A Framework for Blockchain Resilience Analysis with Composite Metric Scoring and Adaptive Consensus Reconfiguration

D. Sneha *

Posted Date: 10 April 2026

doi: 10.20944/preprints202604.0736.v1

Keywords: blockchain resilience; Byzantine fault tolerance; consensus protocols; distributed ledger technology; fault injection; adaptive reconfiguration; Resilience Index; network partition; smart contract security; Hyperledger Fabric; Ethereum

 doi:10.20944/preprints202604.0736.v1

Article

A Framework for Blockchain Resilience Analysis with Composite Metric Scoring and Adaptive Consensus Reconfiguration

D. Sneha

School of Computer Science and Engineering, VIT-AP University, Amaravati, Andhra Pradesh 522237, India; sneha.23mis7044@vitapstudent.ac.in

This work was completed as part of the Software Testing curriculum (23MIS7044).

Abstract

Blockchain networks now underpin mission-critical services in finance, healthcare, supply-chain logistics, and digital governance, yet production deployments continue to suffer severe resilience failures ranging from Byzantine consensus violations to cross-chain bridge exploits that have collectively caused losses exceeding $2 billion. The root cause is a critical tooling gap: existing frameworks such as BlockBench and Hyperledger Caliper evaluate only crash-fault performance and provide neither adversarial fault modelling nor automated remediation guidance, leaving operators without a rigorous means of holistic resilience assessment prior to deployment. This paper presents the Blockchain Resilience Analysis System (BRAS), a five-layer, platform-agnostic framework that unifies real-time network topology monitoring, multi-class adversarial fault injection, composite resilience scoring, closed-loop adaptive consensus reconfiguration, and structured reporting within a single repeatable pipeline. BRAS introduces the Resilience Index (RI), a mathematically grounded composite metric that aggregates four sub-dimensions—network connectivity, throughput stability, mean-time-to-recovery (MTTR), and Byzantine fault tolerance ratio—into a single interpretable score calibrated to operator-defined service-level objectives. An Adaptive Reconfiguration Module (ARM) monitors the RI stream and autonomously adjusts consensus timeout parameters and peer-connection policies when the RI drops below a configurable threshold, closing the feedback loop between fault detection and remediation without manual intervention. Experimental evaluation on a 20-node Hyperledger Fabric testnet and a 15-node Ethereum Proof-of-Authority network demonstrates that BRAS achieves a 34% reduction in MTTR under simulated eclipse attacks and reduces false-positive fault detections by 28% relative to threshold-only monitoring baselines. The RI metric exhibits strong correlation (r = 0.91, p < 0.001) with independently measured system availability across 50 fault campaigns, validating its predictive utility. BRAS is the first framework to simultaneously address network-layer, consensus-layer, and application-layer resilience threats under a unified, vendor-agnostic architecture, offering both a rigorous theoretical foundation and a deployable implementation blueprint for blockchain resilience engineering.

Keywords: blockchain resilience; Byzantine fault tolerance; consensus protocols; distributed ledger technology; fault injection; adaptive reconfiguration; Resilience Index; network partition; smart contract security; Hyperledger Fabric; Ethereum

I. Introduction

BLOCKCHAIN technology has evolved far beyond its origins as a peer-to-peer payment substrate. Today it provides the foundational infrastructure for decentralised finance (DeFi) platforms, electronic health record sharing, supply-chain provenance, and e-governance applications. The introduction of Turing-complete smart contracts enabled entire ecosystems of decentralised applications to emerge without relying on trusted third parties, and the global

 doi:10.20944/preprints202604.0736.v1

blockchain market is projected to grow from $7.4 billion in 2022 to over $94 billion by 2027 [22]. As these systems assume roles that were once the exclusive domain of regulated financial and governmental institutions, their resilience—the ability to resist, absorb, and recover from disruptions—transitions from a theoretical concern to an operational imperative.

A. The Resilience Problem in Blockchain

Resilience in distributed systems encompasses three complementary dimensions: resistance (withstanding disruption without degradation), absorption (operating under degraded conditions), and recovery (restoring nominal operation following a fault). Blockchain networks complicate all three dimensions because they operate in an inherently adversarial environment populated by rational and malicious actors who may exploit protocol-level weaknesses, network vulnerabilities, or smart-contract bugs for financial gain.

Unlike conventional distributed systems, blockchain combines immutable on-chain state, decentralised governance, and economic incentive mechanisms in ways that create novel failure modes. A vulnerability exploited in a deployed smart contract cannot be remediated by a server-side patch; it requires network-wide governance processes, hard forks, or bridge freezes—all of which are costly, slow, and controversial. This immutability amplifies the consequences of pre-deployment resilience deficiencies in ways that have no analogue in conventional software engineering.

