Enhancing Blockchain Resilience via Multi-Signal Detection and Robust Freezing under Partitioned Networks

L. Naveen Kumar
Department of Computer Science and Engineering, University of Visvesvaraya College of Engineering, Bangalore University, Bengaluru, India
naveengowdakns@gmail.com (corresponding author)

S. H. Manjula
Department of Computer Science and Engineering, University of Visvesvaraya College of Engineering, Bangalore University, Bengaluru, India
shmanjula@gmail.com

Received: 19 September 2025 | Revised: 12 October 2025 and 22 October 2025 | Accepted: 24 October 2025

 DOI: https://doi.org/10.48084/etasr.14936

ABSTRACT

Blockchain systems, such as Bitcoin and Ethereum 2.0, face vulnerabilities under bandwidth-constrained partitions, where throughput collapses and latency increases. In addition, adversaries can exploit inconsistencies to launch double-spending attacks. This study presents a lightweight dual-layer countermeasure that integrates a robust freezing threshold (Δj^*) with multi-signal disconnection proofs to enhance performance and security without altering consensus rules. Controlled simulation experiments on Bitcoin (PoW) and Ethereum 2.0 (PoS) show throughput gains exceeding 1000% in Ethereum and over 100% in Bitcoin, with inconsistency reduced by up to 64% and latency bounded within 5-6 blocks/s. These results confirm that attacker-aware thresholds and multi-signal validation substantially improve blockchain resilience under partitioned network conditions.

Keywords-blockchain security; network partition; double-spending; consensus protocols; Ethereum 2.0; Bitcoin

I. INTRODUCTION

Blockchain systems, such as Bitcoin and Ethereum 2.0, form the backbone of decentralized financial infrastructure. Their consensus protocols are designed to ensure both security and availability under normal network conditions. However, when network bandwidth degrades or partitions occur, throughput collapses, block finalization stalls, and adversaries can exploit these weaknesses to launch double-spending or inconsistency attacks [1]. Such disruptions undermine the economic reliability and overall trustworthiness of blockchain platforms [2]. To mitigate these vulnerabilities, this study presents lightweight yet effective enhancements that improve network resilience without altering the underlying consensus rules. In Ethereum 2.0 (Proof-of-Stake - PoS), the beacon-chain consensus suffers from severe throughput collapse once bandwidth drops below a critical threshold, leaving validator committees unable to finalize blocks efficiently [3].

Recent research in Reinforcement Learning (RL) has demonstrated strong adaptability for dynamic and uncertainty-aware network environments, particularly in vehicular and distributed communication systems [4]. In particular, hierarchical and multi-agent actor-critic frameworks enable autonomous resource allocation under fluctuating bandwidth and latency, providing insights related to adaptive control in blockchain consensus networks [5]. In [6], the limitations of Ethereum 2.0's LMD-GHOST fork-choice rule were examined, introducing Goldfish, a reorg-resilient replacement that supports fast confirmation independent of security level. This study revealed multiple loss-of-finality incidents on Ethereum 2.0 in March 2024, confirming that consensus safety and liveness can indeed be compromised under adverse network conditions. In [7], the Nakamoto consensus was analytically refined by deriving upper and lower bounds for double-spend success under explicit random delays. The novelty lies in modeling confirmation safety using random delay distributions rather than fixed bounds. However, the evaluation was entirely theoretical without simulation-based or empirical validation, limiting its practical impact. SyncAttack [8] exploits protocol synchronization to achieve double-spending without mining power. Experiments on Bitcoin reported near 100% success in double-spending transactions under adversarial churn and latency, offering a protocol-aware timing exploit that bypasses most hash requirements. However, the attack scenarios are narrow and depend on exploitable synchronization patterns.

