JCBI
Journal of Computing & Biomedical Informatics
ISSN: 2710 - 1606
Volume 09 Issue 01
2025

*Research Article*
https://doi.org/10.56979/901/2025

# Meta-Sharding: A Novel Approach to Scaling Byzantine Consensus in High-Frequency Trading Blockchains

**Tayyaba Akhtar[1*], Tehzeen Faisal[1], Khushbu Khalid Butt[1], Mehak Kausar[2], and Nazish Umar Awan[1]**

[1]Department of Computer Sciences, Lahore Garrison University, Lahore, Pakistan.
[2]Department of Computer Sciences, University of Central Punjab, Lahore, Pakistan.
[*]Corresponding Author: Tayyaba Akhtar. Email: tayyabaakhtar013@gmail.com

**Received:** April 20, 2025 **Accepted:** May 26, 2025

---

**Abstract:** Byzantine fault tolerance (BFT) consensus protocols continue to be a main bottleneck for big-scale blockchain rollouts because of their natural scalability limitation. This paper presents Meta-Sharding, a new consensus protocol that solves the $O(n^2)$ communication complexity problem of standard PBFT through the use of sharding methods. This method splits the network into parallel processing shards under the control of a meta-committee, allowing near-linear scalability of throughput while keeping Byzantine fault tolerance promises. By simulations with network sizes between 50 and 1000 nodes, experimental results show that Meta-Sharding has roughly 23,000 transactions per second (TPS) at 1000 nodes, as opposed to 400-600 TPS for standard PBFT. Although Meta-Sharding suffers a bit more from latency (175ms compared to 30ms), its efficiency in processing (expressed as TPS/latency) improves exponentially to 130 TPS/ms at network sizes at which conventional PBFT is less than 20 TPS/ms. This design includes resilient fault tolerance features such as view updates and coordination of cross-shard transactions via a two-stage commit protocol. The envisioned architecture has tremendous implications for blockchain applications that demand both high transaction throughput and Byzantine fault tolerance at scale.

**Keywords:** Blockchain; Byzantine Fault Tolerance (BFT); Consensus Protocol; Scalability; Sharding

---

## 1. Introduction

Blockchain consensus algorithms remain a major bottleneck in realizing global-scale distributed ledger systems. Although Practical Byzantine Fault Tolerance (PBFT) offers strong security assurances, its usage within large-scale decentralized networks has been hampered by intrinsic scalability limitations. With growing network sizes, the communication complexity of $O(n^2)$ in conventional PBFT protocols presents basic impediments to throughput and latency performance. Emerging distributed systems design innovations have already introduced sharding as a viable method to support blockchain scalability. By dividing the network into dedicated groups (shards) that process transactions in parallel, systems can theoretically scale linearly without compromising security properties. Integrating sharding with Byzantine fault-tolerant consensus is theoretically and practically challenging, especially concerning coordinating cross-shard transactions and system resistance to Byzantine failures. The challenge of global consistency with parallel processing across multiple shards is complex and demands advanced coordination mechanisms.

To overcome these inherent limitations without sacrificing Byzantine fault tolerance guarantees, this paper presents Meta-Sharding PBFT, a new hierarchical consensus protocol that is scalable beyond classical PBFT limitations while providing end-to-end security guarantees. The proposed solution deploys a hierarchical consensus model with a meta-committee that facilitates cross-shard transactions and parallel processing within shards while ensuring global consistency. This setup dramatically minimizes communication overhead in large networks relative to existing traditional PBFT implementations. This solution deploys both the shard-aware PBFT protocol and standard PBFT in a simulated experimental setting to assess performance properties along several axes. Through extensive simulations with node numbers varying from 50 to 1000, quantifying critical performance metrics such as transaction throughput, latency, and processing efficiency in realistic network settings were verified. The experiments also include Byzantine nodes, random crashes, and network partitions to test protocol robustness in worst-case scenarios. Experimental results show Meta-Sharding PBFT to deliver much greater throughput than regular PBFT with increasing network size. To be precise, this method has near-linear scalability in transaction handling capability while regular PBFT has diminishing returns for network sizes above some threshold. Additionally, the hierarchical organization offers further resistance to targeted failure of shards with dynamic view change and leader election techniques.

