contracts on the blockchain.

2.4.4. Deploy Smart Contract

The process of deploying a smart contract to a blockchain involves several steps. The smart contract is compiled into byte code once it has been created in a language that works with the chosen blockchain platform (in this case, Solidity for Ethereum). The platform's libraries and development tools are set up concurrently with the creation of the smart contract code, and the blockchain network to which it will be deployed is defined.

2.4.5. Test

The practical implications of each consensus mechanism for running smart contracts are tested. With an emphasis on the crucial interaction between theoretical ideas and their real-world implementation, this paper offers a realistic look at the practical implications of consensus mechanisms in the context of blockchain development. Exploring the intricacies of Sepolia, Callisto, and Tron Nile reveals a variety of possible consequences that could affect the creation of decentralized applications.

2.5. Evaluation

The following elements were the main focus of this study due to their significance and influence on overall performance:

- Block Time: How long does it typically take to add a new block to the blockchain? One important determinant of how quickly transactions is verified and added to the distributed ledger is the block time. Faster transaction confirmations are the outcome of shorter block

periods; however, the network's overall security may be jeopardized. However, higher block periods can result in slower transaction processing even while they improve security.

- Deploy Gas Fee: This refers to the costs of the cryptocurrency (gas) required for a smart contract to be published or a transaction to be carried out on the blockchain. Miners or validators receive payment for their computational efforts in the form of a transaction fee. Users and developers benefit from low gas costs since they reduce the costs of implementing and interacting with smart contracts. Increased demand for blockchain resources or consensus techniques that need more resources can be connected to higher gas prices.
- Block Gas Limit: The maximum amount of gas that can fit inside the mass. The amount of computing power required to execute transactions or smart contracts is measured by a metric called a block gas. By allowing more complex transactions or calculations to be included in each block, the maximum gas value per block enhances the network's overall performance. On the other hand, the minimal restrict limits the complexity or range of transactions that can be completed in a single block.
- Block Size: This refers to an unmarried block's ability to store data. Bytes are used as the unit of measurement, and this includes smart contract code, transaction records, and other information. Blocks with larger sizes have a greater potential to handle transactions or complex smart contracts, but they may also cause longer propagation delays. Lowering block sizes, on the other hand, results in faster block propagation but limits the quantity and complexity of transactions that can be included.

3. Results and Discussion

The results of contract deployment across three test networks using different consensus procedures are shown in Table 2. While Sepolia and Tron Nile have shorter block-to-block times but higher transaction costs, Callisto has longer block-to-block times but lower transaction costs.

Table 2. Comparing the results of the three consensus algorithms.

	POW	**POS**	**DPOS**
Test Network	Callisto	Sepolia	Tron Nile
Block Time	20 sec	7 sec	2 sec
Deploy Gas Fee	0.55294618CLO= 0.000383usd	0.00227106ETH= 5.33usd	216.57186TRX= 22.90usd
Block Gas Limit	8,000,000	30,000,000	400,000,000
Block Size	2,963 bytes	97,009 bytes	3,213 Bytes

Performance characteristics of any test network can be influenced considerably by the consensus methods being utilized. For the test networks under consideration, Block Time shows that the speed of Tron Nile is quite impressive of two seconds per block, while Callisto emphasizes security as the block time is longer, taking 20 seconds, and Sepolia is in the middle at 7 seconds, as shown in Fig. 9.

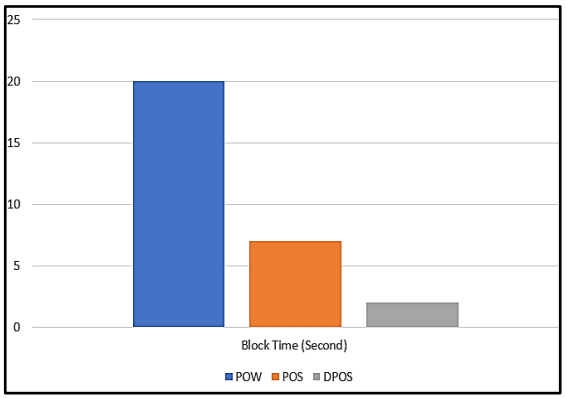

Fig. 9. Blocking Time for the three techniques.