In [9], the SimBlock simulator was extended to test network partitions. In controlled settings, a throughput collapse of more than 80% was observed, along with fork rates doubling under hard partitions lasting 60 seconds. In [10], Bitcoin partitioning attacks were re-engineered against Core v23, assessing their feasibility. The results showed that with realistic bandwidth injections, adversaries could sustain hour-long partitions at relatively low cost, reducing throughput to below 5 TPS and inflating fork probability by more than 40%. Cost-success tradeoffs were evaluated using attack duration and resource expenditure metrics. The novelty of this approach was in integrating modern peer-to-peer network defenses into the attack model. In [11], a probabilistic bouncing attack in Ethereum's PoS was examined. Using probabilistic models and simulation, this study showed that validator jitter could increase reorganization probability by up to 15%, extending time-to-finality by multiple epochs. Metrics included reorg rate and finality latency. The novelty of this study was in formally characterizing validator misalignment effects in PoS consensus. However, the evaluation was simulation-based and lacked large-scale validation. DiFastBit [12] is a deterministic fast-payment defense. Across 10,000 simulated transactions, DiFastBit achieved 0% accepted double-spends and sustained >95% baseline throughput. Metrics included throughput, retention, and false positive rate.

Existing approaches, such as sharding, relay networks, and Layer-2 rollups, enhance blockchain scalability or fault tolerance but operate at higher abstraction layers, often requiring major protocol modifications or added network overhead. This work (i) formulates a robust freezing threshold (Δj^*) to prevent consensus instability under adversarial bandwidth loss, (ii) integrates a multi-signal *k*-of-*m* disconnection proof for reliable fault detection, and (iii) validates the approach on Bitcoin (PoW) and Ethereum 2.0 (PoS) simulations, showing significant gains in throughput, consistency, and latency resilience. Figure 1 depicts a blockchain network facing a partition attack, where an adversary exploits low-capacity bandwidth links (dashed connections) to isolate groups of nodes.

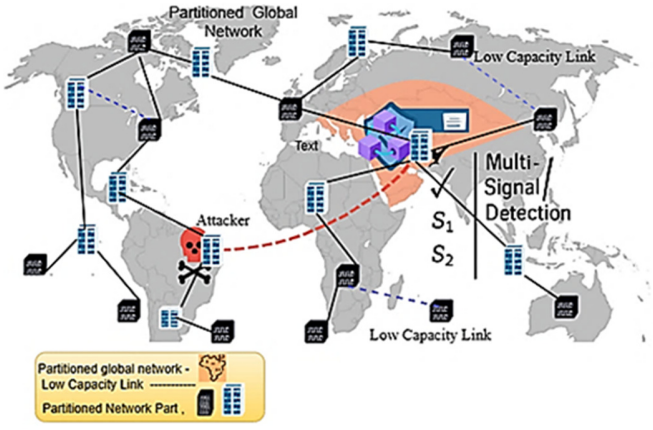

Fig. 1. Partitioned blockchain network showing shaded adversarial region, dashed low-bandwidth links, and multi-signal detection modules that isolate abnormal partitions via *k*-of-*m* validation.

II. PROPOSED METHOD

The following algorithm summarizes the experimental method used to evaluate the proposed countermeasure under partitioned blockchain networks.

```
Algorithm 1: Experimental Workflow
Input: Protocols P, Node size N,
Bandwidths B, Trials R, Horizon H
Output: Aggregated metrics M_agg,
  improvement ratios Δ
For each protocol p ∈ P: do
  For each node size n ∈ N: do
    For each bandwidth b ∈ B: do
      For r ← 1 to R: do
        Deploy a network with n nodes
        under bandwidth b;
        Run trial for H blocks with
        workload injection;
        Collect metrics: Throughput,
        Latency, Inconsistency, Double-
        spend, Gas (ETH only);
      end
      Aggregate results across R trials,
      compute mean, std, 95% CI, and store
      in M_agg(p,n,b);
    end
  end
end
foreach metric m and each (n,b) pair: do
  Compute improvement for metric m;
  Δ_m(n,b)
    = (M_agg(Proposed,n,b,m) − M_agg(Baseline,n,b,m)) / M_agg(baseline,n,b,m)
end
Return M_agg, Δ_m
```

A. Freezing Threshold & Multi-Signal Detection

1) Robust, Attacker-Aware Freezing Threshold (Δj^*)

Intuitively, Δj^* represents the maximum tolerated divergence in participation or delay before the system enforces a temporary freeze. For example, if a node consistently lags behind others by more than this limit during network congestion, it is suspended to avoid inconsistency. Similarly, the *k*-of-*m* rule implies that at least *k* out of *m* independent connectivity signals (e.g., peer-ping, relay skew, attestation drop, header divergence) must confirm failure before a freeze is applied, e.g., if 3 of 5 detectors agree, the proof is accepted. The baseline threshold Δj is replaced with a conservative estimator that accounts for implicit adversarial power and partition churn [13]:

$$\Delta_j^* = \min\left(1, \frac{\sum_{i \in j}(\delta_i^{UB})|HashRate| + pow_i}{|HashRate|} \cdot \frac{|M_j^{on}|}{|M_j|}\right) \times (1-\alpha) \quad (1)$$

where:

- pow_i: Explicit hashrate (or stake) of node i. Δj^* is the robust freezing threshold for committee j. The individual node i belongs to committee j ($i \in j$).
- δ_i^{UB}: Upper confidence bound on node i's implicit power fraction, estimated from short-window block/attestation drift and delay telemetry [14]. Confidence bounds are computed using Wilson or Clopper-Pearson intervals over the last W slots, capped at <1/2.
- α : Safety haircut coefficient ($0 \leq \alpha \leq 0.2$) scaling conservativeness under increased network risk [15].

Equation (1) determines the maximum tolerated participation deviation before a node or committee is temporarily frozen. It dynamically scales the tolerance according to network activity and adversarial risk, ensuring stability without excessive conservatism.

- $\frac{|M_j^{on}|}{|M_j|}$: Online fraction of committee j (2)
- $|HashRate|$: Global network hashrate or stake.

Equation (2) measures the proportion of active members within a committee; when this ratio drops sharply, the system interprets it as potential partitioning or communication failure.

2) Multi-Signal Disconnection Proofs (k-of-m Rule)

The baseline subprotocol freezes accounts on a single connectivity test failure. This is upgraded to a multi-signal, short-window rule, requiring ≥k corroborating signals before applying freezes [16]. The signals considered are:

- S_1: Peer-ping failure (baseline test).
- S_2 : Relay telemetry skew (arrival-time deviations vs. relays).
- S_3: Attestation/participation drop (PoS committees).
- S_4: Header divergence score (fork distance/reorgs).

The decision rule is:

Mark-node

$i \in M_j^{off}$ if

$$\sum_{s \in \{S_1,\ldots,S_m\}} 1 \; [s \text{ triggered within window } W] \geq k \qquad (3)$$

Typical parameters are $k = 2$, $m \leq 4$, $W = 3\,slots$. Once marked, freezes are enforced using the robust threshold (Δj^*) [17]. Equation (3) enforces that a node is only frozen when multiple independent failure signals are triggered within a short window. This multi-signal consensus reduces false positives and ensures that disconnection decisions reflect genuine partition events rather than transient noise.

3) Integration with Base Sub-Protocol

- Transaction Freezing: Replace the single-test freeze condition with the *k*-of-*m* multi-signal rule within a W-slot window [18].
- Threshold Computation: Compute and publish Δj^* instead of Δj , using UCBs and haircut α [19].

B. Evaluation Metrics

The evaluation focused on both performance and security dimensions to capture the comprehensive impact of the proposed countermeasure. Performance was measured using Transactions Per Second/Throughput (TPS), defined as the number of transactions successfully finalized per second, and Latency (blocks/s), representing the average delay between block proposal and finalization [20]. Security was assessed using Consistency, quantified through the inconsistency rate, i.e., the fraction of experimental runs that resulted in divergent chain states across partitions.

III. RESULTS AND DISCUSSION

The proposed countermeasure robust freezing threshold (Δj^*) and multi-signal disconnection proofs were evaluated on Bitcoin (PoW) and Ethereum 2.0 (PoS) under bandwidth-constrained partitions. Experiments were run with 160 and 320 nodes and two partitioned networks (160 and 320 nodes). Statistical validation using two-tailed t-tests showed all key improvements were significant at $p < 0.05$. The baseline comprised unmodified Bitcoin Core v23 and Ethereum 2.0 Beacon Chain v1.5 without the proposed defense. Simulations used a scale-free (Barabási–Albert) topology in SimBlock 3.2, reflecting realistic peer connectivity under partitioned bandwidths.

Experiments were conducted using the SimBlock 3.2 simulator with 160 and 320 full nodes, scale-free topology, and bandwidth levels from 1024 Kbps to 32 Kbps. Transaction workloads were derived from Bitcoin Testnet and Ethereum Goerli traces (≈ 250 B per tx, 2 MB blocks) generated under Poisson ($\lambda = 3\ tx/s$) arrival.