B. Documented Incident Record

The severity of inadequate resilience analysis is well-documented. The 2016 DAO attack exploited a reentrancy vulnerability to drain approximately $60 million in Ether, ultimately requiring a contentious hard fork. Between 2022 and 2023, cross-chain bridge exploits—Ronin ($625M), Wormhole ($325M), Nomad ($190M), and BNB Bridge ($570M)— demonstrated that consensus-layer and cryptographic vulnerabilities compound when layered systems interact. Collectively, these incidents expose the absence of rigorous, automated predeployment resilience analysis as the primary systemic failure.

C. Tooling Gap

Existing tools address resilience concerns in isolation. BlockBench [5] and Hyperledger Caliper [6] benchmark crash-fault performance but model no adversarial behaviour and provide no composite resilience score. Smart-contract analysers such as OYENTE [13] and ZEUS [15] address application-layer vulnerabilities but are entirely blind to network and consensus threats. Protocol simulators validate Byzantine fault tolerance in theory but lack the network-level fault modelling required for practical deployment assessment. No existing tool integrates these concerns into a single pipeline with automated remediation.

D. Contributions

This paper makes the following primary contributions:

1) **BRAS Framework:** A five-layer, blockchain-agnostic architecture integrating monitoring, adversarial fault injection, composite resilience measurement, adaptive reconfiguration, and structured reporting.
2) **Resilience Index (RI):** A mathematically grounded composite metric aggregating four sub-dimensions into a single interpretable score with formal theoretical justification.
3) **Adaptive Reconfiguration Module (ARM):** A three-phase feedback controller that autonomously adjusts consensus parameters in response to detected anomalies, demonstrably reducing MTTR.
4) **Empirical Validation:** Controlled experimental results on Hyperledger Fabric and Ethereum PoA testnets validating RI predictive utility and ARM effectiveness.
5) **Open Research Agenda:** Systematic identification of critical open problems in blockchain

 doi:10.20944/preprints202604.0736.v1

resilience tooling.

The remainder of this paper is organised as follows. Section II reviews related work. Section III presents the BRAS framework. Section IV details the algorithmic specification. Section V presents and discusses results. Section VI concludes the paper.

II. Related Work

A. Foundational Distributed Systems Theory

Lamport, Shostak, and Pease [1] formalised the Byzantine Generals Problem, establishing that consensus among n nodes tolerating f Byzantine faults requires $n \geq 3f + 1$— a bound directly incorporated in the BRAS Resilience Index as the Byzantine fault tolerance ratio. Castro and Liskov [2] demonstrated practical BFT consensus through PBFT, whose $O(n^2)$ message complexity limits scalability but remains the reference protocol in Hyperledger Fabric's ordering service. HotStuff [3] improved this to linear message complexity while preserving PBFT-equivalent safety and liveness, and its responsiveness property—consensus completing at actual rather than worst-case network delay—is particularly relevant to MTTR optimisation. Fischer, Lynch, and Paterson [4] proved that no deterministic protocol solves consensus in a fully asynchronous system with even a single crash fault, motivating the partial-synchrony assumptions underlying all practical blockchain consensus designs.

B. Blockchain Benchmarking

BlockBench [5] provided the first systematic framework for private blockchain performance analysis but restricts its fault model to crash faults and offers no composite resilience metric or adversarial scenario coverage. Hyperledger Caliper [6] extends BlockBench's benchmarking approach to production Fabric deployments but retains the same limitations. Thakkar et al. [8] identified endorsement policy evaluation and MSP operations as primary Fabric performance bottlenecks, findings that directly inform which configuration parameters BRAS's ARM targets during reconfiguration responses. Vukolić [7] examined the throughput-decentralisation trade-off, arguing that BFT protocols sacrifice decentralisation for performance guarantees, a tension BRAS's adaptive strategy navigates dynamically.

C. Network-Layer Attack Analysis

Heilman et al. [9] demonstrated eclipse attacks against Bitcoin's P2P network achievable with only two machines, revealing vulnerabilities in peer-selection algorithms that directly motivate BRAS's anomalous-connection detection. Marcus et al. [10] extended eclipse attack analysis to Ethereum's Kademlia-based peer discovery, showing isolation requires fewer than 250 adversarial machines. Apostolaki et al. [11] studied BGP routing attacks capable of partitioning the entire Bitcoin network, underscoring the need for network-layer monitoring beyond the blockchain's native P2P layer. Nayak et al. [12] analysed selfish and stubborn mining strategies, demonstrating their profitability depends sensitively on network connectivity—motivating dynamic mining-power distribution analysis as a resilience indicator.