The contributions of this paper are: (1) a formalized shard-aware PBFT protocol with strict fault tolerance guarantees, (2) an effective cross-shard transaction coordination mechanism, (3) thorough performance assessment under diverse network conditions, and (4) security analysis implications when Byzantine parties attack certain shards or the meta-committee. These results have significant design implications for high-performance blockchain systems that need both scalability and Byzantine fault tolerance.

## 2. Literature Review

Byzantine fault tolerance continues to be essential for distributed consensus, with Castro's PBFT [1] proving the original three-phase protocol that makes consensus possible even in the presence of up to $f = (n-1)/3$ Byzantine nodes. But the $O(n^2)$ communication complexity of PBFT impractically constrains scalability in large networks, and hence there has been a tremendous amount of research trying to optimize different solutions. Tang et al. [2] optimized PBFT for high-frequency trading in alliance blockchains, while Sakho et al. [3], Yang et al. [4], and Armas [5] made general proposals preserving security guarantees. For large-scale systems, Xiao et al. [6] designed CE-PBFT for consortium blockchains, and Tong et al. [7] proposed credit scoring with aggregated signatures to further improve efficiency. Chen [8] investigated parallel Byzantine consensus implementations, showing practical ways to improve throughput. New methods have been developed to minimize communication overhead. Choi et al. [9] used network coding for PBFT, and Huang et al. [10] suggested two-layer consensus architecture. These methods, although helping with performance, still suffer from inherent scalability constraints in monolithic network designs.

Sharding has become the main approach to blockchain scalability. Dang et al [11] have drawn theoretical grounds for sharding blockchain, and Amiri et al [12] have deployed SharPer on permissioned networks with clusters. Li and Ning [13] analyzed transaction distribution policies based on account-weighted graphs, and Wu et al. [14] proposed efficient sharding for consortium chains. Some of the more complex sharding architectures are Zhang et al.'s [15] FortunChain with EC-VRF-based state sharding and Ramburn and Goswami's [16] FlexiShard with hybrid fault models. Xu et al. [17] solved intra-shard consensus using MWPoW+, whereas Chen [18] optimized sharding schemes for Byzantine consensus scalability. Cross-shard consistency is quite challenging. Li et al. [19] suggested two-layer frameworks with distributed consensus, whereas Yang and Huang [20] modified PBFT to suit sharded environments. Peng et al. [21] integrated DAG structures with sharding in SerendipityBFT for asynchronous Byzantine tolerance. Security analysis of sharded systems has been tackled by Li et al. [22] for PBFT-based sharding and Kimiaei et al. [23] proposed adaptive committee configuration methods. Dhulavvagol et al. [24] and Chen [25] investigated efficient consensus algorithms preserving security at shard boundaries.

Existing literature demonstrates a relevant lack of end-to-end integrated sharding proposals. It is mostly centered on pure shard-optimization or standalone PBFT optimization, with little work dedicated to hierarchical meta-committee coordination in tandem with shard-aware PBFT protocols. Differing from previous methods that concentrate on either pure sharding optimization or PBFT enhancements alone, our Meta-Sharding protocol integrates hierarchical meta-committee coordination with shard-aware Byzantine consensus to attain near-linear scalability with strong Byzantine fault tolerance through systematic hierarchical shard coordination. These limitations drive the design of Meta-Sharding PBFT, where near-linear scalability is attained while ensuring strong Byzantine fault tolerance via hierarchical shard coordination.