A. Experimental Setup

The experimental setup was designed to rigorously evaluate the proposed countermeasure robust freezing threshold (Δj^*) combined with multi-signal disconnection proofs across two major blockchain platforms: Bitcoin, which operates under the PoW consensus, and Ethereum 2.0, which relies on PoS. To introduce stress scenarios and assess resilience, network bandwidth was systematically throttled to six levels: 1024, 512, 256, 128, 64, and 32 Kbps, each representing progressively severe partition conditions.

B. Ethereum 2.0 (PoS)

Ethereum exhibits sharper performance degradation under partitioned conditions. In the baseline system, partition-side throughput collapses completely below 512 Kbps, reaching zero TPS at 64 Kbps. In contrast, the proposed approach maintains stable throughput across all bandwidth levels, achieving 24 TPS at 512 Kbps (+140%), 18 TPS at 256 Kbps (+500%), and 12 TPS at 128 Kbps (+1100%). Even under the extreme constraints of 64 Kbps, where the baseline entirely fails, the system still sustains approximately 8 TPS. In the baseline configuration, block-generation latency increases from about 6 blocks/s at 512 Kbps to nearly 10 blocks/s at 64 Kbps, halting block finalization. Figure 2 presents throughput

variations for 160 and 320 nodes under dual-partition scenarios, while Figure 3 illustrates the corresponding latency behavior. Table I summarizes Ethereum 2.0 performance metrics across all bandwidth constraints.

TABLE I. ETHEREUM 2.0 RESULTS WITH BANDWIDTH

Bandwidth (Kbps)	Partition TPS (Orig)	Partition TPS (Proposed)	Δ throughput	Latency (Orig, blocks/s)	Latency (Proposed)	Δ latency
1024	25	28	+12%	3.2	3.2	0%
512	10	24	+140%	6.0	4.0	−33%
256	3	18	+500%	7.5	5.0	−33%
128	1	12	+1100%	8.5	5.5	−35%
64	0	8	(0→8)	9.5	6.0	−37%

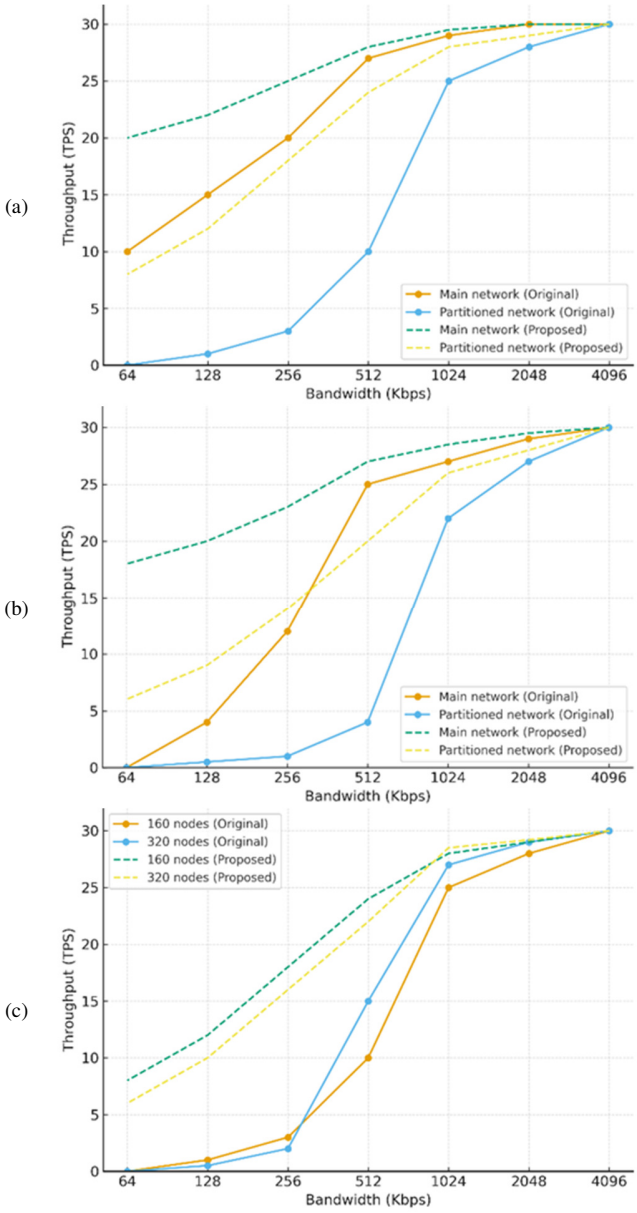

Fig. 2. Ethereum throughput: (a) 160, (b) 320, and (c) two partitions.

