Penerbit Akademia Baru

Journal of Advanced Research in Computing and Applications

Journal homepage:
https://akademiabaru.com/submit/index.php/arca/index
2462-1927


# Elevating Blockchain Consensus: A Comparative Performance Analysis on Proof of Stake, Delegated Proof of Stake, Practical Byzantine Fault Tolerance and Casper

Sarwen Kumar Naidu Kumrasen[1], Syarifah Bahiyah Rahayu[2,3,*], Stefan Wolfgang Pickl[4]

[1] Department of Computer Science, Faculty of Defence Science and Technology, National Defence University of Malaysia (UPNM), 57000 Kuala Lumpur, Malaysia
[2] Cyber Security and Industry Revolution Digital Centre, Institute of CyberSecurity and Electronic Systems, National Defence University of Malaysia, (UPNM) 57000 Kuala Lumpur, Malaysia
[3] Department of Defence Science, Faculty of Defence Science and Technology, National Defence University of Malaysia, 57000 Kuala Lumpur, (UPNM) Malaysia
[4] Department of Computer Science, University of Bundeswehr Munich, Neubiberg, 85579, Germany

**ARTICLE INFO**

***Article history:***
Received 23 September 2025
Received in revised form 20 November 2025
Accepted 21 November 2025
Available online 9 December 2025

***Keywords:***
Consensus Algorithms; Decentralization; Latency; Security; Throughput; Blockchain

**ABSTRACT**

The growth of Blockchain technology has fundamentally impacted how digital transactions are made by making them more secure and decentralized while preserving data integrity and transparency. Traditional consensus algorithms, such as Casper, Delegated Proof of Stake (DPoS), Proof of Stake (PoS), and Practical Byzantine Fault Tolerance (PBFT), have several drawbacks, including inadequate network scalability, insufficient security, and inefficient transaction processing. By optimizing the existing algorithms Proof of Stake Plus (PoS+), Randomized Delegated Proof of Stake (RDPoS), Flexible Byzantine Fault Tolerance (FBFT), and Casper + Secure, this study on blockchain consensus methods seeks to increase network performance, decentralization, and reliability. A structured methodology is designed to conduct data gathering, algorithm design, implementation, and evaluation. The evaluation is based on throughput and latency. The findings shows PoS+ significantly outperforms PoS resulting in higher throughput and lower latency. Due to the dynamic delegate selection process, RDPoS outperforms DPoS in terms of fairness, decentralization, and transaction speed. However, FBFT performs the poorest, making it unsuitable for settings where quick transactions are common. Casper+ Secure preserves the trade-off between security and performance despite its reduced throughput. Future study is to build self-adaptive blockchain networks with enhanced versatility, resilience, and decision-making.

## 1. Introduction

Blockchain technology has significantly transformed the digital landscape by upgrading from traditional centralized systems to decentralized, secure networks. An essential component of this

* *Corresponding author.*
*E-mail address: syarifahbahiyah@upnm.edu.my*

https://doi.org/10.37934/arca.41.1.112128

technology is the consensus algorithms, which guarantee that all users in a distributed network agree on the ledger's current state. Blockchain has also gained traction as a foundation for cybersecurity applications, especially where data integrity is critical [1]. Blockchain continues to evolve as a secure, append-only distributed ledger used across industries ranging from finance to logistics [2]. Without a central authority, these algorithms allow nodes to work together efficiently while preserving the integrity and validity of transactions [3]. Despite its revolutionary transformations, blockchain technology faces significant obstacles in scalability, energy consumption, and security concerns with current consensus algorithms. Consensus algorithms generally serve as the decision-making backbone of distributed ledgers, ensuring agreement among nodes in the absence of central authority [4]. A consensus algorithm can be broadly defined as any mechanism that enables distributed nodes to produce a unified state agreement [5].

There are drawbacks to the consensus algorithms used today, such as Proof of Stake (PoS), Delegated Proof of Stake (DPoS), Practical Byzantine Fault Tolerance (PBFT), and Casper. PoS is recognized for its improved scalability and lower energy usage, but because nodes with greater stakes have more influence, it has centralization problems and security threats [6]. Economic analyses suggest that PoS systems behave differently under rational participation, influencing validator strategies and network stability [7]. Comparative analyses continue to underscore unresolved challenges in current implementations, particularly regarding decentralization and validator fairness [8]. DPoS increases transaction speed and scalability, but due to fewer delegates and the risk of overlapping roles, it puts centralization in danger [9]. To address these issues, improved versions of DPoS have been proposed, introducing more transparent and adaptive delegate selection mechanisms [10]. PBFT is a better fit for permissioned blockchain networks because of its excellent security and fault tolerance, but scalability issues [11]. Casper is currently in progress and has unresolved implementation and security issues. It combines PoW and PoS to improve security issues and scalability [12]. Recent reviews also emphasize that choosing an appropriate consensus mechanism depends heavily on the target application environment and its performance constraints [13].

This study aims to improve the consensus algorithms PoS+, RDPoS, FBFT, and Casper + Secure based on the limitations of traditional consensus algorithms, such as PoS, DPoS, PBFT, and Casper to improve blockchain security. The study also aims to assess the performance of consensus algorithms in terms of latency and throughput. To guarantee these novel mechanisms' applicability in real-world scenarios and enhance blockchain technology's general resilience and effectiveness, it is intended to assess them based on latency and throughput [14].

The motivation of this study is to improve the latency and throughput of traditional consensus algorithms. The objective is to introduce new consensus algorithms that enhance performance and scalability in blockchain networks. Comprehending the findings of consensus algorithms is essential to developing blockchain systems that are more robust and effective. For example, PoS and DPoS increase energy efficiency and scalability but come with centralization and security problems. Even with its high level of security, PBFT has scaling issues, which makes it less appropriate for open networks. A number of comparative studies also categorize consensus mechanisms by performance, complexity, and suitability for permissioned or public networks [15]. Emerging studies further highlight that hybrid consensus frameworks combined with machine-learning-based anomaly detection can significantly strengthen blockchain security [16]. Although promising, Casper's reliability and security have not yet been thoroughly verified in real-world settings, which raises questions. Some authors argue that widely adopted protocols obscure underlying design weaknesses that impact transaction ordering and fairness [17]. Machine-learning-oriented approaches such as anomaly detection have also been suggested to enhance validator behaviour monitoring [18].