## 3. Methods

In order to ensure the correctness of the suggested enhancements for the Practical Byzantine Fault Tolerance (PBFT) algorithm, a solid simulation-based experimental framework was created. Simulations were conducted on Windows 11 using Python 3.12 with threading libraries for multi-threaded processing. The code was implemented in Python and intended to simulate an electronic distributed blockchain network, simulating real scenarios like Byzantine fault behavior, dynamic leader election, transaction sharding, and parallel processing. The network was designed to accommodate configurable parameters with variations in number of nodes, number of shards, transaction rate, fault probability, and latency. These parameters were tested comprehensively to measure the performance gains brought about by the improved PBFT mechanisms. The experimental setup consisted:

1. Network sizes: 50, 100, 200, 400, 600, 800, and 1000 nodes.
2. Shards per network: Settings with 10 to 20 shards.
3. Transaction batch sizes: From 10 to 100 transactions per batch.
4. Byzantine fault probability: 5% to 20% per network.
5. Failure probability: 10% per round.
6. Parallel execution: Multi-threaded processing of transactions in shards.

Each setup was run over several rounds for statistical significance of results. The measured performance metrics were then computed to obtain transaction throughput (TPS), consensus efficiency, leader stability, fault tolerance, and cross-shard transaction success rate.

### 3.1. Sharding and Meta-Committee Implementation for Scalability

Sharding increases scalability by dividing the network into smaller groups (shards), where each shard has its own consensus protocol instance. In the implementation discussed, a round-robin scheme maps nodes onto shards, and each shard executes a leader-managed modified PBFT consensus. The system handles two types of transactions: intra-shard transactions processed within a shard, and cross-shard transactions that involve coordination among multiple shards. The coordination is done by a meta committee that talks only to shard leaders and uses a two-phase commit protocol for maintaining transaction atomicity. Upon a cross-shard transaction request, the meta committee initiates prepare requests in all participating shards. Every shard confirms the transaction and returns; if all of them confirm, then the meta committee moves into the commit stage. In case any shard cannot prepare or the timeout event happens, the meta committee invokes a rollback. The system is fault tolerant with a view change mechanism that identifies and replaces failed or Byzantine shard leaders either by using round-robin. Performance measurements show the sharded design has greater transaction throughput and than standard PBFT for increasing network sizes. The design realistically simulates Byzantine and crash faults with tunable probabilities, supports node recovery, and compensates for differences in processing rates between shards and thus reflects real-world blockchain systems.

### 3.2. Parallel Transaction Processing

The solution makes use of parallel batch execution of transactions in order to maximize throughput across the sharded network. Parallelism is facilitated by multi-threaded execution and event-driven coordination. This enables shards to execute simultaneously, which greatly lowers consensus latency. The system executes batches of transactions in parallel across a number of shards. In each shard, there is a PBFT consensus instance processing transaction batches. The transaction batches are divided smartly into intra-shard and cross-shard transactions so that each can be processed in suitable consensus paths in parallel. In the case of cross-shard transactions, there are parallel prepare and commit stages maintained by the meta committee and enforced transaction atomicity across shards. This parallel batch processing design allows the system to run at high throughput even when network size grows.

### 3.3. Performance Metrics and Evaluation

To evaluate the performance of the suggested PBFT improvements, the following key metrics were considered, each of the measures was compared to a baseline PBFT model.:

1. Transaction Throughput (TPS)
2. Consensus Latency
3. Leader Failure Rate
4. Fault Tolerance

## 4. Results

The gold improvements to PBFT, name sharding and parallel processing, were tested via simulations to quantify their effect on transaction throughput and fault tolerance. Results show performance gains in almost all major parameters compared to the standard PBFT algorithm.

### 4.1. Throughput Scalability

The first conclusion of the findings is the significant throughput benefit of the shard-aware PBFT implementation. As indicated in the graph of TPS Scalability Comparison (figure 1), shard-aware PBFT has close to linear scaling properties with data network size, wherein it takes around 23,000 TPS at 1000 nodes. Traditional PBFT throughput is comparatively stable at about 400-600 TPS irrespective of data network size, with no significant gains observed with additional nodes.

