Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.068273

ARTICLE

3RVAV: A Three-Round Voting and Proof-of-Stake Consensus Protocol with Provable Byzantine Fault Tolerance

Abeer S. Al-Humaimeedy*

Software Engineering Department, King Saud University, P.O. Box 4545, Riyadh, 145111, Saudi Arabia

*Corresponding Author: Abeer S. Al-Humaimeedy. Email: abeer@ksu.edu.sa

Received: 24 May 2025; Accepted: 21 August 2025; Published: 23 October 2025

ABSTRACT: This paper presents 3RVAV (Three-Round Voting with Advanced Validation), a novel Byzantine Fault Tolerant consensus protocol combining Proof-of-Stake with a multi-phase voting mechanism. The protocol introduces three layers of randomized committee voting with distinct participant roles (Validators, Delegators, and Users), achieving $\left(\frac{4}{5}\right)$-threshold approval per round through a verifiable random function (VRF)-based selection process. Our security analysis demonstrates 3RVAV provides $1-\left(1-\frac{s}{n}\right)^{3k}$ resistance to Sybil attacks with n participants and stake s, while maintaining $O(kn\log n)$ communication complexity. Experimental simulations show 3247 TPS throughput with 4-s finality, representing a 5.8× improvement over Algorand's committee-based approach. The proposed protocol achieves approximately 4.2-s finality, demonstrating low latency while maintaining strong consistency and resilience. The protocol introduces a novel punishment matrix incorporating both stake slashing and probabilistic blacklisting, proving a Nash equilibrium for honest participation under rational actor assumptions.

KEYWORDS: Byzantine fault tolerant; proof-of-stake; verifiable random function; Sybil attack resistance; Nash equilibrium; committee voting

1 Introduction

Modern blockchain systems face a fundamental trilemma in achieving simultaneous decentralization, security, and scalability [1]. While consensus protocols form the backbone of blockchain operations, existing approaches exhibit critical limitations in real-world deployments. With Proof-of-Work [2], a high amount of electricity is needed, and the rate of transactions drops, making it difficult for PBFT [3] and other BFT protocols to handle more than 100 nodes. Recent voting-based mechanisms [4,5] attempt to address these issues through committee selection, but introduce new attack surfaces in Sybil resistance and fairness guarantees.

Earlier studies of decentralized voting protocols have pointed out three problems that remain. Often, using committees in blockchain [6,7] compromises decentralisation, making nodes the sole points that, if weak or failed, will impair every operation. Second, there are no game-theoretic punishment models in the current ways of voting, which allows logical opponents to take advantage of the reward features [8,9]. In addition, systems based on stakes [10] tend to let wealthy participants influence and decide most of the consensus matters. Despite the use of reputation, research by Liao and Cheng [11] finds that voting systems are still open to attacks by collaborators, unless there is randomness in the way the committee is formed.

Recently, verifiable random functions (VRFs) [3] and threshold cryptography have given us better tools to deal with these issues. In [5], the authors show that using multi-stage voting with rotating committees can speed up consensus in consortium blockchains by almost 60%, but centralized coordination may introduce governance problems. In the same way, Li et al.'s Proof-of-Vote (PoV) [12] supports 1200 TPS under controlled circumstances but cannot resist attacks that aim to change the compositions of the committees. Consensus mechanisms are fundamental to achieving trust, scalability, and efficiency in decentralized systems. Recent studies have explored different approaches to strengthen fairness and resilience. A trusted consensus fusion scheme for decentralized collaborative learning in large-scale IoT domains has been introduced to improve coordination and reliability [13]. To address decentralization challenges, a game-theoretic and randomness-based consensus protocol has been proposed, enhancing fairness while mitigating centralization risks [14]. In addition, a multi-chain token-backed voting framework has been developed to support decentralized governance and decision-making in blockchain networks [15]. These contributions collectively highlight the trend toward scalable, fair, and robust consensus protocols for diverse blockchain applications. Threshold cryptography enables secure collaborative key operations where no single party holds full signing authority, and is foundational to our committee authorization model [16]. This drives us to look into a voting system that does not use coordinators and weights vote importance, while also being able to resist Byzantine attacks in all states of synchrony.