These realizations highlight how difficult it is to create consensus algorithms that strike a balance between efficiency, scalability, and security. Blockchain platforms also face recurring security threats such as double spending, node compromise, and incentive manipulation [19]. The results point to the necessity of ongoing innovation and study to get over these obstacles and improve the functionality of blockchain networks. This research is significant in the advancement of blockchain technology by addressing the inherent limitations of existing algorithms and focusing on practical performance metrics. The consensus algorithm's security and dependability of blockchain technology is a major aspect for a wide range of applications across industries, including healthcare and finance [20].

## 2. Methodology

The study methodology takes a methodical approach to investigating and improving blockchain consensus algorithms. Data collection, algorithm design, implementation, and evaluation make up its four primary stages. The data-gathering stage employs a thorough literature review to analyse existing studies and research papers on blockchain technology, consensus algorithms, and their applications. The primary focus is on identifying limitations in current consensus algorithms such as PoS, DPoS, PBFT, and Casper. Key topics reviewed include performance metrics, security issues, and optimization strategies for consensus algorithms. This stage is crucial for understanding the gaps in current knowledge and narrowing down specific focus areas for the study.

In the Algorithm Design phase, the architecture of the enhanced consensus algorithm is developed. This includes designing block diagrams and pseudocode to outline the proposed system. The design addresses the limitations identified in the data-gathering stage, focusing on solutions to enhance efficiency, security, and scalability. The pseudocode for each proposed algorithm is meticulously crafted to capture the nuances of the improved consensus mechanism.

The block diagram in Figure 1 illustrates the overall process of the study, starting with identifying limitations in existing consensus algorithms (PoS, DPoS, PBFT, and Casper). A structure is developed to address these limitations, followed by a comprehensive performance evaluation to assess the proposed algorithm's efficiency in terms of latency, throughput, and security. Data collected is structured into CSV format for subsequent comparison analysis, enabling a detailed study of the enhancements.

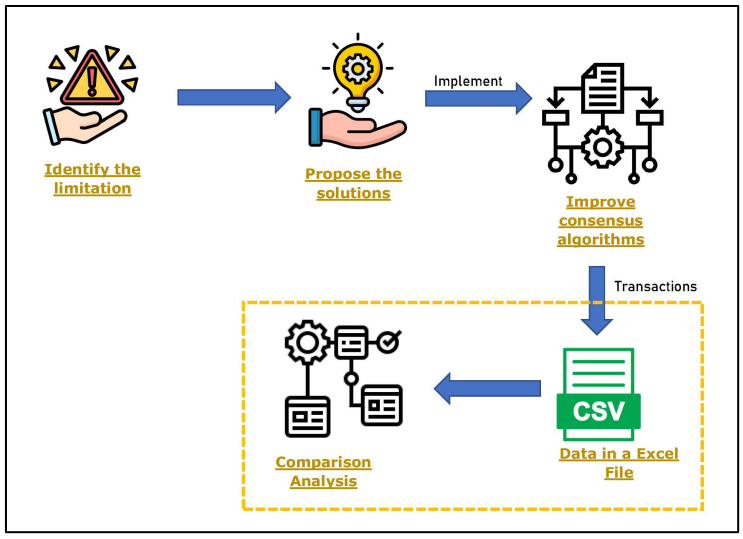


**Fig. 1.** Block Diagram of The System Architecture

### *2.1 Proposed Consensus Algorithms*

By optimizing the existing algorithms Proof of Stake Plus (PoS+), Randomized Delegated Proof of Stake (RDPoS), Flexible Byzantine Fault Tolerance (FBFT), and Casper + Secure, this study on blockchain consensus methods seeks to increase network performance, decentralization, and reliability. Below are the proposed consensus algorithms:

A. Algorithm workflow for proof of stake plus (PoS+)

The PoS+ Algorithm Workflow in Figure 2 outlines a blockchain represented by an instance of the PoS class, with validators stored in an array. Validators are added and sorted by stake, and transactions are processed by randomly selected validators. This Algorithm Workflow addresses the initial distribution and nothing-at-stake attacks by implementing checks for validator eligibility and fair distribution.

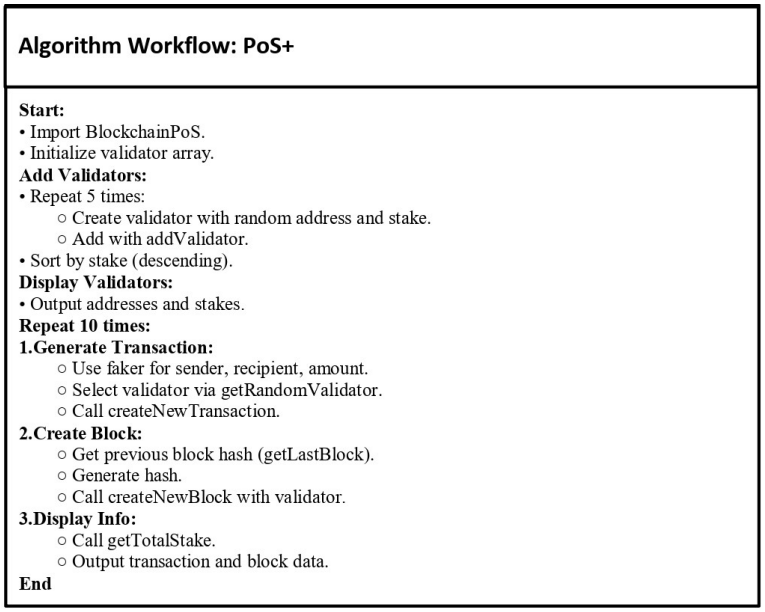


**Fig. 2.** Algorithm Workflow for PoS+

B. Algorithm workflow for Randomized Delegated Proof Of Stake (RDPoS)

The RDPoS Algorithm Workflow in Figure 3 initializes delegates and a blockchain instance, adding delegates with unique stakes and addresses. A dynamic selection algorithm randomly chooses a delegate from the top performers for each transaction, enhancing transparency and reducing centralization risks.