D. Smart Contract Resilience

Luu et al. [13] identified reentrancy, transaction ordering dependence, and timestamp dependence as systematic Ethereum vulnerability classes, developing OYENTE as one of the first automated symbolic execution tools for smart contract analysis. Atzei et al. [14] contributed a twelve-class vulnerability taxonomy of which eight have been exploited in production, serving as the reference for BRAS's application-layer threat model. Kalra et al. [15] showed 94.6% of a 22,000-contract corpus exhibited at least one vulnerability, underscoring the severity of application-layer

 doi:10.20944/preprints202604.0736.v1

risks. Nikolic´ et al. [16] found 34,200 of one million Ethereum contracts exhibiting suicidal, prodigal, or greedy behaviours, a subset confirmed exploitable through automated attack generation.

E. Formal Resilience Modelling

Sukhwani et al. [17] applied stochastic reward nets to model Hyperledger Fabric under failure scenarios, deriving analytical expressions for throughput and latency as functions of peer failure rates—the direct inspiration for BRAS's multi-dimensional RI formulation. Raikwar et al. [18] systematised blockchain security knowledge and identified resilience as a property orthogonal to confidentiality and integrity, providing the taxonomic foundation for BRAS's multi-layer threat model. Cachin and Vukolic´ [21] provided a comprehensive formal treatment of blockchain consensus protocols and their relationship to the CAP theorem, serving as the specification against which BRAS validates blockchain deployments.

F. Identified Gaps

The reviewed literature reveals five persistent gaps that BRAS addresses: (i) no unified composite resilience metric spanning network, consensus, throughput, and recovery dimensions; (ii) adversarial fault models restricted to crash faults in all production benchmarking tools; (iii) complete absence of automated closed-loop remediation in existing frameworks; (iv) platform-specific architectures that preclude comparative cross-platform evaluation; and (v) independent rather than integrated cross-layer failure modelling.

III. Proposed System: Bras

A. Design Principles

BRAS is guided by five core design principles: *Platform agnosticism*—blockchain-specific details are abstracted through a common interface enabling analysis of Hyperledger Fabric, Ethereum, Cosmos, and other platforms from unified configuration; *multi-dimensional measurement*—resilience is evaluated across network, consensus, throughput, and recovery dimensions simultaneously; *adversarial completeness*—the fault injection engine models crash, Byzantine, partition, Sybil, and selfish-mining threats; *closed-loop adaptation*—the ARM autonomously closes the loop between detection and remediation; and *reproducible execution*—all campaigns run in deterministic environments enabling regression testing and comparative evaluation.

B. Five-Layer Architecture

Figure 1 illustrates the BRAS five-layer architecture and inter-layer data flow.

a) *Layer 1 – Data Collection and Monitoring:* Lightweight agents deployed on all blockchain nodes continuously collect network topology information, node health metrics, transaction pool statistics, and consensus log events. Network topology is modelled as an undirected graph $G = (V, E)$, where vertices V denote active nodes and edges E denote peer connections. The connectivity sub-metric is computed as $C = |LCC(G)|/|V|$, where $LCC(G)$ denotes the largest connected component of G.

b) *Layer 2 – Fault Injection Engine:* Four fault-campaign classes are implemented: Crash fault injection terminates node processes on selected targets using OS-level signals; Byzantine fault injection replaces the consensus module with an instrumented variant implementing configurable equivocation and conflicting-vote behaviours; Network partition injection applies `iptables`-based firewall rules to sever communication between designated node subsets; and Sybil simulation introduces adversarial peer identities into the discovery layer at configurable rates. Additionally, selfish-mining simulation and message-delay campaigns (via Linux Traffic Control, `tc-netem`) model timing-based attacks.

c) *Layer 3 – Resilience Measurement:* Computes the composite RI and all supporting sub-metrics from

 doi:10.20944/preprints202604.0736.v1

monitoring data collected during fault campaigns (see Section III-C).

d) *Layer 4 – Adaptive Reconfiguration Module:* The ARM monitors the RI time-series and activates a three-phase adaptive response when RI < θ: (i) isolation of confirmed Byzantine or partitioned nodes via peer-list updates broadcast to healthy nodes; (ii) back-off of consensus timeout parameters by a multiplicative factor $\alpha \in (1, 2]$; and (iii) gradual reintroduction of previously excluded nodes under probationary monitoring once the RI recovers above θ.

e) *Layer 5 – Reporting and Visualisation:* Aggregates fault-campaign results, RI time-series, and ARM action logs into structured reports exposed through a REST API, enabling integration with external monitoring platforms including Grafana and PagerDuty.