As for Deploy Gas Fee, Callisto has the least cost, charging only 0.000383 USD which is quite economical for transacting, while Sepolia's moderate cost of 5.33 USD is justifiable in terms of PoS costs, and high charges of DPoS on Tron Nile of 22.90 USD can also be explained by the resource requirements, as shown in Fig. 10.

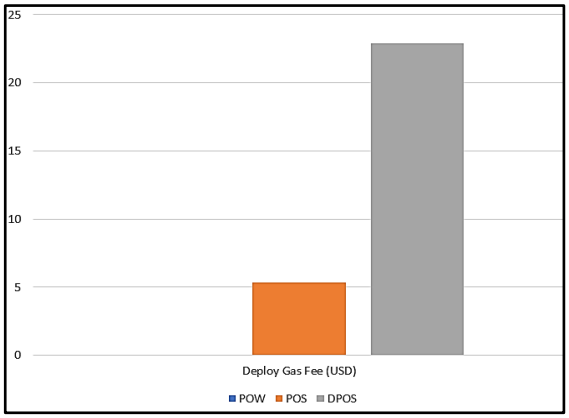

Fig. 10. Deploy gas fee for the three techniques.

For Block Gas Limit, Tron Nile didn't disappoint with 400,000,000 a vast amount showing an edge in the number of transactions, as opposed to the 30,000,000 placed against Sepolia, while Callisto remained low with 8,000,000, as shown Fig. 11.

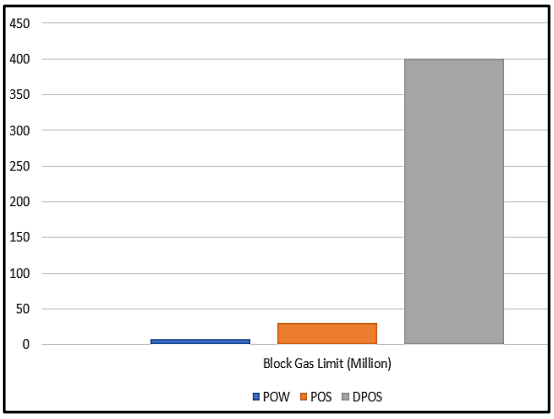

Fig. 11. Block gas limit for the three techniques.

The block size in Fig. 12 indicates Sepolia's capacity at 97,009 bytes, an average size for Tron Nile at 3,213 bytes, and 2,963 bytes for Callisto, which at best enhances processing capacity. From these parameters, it can be seen how different networks balance speed, cost, capacity and security.

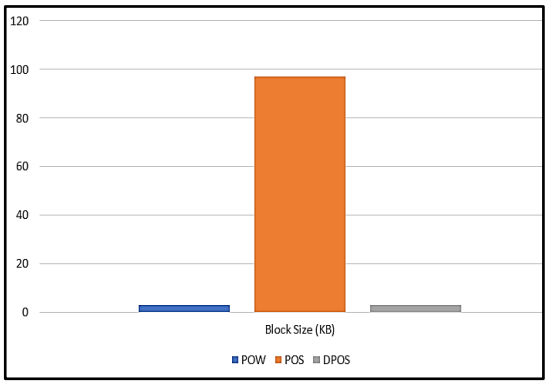

Fig. 12. Block size for the three techniques.

Tron Nile, Sepolia, and Callisto develop different fortes which respond to different project requirements. For instance, Tron Nile is beneficial to applications that require speed and efficiency because it has a low 2-second blocktime combined with a high gas limit of 400 million. On the contrary, it does have some drawbacks as it has high gas fees which could be a turn down for some projects. Sepolia on the other hand has a reasonable equilibrium of speed, fees, and capacity with its 7 second block time, reasonable gas fee of 5.33 USD and a 30 million gas limit which will allow it to be used in places where there is a compromise between cost and performance. In comparison, Callisto is least effective for applications requiring speed and scalability as it has high block time of 20 seconds and 8 million block gas limits however it has the lowest gas fee of 0.000383 USD.