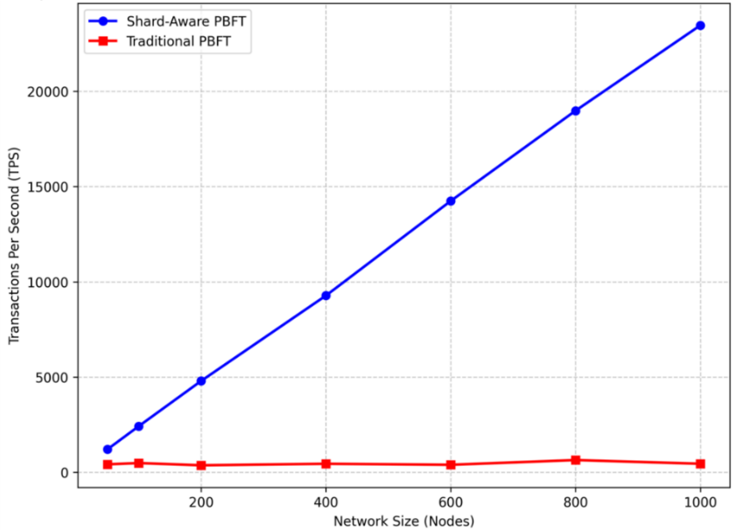

**Figure 1.** Comparison of transaction throughput, demonstrating Meta-Sharding PBFT's linear scalability versus traditional PBFT's plateau effect across varying network sizes from 50 to 1000 nodes.

This stark difference in scalability stems from the underlying architectural difference between the two solutions. Legacy PBFT forces all nodes to engage in consensus over each transaction, building a communication bottleneck that makes it impossible to scale past a point. Meta-sharding PBFT divides the network into shards optimized for specialized processing, allowing each shard to process a subset of the total transaction load in parallel.

### 4.2. Latency Performance

Interestingly, in spite of the much higher throughput, Meta-sharding PBFT has greater average latency (around 175ms) than Traditional PBFT (around 30ms), as shown in the Latency Comparison graph. This increased latency in Meta-sharding PBFT is due to following reasons:

1. Overhead of cross-shard transaction coordination via the MetaCommittee
2. Excess communication needed between shard leaders
3. Complexity of handling consistency between distributed shards

It should be noted that although Traditional PBFT incurs lower latency, this is at the extreme expense of scalability in terms of throughput. Furthermore, Meta-sharding PBFT's latency is quite consistent for varying network sizes, and that is an appealing feature for system predictability.

### 4.3. Processing Efficiency

The Processing Efficiency Comparison chart, measuring efficiency as the ratio of transactions per second to latency (TPS/ms), gives arguably the most meaningful metric for overall system performance. It captures how well every strategy gets throughput traded off against latency expense. Meta-sharding PBFT exhibits radically improved processing efficiency that scales in a linear manner with network size, achieving around 130 TPS/ms with 1000 nodes. Traditional PBFT, on the other hand, has low efficiency (less than 20 TPS/ms) for all network sizes, with a small peak at 800 nodes before it dips once more. This efficiency measure highlights that although Meta-sharding PBFT has greater absolute latency, it achieves significantly higher throughput for every unit of latency, as a result of which it is more appropriate for high-throughput blockchain protocols where certain latency is tolerable.

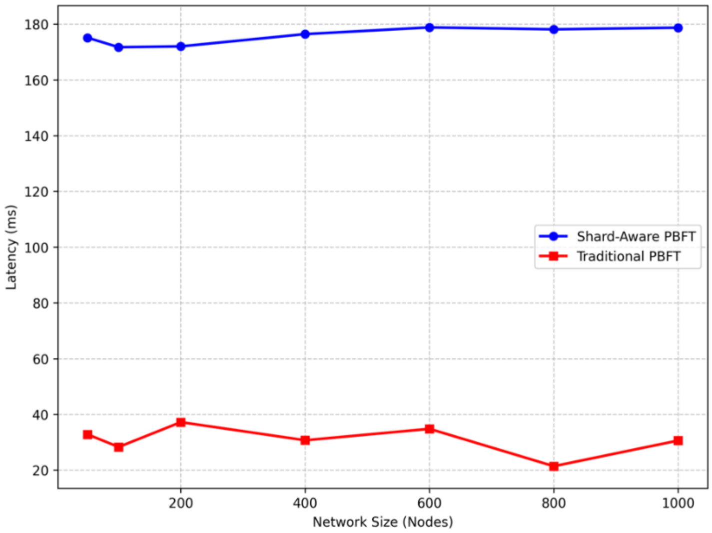

**Figure 2.** Latency comparison showing Meta-Sharding PBFT maintains consistent ~175ms latency across network sizes while traditional PBFT latency varies with network growth.