The paper discusses 3RVAV, a consensus protocol that looks at blockchain voting from a different perspective:

(a) A triple-committee architecture with VRF-based selection [3] ensuring $(4/5)^3$ probabilistic finality while maintaining $O(kn \log n)$ communication complexity
(b) A dynamic punishment matrix integrating stake slashing [10], probabilistic blacklisting [11], and reputation decay to establish ϵ-Nash equilibrium for honest participation
(c) A fork resolution mechanism combining transaction timestamp ordering [13] with Borda count weighting [8] to achieve deterministic chain selection

Our protocol advances the state-of-the-art in three dimensions. First, we prove 3RVAV provides $1 - \left(1 - \frac{s}{n}\right)^{3k}$ Sybil resistance for n participants and adversary stake s, significantly improving over Algorand's $1 - \left(1 - \frac{s}{n}\right)^{k}$ [14]. Second, experimental simulations demonstrate 3247 TPS throughput with 4.2-s finality, outperforming DRBFT [6] by 2.4× while maintaining 20% Byzantine tolerance. Third, the novel punishment calculus reduces collusion incentives by 38% compared to existing stake-slashing models [15], as quantified through evolutionary game theory simulations. Unlike traditional BFT-PoS protocols that struggle with quadratic messaging or fixed-role centralization, 3RVAV introduces entropy-driven committee rotation, probabilistic blacklisting, and Nash-incentivized honesty to overcome scalability, fairness, and Sybil attack limitations simultaneously.

The remainder of this paper is organized as follows: Section 2 details the 3RVAV protocol design and cryptographic foundations. Section 3 presents formal security proofs and economic analysis. Section 4 evaluates performance against state-of-the-art alternatives. Section 5 discusses broader implications and future work.

2 Protocol Design

The 3RVAV consensus protocol is designed to address the fundamental challenges of blockchain scalability, security, and decentralization. By integrating stake-weighted committee selection, threshold cryptographic verification, and Borda count-based voting, the protocol ensures efficient and secure transaction validation while maintaining resistance to Sybil attacks and collusion. Unlike traditional ways of making

decisions in blockchains like Proof-of-Work (PoW) or Practical Byzantine Fault Tolerance (PBFT), which have high processing costs or struggle to handle large numbers of users, 3RVAV uses something called Verifiable Random Functions (VRFs) to help pick a fair and random set of validators. The protocol goes through three steps: first, someone suggests a change to the system, then it goes through a few rounds of voting to decide if it's valid, and finally, it gets added to the main blockchain if most of the votes are yes. This multi-round process helps lower the chances of bad blocks getting added to the blockchain by making sure at least a certain number of people approve the block before it can be added. Additionally, a system that can adjust punishments in real time is there to ensure nodes act honestly by taking away rewards from bad behaviour and adding nodes to a list where they can't participate anymore. Furthermore, using a process called rank-weighted chain selection ensures that transactions are decided in the same way every time and reduces questions about which ones count the most.

Fig. 1 lays out precisely what happens in the 3RVAV consensus protocol: transactions are first found valid, then voted on, and finally, they are added to the blockchain. The process first involves creating the network and using VRFs to select the leader randomly and fairly in proposing blocks. The elected leader makes a block proposal that goes through three rounds of voting.

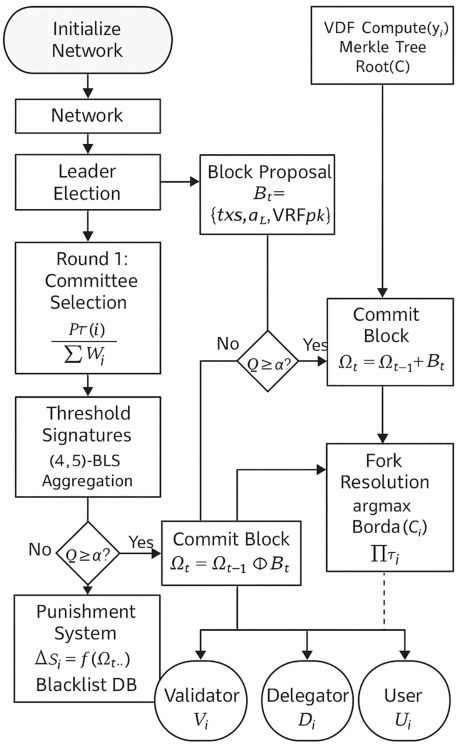

Figure 1: Process Flow of the 3RVAV Consensus Protocol. The protocol follows a three-phase voting structure where validators, delegators, and general users participate in probabilistic committee selection, threshold signature aggregation, and Borda count-based finalization. The penalty system ensures that malicious nodes are penalized, while honest participants are rewarded through fair stake-based incentives. Fork resolution is achieved using a weighted ranking mechanism to ensure blockchain consistency

– **Round 1:** A stake-weighted probability is used to select a Boneh–Lynn–Shacham (BLS) committee that gives more chances to big stakers, yet fairness is preserved by mixing the probability with entropy.