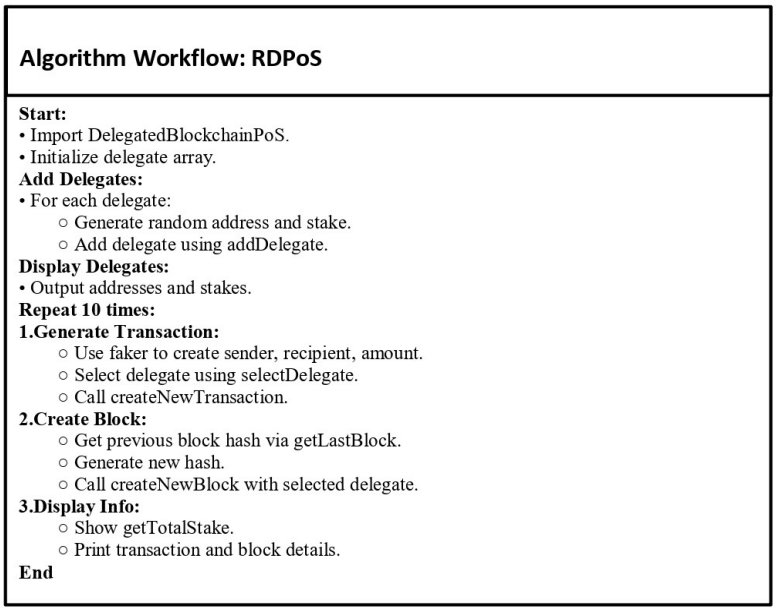


**Fig. 3.** Algorithm Workflow for RDPoS

C. Algorithm workflow for Flexible Byzantine Fault Tolerance (FBFT)

The FBFT Algorithm Workflow in Figure 4 begins with initializing replicas and assigning stakes, followed by dynamic management of replicas and computing total stakes. This approach addresses network assumptions and high resource requirements, providing flexibility through flags for specific use cases.

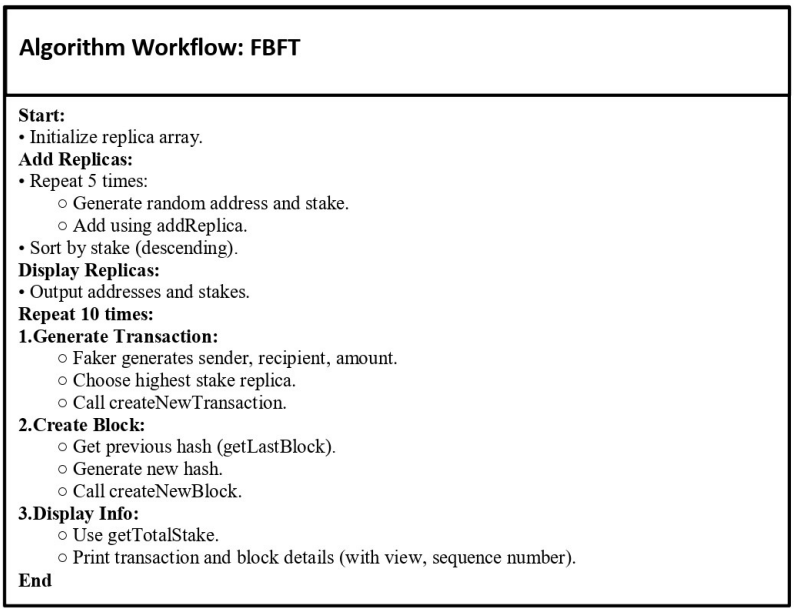


**Fig. 4.** Algorithm Workflow for FBFT

D. Algorithm workflow for Casper + Secure

Earlier implementations of Casper focused on introducing the Friendly Finality Gadget to improve validator safety and block finalization [21]. The Casper + Secure Algorithm Workflow in Figure 5 includes initializing validators and implementing a slashing mechanism to penalize malicious

behavior. Transactions are processed by validators ranked by stake, with dynamic thresholds for selection, enhancing security and reducing exploitation risks.

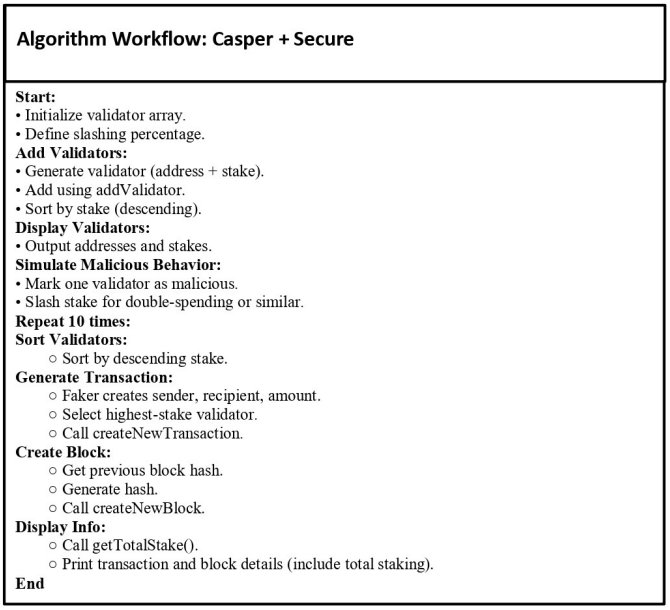


**Fig. 5.** Algorithm Workflow for Casper + Secure

*2.2 Implementation of Proposed Consensus Algorithms*

The Implementation phase translates the pseudocode into functional programming modules for the proposed algorithms: PoS+, RDPoS, FBFT, and Casper + Secure. This phase addresses drawbacks identified in traditional algorithms, enhancing security, fairness, and efficiency. The enhanced algorithms are tested with transaction datasets of varying sizes, generating CSV files for evaluation.