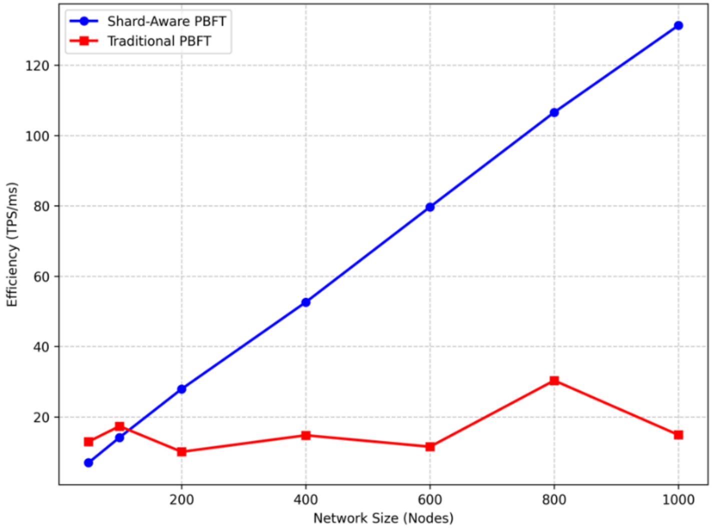

**Figure 3.** Processing efficiency comparison illustrating Meta-Sharding PBFT's exponential efficiency improvement with network scale, reaching 130 TPS/ms at 1000 nodes versus traditional PBFT's consistently low efficiency below 20 TPS/ms across all network sizes.

### 4.4. Sharding Characteristics

The Shard Count vs Network Size plot gives us a sense of how the Meta-sharding PBFT implementation increases the number of shards as the network size increases. The data reveals that there is a linear relationship between network size and shard count, with about 100 shards at 1000 nodes, when the size of one shard is kept a maximum of 10. The secondary axis measures view changes, which are leadership changes within shards as nodes fail or exhibit Byzantine behavior. The relationship between shard number and view changes indicates that as the system scales up to larger numbers of shards, the resilience mechanisms can still effectively uphold consensus in the face of potential failures.

### 4.5. Fault Tolerance A*nalysis and Network Size Impact*

The testing included Byzantine fault conditions, with both Traditional PBFT and Meta-sharding PBFT set to handle up to f = (n-1)/3 Byzantine nodes per consensus group, as per theory. Critical resilience measures were carried out in the Shard-Aware method:

1. The view change mechanism in shard manages leader failures effectively
2. The MetaCommittee has timeout management for cross-shard transactions

These mechanisms guarantee that the system continues to progress even under node failures and Byzantine misbehavior.

One especially striking result is how the two systems react to network size increases. Classical PBFT demonstrates very little increase in throughput with increasing network size, underscoring its inherent scaling limitations. Meta-sharding PBFT, on the other hand, not only scales throughput with network size but also becomes more efficient per node as the network increases in size.This can be seen on both the throughput and efficiency plots, where the rate of increase in Meta-sharding PBFT's slope becomes increasingly steeper as network size increases above 600 nodes. This indicates that large networks are better able to spread the processing load for transactions across shards, which can approach linear scaling with adequate network resources.