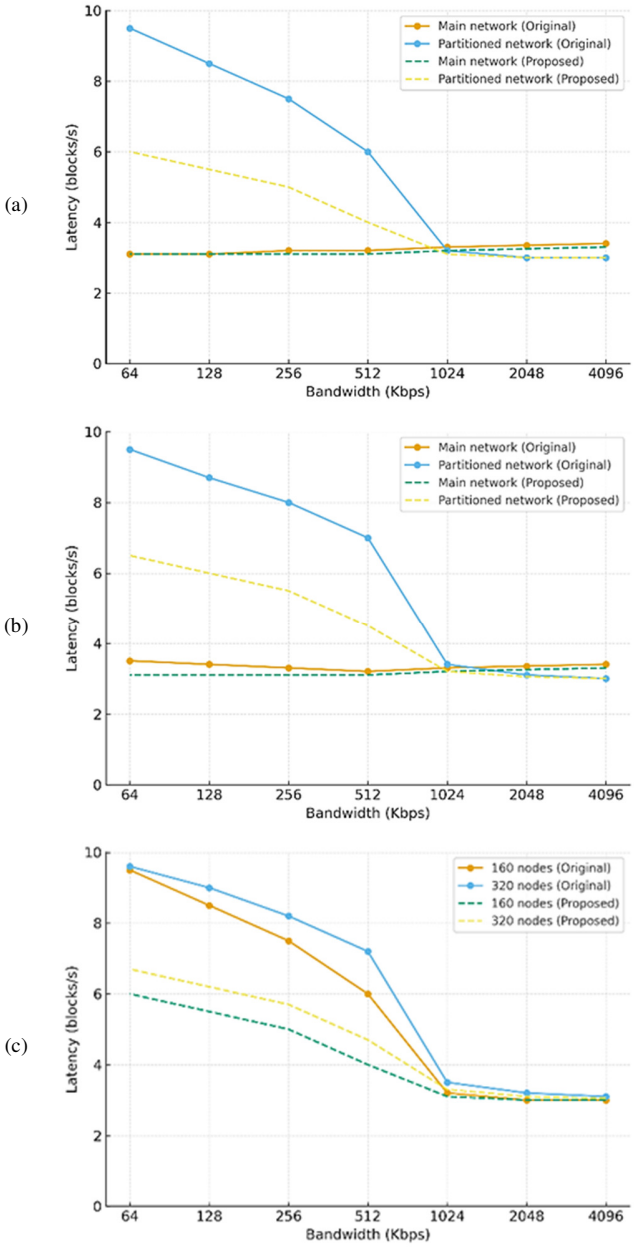

Fig. 3. Ethereum latency: (a) 160, (b) 320, and (c) two partitions.

C. Bitcoin (PoW)

In the baseline system, throughput on the partitioned side collapses sharply under constrained bandwidth. At 128 Kbps, throughput drops to only 2 TPS, and reaches zero at 32 Kbps. With the proposed improvements, throughput remains significantly higher, maintaining 3.6 TPS at 128 Kbps (+80%) and 2.2 TPS at 64 Kbps (+120%). These results indicate that the method transforms catastrophic collapse into a more gradual and manageable performance decline. Consistency improves correspondingly. Under the baseline setup, the inconsistency rate rises above 0.7 at 128 Kbps and reaches 1.0 at 64–32 Kbps, meaning that every run diverges. The proposed countermeasure reduces inconsistency to 0.25 at 128 Kbps and

0.35 at 64 Kbps, yielding an overall reduction of about 60%. Security also benefits directly. In the baseline, double-spend attacks succeed with a probability approaching 100% once bandwidth falls below 128 Kbps. Under the proposed design, the success probability never exceeds 15% at 64 Kbps. Figure 4 illustrates throughput variations for 160- and 320-node networks under dual-partition scenarios. Figure 5 presents the corresponding inconsistency trends, highlighting a clear reduction in fork occurrences and validation errors relative to the baseline configuration.