The PoS+ algorithm addresses the initial distribution and nothing-at-stake attacks by ensuring fair distribution of validators and implementing checks for transaction eligibility. Key improvements include:

- Add the first validator if none exists, without stakeholder concerns.
- Adding new validators only if their stake exceeds the median stake.
- Implementing a fallback method to ensure at least one validator is present.
- Checking for eligible validators and skipping transactions if none are valid.

The RDPoS algorithm introduces a dynamic delegate selection algorithm and enhanced logging for transparency and accountability. Key improvements include:

- Selecting a delegate at random from a subset of top performers.
- Logging delegate information during transaction and block creation.
- Enhancing transparency and reducing centralization risks.

The FBFT algorithm addresses high resource requirements and network assumptions through dynamic replica management and flexible flags. Key improvements include:

- Using an object for dynamic replica management for faster lookup and removal.
- Dynamically computing total stake to avoid unnecessary iteration.
- Providing flexibility through flags for specific resource and network assumptions.

The Casper + Secure algorithm implements a slashing mechanism and dynamic validator selection. Key improvements include:

- Introducing a 10% slashing mechanism for malicious behaviour.
- Considering stakeholder and randomness during validator selection.

- Using dynamic thresholds for validator selection.
- Enhancing security and reducing exploitation risks.

*2.5 Evaluation*

The evaluation phase assesses the performance of the improved consensus algorithms by comparing them with traditional ones using key performance metrics such as latency and throughput. The algorithms are tested with transaction sets of 500, 1000, 2000, and 3000 transactions.

Latency is calculated using the following formula (Eq. 1):

$$Latency = \frac{(t_{end} - t_{start})}{T} \quad (1)$$

where $t_{end}$ is the completion time of the last transaction, $t_{start}$ is the initiation time of the first transaction, and T is the total number of transactions processed.

Throughput is measured as:

$$Throughput = \frac{I}{T} \quad (2)$$

Throughput in Eq.2 is where $I$ represent the total number of transactions, and $T$ is the duration over which the transactions were completed.

Because they accurately represent the effectiveness and efficiency of consensus algorithms in actual blockchain contexts, latency and throughput are crucial to this study. For blockchain systems to scale efficiently, manage increasing user demands, and remain responsive under high transaction loads, high throughput and low latency are crucial. These indicators are especially important for maintaining confidence and usability in applications that are time-sensitive and security-sensitive, such supply chain, healthcare, or finance. Recent studies have proposed optimized consensus frameworks such as lightweight protocols for edge devices and hybrid DAG-PBFT structures that significantly reduce latency while preserving fault tolerance, further validating the importance of selecting metrics that reflect real-world conditions [21, 22]. The study intends to assess how effectively improved consensus algorithms satisfy the real-world requirements of contemporary decentralized networks by concentrating on these two-performance metrics.

The improved algorithms are run multiple times with different transaction sets to ensure robust data for comparative analysis. Python-based tools are used for detailed analysis, focusing on latency and throughput to assess performance improvements (Fig. 6).

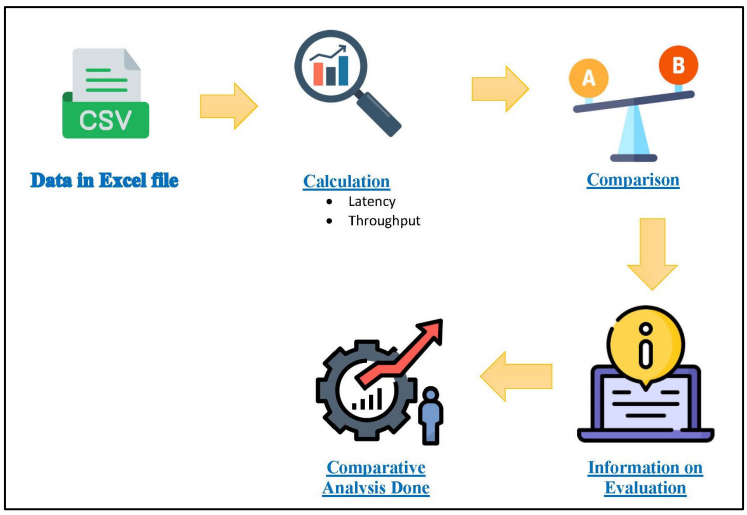


**Fig. 6.** Comparison Analysis Framework

## 3. Results

Transactional factors such as latency and throughput have a vital impact on the speed and efficiency of a of blockchain systems. Throughput refers to the number of completed transactions per unit time, while latency is the time for a transaction to be initiated and finalized. Sustaining a transactional environment that is scalable and responsive involves maintaining low latency and high throughput. Advanced consensus algorithms aim to enhance these factors to ensure secure, efficient, and scalable transaction processing.

Based on Table 1, PoS+ algorithm demonstrates consistently low average latency, ranging from 0.000594 to 0.000721 seconds, ensuring swift transaction processing. The algorithm also exhibits a robust mean throughput of 1543.8 transactions per second, highlighting its efficiency in handling diverse workloads. These results confirm PoS+ as an effective and reliable consensus algorithm suitable for real-world applications where speed and stability are critical.

**Table 1**
Average Latency And Average Throughput Of Pos+

| Metric | Transactions | Average Latency (s/transaction) | Average Throughput (transactions/s) |
|---|---|---|---|
| count | 4.000000 | 4.000000 | 4.000000 |
| mean | 1625.000000 | 0.000654 | 1543.800263 |
| std | 1108.677891 | 0.000252 | 117.033093 |
| min | 500.000000 | 0.000594 | 1398.026790 |
| 25% | 875.000000 | 0.000634 | 1508.615210 |
| 50% | 1500.000000 | 0.000651 | 1546.283238 |
| 75% | 2250.000000 | 0.000670 | 1581.470081 |
| max | 3000.000000 | 0.000721 | 1684.699606 |

Based on Table 2, RDPoS algorithm showcases impressive performance with average latency ranging from 0.000031 to 0.000047 seconds and an exceptional average throughput of 26352.07 transactions per second. These metrics underline RDPoS's capability to handle transactions swiftly and reliably, making it a potent algorithm for applications requiring low latency and high throughput.

