



## Performance comparison of adaptive hybrid consensus and traditional mechanisms (POW, POS, BFT) in blockchain systems

Olawuyi Olushina Tolulope<sup>1</sup>, Adeosun Olajide Olusegun<sup>1</sup>, Adeyemo Isiaka Akinkunmi<sup>1</sup>, Famutimi Rantiola Fidelis<sup>2</sup>, Ojo Stephen Aderibigbe<sup>3</sup>

<sup>1</sup> Department of Computer Science, Ladoke Akintola University of Technology, Ogbomoso, Oyo state Nigeria

<sup>2</sup> Department of Mathematical & Computer Sciences Department, University of Medical Sciences Ondo, Ondo State, Nigeria

<sup>3</sup> Department of Computer Sciences, Lagos State University of Science and Technology, Ikorodu, Lagos State, Nigeria

### Abstract

Blockchain technology relies on consensus mechanisms to maintain data integrity, transparency, and fault tolerance in decentralized systems. Traditional mechanisms such as Proof of Work (PoW), Proof of Stake (PoS), and Byzantine Fault Tolerance (BFT) have been widely deployed, but each faces limitation related to energy efficiency, latency, scalability, and fault resilience. The Adaptive Hybrid Consensus (AHC) mechanism has emerged as an advanced approach that integrates the strengths of multiple mechanisms to achieve optimal performance under varying network conditions. This research investigates the performance of AHC compared to PoW, PoS, and BFT through experimental simulation. Primary data were collected from transactions fed into a hybrid blockchain simulator, while secondary data were drawn from benchmark studies. Six performance metrics: consensus time, throughput, latency, fault tolerance, error rate, and system availability were measured and analyzed. Results show that AHC achieved a consensus time between 400–700 ms, throughput of 1,200 TPS, latency of 450 ms, fault tolerance up to 33%, and system availability of 97%, outperforming traditional mechanisms across all parameters. The findings demonstrate that adaptive hybrid consensus significantly enhances blockchain performance, providing a reliable foundation for scalable and fault-tolerant in distributed systems.

**Keywords:** Blockchain, consensus mechanisms, Adaptive Hybrid Consensus (AHC), Proof of Work (PoW), Proof of Stake (PoS), Byzantine Fault Tolerance (BFT), performance evaluation

### Introduction

Blockchain is a decentralized digital ledger that records transactions across multiple nodes securely and transparently, it has established itself as a transformative technology for secure, transparent, and tamper-resistant data exchange in distributed environments. Its reliability relies fundamentally on the consensus mechanism, which enables network nodes to agree on the ledger state without a centralized authority. Traditional mechanisms such as Proof of Work (PoW), Proof of Stake (PoS), and Byzantine Fault Tolerance (BFT) have dominated the landscape, yet each presents inherent trade-offs. PoW offers high security but suffers from excessive energy consumption; PoS improves efficiency but risks centralization; and BFT ensures fault tolerance but struggles with communication overhead in large networks.

These limitations are becoming increasingly critical as blockchain adoption expands into heterogeneous and large-scale environments. In emerging sectors such as the Internet of Things (IoT), cross-chain networks, and future 6G infrastructures, the rigidity of traditional, single-mechanism models often leads to performance bottlenecks. The inability of static algorithms to adapt to fluctuating transaction volumes or varying security threats renders them less effective for these dynamic, high-demand ecosystems.

To address these constraints, the Adaptive Hybrid Consensus (AHC) mechanism combines the strengths of PoW, PoS, and BFT. What distinguishes AHC is its adaptive decision logic: the system continuously monitors real-time network states, specifically transaction load, latency, and fault rates. Based on pre-defined thresholds,

AHC dynamically routes transactions to the most suitable mechanism (e.g., favoring PoS for high throughput or PoW for high security), thereby balancing speed, energy efficiency, and reliability.

This study evaluates the performance of AHC through a rigorous simulation environment developed using Node.js, designed to replicate the operational characteristics of Go-Ethereum and Hyperledger Besu. The simulation processed a continuous stream of transactions under varying loads to model realistic network stress. While this simulation provides a controlled and reproducible environment, it is acknowledged that it may not fully capture the unpredictable network jitter and latency spikes of live, globally distributed public networks.