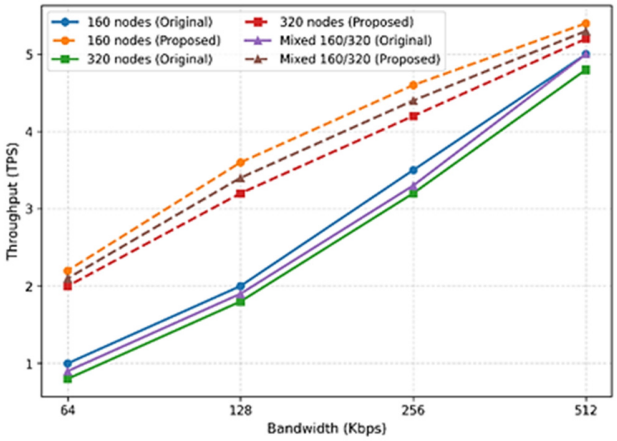

Fig. 4. Bitcoin throughput (160, 320, and two partitions).

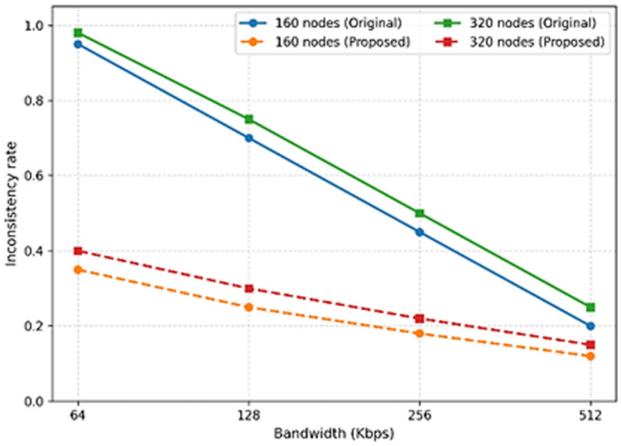

Fig. 5. Bitcoin inconsistency rates (160 and 320 nodes).

Figure 6 shows the probability of double-spending success. In the baseline, the success approaches 100% below 128 Kbps. The proposed method keeps the success probability near zero, at less than 15%, even at 64 Kbps. This confirms that double-spends are no longer economically viable. Table III summarizes Bitcoin outcomes. Partition throughput improves by +31-120%, while inconsistency falls by 40-64% across bandwidth levels. Table II shows the comparative throughput and consistency improvements of the proposed framework against existing blockchain resilience methods.

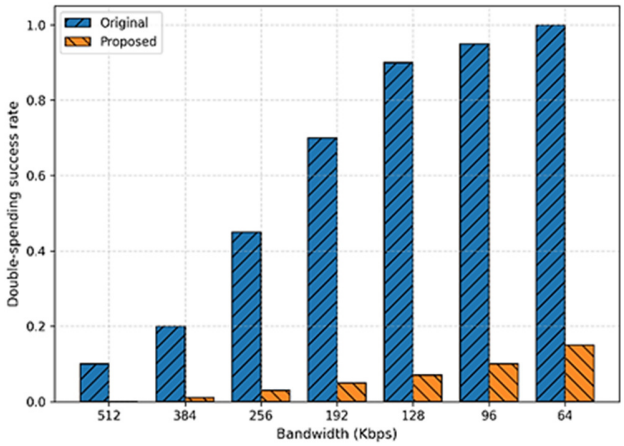

Fig. 6. Bitcoin double-spend success rates.