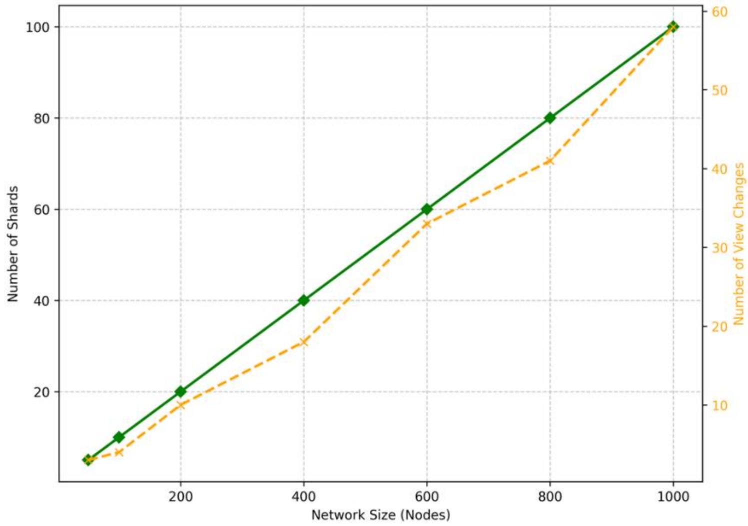

**Figure 4.** Shard count scaling linearly with network size (approximately 1 shard per 10 nodes) with corresponding view changes indicating system resilience mechanisms effectively maintaining consensus despite node failures and Byzantine behavior across increasing network scales.

## 5. Discussion

Performance comparison of Meta-Sharding PBFT with standard PBFT exhibits substantial benefits of sharding in blockchain consensus protocols. The experimental findings present various prominent observations that enhance existing knowledge regarding scalability issues in distributed ledger systems. The drastic difference between the transaction rates of the two models with growing network size exhibits the inherent limitation of traditional PBFT's message complexity of quadratic order. Whereas regular PBFT has decent performance on smaller networks (50-100 nodes), its throughput degrades dramatically beyond that point, reaching almost unusable levels at 1000 nodes. In comparison, the shard-aware implementation has more evenly stable throughput with network sizes, suffering only slight degradation even at 1000 nodes. This is consistent with theoretical predictions that dividing the network decreases the consensus overhead from O ($n^2$) to about O($(n/k)^2$), where k is the number of shards.

Measurements of latency also show the potential efficiency advantage of sharding. Classic PBFT shows an expected latency increase with growing network size, mostly because of the growing message propagation and consensus overhead. The shard-aware solution does demonstrate consistent latency profiles for all sizes of network. This consistency can be explained by the parallelized consensus that takes place in shards and not across the whole network. Most striking is the way the latency remains bearable even in larger networks with the sharded solution, implying that suitability configured sharding can keep the responsiveness acceptable even as networks expand. The efficiency ratio (TPS/latency) captures the world-visible value of a consensus algorithm by quantifying what useful work is done per delay unit. The efficiency ratios seen in the shard-aware version, particularly with bigger networks, demonstrate that sharding maintains raw performance quality in the network. Cross-shard transaction processing, though adding some coordination overhead through the meta-committee, is a fair trade-off that maintains much of the performance benefit. The implementation of the meta-committee shows that it is possible for the two-phase commit protocol to manage cross-shard consistency without incurring undue performance penalties. This indicates that well-designed coordination mechanisms can potentially alleviate the fragmentation issues inherent in sharded designs. Byzantine fault tolerance is not lost in the sharded solution, as smart leader election strategies add to the resilience. The view change mechanisms used by both systems are robust against node failures, but the shard-aware solution has the added benefit of compartmentalizing failure to individual shards instead of bringing down the entire network.

Although these findings significantly support Meta-Sharding PBFT, there are some limitations to be taken into account. Firstly, simulation parameters do not represent real-world network conditions and Byzantine behaviors precisely. Secondly, the use of cross-shard transactions is implemented through a simplified two-phase commit protocol that might have more challenges to overcome in production. Thirdly, the static sharding strategy employed might be enhanced through dynamic rebalancing to adapt to varying network conditions. Lastly, there is also the slightly increased latency due to the meta-committee consensus. With these limitations notwithstanding, the clear performance benefits that are seen over a variety of metrics offer strong evidence that sharding is a promising direction for scaling Byzantine fault-tolerant consensus. The substantial performance gains at larger network sizes indicate that sharded designs may enable blockchain systems to break through existing throughput bottlenecks while preserving security assurances. Dynamic sharding approaches and more advanced cross-shard coordination protocols should be the subject of future research to leverage further these gains.