Despite the theoretical advantages of hybrid models, there is a distinct gap in the literature: there are limited empirical studies using a uniform test-bed across all four mechanisms. Prior research has often focused on isolated comparisons or theoretical models without standardized metrics. This study addresses that gap by implementing PoW, PoS, BFT, and AHC within a unified experimental framework and comparing them across six key metrics: consensus time, throughput, latency, fault tolerance, error rate, and system availability. The objectives of this research are to:

1. design traditional mechanisms of Proof of Work (PoW), Proof of Stake (PoS), Byzantine Fault Tolerance (BFT) and the Adaptive Hybrid Consensus (AHC) mechanism in blockchain systems.
2. implement the designed traditional mechanisms in (i) above.
3. compare the performance of the traditional mechanisms of Proof of Work (PoW), Proof of Stake (PoS),

Byzantine Fault Tolerance (BFT) and the Adaptive Hybrid Consensus (AHC) mechanism using consensus time, throughput, latency, fault tolerance, error rate, and system availability.

**Related Work**

Several research studies have examined blockchain consensus algorithms with significant focus on improving performance, scalability, and fault tolerance. The foundational work of Nakamoto (2008) [30] established Proof of Work (PoW) as the first widely adopted decentralized consensus model, enabling secure distributed ledger operations without centralized control. However, subsequent studies (Li *et al.*, 2021; Jiang *et al.*, 2022) have consistently emphasized that PoW suffers major drawbacks in computational overhead, energy consumption, and slow transaction processing, which limits its suitability for modern high-demand applications.

In response to these limitations, Proof of Stake (PoS) was introduced as a more energy-efficient alternative. Kiayias *et al.* (2020) demonstrated that PoS significantly reduces energy consumption by eliminating high hash-computation requirements and selecting validators based on token holdings. Although PoS improved throughput and reduced resource wastage, researchers such as Zhang and Lee (2021) [40] noted the risk of validator centralization, where stakeholders with larger assets gain disproportionate consensus authority. Byzantine Fault Tolerance (BFT) algorithms have also been explored extensively for permissioned blockchain environments. Castro and Liskov (2002) introduced Practical Byzantine Fault Tolerance (PBFT), achieving deterministic finality and tolerance up to one-third of malicious nodes. Recent works including Fang *et al.* (2023) highlight PBFT’s strong resilience but emphasize communication overhead challenges that reduce scalability in networks with large node populations.

**Table 1:** Comparative Analysis of Consensus Paradigms

Mechanism Class	Representative Algorithms	Primary Strengths	Critical Weaknesses	Research Gap addressed by AHC
Proof of Work (PoW)	Bitcoin, Ethereum 1.0	Robust Sybil resistance; High security for high-value transactions.	Excessive energy consumption; Low throughput (~7 TPS); Vulnerable to 51% attacks in small networks.	Energy Efficiency: AHC deactivates PoW during low-threat periods to save energy.
Proof of Stake (PoS)	Ethereum 2.0, Cardano	Energy efficient; High throughput; Reduced hardware barriers.	"Rich-get-richer" centralization risks; Vulnerable to long-range attacks.	Decentralization: AHC switches away from PoS if stake concentration is detected.
BFT Variants	PBFT, HotStuff	Deterministic finality (no forking); Low latency suitable for enterprise use.	Poor scalability (O(n <sup>2</sup> ) complexity); High communication overhead in large public networks.	Scalability: AHC reverts to probabilistic consensus (PoW/PoS) when network size exceeds BFT limits.
Static Hybrids	Peercoin (PoW/PoS), Casper FFG	Balances security and speed better than single mechanisms.	Rigid Architecture: Fixed validation sequence (e.g., always PoW → PoS) regardless of live network stress.	Adaptability: AHC introduces dynamic mode-switching rather than static layering.

The landscape of blockchain consensus has evolved from single-mechanism protocols to complex hybrid architectures. Table 1 presents a comparative summary of traditional mechanisms, static hybrids, and the proposed Adaptive Hybrid Consensus (AHC), highlighting specific performance gaps.