There are certain parameters which can influence the choice of a particular protocol. For applications that can potentially benefit from speedy confirmations, DPOS as used by Tron Nile will be efficient as it registers fast transaction confirmations. On the other hand, for applications where cost matters, POW as Callisto has demonstrated does help in keeping the cost of deployment low. On the contrary, POS as evident from the use of Sepolia will give a better option in terms of security while providing reasonable performance. Furthermore, the use of Sepolia is further enhanced by the extensive block size of 97,009 bytes designated for use in data-intensive or high-activity transactions.

Deciding which protocol to use comes down to a set of decisions and selection depending on the aims of the particular project. Both voting systems as well as design models change the final set of attributes, hence achieving special requirements for their specific application. Based on these characteristics, the developers are able to pick the most suitable blockchain which fulfills their specific needs.

4. Conclusion

This study emphasizes how crucial consensus algorithms are in determining how well blockchain networks and decentralized applications operate. Through a comparative investigation of Proof of Work (PoW), Proof of Stake (PoS), and Delegated Proof of Stake (DPoS) on real-world test networks, the paper reveals how each technique addresses critical factors like as security, scalability, and efficiency. PoS and DPoS provide notable benefits in energy efficiency and transaction throughput, whereas PoW is superior in security and decentralization. The results emphasize that the consensus algorithm selection should be in line with the particular goals and specifications of a project. Comparative analysis can be expanded to include analysis of elements of the user experience, such as the time it takes to receive confirmation and the level of complexity in user interactions, in order to understand the practical consequences for participants in each network.

5. References

[1] D. Berdik, S. Otoum, N. Schmidt, D. Porter, and Y. Jararweh, "A survey on blockchain for information systems management and security," Information Processing & Management, vol. 58, no. 1, pp. 102397, Jan. 2021. DOI: 10.1016/j.ipm.2020.102397.

[2] M. Javaid, A. Haleem, R. P. Singh, S. Khan, and R. Suman, "Blockchain technology applications for Industry 4.0: A literature-based review," Blockchain: Research and Applications, vol. 2, no. 4, pp. 100027, Dec. 2021. DOI: 10.1016/j.bcra.2021.100027.

[3] I. Eyal, A. E. Gencer, E. G. Sirer, and R. Van Renesse, "Bitcoin-NG: A scalable blockchain protocol," in 13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16), Santa Clara, CA, 2016, pp. 45–59. DOI: 10.5555/2930611.2930616.

[4] S. Alsaqqa and S. Almajali, "Blockchain technology consensus algorithms and applications: A survey," 2020. DOI: 10.36227/techrxiv.13267398.v1.

[5] S. Bano, A. Sonnino, M. Al-Bassam, S. Azouvi, P. McCorry, S. Meiklejohn, and G. Danezis, "SoK: Consensus in the age of blockchains," in Proceedings of the 1st ACM Conference on Advances in Financial Technologies, Zurich, Switzerland, Oct. 2019, pp. 183–198. DOI: 10.1145/3318041.3355458.

[6] B. Liskov, "From viewstamped replication to Byzantine fault tolerance," in Replication: Theory and Practice, Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 121–149. DOI: 10.1007/978-3-642-11294-3_6.

[7] M. Jakobsson and A. Juels, "Proofs of work and bread pudding protocols," in Secure Information Networks: Communications and Multimedia Security IFIP TC6/TC11 Joint Working Conference on Communications and Multimedia Security (CMS'99), Leuven, Belgium, Sep. 1999, pp. 258–272. Boston, MA: Springer US. DOI: 10.1007/978-0-387-35568-6_19.

[8] Y. Wei, Q. Xu, and H. Peng, "An enhanced consensus algorithm for blockchain," Scientific Reports, vol. 14, no. 1, pp. 17701, 2024. DOI: 10.1038/s41598-024-24324-7.

[9] K. Wu, Y. Ma, G. Huang, and X. Liu, "A first look at blockchain-based decentralized applications," Software: Practice and Experience, vol. 51, no. 10, pp. 2033–2050, Oct. 2021. DOI: 10.1002/spe.3037.

[10] S. Fahim, S. K. Rahman, and S. Mahmood, "Blockchain: A comparative study of consensus algorithms PoW, PoS, PoA, PoV," International Journal of Mathematical Sciences and Computing, vol. 3, pp. 46–57, 2023. DOI: 10.5815/ijmsc.2023.03.04.