**Table 2**
Average Latency and Average Throughput Of RDPOS

| Metric | Transactions | Average Latency (s/transaction) | Average Throughput (transactions/s) |
|---|---|---|---|
| count | 4.000000 | 4.000000 | 4.000000 |
| mean | 1625.000000 | 0.000040 | 26352.070659 |
| std | 1108.677891 | 0.000007 | 4230.161382 |
| min | 500.000000 | 0.000031 | 22656.738831 |
| 25% | 875.000000 | 0.000037 | 23989.987827 |
| 50% | 1500.000000 | 0.000041 | 25110.579359 |
| 75% | 2250.000000 | 0.000045 | 27653.737632 |
| max | 3000.000000 | 0.000047 | 32363.822356 |

Based on Table 3, FBFT algorithm exhibits consistent performance with average latency per transaction ranging from 0.002360 to 0.002816 seconds and a mean throughput of 401.35 transactions per second. Although FBFT has slightly higher latency compared to some algorithms, it balances this with dependable throughput, making it suitable for scenarios where both factors are critical.

**Table 3**
Average Latency and Average Throughput of FBFT

| Metric | Transactions | Average Latency (s/transaction) | Average Throughput (transactions/s) |
|---|---|---|---|
| count | 4.000000 | 4.000000 | 4.000000 |
| mean | 1625.000000 | 0.002525 | 401.352756 |
| std | 1108.677891 | 0.000200 | 25.492478 |
| min | 500.000000 | 0.002360 | 365.611742 |
| 25% | 875.000000 | 0.002463 | 406.956959 |
| 50% | 1500.000000 | 0.002463 | 406.595959 |
| 75% | 2250.000000 | 0.002558 | 413.103184 |
| max | 3000.000000 | 0.002816 | 425.871663 |

Based on Table 4, Casper + Secure algorithm shows average latency per transaction between 0.002381 and 0.002793 seconds, with a mean throughput of 393.67 transactions per second. Despite slightly higher latency, it maintains consistent performance, making it effective in scenarios balancing security and efficient transaction processing.

**Table 4**
Average Latency and Average Throughput of Casper + Secure

| Metric | Transactions | Average Latency (s/transaction) | Average Throughput (transactions/s) |
|---|---|---|---|
| count | 4.000000 | 4.000000 | 4.000000 |
| mean | 1625.000000 | 0.002576 | 393.667750 |
| std | 1108.677891 | 0.000214 | 32.174685 |
| min | 500.000000 | 0.002381 | 363.001000 |
| 25% | 875.000000 | 0.002393 | 367.321000 |
| 50% | 1500.000000 | 0.002565 | 394.827500 |
| 75% | 2250.000000 | 0.002743 | 421.174250 |
| max | 3000.000000 | 0.002793 | 422.015000 |

Based on Figure 7 and Figure 8, the algorithm shows much lower latency, operating at approximately $10^{-5}$ seconds compared to PoS's latency of around $10^{-3}$ second and significantly higher throughput, ranging from 14510.5 to 27858.6 transactions per second, versus PoS's relatively limited range of 168.2 to 206.8 transactions per second.

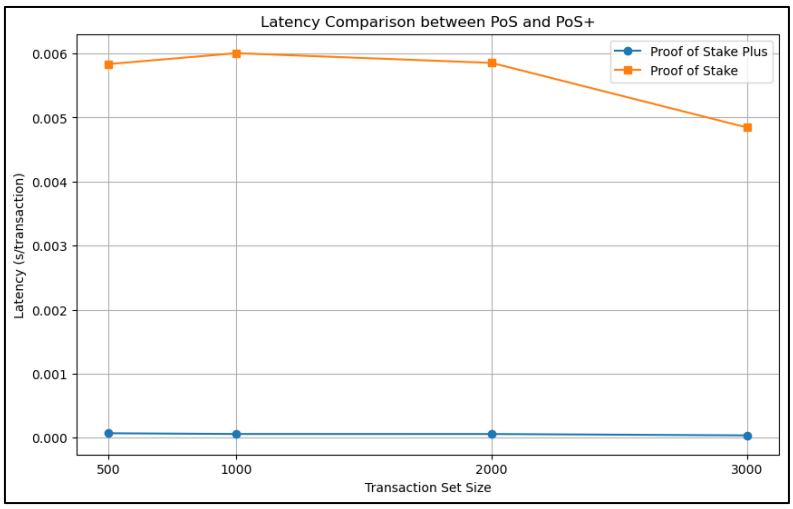


**Fig. 7.** Latency Comparison Between Pos and PoS+

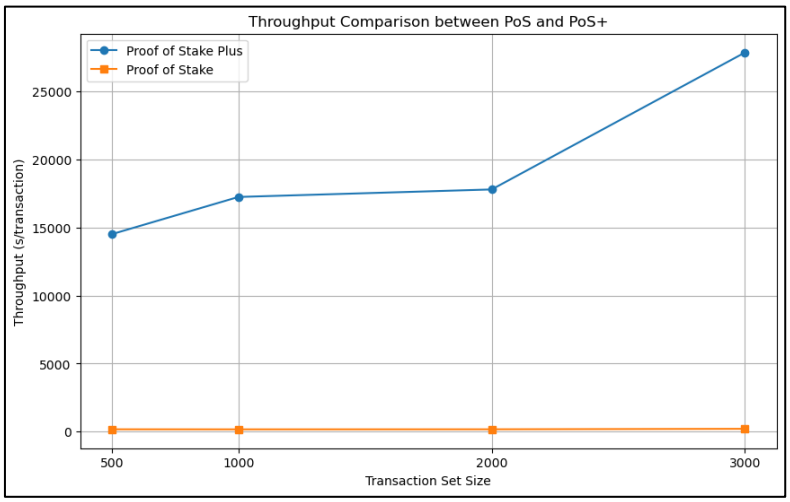


**Fig. 8.** Throughput Comparison Between PoS and PoS+

Based on Figure 9 and Figure 10, the RDPoS algorithm outperforms DPoS in terms of both latency and throughput. RDPoS exhibits lower average latency and significantly higher throughput, indicating its enhanced transaction processing speed and efficiency.