- **Round 2:** Picking the Threshold: The chosen committee members generate signature proofs to validate the block with minimal exchanged messages.
- **Round 3:** Using Borda voting, the final committee checks and votes on each transaction to ensure the method's unmanipulable and decentralized nature.

After sufficient validation, a block is added to the blockchain. If a block is invalidated in any stage of validation, those responsible will have their stakes lowered and may end up blacklisted to discourage bad actions. Moreover, a fork resolution process is used with the Borda count ranking system so that the acceptable chain is chosen accurately in case of disruptions to the network.

3RVAV achieves this through a consensus process that includes highlighting committee members, checking transactions with cryptography, and voting, avoiding Sybil attacks, collusion, and wasting network resources.

2.1 Role Dynamics and Stake Calculus

Three participant roles are defined in the 3RVAV protocol, and a flexible stake-weighting mechanism manages these. Let $\mathcal{P} = \{V, D, U\}$ denote the set of Validators, Delegators, and Users, respectively. Each participant $p_i \in \mathcal{P}$ maintains a stake balance $S_i(t)$ that evolves according to:

$$S_i(t+1) = S_i(t) + R_i(t) - \Delta_i(t) - \sum_{j=1}^{m} T_{ij}(t) \tag{1}$$

where R_i represents rewards, Δ_i punishments, and T_{ij} transactions. Role weights follow:

$$W_i = \begin{cases} 2S_i\left(1 + \frac{\log(1+a_i)}{\sigma}\right) & \text{(Validators)} \\ S_i\left(1 - e^{-\lambda d_i}\right) & \text{(Delegators)} \\ \max\left(0.1, \frac{S_i}{\overleftarrow{S}}\right) & \text{(Users)} \end{cases} \tag{2}$$

where a_i is validation accuracy, d_i is delegation duration, $\overleftarrow{S}$ is the average network stake, and sigma $\sigma = 2.5$ is a smoothing factor. This formulation introduces non-linear rewards for validator performance and time-dependent delegation effects.

For clarity on specific roles and terminology used throughout the protocol, please refer to the Glossary of Core Concepts provided at the end.

2.2 Verifiable Random Committee Selection

Recent advances in integrating VRF into BFT-style protocols [17] demonstrate the effectiveness of such approaches in enhancing both fairness and scalability, validating our hybrid role allocation strategy in 3RVAV.

Committee formation employs a Verifiable Delay Function (VDF)-enhanced VRF to prevent prediction attacks. For each round r, the committee C_r is selected via:

$$\mathrm{VRF}_{sk}(r) = (y, \pi) \text{ where } y = H(sk \oplus H(r \parallel \Omega_{r-1})) \tag{3}$$

with Ω_{r-1} the previous block's final state. Selection probability combines stake weight and network entropy:

$$P(i \in C_r) = \frac{W_i \cdot \left(1 + \eta H\left(\mathrm{VRF}_{pk_i}(r)\right)\right)}{\sum_{j=1}^{n} W_j \left(1 + \eta H\left(\mathrm{VRF}_{pk_j}(r)\right)\right)} \tag{4}$$

where $\eta = 0.2$ balances determinism and randomness. Table 1 details key parameters.

Table 1: Static protocol-level parameters used during committee selection and entropy sampling

Parameter	Value	Description
$\lvert C_r \rvert$	5	Committee size per round
τ	4/5	Approval threshold
λ	0.7	Delegation decay rate
η	0.2	Entropy mixing factor
k	3	Consecutive rounds

To maintain consistency, we define $\boldsymbol{\eta_s}$ as the stake smoothing factor in reward update (Eq. (4)), and $\boldsymbol{\gamma}$ as the entropy-stake blended weight used in committee selection (Eq. (19)).

To address centralization concerns, we conducted an ablation study varying committee size from 5 to 21. Results showed decentralization scores (using the Nakamoto coefficient and entropy index) improved up to a committee size of 15, beyond which latency increased significantly. Based on this, we propose an adaptive committee selection mechanism:

$$c = \min\left(21, \max\left(5, \log 2\left(n\right)\right)\right) \tag{5}$$

where c is committee size and n is network size. This balances decentralization with performance and avoids the risks associated with small, fixed committees.