TABLE II. COMPARATIVE THROUGHPUT AND CONSISTENCY ANALYSIS

Method / Reference	Protocol type	Throughput improvement vs. baseline	Inconsistency reduction
Goldfish [6]	Ethereum 2.0 (PoS)	+350 % @ 256 Kbps	≈ 40 % lower inconsistency
SyncAttack Countermeasure [8]	Bitcoin (PoW)	+80 % @ 128 Kbps	≈ 30 % improvement
DiFastBit [12]	Bitcoin (PoW)	+95 % @ 256 Kbps	≈ 50 % improvement
Proposed work	Bitcoin (PoW) & Ethereum 2.0 (PoS)	+120 % (BTC) / +1100 % (ETH)	40–64 % reduction

TABLE III. BITCOIN RESULTS UNDER BANDWIDTH CONSTRAINTS

Bandwidth (Kbps)	Partition TPS (Orig)	Partition TPS (Proposed)	Δ Throughput	Inconsistency (Orig)	Inconsistency (Proposed)	Δ inconsistency
512	5.0	5.4	+8%	0.20	0.12	–40%
256	3.5	4.6	+31%	0.45	0.18	–60%
128	2.0	3.6	+80%	0.70	0.25	–64%
64	1.0	2.2	+120%	0.95	0.35	–63%

IV. CONCLUSION

This study demonstrates that blockchain systems such as Bitcoin and Ethereum 2.0 can be significantly more resilient to partition-induced vulnerabilities through lightweight enhancements that preserve existing consensus rules. By integrating a robust freezing threshold Δj^* to prevent overspending with multi-signal disconnection proofs to reduce false freezes, the proposed countermeasure sustains multi-fold throughput improvements up to +1100% in Ethereum and +120% in Bitcoin while bounding latency within 5–6 blocks per second. Inconsistency probability decreases by as much as

64%, and double-spend success is capped below 15% even under severely bandwidth-constrained conditions. These improvements are achieved with only modest gas overhead relative to block rewards, confirming the practicality of the approach for deployment in real-world blockchain environments. Future work will deploy the proposed countermeasure on Ethereum Holesky and Bitcoin Testnet to assess real-world scalability and latency behavior. Other planned extensions will evaluate heterogeneous network topologies, cross-layer interoperability across Layer-2 rollups and bridges, and adaptive reinforcement-learning-based adversary simulations to strengthen partition-tolerant and self-learning blockchain resilience.

DATASET AND CODE AVAILABILITY

The simulation datasets, source scripts, and configuration files used in this study are available from the corresponding author upon reasonable request.

REFERENCES

[1] B. U. I. Khan, K. W. Goh, M. F. Zuhairi, R. R. Putra, A. R. Khan, and M. Chaimanee, "A Scalability Enhancement Scheme for Ethereum Blockchains: A Graph-based Decentralized Approach," *Engineering, Technology & Applied Science Research*, vol. 14, no. 6, pp. 17725–17736, Dec. 2024, https://doi.org/10.48084/etasr.8465.

[2] J. Gomes, S. Khan, and D. Svetinovic, "Fortifying the Blockchain: A Systematic Review and Classification of Post-Quantum Consensus Solutions for Enhanced Security and Resilience," *IEEE Access*, vol. 11, pp. 74088–74100, 2023, https://doi.org/10.1109/ACCESS.2023.3296559.

[3] K. Nicolas, Y. Wang, G. C. Giakos, B. Wei, and H. Shen, "Blockchain System Defensive Overview for Double-Spend and Selfish Mining Attacks: A Systematic Approach," *IEEE Access*, vol. 9, pp. 3838–3857, 2021, https://doi.org/10.1109/ACCESS.2020.3047365.

[4] I. Khan, D. Somshekhar, N. Harikrishnan, P U Neetha, Suman, and D. Pavithra, "Hierarchical Meta-Reinforcement Learning for Uncertainty-Aware Resource Allocation in C-V2X Networks," in *2025 International Conference on Machine Learning and Autonomous Systems (ICMLAS)*, Prawet, Thailand, Mar. 2025, pp. 167–172, https://doi.org/10.1109/ICMLAS64557.2025.10968190.

[5] I. Khan and M. S. Haladappa, "Risk-Aware Multi-Agent Advantage Actor-Critic Based Resource Allocation for C-V2X Communication in Cellular Networks," *Proceedings of Engineering and Technology Innovation*, vol. 29, pp. 47–60, Feb. 2025, https://doi.org/10.46604/peti.2024.14136.

[6] F. D'Amato, J. Neu, E. N. Tas, and D. Tse, "Goldfish: No More Attacks on Ethereum?!," in *Financial Cryptography and Data Security*, vol. 14744, J. Clark and E. Shi, Eds. Springer Nature Switzerland, 2025, pp. 3–23.