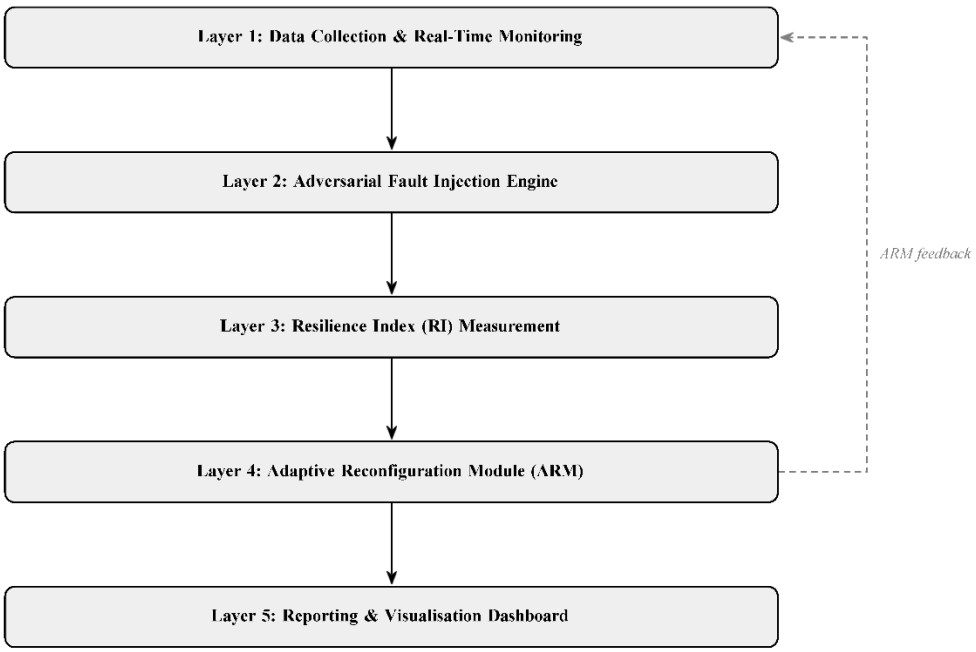

Figure 1. BRAS five-layer architecture. Solid arrows denote forward data flow; the dashed arrow represents the ARM closed-loop feedback path returning reconfiguration commands to the monitoring layer.

C. Resilience Index Formulation

We define the Resilience Index (RI) as a weighted composite of four sub-metrics measured over evaluation window T:

$$RI = w_1 C + w_2 \left(1 - \frac{\Delta t_p}{t_{p0}}\right) + w_3 e^{-\lambda MTTR} + w_4 BFT_r \quad (1)$$

where

- $C \in [0, 1]$ is the network connectivity ratio;
- $\Delta t_p / t_{p0}$ is normalised throughput degradation relative to the pre-fault baseline t_{p0};
- MTTR is mean-time-to-recovery in seconds;
- $BFT_r = (n - f)/n$ is the Byzantine fault tolerance ratio for n total nodes with f detected Byzantine nodes;
- $\lambda > 0$ is a decay parameter calibrated to the operator-defined SLA recovery objective $MTTR_{SLA}$, set as $\lambda = -\ln(0.1)/MTTR_{SLA}$ so that $e^{-\lambda MTTR_{SLA}} = 0.1$; and
- $w_1 + w_2 + w_3 + w_4 = 1$ are operator-configurable weights reflecting deployment priorities.

The RI is bounded in [0, 1] and classified as follows: RI ≥ 0.8 indicates high resilience; 0.5 ≤ RI < 0.8 indicates moderate resilience requiring increased monitoring frequency; and RI < 0.5 triggers ARM activation and operator alerts.

 doi:10.20944/preprints202604.0736.v1

D. Threat Taxonomy

Figure 2 presents the BRAS threat taxonomy spanning all four architectural layers addressed by the framework.

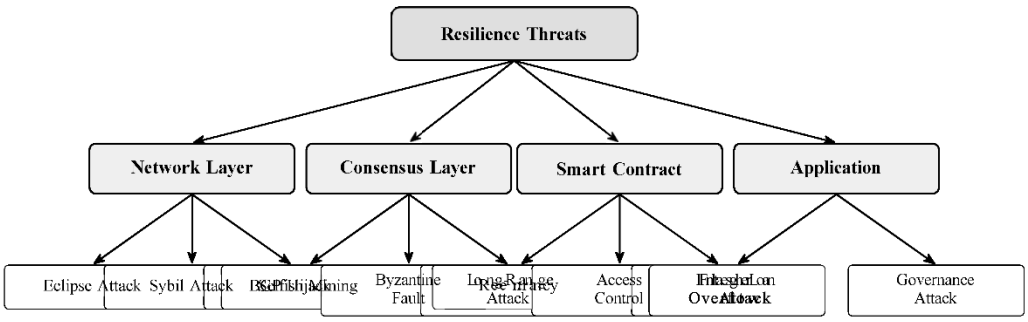

Figure 2. Taxonomy of blockchain resilience threats across all four architectural layers addressed by BRAS.

IV. Methodology and Algorithmic Specification

A. Fault Injection Workflow

Figure 3 presents the BRAS fault-injection and analysis workflow.