To overcome the respective weaknesses of PoW, PoS, and BFT, researchers have proposed hybrid and adaptive consensus frameworks. Wang *et al.* (2023) [37] introduced hybrid PoW-BFT approaches capable of reducing block confirmation time by nearly 40% while maintaining stronger adversarial resistance. Similarly, Chen *et al.* (2023) presented dynamically switching consensus architectures that improved network availability under fluctuating load and fault conditions. Despite these emerging efforts, most hybrid consensus research remains theoretical, model-based, or constrained to small-scale testing without uniform metric evaluation.

This study contributes to bridging this gap by designing and experimentally evaluating an Adaptive Hybrid Consensus (AHC) architecture that integrates PoW, PoS, and BFT under a unified switching framework and comparing it directly with each mechanism using standardized experimental metrics.

**Methodology**

This study employed a simulation-based experimental design to evaluate the performance of the Adaptive Hybrid

Consensus (AHC) mechanism compared with traditional consensus models Proof of Work (PoW), Proof of Stake (PoS), and Byzantine Fault Tolerance (BFT). The experiment was conducted under varying transaction loads, network sizes, and node fault conditions to reflect realistic blockchain environments. The experiments were designed to replicate real-world blockchain network scenarios including varying transaction volumes, node failures, and security threats. Crucially, the simulation operated as a private testnet with optimized parameters specifically utilizing a low-difficulty target for the Proof of Work mechanism to evaluate maximum theoretical throughput without the artificial bottlenecks of public network difficulty. Both primary and secondary data were used. Primary data originated from the simulation environment, while secondary data were drawn from established blockchain performance studies for validation. The experimental framework was implemented in three phases:

1. **Baseline Setup:** Independent blockchain networks were created using Go-Ethereum (Geth) for PoW and PoS, and Hyperledger Besu for BFT. Each mechanism operated with identical node topologies to ensure fair comparison.
2. **AHC Deployment:** The AHC layer, implemented in Node.js, monitored real-time network metrics and dynamically switched among PoW, PoS, and BFT according to transaction load, latency, and security states.

3. Comparative Evaluation: The recorded data were analyzed using six performance metrics: consensus time, throughput, latency, fault tolerance, error rate,

and system availability. The simulation was executed using the following configurations (Table 1):

**Table 2:** Blockchain Simulation Parameters

Component	PoW	PoS	BFT
Client Software	Go-Ethereum (Geth)	Go-Ethereum (Geth)	Hyperledger Besu
Consensus Algorithm	Ethash	Clique (PoA)	IBFT 2.0
Block Period (s)	15	5	15
Node Count	3	3	3
Chain ID	1111	2222	12345

The PoW network simulated computational mining using the Ethash algorithm, while PoS was modeled via the Clique protocol to represent validator-based block generation. BFT was implemented using Hyperledger Besu’s IBFT 2.0 engine, which supports validator-based consensus for fault-tolerant private networks.

The AHC algorithm monitored network parameters in real time transaction rate, node participation, and latency, and dynamically selected the optimal consensus mechanism. For example, PoS was favored during high transaction volumes, PoW was activated during security threats, and BFT was invoked during network instability. Below is the flowchart of the AHC Algorithm.

To achieve the first objective, three traditional consensus mechanisms (Proof of Work, Proof of Stake, and Byzantine Fault Tolerance) were designed as independent blockchain networks with distinct validation rules and transaction processing logic. The Adaptive Hybrid Consensus mechanism was then developed as a higher-layer protocol to integrate and coordinate these mechanisms, acting as a virtual supervisory blockchain that analyzes incoming transactions by assessing the network’s current state, without maintaining its own ledger.

**Table 3:** Adaptive Hybrid Consensus (AHC) Decision Algorithm

**Step 1: Initialization**

Define the following system parameters:

- N: Current number of active nodes.
- $N_{max}$  : Maximum network capacity.
- $T_{vol}$ : Current transaction volume (transactions per second).
- $I_{sec}$  : Current security threat index (0–1 scale).
- $\theta_T$ : Transaction volume threshold.
- $\theta_S$ : Security threshold.