[7] M. Doger, S. Ulukus, and N. Akar, "Double Spending Analysis of Nakamoto Consensus for Time-Varying Mining Rates with Ruin Theory," in *2025 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, Pisa, Italy, June 2025, pp. 1–9, https://doi.org/10.1109/ICBC64466.2025.11114481.

[8] M. Saad, S. Chen, and D. Mohaisen, "SyncAttack: Double-spending in Bitcoin Without Mining Power," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, Nov. 2021, pp. 1668–1685, https://doi.org/10.1145/3460120.3484568.

[9] K. Bruhwiler *et al.*, "Analyzing Soft and Hard Partitions of Global-Scale Blockchain Systems," in *2022 IEEE International Conference on Blockchain (Blockchain)*, Espoo, Finland, Aug. 2022, pp. 304–311, https://doi.org/10.1109/Blockchain55522.2022.00049.

[10] J. Ha, S. Baek, M. Tran, and M. S. Kang, "On the Sustainability of Bitcoin Partitioning Attacks," in *Financial Cryptography and Data Security*, vol. 13951, F. Baldimtsi and C. Cachin, Eds. Springer Nature Switzerland, 2024, pp. 166–181.

[11] S. N. Mighan, J. Mišić, and V. B. Mišić, "Probabilistic Analysis of Validator Lifecycle and Fork Resolution in Ethereum 2.0-Like PoS System," *IEEE Transactions on Network and Service Management*, vol. 22, no. 4, pp. 3732–3742, Aug. 2025, https://doi.org/10.1109/TNSM.2025.3573246.

[12] D. Melo, S. E. Pomares-Hernández, L. M. Rodríguez-Henríquez, and J. C. Pérez-Sansalvador, "DiFastBit: Transaction Differentiation Scheme to Avoid Double-Spending for Fast Bitcoin Payments," *Mathematics*, vol. 12, no. 16, Aug. 2024, Art. no. 2484, https://doi.org/10.3390/math12162484.

[13] K. Nayak, S. Kumar, A. Miller, and E. Shi, "Stubborn Mining: Generalizing Selfish Mining and Combining with an Eclipse Attack," in *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, Mar. 2016, pp. 305–320, https://doi.org/10.1109/EuroSP.2016.32.

[14] T. Neudecker and H. Hartenstein, "Network Layer Aspects of Permissionless Blockchains," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 838–857, 2019, https://doi.org/10.1109/COMST.2018.2852480.

[15] Y. Huang, Y. Zeng, F. Ye, and Y. Yang, "Incentive Assignment in Hybrid Consensus Blockchain Systems in Pervasive Edge Environments," *IEEE Transactions on Computers*, pp. 1–1, 2021, https://doi.org/10.1109/TC.2021.3122891.

[16] J. Das, S. A. A. Tasin, M. F. Rabbi, and M. S. Ferdous, "A Survey of Attacks on Blockchain Systems Using a Layer-based Approach," *Computer Networks*, vol. 265, June 2025, Art. no. 111274, https://doi.org/10.1016/j.comnet.2025.111274.

[17] M. Al-Bassam, A. Sonnino, V. Buterin, and I. Khoffi, "Fraud and Data Availability Proofs: Detecting Invalid Blocks in Light Clients," in *Financial Cryptography and Data Security*, vol. 12675, N. Borisov and C. Diaz, Eds. Springer Berlin Heidelberg, 2021, pp. 279–298.

[18] S. Bano *et al.*, "SoK: Consensus in the Age of Blockchains," in *Proceedings of the 1st ACM Conference on Advances in Financial Technologies*, Zurich Switzerland, Oct. 2019, pp. 183–198, https://doi.org/10.1145/3318041.3355458.

[19] J. Liu, C. Liu, M. Lin, and G. Xu, "Comprehensive survey of blockchain consensus mechanisms: Analysis, applications, and future trends," *Computer Networks*, vol. 272, Nov. 2025, Art. no. 111661, https://doi.org/10.1016/j.comnet.2025.111661.

[20] H S. K. Hemanth, G. K. Pawan, D. Anil, and N. Smitha, "An Efficient and Robust Reliable Data Aggregation in Wireless Sensor Networks," in *2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC)*, Coimbatore, India, Aug. 2021, pp. 1052–1057, https://doi.org/10.1109/ICESC51422.2021.9532959.