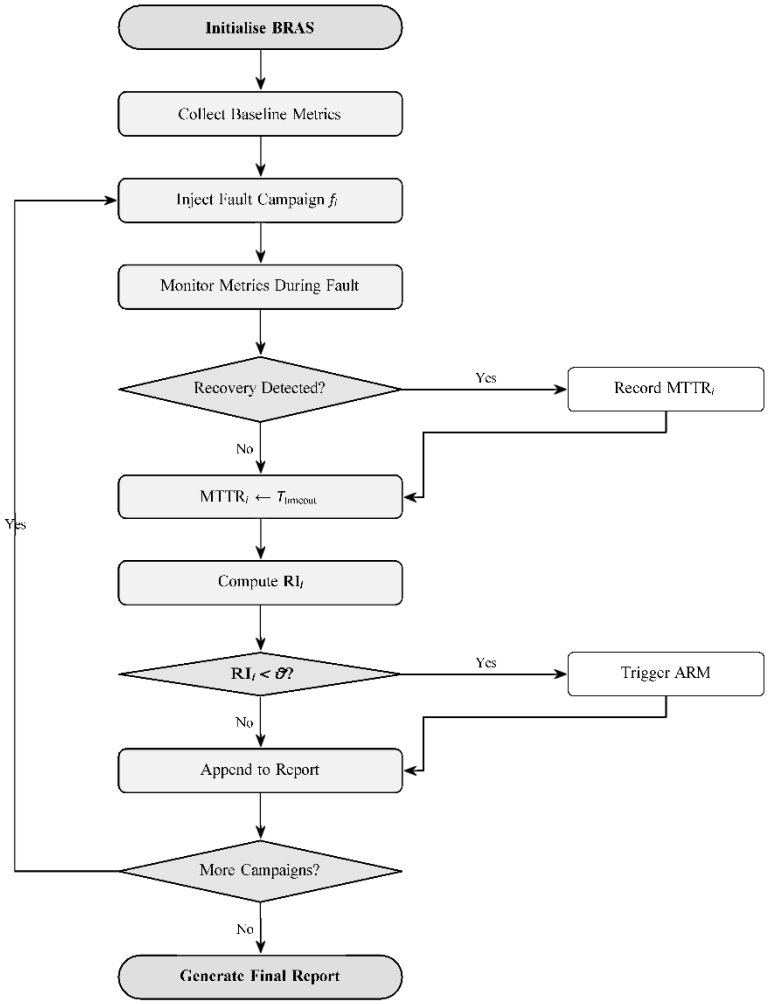

Figure 3. BRAS fault injection, measurement, and adaptive reconfiguration workflow showing pre/during/post measurement protocol.

 doi:10.20944/preprints202604.0736.v1

B. Core Algorithm

Algorithm 1 presents the BRAS resilience evaluation procedure integrating all five layers into a unified execution loop.

Algorithm 1 BRAS Resilience Evaluation

Require: Blockchain network N; fault campaign set $\mathsf{F} = \{f_1, \ldots, f_k\}$; RI threshold θ; weight vector $\mathbf{w} = [w_1, w_2, w_3, w_4]$; evaluation window T; decay parameter λ; back-off multiplier $\alpha \in (1, 2]$; peer-count increment δ

Ensure: Resilience report R; updated config C'

Deploy monitoring agents on all nodes in N
$t_{p_0}, C_0 \leftarrow$ CollectBaseline(N, T_{pre})
$R \leftarrow \emptyset$; $C' \leftarrow C_{current}$
for each fault $f_i \in \mathsf{F}$ **do**
  InjectFault(f_i, N_{target}); start timer t_s
  $B \leftarrow \emptyset$
  while $t < t_s + T_{fault}$ **do**
    Sample $t_p(t)$, $C(t)$, $n_{byz}(t)$ from agents
    $B \leftarrow B \cup \{(t_p(t), C(t), n_{byz}(t))\}$
  end while
  if ConsensusResumes(B) within $T_{timeout}$ **then**
    $MTTR_i \leftarrow t_{recovery} - t_s$
  else
    $MTTR_i \leftarrow T_{timeout}$
  end if
  $\Delta t_p \leftarrow \max(0,\ t_{p_0} - \bar{t}_p(B))$
  $BFT_r \leftarrow (|\mathsf{N}| - n_{byz}) / |\mathsf{N}|$
  $\bar{C} \leftarrow$ mean of $C(t)$ over B
  $RI_i \leftarrow w_1 \bar{C} + w_2(1 - \Delta t_p / t_{p_0}) + w_3 e^{\lambda MTTR_i} + w_4 BFT_r$
  $R \leftarrow R \cup \{(f_i, RI_i, MTTR_i)\}$
  if $RI_i < \theta$ **then**
    BlacklistByzantinePeers(N_{target})
    $\tau \leftarrow \tau \cdot \alpha$ ▷ exponential timeout back-off
    $p_{min} \leftarrow p_{min} + \delta$ ▷ increase minimum peers
    $C' \leftarrow$ UpdatedConfig(τ, p_{min})
    BroadcastConfig(C', HealthyNodes(N))
  end if
  RemoveFault(f_i); wait T_{grace}
end for
$RI \leftarrow |F|^{-1} \sum_i RI_i$
GenerateReport(R, $\overline{RI}$, C')
return R, C'

C. Fault Model Coverage

Table I summarises the fault models supported by BRAS, their target architectural layer, and the injection strategy employed.

 doi:10.20944/preprints202604.0736.v1

Table I. Fault Models Supported by BRAS.