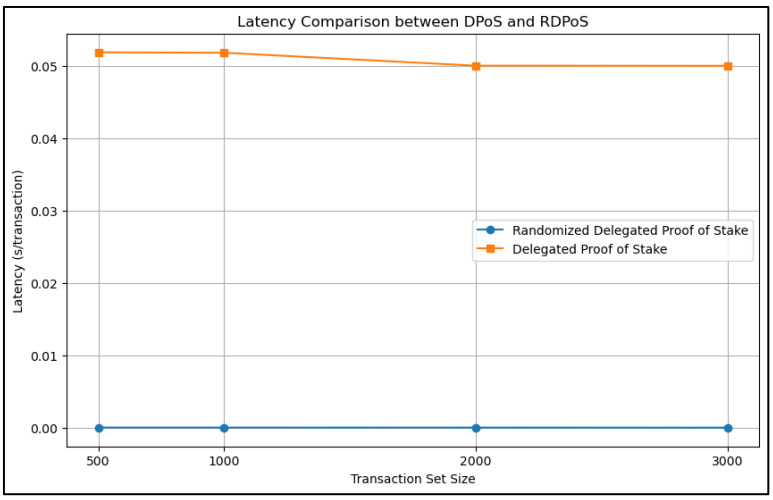


**Fig. 9.** Latency Comparison Between DPoS and RDPoS

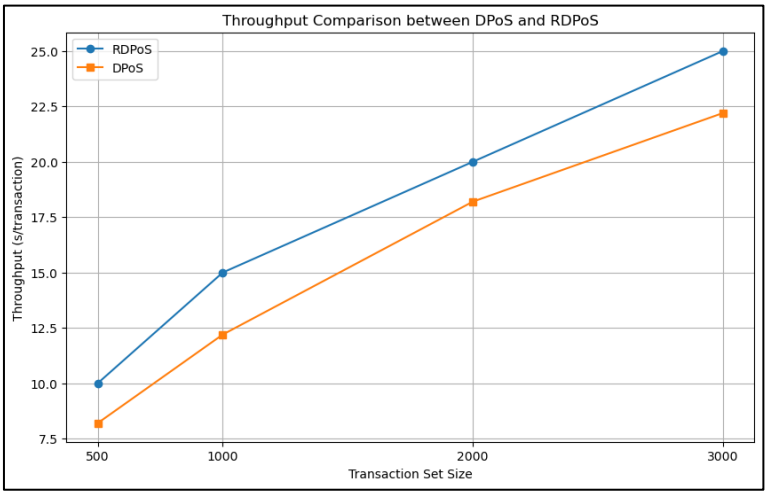


**Fig. 10.** Throughput Comparison Between DPoS and RDPoS

Based on Figure 11 and Figure 12, FBFT and PBFT perform similarly in terms of throughput. However, FBFT has marginally higher latency, which becomes more noticeable as transaction size increases. Both algorithms maintain effective consensus mechanisms with strong throughput.

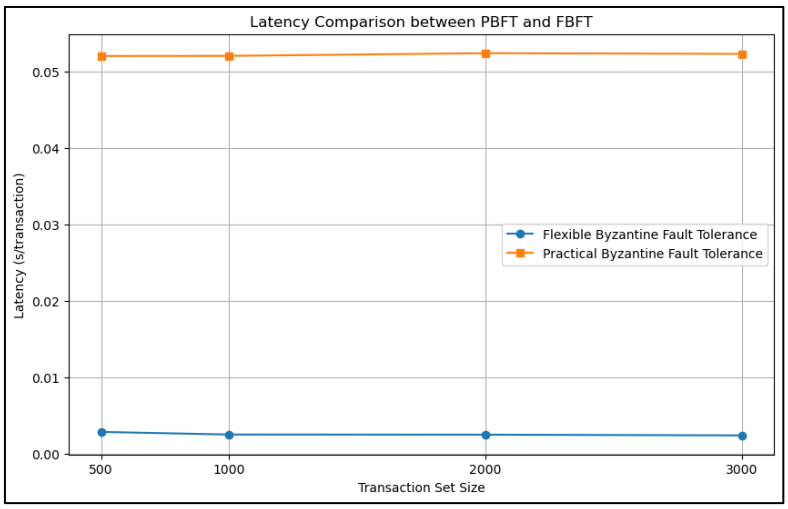


**Fig. 11.** Latency Comparison Between PBFT and FBFT

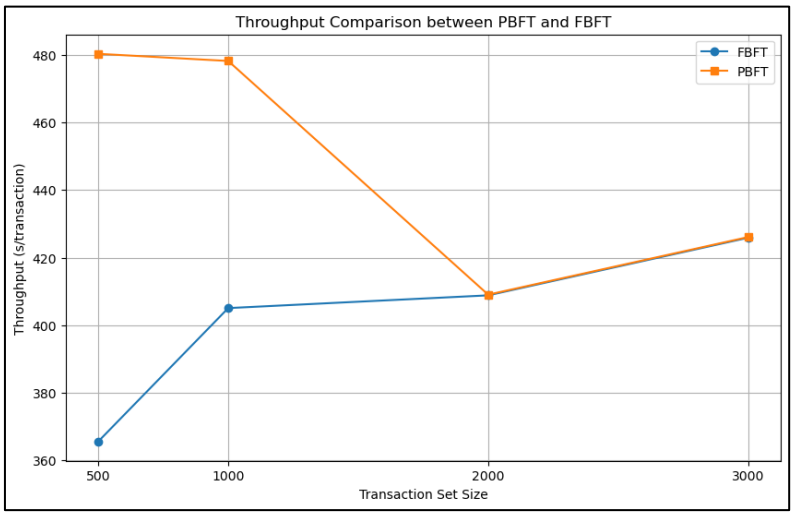


**Fig. 12.** Throughput Comparison Between PBFT and FBFT

Based on Figure 13 and Figure 14, Casper + Secure shows similar latency and throughput values compared to Casper, indicating its ability to balance enhanced security measures with efficient transaction processing.