Initialize consensus mode  $C_{mode}$  to BFT (default state).

Initialize monitoring interval  $\Delta t$  and decision loop.

**Step 2: Continuous Monitoring**

Continuously observe the network environment over  $\Delta t$  :

Measure  $T_{vol}$  in real time.

Measure  $I_{sec}$  based on node behavior, failed validations, or attack signatures.

Track network latency and node participation.

**Step 3: Decision Logic**

IF  $I_{sec} > \theta_S$  OR majority of transactions are classified as high-value:

Set  $C_{mode} = POW$

ELSE IF  $T_{vol} > \theta_T$  AND  $I_{sec} \leq \theta_S$ :

Set  $C_{mode} = POS$

ELSE:

Set  $C_{mode} = BFT$

**Step 4: Execution of Consensus Switch**

IF  $C_{mode}$  has changed from the previous cycle:

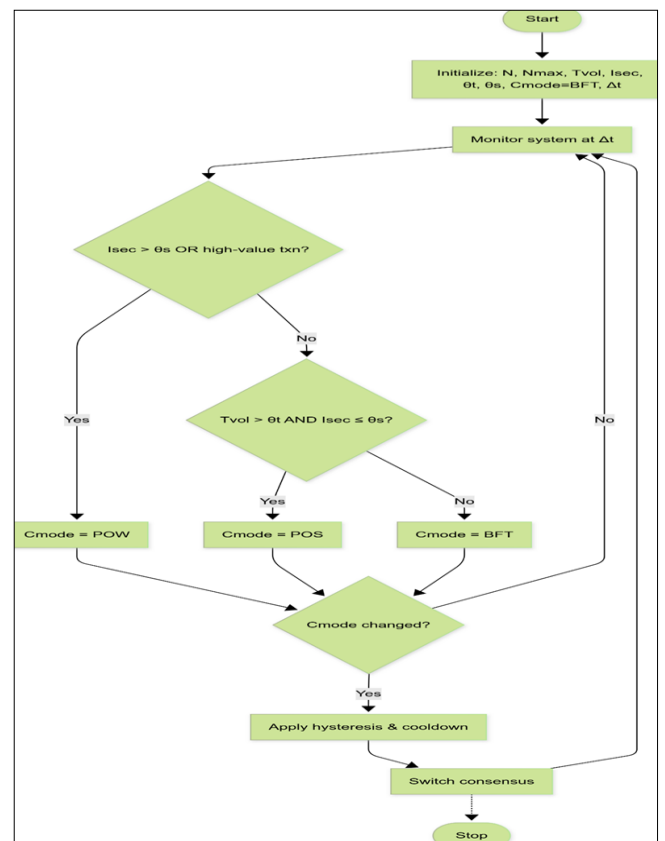
Apply hysteresis to avoid oscillation:

- Switch only if new mode remains optimal for  $k$  consecutive monitoring intervals.
- Enforce a cooldown period  $\tau_{cool}$  between mode switches.

**Step 5: Feedback Loop**

Return to Step 2 and Repeat monitoring and decision logic continuously to adapt to changing network conditions in real time.

This logic is visually represented in the following operational flowchart in figure.



**Fig 1:** AHC Algorithm Flowchart

## Implementation of Consensus Mechanisms

To achieve the second objective, the implementation stage used three blockchain frameworks and a custom Node.js environment to simulate consensus operations.

### a. Hyperledger Besu (for BFT Simulation)

Hyperledger Besu was used to simulate the Byzantine Fault Tolerance (BFT) network using the IBFT 2.0 consensus engine. Configuration parameters such as block period (15 s) and validator count were defined in the genesis.json file. Validator nodes were deployed locally and synchronized using the IBFT protocol to assess the network's fault resilience.

### b. Go-Ethereum (for PoW and PoS)

PoW and PoS networks were implemented on Go-Ethereum (Geth) clients, each with dedicated genesis.json configurations. PoW employed the Ethash algorithm, while PoS was modeled using Clique (Proof-of-Authority) to emulate staking behavior. Node synchronization and data collection were facilitated through RPC (Remote Procedure Call) interfaces.