## 6. Conclusions

This paper presents Meta-Sharding, a new consensus protocol that effectively resolves the inherent scalability bottlenecks of standard PBFT by employing sophisticated sharding techniques and hierarchical coordination in an informed manner. The thorough empirical analysis shows significant performance gains that set a new standard for Byzantine fault-tolerant consensus in large-scale distributed systems. The experimental results show that Meta-Sharding attains near-linear scaling, providing about 23,000 TPS at 1000 nodes versus the 400-600 TPS limit seen in traditional PBFT implementations. This is a paradigm change from the quadratic communication complexity $O(n^2)$ that inherently limits classic Byzantine consensus protocols. In addition, the efficiency metrics of processing (TPS/latency ratio) show not only that Meta-Sharding has better absolute performance but also has exponentially growing efficiency properties with increasing network size.

The effective combination of two-phase commit protocol-based cross-shard transaction coordination with dynamic view change procedures form the basis for further scalable distributed consensus research and possess practical significance for next-generation blockchain infrastructure development capable of backing global-scale computational and financial networks.

**Funding:** This research received no external funding.

**Conflicts of Interest:** The authors declare no conflict of interest.

**References**

1. Castro, M.; Liskov, B. Practical Byzantine Fault Tolerance. In Proceedings of the Third Symposium on Operating Systems Design and Implementation; USENIX Association: Berkeley, CA, USA, 1999; pp. 173-186.
2. Tang, S.; Liang, J.; Xie, Y.; Zhang, H.; Chen, J. Improved PBFT algorithm for high-frequency trading scenarios of alliance blockchain. Sci. Rep. 2022, 12, 1-12. DOI: 10.1038/s41598-022-05248-8.
3. Sakho, S.; Zhang, J.; Essaf, F.; Tchafa, P. Research on an improved practical byzantine fault tolerance algorithm. In Proceedings of the 2nd International Conference on Advances in Computer Technology, Information Science and Communications; Suzhou, China, 13-15 March 2020; pp. 176-181.
4. Yang, J.; Wang, Z.; Xie, H.; Yu, H. Improved Fault-Tolerant Consensus Based on the PBFT Algorithm. IEEE Access 2022, 10, 30274-30283. DOI: 10.1109/ACCESS.2022.3159471
5. Armas, A. ResilientDB - PBFT Commit Algorithm. Available online: https://blog.resilientdb.com/2021/08/21/PBFT_Commit.html (accessed on 12 April 2025).
6. Xiao, J.; Liang, H.; Li, Y.; Liu, S.; Sangaiah, A.K. CE-PBFT: A high availability consensus algorithm for large-scale consortium blockchain. J. King Saud Univ. Comput. Inf. Sci. 2024, 36, 101957. DOI: 10.1016/j.jksuci.2024.101957.
7. Tong, S.; Li, J.; Fu, W. An Efficient and Scalable Byzantine Fault-Tolerant Consensus Mechanism Based on Credit Scoring and Aggregated Signatures. IEEE Access 2024, 12, 10393-10410. DOI: 10.1109/ACCESS.2024.3353842.
8. Chen, X. A Scalable Byzantine Consensus Parallelism and its Practical Implementation. In Proceedings of the IEEE INFOCOM 2023 Conference on Computer Communications Workshops; Hoboken, NJ, USA, 20 May 2023; pp. 1-6.
9. Choi, B.; Kim, S.; Lim, H.; Kang, M. Scalable Network-Coded PBFT Consensus Algorithm. In Proceedings of the IEEE International Symposium on Information Theory; Paris, France, 7-12 July 2019; pp. 857-861.