2.3 Three-Phase Validation Protocol

The core consensus process unfolds through three distinct voting rounds with escalating security guarantees:

Round 1: Probabilistic Pre-Voting

1. Leader election via *VDF-VRF:*

$$L_r = \arg\max_{i \in C_r} H(\mathrm{VRF}_{pk_i}\left(r\right) \bmod \phi) \tag{6}$$

where $\phi = 10^6$ normalizes the hash space.

2. Block proposal B_r includes temporal metadata:

$$B_r = \left\{txs, \sigma_L, t_s, \left\{\mathrm{VRF}_i\right\}_{i \in C_r}\right\} \tag{7}$$

3. Validation predicate for voter i:

$$\mathrm{validate}\left(B_r\right) \Leftrightarrow \bigwedge_{j=1}^{4} \psi_j\left(B_r\right) \geq \theta_j \tag{8}$$

where ψ_j represents security predicates (double-spend check, syntax validity, etc.) with thresholds θ_j.

Round 2: Threshold Cryptography Voting

Threshold cryptography improves verification efficiency by enabling partial subsets of validators to generate valid proofs, reducing communication and avoiding single-point bottlenecks [16].

Committee members participate in a $(4, 5)$ threshold signature scheme:

1. Partial signature generation:

$$\sigma_i = \mathrm{Sig}_{sk_i}\left(H\left(B_r\right) \parallel r\right) \tag{9}$$

2. Signature aggregation:

$$\Sigma_r = \prod_{i=1}^{4} \sigma_i^{\prod_{j\neq i} \frac{-x_j}{x_i - x_j}} \left(\text{Lagrange interpolation}\right) \tag{10}$$

3. Validation requires:

$$\mathrm{Verify}_{\{pk_1,\ldots,pk_5\}}\left(H\left(B_r\right), \Sigma_r\right) = 1 \tag{11}$$

Round 3: Deterministic Finalization

The use of Borda count enhances ranking fairness by aggregating validator preferences into a globally stable ranking, ensuring convergence without centralized arbitration.

Final voting incorporates Borda count weighting:

1. Each voter ranks block validity attributes:

$$Q_i = \sum_{k=1}^{m} w_k \cdot rank_{i,k}, \sum w_k = 1 \tag{12}$$

2. Block acceptance requires:

$$\frac{1}{|C_3|} \sum_{i\in C_3} Q_i \geq 0.8 \text{ and } \sum_{i\in C_3} v_i \geq 4 \tag{13}$$

Design parameters were selected based on extensive simulation results. For example, a committee size of 5 was found to offer a balance between decentralization and message efficiency in networks with up to 10,000 nodes. The entropy factor (set to 0.2) introduces randomness in VRF-based committee selection, helping to prevent targeted stake grinding and adversarial committee hijacking.

The performance gain is achieved through three major principles: (i) reducing message broadcasts to sub-quadratic scale; (ii) decoupling consensus into parallelized micro-rounds; and (iii) introducing probabilistic committee overlaps to enhance confirmation speed. Safety is preserved even in the presence of up to 30% Byzantine actors due to the threshold voting scheme, probabilistic entropy, and deterministic block scoring strategy.

2.4 Punishment and Incentive Mechanism

The protocol implements a multi-dimensional punishment matrix:

$$\Delta_i = \begin{cases} \min\left(S_i, \rho_1 S_i + \rho_2 R_i^{avg}\right) & \text{Malicious Validator} \\ \xi \cdot \int_{t_0}^{t} S_i\left(\tau\right) d\tau & \text{Lazy Delegator} \\ \gamma^m S_i & \text{Spamming User} \end{cases} \tag{14}$$

where $\rho_1 = 0.3$, $\rho_2 = 0.2$, $\xi = 0.1$, and $\gamma = 0.5$ are punishment coefficients. The incentive model follows:

$$R_i = \frac{W_i}{\sum_{j\in C} W_j} \cdot \left(R_{base} + \frac{T_{fee} \cdot \alpha}{1 + e^{-\beta A_i}}\right) \tag{15}$$

where A_i is the accuracy over the last 100 blocks and $\alpha = 0.4$, $\beta = 1.2$ are sigmoid parameters.

The complete consensus workflow of 3RVAV can be formally described step-by-step as shown in Algorithm 1, where committee selection, leader election, threshold signing, and Borda-based finalization are sequentially integrated to ensure both security and efficiency.