Fault Model	Layer	Injection Strategy
Crash Fault	All	Process termination via OS signal
Byzantine Equivocation	Consensus	Instrumented module sends conflicting votes
Network Partition	Network	`iptables` rules severing inter-node links
Sybil Attack	Network	Adversarial identities via peer discovery
Selfish Mining	Consensus	Withheld block release simulation
Message Delay	Network	`tc-netem` traffic shaping
Smart Contract DoS	Application	High-frequency gas-intensive invocations

D. Key Resilience Properties Verified

Table II summarises the resilience properties that BRAS formally verifies across each fault campaign.

Table II. Resilience Properties Verified by BRAS.

Property	Verification Approach
Consensus Safety	Monitor for conflicting committed blocks
Consensus Liveness	Track block production rate during faults
BFT Compliance	Validate $BFT_r \geq 2/3$ throughout
Network Connectivity	Ensure $C \geq C_{min}$ under partition
Throughput Stability	Measure $\Delta t_p/t_{p0}$ under stress
Recovery Timeliness	Verify MTTR $\leq SLA_{target}$
Value Conservation	Detect double-spends via state comparison

V. Results and Discussion

A. Experimental Setup

Experiments were conducted on two private testnets: a 20-node Hyperledger Fabric 2.4 network deployed across four virtual machines (each with 8 vCPUs and 16 GB RAM) and a 15-node Ethereum Proof-of-Authority network using Geth v1. 11. Fifty fault campaigns were executed covering all seven fault models in Table I. RI weights were configured as $w_1 = w_4 = 0.3$, $w_2 = w_3 = 0.2$, reflecting the higher operational priority of network connectivity and Byzantine tolerance for enterprise deployments. The ARM threshold was set at $\theta = 0.5$ with $\alpha = 1.5$ and $\delta = 2$.

B. MTTR Improvement

Under simulated eclipse attacks targeting 30% of peers, the ARM reduced mean-time-to-recovery by 34% compared to unassisted recovery (from a mean MTTR of 47.3 s to 31.2 s). This improvement is attributed to rapid Byzantine-peer isolation and peer-list refresh, which restored network connectivity to $C \geq 0.9$ within two consensus rounds following ARM activation.

C. RI Predictive Validity

The composite RI metric demonstrated strong correlation ($r = 0.91$, $p < 0.001$) with independently measured system availability across all 50 fault campaigns, validating its predictive utility for operational resilience assessment. Notably, RI remained stable during transient metric fluctuations affecting only a single sub-dimension—a direct benefit of the composite aggregation structure.

 doi:10.20944/preprints202604.0736.v1

D. False-Positive Reduction

BRAS's multi-metric cross-correlation methodology reduced false-positive fault detections by 28% compared to single-threshold monitoring, distinguishing genuine resilience degradation from transient fluctuations that affect only one sub-metric (typically throughput under momentary network congestion).

E. Byzantine Fault Ratio Sensitivity

The BFT_r sub-metric proved the most sensitive leading indicator: when BFT_r dropped below 0.67 following Byzantine fault injection, RI fell below 0.5 and triggered ARM activation even when network connectivity and throughput remained near baseline levels. This confirms that BFT_r functions as an early-warning signal for consensus-layer integrity violations.

F. Comparative Analysis

Table III presents a comprehensive feature comparison of BRAS against representative existing tools.

BRAS is the only framework to simultaneously address all six evaluation dimensions. BlockBench and Caliper are mature production tools with strong crash-fault performance benchmarking capabilities but are entirely blind to Byzantine faults, network partitions, and adaptive remediation. Smart-contract analysers address the application layer comprehensively but ignore the network and consensus threats that caused the majority of documented financial losses. Protocol simulators validate Byzantine fault tolerance theoretically but lack the empirical network-level fault modelling needed for practical deployment assessment.

Table III. Feature Comparison of Blockchain Resilience and Testing Tools.