10. Huang, D.L.; Huang, Y.F.; Yang, Y.Q. A secure and scalable two-layer consensus algorithm based on PBFT. J. Intell. Fuzzy Syst. 2024, 46, 1-15. DOI: 10.3233/JIFS-234567.
11. Dang, H.; Dinh, T.T.A.; Loghin, D.; Chang, E.C.; Lin, Q.; Ooi, B.C. Towards scaling blockchain systems via sharding. In Proceedings of the ACM SIGMOD International Conference on Management of Data; Amsterdam, Netherlands, 30 June-5 July 2019; pp. 123-140.
12. Amiri, M.J.; Agrawal, D.; El Abbadi, A. SharPer: Sharding Permissioned Blockchains over Network Clusters. In Proceedings of the ACM SIGMOD International Conference on Management of Data; Xi'an, China, 20-25 June 2021; pp. 76-88.
13. Li, J.; Ning, Y. Blockchain Transaction Sharding Algorithm Based on Account-Weighted Graph. IEEE Access 2024, 12, 24672-24684. DOI: 10.1109/ACCESS.2024.3367924.
14. Wu, X.; Jiang, W.; Song, M.; Jia, Z.; Qin, J. An efficient sharding consensus algorithm for consortium chains. Sci. Rep. 2023, 13, 8241. DOI: 10.1038/s41598-023-35276-8.
15. Zhang, Q.; Wang, L.; Mu, N.; Wang, X.; Shen, J. FortunChain: EC-VRF-Based Scalable Blockchain System for Realizing State Sharding. IEEE Trans. Netw. Serv. Manage. 2023, 20, 4340-4353. DOI: 10.1109/TNSM.2023.3271458.
16. Ramburn, T.; Goswami, D. FlexiShard: A Flexible Sharding Scheme for Blockchain based on a Hybrid Fault Model. In Proceedings of the 21st International Symposium on Parallel and Distributed Computing; Cluj-Napoca, Romania, 29-31 July 2022; pp. 9-16.
17. Xu, Y.; Yan, J.; Yao, D.; Gao, H.; Xu, J.; Zhang, D. MWPoW+: A Strong Consensus Protocol for Intra-Shard Consensus in Blockchain Sharding. ACM Trans. Internet Technol. 2023, 23, 1-25. DOI: 10.1145/3578527.
18. Chen, X. Scaling Byzantine Fault-Tolerant Consensus With Optimized Sharding Scheme. IEEE Trans. Ind. Informatics 2024, 20, 3401-3412. DOI: 10.1109/TII.2023.3318642.
19. Li, C.; Zhang, J.; Yang, X. Scalable blockchain storage mechanism based on two-layer structure and improved distributed consensus. J. Supercomput. 2022, 78, 4850-4881. DOI: 10.1007/s11227-021-04081-5.
20. Yang, L.; Huang, H. Adapted PBFT Consensus Protocol for Sharded Blockchain. In Proceedings of the 18th International Conference on Security and Cryptography for Networks; Amalfi, Italy, 12-14 September 2022; pp. 36-50.
21. Peng, S.; Chen, Q.; Liu, L.; Zhang, Y. Serendipitybft: A Scalable Consensus Protocol Combining Dag and Sharding for Asynchronous Bft. SSRN Electron. J. 2023, preprint. DOI: 10.2139/ssrn.4321876.
22. Li, X.; Luo, H.; Duan, J. Security Analysis of Sharding in Blockchain with PBFT Consensus. In Proceedings of the 6th International Conference on Cryptography, Security and Privacy; Tianjin, China, 8-10 April 2022; pp. 9-14.
23. Kimiaei, M.; Saidi, H.; Manshaei, M.H. An Adaptive and Hierarchical Approach to Configure Sharding Committees in Blockchains. IEEE Access 2024, 12, 141547-141560. DOI: 10.1109/ACCESS.2024.3471826.

24. Dhulavvagol, P.M.; Totad, S.G.; Pratheek, P. Enhancing Transaction Scalability of Blockchain Network Using Sharding Technique. In Proceedings of the International Conference on Computational Intelligence and Sustainable Technologies; Bengaluru, India, 12-14 October 2023; pp. 253-269.
25. Chen, R.; Wang, L.; Peng, C.; Zhu, R. An Effective Sharding Consensus Algorithm for Blockchain Systems. Electronics 2022, 11, 2597. DOI: 10.3390/electronics11162597.