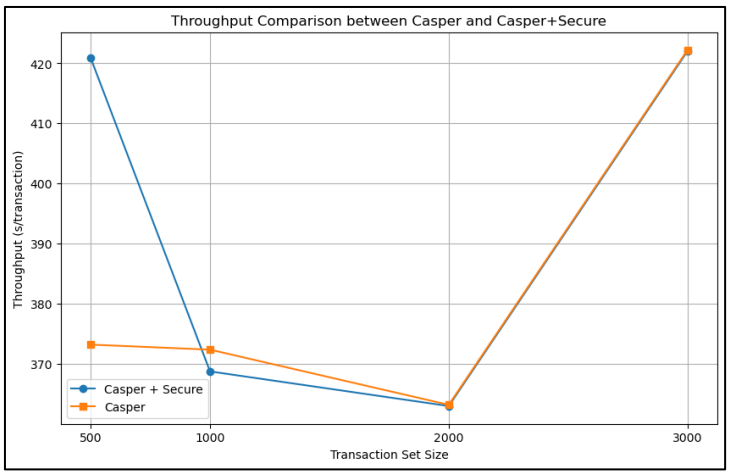


**Fig. 13.** Latency Comparison Between Casper And Casper+Secure

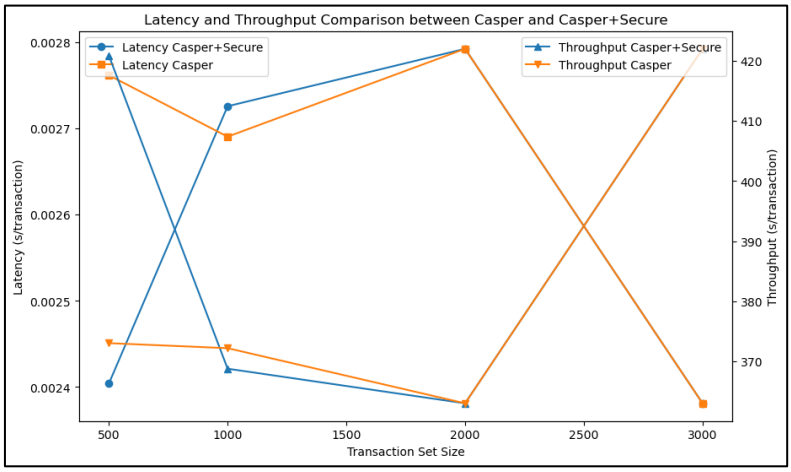


**Fig. 14**. Throughput Comparison Between Casper And Casper+Secure

Table 5 presents a comprehensive comparison of latency and throughput across all studied consensus algorithms. PoS+ stands out with the lowest latency and highest throughput, indicating its superior efficiency and scalability. RDPoS also shows remarkable performance, while other algorithms like FBFT and Casper + Secure demonstrate balanced trade-offs between latency, throughput, and additional features.

**Table 5**
Overall Comparison Of Latency And Throughput

| Consensus Algorithm | Average Latency (s/transaction) | Average Throughput (transactions/s) |
|---|---|---|
| PoS | 0.005840 | 171.94 |
| PoS+ | 0.000040 | 26352.07 |
| DPoS | 0.002250 | 448.20 |
| RDPoS | 0.000047 | 32336.82 |
| PBFT | 0.002085 | 480.10 |

| | | |
|---|---|---|
| FBFT | 0.002525 | 401.35 |
| Casper | 0.002576 | 393.67 |
| Casper + Secure | 0.002657 | 382.55 |

In conclusion, PoS+ emerges as the most efficient and scalable consensus algorithm in terms of latency and throughput. Algorithms like Casper + Secure, Casper, and PBFT provide reliable transaction processing but may have drawbacks when compared to PoS+. FBFT, meanwhile, demonstrates a more nuanced performance with slightly higher latency and lower throughput, suggesting trade-offs between different performance metrics. The findings emphasize the importance of selecting consensus algorithms based on specific needs and priorities in real-world applications. As supported by recent evaluations [23, 24], achieving the best trade-off between performance and decentralization often requires selecting or customizing consensus mechanisms based on the intended blockchain environment and workload characteristics.

## 4. Conclusions

With an emphasis on latency and throughput, we investigated and contrasted four consensus algorithms in this study: PoS+, RDPoS, FBFT, and Casper+Secure. PoS+ was the most effective of them all, providing remarkable throughput performance, low latency, and scalability.

In terms of handling transactions consistently, traditional methods like PBFT and Casper were still effective, but they still had observable drawbacks. There is room for improvement as FBFT in particular displayed somewhat higher latency and poorer throughput. Lightweight high-throughput consensus models for edge computing further highlight the growing need for scalable blockchain solutions [25].

Our investigation provided insights that can direct decision-making based on particular system requirements by illuminating each algorithm's strengths and potential weaknesses. However, it's crucial to remember that our conclusions are based only on a survey of the research. This method might have overlooked new developments or subtleties in practice, and because we only looked at four algorithms and two performance indicators, the findings shouldn't be interpreted as broadly applicable.

To provide a more comprehensive perspective, future research should take into account more comprehensive evaluation criteria, such as energy consumption, resource efficiency, and environmental effect. Future studies should look into machine learning-driven consensus algorithms [21, 13], hybrid models such as DAG-PBFT [9], and novel consensus mechanisms optimized for latency and throughput [24]. Applying machine learning approaches to jobs like optimization, anomaly detection, and validator selection has a lot of promise. In particular, reinforcement learning may pave the way for blockchain systems that are more flexible and self-regulating. Machine-learning-based prediction and decision-making models have shown potential in optimizing blockchain-related computational tasks [20]. By predicting validator behavior patterns and maximizing consensus decisions, predictive machine-learning models—like those employed in financial forecasting tasks—can likewise assist blockchain systems. [26]

In the end, this study emphasizes how crucial consensus processes are to preserving security and trust in decentralized systems and how much scope remains for development and innovation in this area.

## Acknowledgement

This research was funded by the Ministry of Higher Education Malaysia, Fundamental Research Grant Scheme (FRGS), FRGS/1/2024/ICT07/UPNM/02/1

## References

[1] Raina, Shivangi. "Blockchain Use Case–Cybersecurity: A Review." 2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), 277–279. https://doi.org/10.1109/ICACITE57410.2023.10182786. 2023.