[11] S. Das, J. Rout, R. Priyadarshini, and M. Mishra, "A comparative analysis of the consensus algorithms in blockchain technology," in Proceedings of the International Conference on Innovative Computing & Communication (ICICC) 2022, 2022. DOI: 10.2139/ssrn.4091413.

[12] M. S. Ferdous, M. J. M. Chowdhury, M. A. Hoque, and A. Colman, "Blockchain consensus algorithms: A survey," arXiv preprint, arXiv:2001.07091, 2020. DOI: 10.48550/arXiv.2001.07091.

[13] L. Ge, J. Wang, and G. Zhang, "Survey of consensus algorithms for proof of stake in blockchain," Security and Communication Networks, vol. 2022, pp. 1–13, 2022. DOI: 10.1155/2022/4356437.

[14] S. Alam, "The current state of blockchain consensus mechanism: Issues and future works," International Journal of Advanced Computer Science and Applications, vol. 14, no. 8, 2023. DOI: 10.14569/IJACSA.2023.0140872.

[15] S. Kaur, S. Chaturvedi, A. Sharma, and J. Kar, “A research survey on applications of consensus protocols in blockchain,” Security and Communication Networks, vol. 2021, pp. 1–22, 2021. DOI: 10.1155/2021/8824738.

[16] Q. Wang, J. Huang, S. Wang, Y. Chen, P. Zhang, and L. He, “A comparative study of blockchain consensus algorithms,” in Journal of Physics: Conference Series, vol. 1437, no. 1, p. 012007. IOP Publishing, 2020. DOI: 10.1088/1742-6596/1437/1/012007.

[17] S. Fahim, S. K. Rahman, and S. Mahmood, “Blockchain: A comparative study of consensus algorithms PoW, PoS, PoA, PoV,” International Journal of Mathematical Sciences and Computing, vol. 3, pp. 46–57, 2023. DOI: 10.5815/ijmsc.2023.03.04.

[18] V. Gramoli, “From blockchain consensus back to Byzantine consensus,” Future Generation Computer Systems, vol. 107, pp. 760–769, June 2020. DOI: 10.1016/j.future.2020.02.020.

[19] G. Zhang, F. Pan, Y. Mao, S. Tijanic, M. Dang’ana, S. Motepalli, ... and H. A. Jacobsen, “Reaching consensus in the Byzantine empire: A comprehensive review of BFT consensus algorithms,” ACM Computing Surveys, 2022. DOI: 10.1145/3494536.

[20] A. Ahmad, A. Alabduljabbar, M. Saad, D. Nyang, J. Kim, and D. Mohaisen, “Empirically comparing the performance of blockchain's consensus algorithms,” IET Blockchain, vol. 1, no. 1, pp. 56–64, 2021. DOI: 10.1049/iet-blc.2020.0012.

[21] M. S. Ferdous, M. J. M. Chowdhury, and M. A. Hoque, “A survey of consensus algorithms in public blockchain systems for crypto-currencies,” Journal of Network and Computer Applications, vol. 182, pp. 103035, Mar. 2021. DOI: 10.1016/j.jnca.2021.103035.

[22] M. Foti, C. Mavromatis, and M. Vavalis, “Decentralized blockchain-based consensus for Optimal Power Flow solutions,” Applied Energy, vol. 283, pp. 116100, Feb. 2021. DOI: 10.1016/j.apenergy.2020.116100.

[23] Z. Auhl, N. Chilamkurti, R. Alhadad, and W. Heyne, “A Comparative study of consensus mechanisms in blockchain for IoT networks,” Electronics, vol. 11, no. 17, pp. 2694, Sep. 2022. DOI: 10.3390/electronics11172694.

[24] M. Pineda, D. Jabba, W. Nieto-Bernal, and A. Pérez, “Sustainable Consensus Algorithms Applied to Blockchain: A Systematic Literature Review,” Sustainability, vol. 16, no. 23, pp. 10552, Dec. 2024. DOI: 10.3390/su162310552.