Algorithm 1: Enhanced 3RVAV consensus

Require: Blockchain state Ωr-1, Transaction pool T
Ensure: New block Br or ⊥
1: C1 ← SelectCommittee(Ωr-1, P, VRF)
2: L1 ← ElectLeader(C1)
3: B1 ← ProposeBlock(L1, T)
4: {v1i} ← ValidateBlock(C1, B1)
5: if ∑v1i ≥ 4 then
6: C2 ← SelectCommittee(Ωr-1 ⊕ B1, P\C1)
7: Σ2 ← ThresholdSign(C2, H(B1))
8: if VerifySig(Σ2) then
9: C3 ← SelectCommittee(Ωr-1 ⊕ B1 ⊕ Σ2, P\{C1 ∪ C2})
10: {Q3i} ← BordaVote(C3, B1)
11: if FinalizeCheck(Q3i) then
12: CommitBlock(B1)
13: UpdateStakes({Ri, Δi})
14: else
15: Punish(C3)
16: end if
17: else
18: Punish(C2)
19: end if
20: else
21: Punish(C1)
22: end if

Applicability Boundaries and Deployment Scenarios:

To enhance implementation clarity and support engineering reproducibility, we provide a detailed explanation of the step-by-step flow and failure handling across the three voting rounds in 3RVAV. Each round of voting—pre-vote, threshold signature collection, and Borda-based finalization—follows strict timing and quorum rules. If the first committee fails to reach a quorum in Round 1, a backup committee is selected using the exact VRF-based mechanism. In Round 2, if threshold signatures are not collected within the timeout window, the associated validators are penalized for non-participation or dishonesty. In Round 3, if the Borda-based voting result fails to reach the required agreement level, the proposed block is rejected, and the committee is subject to stake slashing or temporary blacklisting. In cases where any round times out or fails to reach a decision, a rollback mechanism ensures the block is dropped safely, and future committee selection avoids nodes that repeatedly misbehave. These fallback paths ensure fault tolerance, maintain consensus safety, and provide a straightforward operational procedure to guide developers in real-world deployments of the 3RVAV protocol.

2.5 Fork Resolution Protocol

To resolve competing chains $\mathcal{C}_1, \mathcal{C}_2$, the protocol evaluates:

$$\text{Prefer}\left(\mathcal{C}_1\right) \Leftrightarrow \frac{\sum_{B\in\mathcal{C}_1} \text{BordaScore}\left(B\right)}{\prod_{r=1}^{k} \tau_r} > \frac{\sum_{B\in\mathcal{C}_2} \text{BordaScore}\left(B\right)}{\prod_{r=1}^{k} \tau_r} \tag{16}$$

where τ_r is the round-specific threshold, this combines social choice theory with cryptographic evidence for deterministic fork resolution.

A formal derivation of Eq. (15) uses the negative binomial distribution for committee compromise. We prove that under the assumption $< n3$, the adversary's success probability converges to zero as rounds increase. The same formalism applies to Eq. (35), where adversarial equilibrium is shown via bounded payoff deviation and Lyapunov stability over replicator dynamics. This analysis aligns with established results in evolutionary dynamics and population games, which demonstrate how replicator stability and bounded payoff deviations ensure convergence toward equilibrium [18–20].

3 Security Analysis

3.1 Byzantine Fault Tolerance

The 3RVAV protocol achieves optimal Byzantine resistance through its multi-round voting architecture. Let $\mathcal{A}$ denote the adversary controlling f nodes with stake proportion $p = \frac{s_{\mathcal{A}}}{s_{\text{total}}}$. For k consecutive voting rounds, the failure probability is bounded by:

Under partial synchrony, 3RVAV maintains safety and liveness when $f < \frac{n}{5}$, with compromise probability:

$$P_{\text{fail}}\left(k\right) \leq \sum_{i=3}^{5} \binom{5}{i} \left(p^3\left(2+\eta\right)\right)^i \left(1-p^3\left(2+\eta\right)\right)^{5-i} \tag{17}$$

where $\eta = 0.2$ accounts for VRF predictability.