[2] Hayes, Adam. "Blockchain Facts: What Is It, How It Works, and How It Can Be Used." Investopedia. https://www.investopedia.com/terms/b/blockchain.asp. 2023.

[3] Gramoli, Vincent. "From Blockchain Consensus Back to Byzantine Consensus." Future Generation Computer Systems 107 (2020): 760–769. https://doi.org/10.1016/j.future.2017.09.023.

[4] GeeksforGeeks. "Consensus Algorithms in Blockchain." https://www.geeksforgeeks.org/consensus-algorithms-in-blockchain/. 2019.

[5] TechTarget. "What Is a Consensus Algorithm?" https://www.techtarget.com/whatis/definition/consensus-algorithm. n.d.

[6] Bamakan, Seyed Mohammad Hosseini, Amir Motavali, and Ali Babaei Bondarti. "A Survey of Blockchain Consensus Algorithms Performance Evaluation Criteria." Expert Systems with Applications 154 (2020): 113385.

[7] Saleh, Fahad. "Blockchain Without Waste: Proof-of-Stake." SSRN Electronic Journal. https://doi.org/10.2139/ssrn.3183935. 2018.

[8] Chaudhry, Nouman, and Muhammad Yousaf. "Consensus Algorithms in Blockchain: Comparative Analysis, Challenges and Opportunities." 2023.

[9] Ferdous, Sadia, Md Jabed, Md Chowdhury, Md Hoque, and Asif Colman. "Blockchain Consensus Algorithms: A Survey." 2020.

[10] Hu, Qinghua, Bin Yan, Yang Han, and Jianping Yu. "An Improved Delegated Proof of Stake Consensus Algorithm." Procedia Computer Science 187 (2021): 341–346.

[11] Hussein, Zeyad, Mohamed A. Salama, and Sherif A. El-Rahman. "Evolution of Blockchain Consensus Algorithms: A Review on the Latest Milestones of Blockchain Consensus Algorithms." Cybersecurity 6, no. 1 (2023). https://doi.org/10.1186/s42400-023-00163-y.

[12] Gimenez-Aguilar, Maria, José M. de Fuentes, Lourdes Gonzalez-Manzano, and David Arroyo. "Achieving Cybersecurity in Blockchain-Based Systems: A Survey." Future Generation Computer Systems 124 (2021): 91–118.

[13] Yadav, Shailesh, Navneet Singh, and D. S. Kushwaha. "Evolution of Blockchain and Consensus Mechanisms and Its Real-World Applications." Multimedia Tools and Applications. https://doi.org/10.1007/s11042-023-14624-6. 2023.

[14] Reiff, Nathan. "What Is a DAO?" Investopedia. https://www.investopedia.com/tech/what-dao/. 2021.

[15] Hattab, Sahar, and I. F. Taha Alyaseen. "Consensus Algorithms Blockchain: A Comparative Study." International Journal of Perceptive and Cognitive Computing 5, no. 2 (2019): 66–71. https://doi.org/10.31436/ijpcc.v5i2.103.

[16] Venkatesan, Kandan, and Syarifah Bahiyah Rahayu. "Blockchain Security Enhancement: An Approach Toward Hybrid Consensus Algorithms and Machine Learning Techniques." Scientific Reports 14 (2024): 1149. https://doi.org/10.1038/s41598-024-51578-7.

[17] Gupta, Suyash, Jan Hellings, Sina Rahnama, and Mohammad Sadoghi. "Blockchain Consensus Unraveled." Proceedings of the 14th ACM International Conference on Distributed and Event-Based Systems, 2020. https://doi.org/10.1145/3401025.3404099.

[18] Kumar, Saurabh. "5 Anomaly Detection Algorithms Every Data Scientist Should Know." Towards Data Science. https://towardsdatascience.com. 2021.

[19] Li, Xin, Peitong Jiang, Ting Chen, Xiaoqi Luo, and Qiang Wen. "A Survey on the Security of Blockchain Systems." Future Generation Computer Systems 107 (2020): 841–853.

[20] Jebamikyous, Hala, Ming Li, Yash Suhas, and R. Kashef. "Leveraging Machine Learning and Blockchain in E-Commerce and Beyond: Benefits, Models, and Application." Discover Artificial Intelligence 3 (2023). https://doi.org/10.1007/s44163-022-00046-0.

[21] Xiong, Hao, Min Chen, Chunhua Wu, Yijun Zhao, and Wen Yi. "Research on Progress of Blockchain Consensus Algorithm." Future Internet 14, no. 2 (2022). https://doi.org/10.3390/fi14020047.

[22] Zhang, Qian, and Priya Patel. "A Comprehensive Review of Blockchain Performance Metrics: Emphasis on Latency and Throughput." Computers & Security 130 (2023): 103191. https://doi.org/10.1016/j.cose.2023.103191.

[23] Supreet, S., and Manisha Naravani. "Performance Evaluation of Consensus Algorithms in Private Blockchain Networks." Proceedings of ICACCM 2020. https://doi.org/10.1109/ICACCM50413.2020.9213019. 2020.

[24] Wang, Yong, Ming Zhong, and Tiancheng Cheng. "Research on PBFT Consensus Algorithm for Grouping Based on Feature Trust." Scientific Reports 12, no. 1 (2022). https://doi.org/10.1038/s41598-022-15282-0.

[25] Xiao, Min, Liyuan Yang, Zhiying Liu, and Jun Ren. "A Lightweight Consensus Protocol for High-Throughput Blockchain in Edge Computing." IEEE Transactions on Network and Service Management 20, no. 2 (2023): 1233–1246. https://doi.org/10.1109/TNSM.2023.3234568.

[26] Soujanya, R., P. Akshith Goud, A. Bhandwalkar, and G. Anil Kumar. "Evaluating Future Stock Value Asset Using Machine Learning." Materials Today: Proceedings 33 (2020): 4808–4813. https://doi.org/10.1016/j.matpr.2020.08.385.