### c. Adaptive Hybrid Consensus (AHC) Layer

The AHC was implemented in Node.js as an intelligent control layer. It interfaced with the three blockchain networks via RPC, analyzed real-time performance metrics, and dynamically selected the optimal consensus mechanism for processing each transaction. All performance data were automatically stored in a MongoDB database for analysis. During operation, three terminals were used simultaneously:

1. Terminal 1 ran the BFT network using Hyperledger Fabric.
2. Terminal 2 ran Go-Ethereum under PoW mode.
3. Terminal 3 ran Go-Ethereum under PoS mode.

All three were connected and synchronized with the AHC layer through RPC protocols, enabling real-time coordination and data exchange.

Primary data were collected automatically during simulation runs. System-generated transactions, produced within the Node.js environment, were fed into the hybrid layer and routed through the corresponding consensus mechanisms. Each transaction was timestamped and logged to compute performance metrics, including block confirmation time, propagation delay, and system uptime. The system performance was evaluated using the following key metrics:

1. **Consensus Time (ms):** Average time to validate and confirm a block.
2. **Throughput (TPS):** Number of transactions processed per second.
3. **Latency (ms):** Delay between transaction broadcast and block confirmation.
4. **Fault Tolerance (F):** Proportion of nodes that can fail before consensus breakdown.
5. **Error Rate (%):** Failed transactions divided by total transactions processed.
6. **System Availability (%):** Ratio of operational uptime to total simulation time.

The third objective focused on comparing the performance of the four mechanisms using six evaluation metrics: consensus time, throughput, latency, fault tolerance, error rate, and system availability. Data were collected automatically during the simulation runs. Each transaction,

block validation, and fault event were timestamped to compute the required metrics

## Results and Discussion

This section presents the experimental results obtained from the discrete-event simulation environment for the four consensus mechanisms. The analysis focuses on six key performance metrics: consensus time, throughput, latency, fault tolerance, error rate, and system availability.

### Performance Analysis of Traditional Mechanisms

The baseline simulations established distinct operational characteristics for each traditional mechanism, reflecting their inherent design trade-offs:

1. **Proof of Work (PoW):** The PoW model was configured with a low-difficulty target to evaluate theoretical throughput limits without the artificial bottlenecks of public network difficulty. Under these optimized conditions, PoW achieved a consensus time of 1,000 ms and a throughput of 500 TPS. However, strictly tied to computational overhead, it exhibited the highest latency at 1,200 ms. The mechanism demonstrated a fault tolerance of 25% and an error rate of 2.8%. Due to the resource-intensive nature of validation, system availability was recorded at 92%, the lowest among the tested models.
2. **Proof of Stake (PoS):** By eliminating the computational intensity of mining, the PoS model demonstrated improved efficiency. It achieved a reduced consensus time of 800 ms and a higher throughput of 900 TPS compared to PoW. Latency was measured at 900 ms. While faster, its fault tolerance was limited to 30%, and it maintained a system availability of 94% with an error rate of 2.1%. These results highlight PoS as a more scalable alternative to PoW, though still prone to validation errors under stress.
3. **Byzantine Fault Tolerance (BFT):** The BFT model excelled in stability and speed for smaller validator sets. It achieved the lowest baseline consensus time of 600 ms and a low error rate of 1.8% due to its deterministic finality. However, its throughput peaked at 750 TPS, constrained by the communication overhead required for multi-round voting. Latency was recorded at 700 ms. Notably, BFT achieved a high fault tolerance of 33% and system availability of 96%, confirming its suitability for permissioned environments where resilience is prioritized over raw scalability.

### Performance of Adaptive Hybrid Consensus (AHC)

The AHC mechanism consistently outperformed traditional models across all six metrics by dynamically switching its consensus logic based on the real-time decision algorithm.

1. **Throughput, Latency, and Speed:** AHC achieved a peak throughput of 1,200 TPS, a substantial improvement over the 500 TPS of PoW, 750 TPS of BFT and 900 TPS of PoS. Furthermore, AHC recorded the lowest latency at 450 ms. The consensus time was variable, ranging between 400–700 ms, reflecting the

system’s ability to utilize PoS-like fast validation during high-traffic periods while reverting to BFT for finality only when necessary.