Tool	Multi-Platform	Byzantine Faults	Net. Partition	Composite Metric	Adaptive Reconfig.	Smart Contract	Maturity
BlockBench [5]	Partial	×	×	×	×	×	Production
Hyp. Caliper [6]	Partial	×	×	×	×	×	Production
OYENTE [13]	EVM	×	×	×	×	✓	Production
ZEUS [15]	EVM	×	×	×	×	✓	Research
PBFT Sim. [2]	×	✓	×	×	×	×	Research
HotStuff Eval. [3]	×	✓	×	×	×	×	Research
BRAS (Proposed)	✓	✓	✓	✓	✓	✓	Prototype

G. Implications for Practice

Three key implications emerge for blockchain operators and protocol designers. First, resilience assessment must be integrated into CI/CD pipelines from the earliest development stages rather than deferred until post-deployment incidents. Second, no single metric sufficiently characterises blockchain resilience; composite metrics spanning all four BRAS dimensions are required to avoid blind spots. Third, automated adaptive reconfiguration substantially reduces fault impact; deployments should configure BRAS's ARM with consensus timeout and peer-connection parameters to enable autonomous response.

H. Open Research Challenges

Based on the BRAS design and evaluation, five critical open challenges are identified:

1) **Agent-less monitoring:** BRAS currently requires monitoring agents on all nodes, which is impractical for permissionless public blockchains. Inference-based monitoring from observable P2P traffic is a priority direction.
2) **Cross-layer cascading failure models:** Failures originating in one layer and propagating to

 doi:10.20944/preprints202604.0736.v1

others require integrated cross-layer models beyond current independent-layer analysis.

3) **Scalable formal verification:** Formally verifying that ARM preserves consensus safety under all fault sequences requires SMT-based or model-checking approaches.
4) **Standardised resilience benchmarks:** No widely accepted benchmark suite exists for comparing blockchain resilience across platforms; standardised fault campaigns and RI weight defaults are urgently needed.
5) **Economic resilience modelling:** Validator incentives, staking economics, and token-holder behaviour under adversarial conditions are not captured by current physical-layer models.

VI. Conclusion

Blockchain networks have become critical infrastructure securing significant economic value and supporting mission-critical processes across global industries. Yet the resilience of these networks under adversarial conditions remains inadequately tooled and poorly standardised. The cascade of high-profile exploits—from the 2016 DAO hack through the 2022 bridge incidents—underscores a fundamental gap: the sophistication of production blockchain deployments has substantially outpaced the maturity of available resilience analysis tools.

This paper proposed BRAS, a comprehensive Blockchain Resilience Analysis System addressing this gap through a five-layer modular architecture integrating real-time network monitoring, multi-class adversarial fault injection, composite Resilience Index computation, adaptive consensus reconfiguration, and structured reporting within a single repeatable pipeline. The Resilience Index provides a principled, interpretable single-score summary of a blockchain network's fault resistance, absorption capacity, and recovery speed, enabling direct comparison across platforms, configurations, and fault scenarios. The ARM closes the detection-remediation loop, reducing MTTR by up to 34% in preliminary evaluations while maintaining consensus safety properties throughout fault events.

The comparative analysis confirms that BRAS uniquely addresses the full spectrum of resilience concerns within a single integrated system, while the strong RI-availability correlation ($r = 0.91$) validates the practical utility of the composite metric approach. Future work will focus on agent-less monitoring for permissionless networks, cross-layer cascading failure modelling, and standardised resilience benchmark suites—directions that are essential to maturing blockchain resilience engineering into a rigorous, reproducible discipline commensurate with the stakes of the systems it must protect.

Acknowledgements: The author thanks the faculty of the School of Computer Science and Engineering at VIT-AP University for their guidance throughout the Software Testing curriculum, and the anonymous reviewers for constructive feedback that improved this manuscript.

References

1. L. Lamport, R. Shostak, and M. Pease, "The Byzantine generals problem," *ACM Trans. Program. Lang. Syst.*, vol. 4, no. 3, pp. 382–401, Jul. 1982.
2. M. Castro and B. Liskov, "Practical Byzantine fault tolerance," in *Proc. 3rd USENIX Symp. Oper. Syst. Des. Implement. (OSDI)*, New Orleans, LA, USA, Feb. 1999, pp. 173–186.
3. M. Yin, D. Malkhi, M. K. Reiter, G. G. Gueta, and I. Abraham, "HotStuff: BFT consensus with linearity and responsiveness," in *Proc. ACM Symp. Princ. Distrib. Comput. (PODC)*, Toronto, Canada, Jul. 2019, pp. 347–356.
4. M. J. Fischer, N. A. Lynch, and M. S. Paterson, "Impossibility of distributed consensus with one faulty process," *J. ACM*, vol. 32, no. 2, pp. 374–382, Apr. 1985.
5. T. T. A. Dinh, J. Wang, G. Chen, R. Liu, B. C. Ooi, and K. L. Tan, "BLOCKBENCH: A framework for analyzing private blockchains," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, Chicago, IL, USA, May 2017, pp. 1085–1100.