[25] M. Anus and A. B. Ngadi, “Comparison Analysis of Blockchain Consensus Algorithms in Decentralized Public Environment: A Review,” Asia Proceedings of Social Sciences, vol. 12, no. 1, pp. 108–112, 2024. DOI: 10.31580/apss.v12i1.268.

[26] H. Jung and H. N. Lee, “ECCPoW: Error-correction code based proof-of-work for ASIC resistance,” Symmetry, vol. 12, no. 6, pp. 988, June 2020. DOI: 10.3390/sym12060988.

[27] Y. Shifferaw and S. Lemma, “Limitations of proof of stake algorithm in blockchain: A review,” Zede Journal, vol. 39, no. 1, pp. 81–95, 2021. DOI: 10.4314/zj.v39i1.7.

[28] S. M. S. Saad and R. Z. R. M. Radzi, “Comparative review of the blockchain consensus algorithm between proof of stake (PoS) and delegated proof of stake (DPoS),” International Journal of Innovative Computing, vol. 10, no. 2, 2020. DOI: 10.11113/ijic.v10n2.212.

Journal of Basrah Researches (Sciences) **50**(2), 267 (2024)
DOI: https://doi.org/10.56714/bjrs.50.2.23

تقييم خوارزميات الإجماع لتقنية Blockchain لتحسين التطبيقات اللامركزية

سجى عدنان عامر

قسم هندسة الحاسوب والاتصالات، كلية الهندسة وعلوم الحاسوب، الجامعة الأميركية للعلوم والتكنولوجيا، بيروت، لبنان.

معلومات البحث

الاستلام 2 تشرين الثاني 2024
المراجعة 10 كانون الاول 2024
القبول 16 كانون الاول 2024
النشر 31 كانون الأول 2024

الكلمات المفتاحية

Blockchain, التطبيقات اللامركزية, Proof of Work, Proof of Work Delegated, Proof of Stake

الملخص

تتطلب الوتيرة المتسارعة لاعتماد التطبيقات اللامركزية استخدام بنيات تحتية لسلسلة الكتل تتسم بالكفاءة وعالية الأداء. يتم دعم سلاسل الكتل هذه من خلال خوارزميات الإجماع التي تعتبر محددات حاسمة لقابلية التوسع وسرعة المعاملات والتكاليف والأمان. لا يزال المطورون غير مدركين للخيارات الأكثر فائدة بسبب وجود فجوة كبيرة في المعلومات المتعلقة بالتقييم الشامل لهذه الخوارزميات في الممارسة العملية. تهدف هذه الدراسة إلى اختبار ومقارنة أداء بروتوكولات توافق blockchain الثلاثة الأكثر شيوعًا - Proof of Work (PoW)، وProof of Work (PoS) وProof of Stake (DPoS) بهدف تحسين تطبيق الأنظمة اللامركزية. ومن خلال نشر العقود الذكية على شبكات اختبار blockchain حقيقية (Callisto for PoW، وSepolia for PoS، وTron Nile for DPoS)، يستكشف البحث مقاييس الأداء الرئيسية مثل وقت الكتلة، وكلفة نشر العملة، والحد الأقصى لكلفة الكتلة، وحجم الكتلة. تستخدم التجارب أدوات مثل Remix IDE وTron-IDE، لتسليط الضوء على الآثار العملية لخوارزميات الإجماع في ظل ظروف العالم الحقيقي المختلفة، بما في ذلك ازدحام الشبكة وتقلب المعاملات. تتفوق Tron Nile في السرعة والسعة ولكنها تتكبد رسوم عالية، وتوازن Sepolia بين الأداء والتكاليف المعتدلة، وتؤكد Callisto على كفاءة التكلفة على حساب السرعة وقابلية التوسع. توفر الرؤى المستمدة من هذه الدراسة إرشادات قيمة للمطورين لاختيار آليات الإجماع المناسبة بناءً على المتطلبات المحددة للتطبيقات اللامركزية.

Citation: S. A. Amenr, J. Basrah Res. (Sci.) **50**(2), 267 (2024). DOI:https://doi.org/10.56714/bjrs.50.2.23

***Corresponding author email :** Sajaadnan2018@gmail.com

ISSN: 1817-2695 (Print); 2411-524X (Online)
Online at: https://jou.jobrs.edu.iq