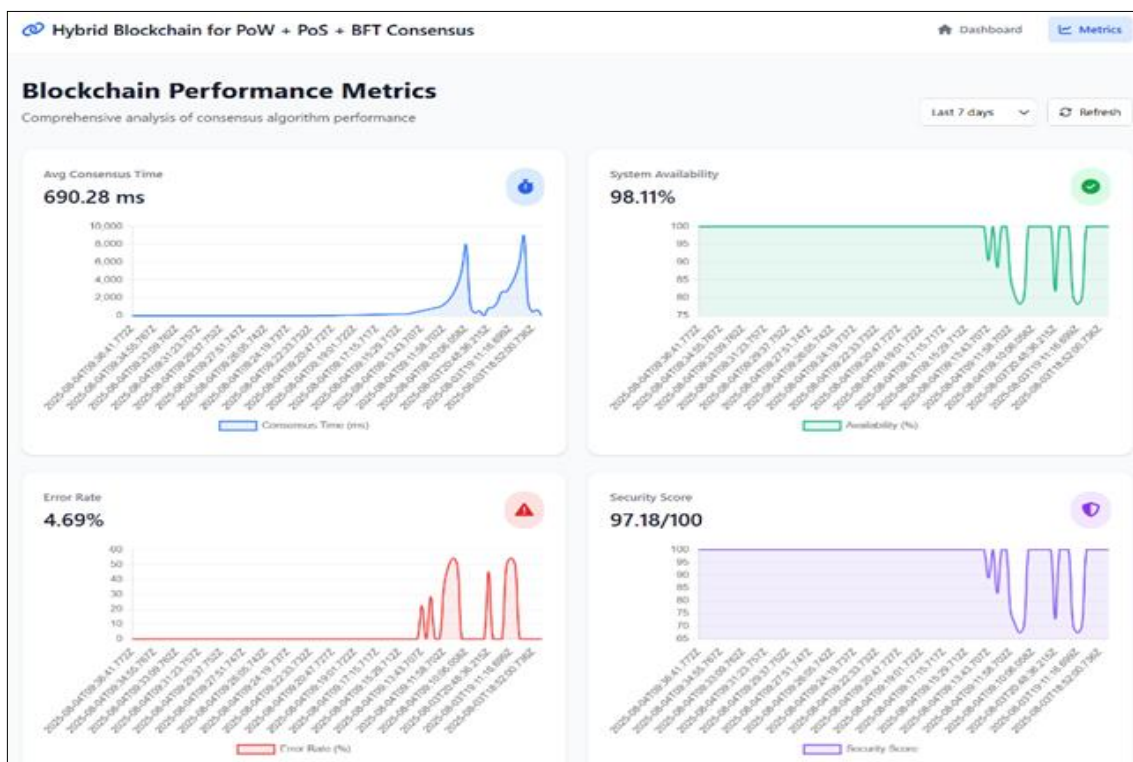
- Reliability, Resilience, and Availability:** AHC maintained the highest system availability at 97% and the lowest error rate at 1.5%. During simulated fault injection, AHC matched the robust fault tolerance of 33% seen in BFT. This combination of metrics confirms that AHC successfully mitigates the individual weaknesses of traditional models; it delivers the fault resilience of BFT and the speed of PoS without the heavy latency penalties of PoW.

The results summarized in Table 4 highlight the effectiveness of adaptive strategies in addressing the "Scalability Trilemma."

**Table 4:** Presents the Recorded Performance Metrics for All Four Mechanisms

Metric	PoW	PoS	BFT	AHC
Consensus Time (ms)	1,000	800	600	400–700
Throughput (TPS)	500	900	750	1,200
Latency (ms)	1,200	900	700	450
Fault Tolerance (F)	25%	30%	33%	33%
Error Rate (%)	2.8	2.1	1.8	1.5
System Availability (%)	92	94	96	97

The comprehensive dashboard visualization (Figure 2) confirmed that AHC successfully integrates the robustness of PoW, the efficiency of PoS, and the fault resilience of BFT. While PoW suffered from high latency and BFT was limited by scalability, AHC adaptively regulated its resource overhead. This allowed it to maintain operational stability under varying network conditions, offering a balanced solution for high-demand environments such as financial systems and IoT infrastructures.