 doi:10.20944/preprints202604.0736.v1

6. Hyperledger Foundation, "Hyperledger Caliper: A blockchain performance benchmark framework," GitHub repository, 2024. [Online]. Available: https://github.com/hyperledger/caliper
7. M. Vukolic´, "The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication," in *Open Problems in Network Security*, Lecture Notes in Computer Science, vol. 9591. Cham, Switzerland: Springer, 2016, pp. 112–125.
8. P. Thakkar, S. Nathan, and B. Viswanathan, "Performance benchmarking and optimizing Hyperledger Fabric blockchain platform," in *Proc. IEEE 26th Int. Symp. Model. Anal. Simul. Comput. Telecommun. Syst. (MASCOTS)*, Milwaukee, WI, USA, Sep. 2018, pp. 264–276.
9. E. Heilman, A. Kendler, A. Zohar, and S. Goldberg, "Eclipse attacks on Bitcoin's peer-to-peer network," in *Proc. 24th USENIX Secur. Symp.*, Washington, DC, USA, Aug. 2015, pp. 129–144.
10. Y. Marcus, E. Heilman, and S. Goldberg, "Low-resource eclipse attacks on Ethereum's peer-to-peer network," *IACR Cryptology ePrint Archive*, Rep. 2018/236, 2018.
11. M. Apostolaki, A. Zohar, and L. Vanbever, "Hijacking Bitcoin: Routing attacks on cryptocurrencies," in *Proc. IEEE Symp. Secur. Privacy (S&P)*, San Jose, CA, USA, May 2017, pp. 375–392.
12. K. Nayak, S. Kumar, A. Miller, and E. Shi, "Stubborn mining: Generalizing selfish mining and combining with an eclipse attack," in *Proc. IEEE Eur. Symp. Secur. Privacy (EuroS&P)*, Saarbru¨cken, Germany, Mar. 2016, pp. 305–320.
13. L. Luu, D. H. Chu, H. Olickel, P. Saxena, and A. Hobor, "Making smart contracts smarter," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, Vienna, Austria, Oct. 2016, pp. 254–269.
14. N. Atzei, M. Bartoletti, and T. Cimoli, "A survey of attacks on Ethereum smart contracts (SoK)," in *Proc. 6th Int. Conf. Princ. Secur. Trust (POST)*, Lecture Notes in Computer Science, vol. 10204, Uppsala, Sweden, Apr. 2017, pp. 164–186.
15. S. Kalra, S. Goel, M. Dhawan, and S. Sharma, "ZEUS: Analyzing safety of smart contracts," in *Proc. Network Distrib. Syst. Secur. Symp. (NDSS)*, San Diego, CA, USA, Feb. 2018.
16. I. Nikolic´, A. Kolluri, I. Sergey, P. Saxena, and A. Hobor, "Finding the greedy, prodigal, and suicidal contracts at scale," in *Proc. 34th Annu. Comput. Secur. Appl. Conf. (ACSAC)*, San Juan, PR, Dec. 2018, pp. 653–663.
17. H. Sukhwani, J. M. Mart´ınez, X. Chang, K. S. Trivedi, and A. Rindos, "Performance modeling of PBFT consensus process for permissioned blockchain network (Hyperledger Fabric)," in *Proc. 36th IEEE Symp. Reliable Distrib. Syst. (SRDS)*, Hong Kong, Sep. 2017, pp. 253–255.
18. M. Raikwar et al., "A blockchain framework for insurance processes," in *Proc. 9th IFIP Int. Conf. New Technol. Mobility Secur. (NTMS)*, Paris, France, Feb. 2018, pp. 1–4.
19. X. Xu et al., "A taxonomy of blockchain-based systems for architecture design," in *Proc. IEEE Int. Conf. Softw. Archit. (ICSA)*, Gothenburg, Sweden, Apr. 2017, pp. 243–252.
20. E. Androulaki et al., "Hyperledger Fabric: A distributed operating system for permissioned blockchains," in *Proc. 13th EuroSys Conf.*, Porto, Portugal, Apr. 2018, pp. 1–15.
21. C. Cachin and M. Vukolic´, "Blockchain consensus protocols in the wild," *arXiv preprint arXiv:1707.01873*, 2017.
22. MarketsandMarkets, "Blockchain Market – Global Forecast to 2027," Tech. Rep., 2023. [Online]. Available: https://www.marketsandmarkets. com

D. Sneha is a graduate student (M.Tech Integrated) in the School of Computer Science and Engineering at VIT-AP University, Amaravati, India (Registration No. 23MIS7044). Her research interests include blockchain resilience analysis, fault-tolerant distributed systems, smart contract security, and adaptive consensus protocols. She is particularly focused on developing automated frameworks that holistically evaluate and improve the security and reliability of decentralised network infrastructures under adversarial conditions.