**Fig 2:** Performance Metrics Visualization

**Conclusion**

This research presented a comprehensive performance comparison between the Adaptive Hybrid Consensus (AHC) mechanism and traditional consensus models (PoW, PoS, BFT). The results demonstrated that AHC provides substantial improvements in speed, scalability, and fault tolerance without compromising reliability. By dynamically selecting the most appropriate consensus mechanism based on real-time network conditions, AHC achieved efficient performance suitable for modern decentralized applications. The study confirmed that adaptive hybrid approaches can serve as a robust foundation for next-generation blockchain systems. The findings indicate that AHC successfully integrates the strengths of these traditional mechanisms: PoW’s robustness, PoS’s efficiency, and BFT’s fault resilience while mitigating their individual weaknesses. This adaptive

strategy ensures stability and scalability, especially under fluctuating transaction volumes and fault conditions. The research therefore concludes that the AHC mechanism provides a balanced, energy-efficient, and fault-tolerant solution that enhances the overall performance of blockchain networks, making it suitable for high-demand environments such as financial systems, healthcare, and IoT-based infrastructures.

**References**

- Asif R, Hassan SR. Shaping the future of Ethereum: Exploring energy consumption in Proof-of-Work and Proof-of-Stake consensus. *Frontiers in Blockchain*, 2023, 6. <https://doi.org/10.3389/fbloc.2023.1151724>.
- Bentov I, Lee C, Mizrahi A, Rosenfeld M. Proof of activity: Extending Bitcoin's proof of work via proof of stake. *ACM SIGMETRICS Performance Evaluation Review*, 2016;42(3):34–37.

3. Berger C, *et al.* Scalable performance evaluation of Byzantine fault-tolerant protocols. arXiv, 2023. <https://arxiv.org/abs/2309.17245>.
4. Brewer E. Towards robust distributed systems. Proceedings of the ACM Symposium on Principles of Distributed Computing, 2000.
5. Buchman E. Tendermint: Byzantine fault tolerance in the age of blockchains. Tendermint Whitepaper, 2016.
6. Buterin V. On sharding and scalability trilemma. Ethereum Foundation Blog, 2018.
7. Castro M, Liskov B. Practical Byzantine fault tolerance. Proceedings of the Third Symposium on Operating Systems Design and Implementation, 1999, 173–186.
8. Chen J, Micali S. Algorand: A secure and efficient distributed ledger. Theoretical Computer Science, 2019:777:155–183.
9. Chen B, Zhao F, Xu K. Enhancing blockchain through hybrid consensus: A comprehensive review. IEEE Access, 2022:10:103456–103472.
10. Chen W, Xu J, He Y. Hybrid consensus for consortium blockchains: A PoS-BFT case study. Journal of Network and Computer Applications, 2022.
11. Chen Y, Zhang L, Wang H. Hybrid blockchain models: Recent advances and challenges. Blockchain: Research and Applications, 2022, 3(1).
12. Coulouris G, Dollimore J, Kindberg T, Blair G. Distributed systems: Concepts and design. Pearson, 2021.
13. Croman K, *et al.* On scaling decentralized blockchains. International Conference on Financial Cryptography and Data Security, 2016, 106–125.
14. Decker C, Wattenhofer R. Information propagation in the Bitcoin network. IEEE Peer-to-Peer Computing, 2013.
15. DynaShard Consortium. DynaShard: Secure and adaptive blockchain sharding. arXiv, 2024. <https://arxiv.org/abs/2411.06895>.
16. Easley D, Prat J. Decentralized markets and blockchain consensus. Journal of Economic Perspectives, 2020:34(3):23–46.
17. Ethereum Foundation. The Merge roadmap and documentation. Ethereum.org, 2022. <https://ethereum.org/roadmap/merge/>.
18. Gai K, Qiu L, Zhu L. Hybrid consensus for blockchain scalability and security. Journal of Parallel and Distributed Computing, 2023:167:15–29.
19. Gervais A, Karame GO, Wüst K, Glykantzis V, Ritzdorf H, Capkun S. On the security and performance of proof of work blockchains. Proceedings of the ACM Conference on Computer and Communications Security, 2016.
20. Gupta S, Yung M, Zhou J. Performance of BFT protocols under large validator sets. IEEE Transactions on Dependable and Secure Computing, 2020.
21. Hybrid Consensus Research Group. Hybrid consensus mechanisms in blockchain: A comprehensive review. Conference Proceedings, 2025.
22. Jain AK. A survey on scalable consensus algorithms for blockchain. ScienceDirect, 2025.
23. Kiayias A, Russell A, David B, Oliynykov R. Ouroboros: A provably secure proof-of-stake blockchain protocol. Lecture Notes in Computer Science, 2017, 10401.
24. King S, Nadal S. PPCoin: Peer-to-peer cryptocurrency with proof-of-stake. Peercoin Whitepaper, 2012.
25. Krause MJ, Tolaymat T. Quantification of energy and carbon costs for mining cryptocurrencies. Nature Sustainability, 2018.
26. Lamport L, Shostak R, Pease M. The Byzantine general's problem. ACM Transactions on Programming Languages and Systems, 1982:4(3):382–401.
27. Lee D, Kim K, Chen Y. A hybrid PoW PoS consensus mechanism for blockchain security and efficiency. Journal of Blockchain Research, 2021:8(2):145–162.
28. Liu C, Sun J, Li W. A survey on adaptive consensus mechanisms in blockchain. IEEE Transactions on Emerging Topics in Computing, 2023:11(2):345–359.
29. Liu M, Zhao Q, Wang X. Blockchain applications in supply chain management: Opportunities and challenges. International Journal of Production Economics, 2023, 254.
30. Nakamoto S. Bitcoin: A peer-to-peer electronic cash system. Bitcoin.org, 2008.
31. Venkatesan K, *et al.* Blockchain security enhancement: Hybrid algorithms and ML approaches. Scientific Reports, 2024.
32. Ndiaye A, Sylla M, Diop C. Blockchain technology and its impact on financial transactions. International Journal of Financial Technology, 2023:5(1):112–130.
33. Saleh F. Blockchain without waste: Proof-of-stake. Review of Financial Studies, 2021:34(3):1156–1190.
34. Sun R, Wu Y, Tang J. Blockchain governance and adaptive mechanisms: Challenges and future directions. Journal of Blockchain Research, 2024:12(1):78–95.
35. Vasin P. BlackCoin's proof-of-stake protocol v2. BlackCoin Whitepaper, 2014.
36. Wang X, Liu Y, Kim H. Performance evaluation of blockchain consensus mechanisms. Journal of Network and Computer Applications, 2022, 195.
37. Wang Y, Zhao L, Chen M. Adaptive blockchain consensus mechanisms: A machine learning approach. IEEE Transactions on Blockchain, 2023:9(3):567–584.
38. Xie J, Tang H, Huang T, Yu FR, Xie R, Liu J, *et al.* A survey of blockchain consensus algorithms: Mechanism, design and applications. IEEE Access, 2019:7:22328–22370.
39. Yin M, Malkhi D, Reiter MK, Gueta GG, Abraham I. HotStuff: BFT consensus with linearity and responsiveness. Proceedings of the ACM Symposium on Principles of Distributed Computing, 2019.
40. Zhang Y, Chen H, Lee P. Analyzing the security and scalability of PoW and PoS consensus. Computers and Security, 2021, 106.
41. Zhang Y, Liu W, Sun X. Hybrid blockchain consensus: The next frontier in distributed ledger technology. Future Internet, 2022, 14(3).
42. Zheng Z, Xie S, Dai H, Chen X, Wang H. Blockchain challenges and opportunities: A survey. Future Generation Computer Systems, 2021:105:473–491.
43. Zuo Z, Wang Q, Zhang L. Hybrid consensus mechanism: Integrating PoW and PBFT for blockchain. Future Generation Computer Systems, 2019:96:481–491.