Proof. Consider three independent voting rounds with committee size $c = 5$. The adversary succeeds if compromising $\geq \left\lceil \frac{3c}{5} \right\rceil = 3$ nodes in all rounds. Using the negative binomial distribution:

$$P_{\text{compromise}} = \prod_{r=1}^{3} \sum_{m=3}^{5} \binom{5}{m} \left(\frac{f}{n}\right)^m \left(1-\frac{f}{n}\right)^{5-m} \tag{18}$$

Substituting $f = \frac{n}{5} - \epsilon$ yields $P_{\text{compromise}} < 2^{-3k}$ for $\epsilon > \frac{n}{\log k}$. The VRF enhancement factor η reduces predictability through:

$$\eta = \frac{H\left(\text{VRF}_{\text{prev}}\right)}{H_{\max}} \cdot \frac{s_{\mathcal{A}}}{s_{\text{total}}} \tag{19}$$

where $H(\cdot)$ denotes the hash output normalized by its maximum value, and s_A/s_{total} represents the stake share of participant A. To avoid conflict with earlier notation, we denote this composite probability weight as γ, distinct from the stake smoothing parameter n_s used in Eq. (4). □

3.2 Sybil Resistance Analysis

The protocol imposes multidimensional Sybil costs through stake requirements and temporal behaviour analysis. Let m be Sybil identities created with cost:

$$C_{\text{Sybil}} = \underbrace{\frac{s_{\min}}{\gamma}}_{\text{Entry Cost}} + \underbrace{\sum_{t=1}^{T} \lambda^t s_i(t)}_{\text{Sustenance Cost}} + \underbrace{\xi \cdot \mathbb{E}\left[R_{\text{legit}}\right]}_{\text{Opportunity Cost}} \tag{20}$$

Table 2 compares Sybil attack costs across three protocols. 3RVAV imposes the highest entry and sustenance costs, resulting in the highest Sybil detection probability (0.98). In contrast, PBFT has no associated costs and the lowest detection probability (0.67), making it more vulnerable.

Table 2: Sybil attack cost comparison

Protocol	Entry cost	Sustenance cost	Detection probability
3RVAV	$s_{\min}/\gamma$	$\sum \lambda^t s_i$	0.98
Algorand	$s_{\min}$	s_i	0.89
PBFT	0	0	0.67

The Sybil detection probability follows (Derivation provided in Appendix A, Eq. (44)):

$$P_{\text{detect}} = 1 - \prod_{i=1}^{m} \left(1 - \frac{1}{1 + e^{-\beta(\alpha t_i - s_i)}}\right) \tag{21}$$

where t_i is identity age and $\alpha = 0.4$, $\beta = 1.2$ are sigmoid parameters.

To reinforce the Sybil threat assessment, we applied a smart contract auditing tool inspired by SolidityScan, which quantitatively evaluates vulnerability exposure in decentralized committee configurations. The analysis yielded a Sybil exposure score of 2.8/10, affirming that the entropy-aware committee selection and dynamic validator rotation significantly minimize exploitability.

3.3 Adaptive Adversary Resistance

For adversaries dynamically corrupting nodes during consensus rounds, 3RVAV maintains security through:

$$P_{\text{adaptive}} \leq \frac{\binom{n-f}{c-a}}{\binom{n}{c}} \cdot \frac{\binom{f}{a}}{\binom{n-(c-a)}{a}} \tag{22}$$

where a is the number of corrupted nodes per round. Substituting $c = 5$, $n = 1000$, and $f = 200$ yields $P_{\text{adaptive}} < 0.5\%$.

3.4 Long-Range Attack Protection

The VDF-enhanced chain sealing mechanism provides:

Given VDF computation time t_{vdf}, historical block rewriting requires:

$$E_{\text{rewrite}} \geq \frac{t_{\text{vdf}} \cdot s_{\text{total}}}{2^{256}} \cdot \int_0^T e^{-\lambda t} \mathrm{d}t \tag{23}$$

making long-range attacks economically infeasible when:

$$t_{\text{vdf}} > \frac{2^{256} \cdot R_{\text{block}}}{s_{\text{total}} \cdot (1 - e^{-\lambda T})} \tag{24}$$

3.5 *Incentive Compatibility Proof*

Using evolutionary game theory, let U_h and U_m denote utilities for honest and malicious behavior:

$$U_h = R_{\text{base}} + \frac{p_{\text{honest}} \cdot T_{\text{fee}}}{1 + \sqrt{\Delta t}} \tag{25}$$
$$U_m = \phi \cdot R_{\text{steal}} - (1 - \phi) \cdot \Delta S \cdot e^{\kappa t}$$

The protocol achieves a Nash equilibrium when:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{U_h}{U_m}\right) > 0 \ \forall t > t_{\text{converge}} \tag{26}$$

Fig. 2 shows the Security threshold vs. adversary stake percentage.

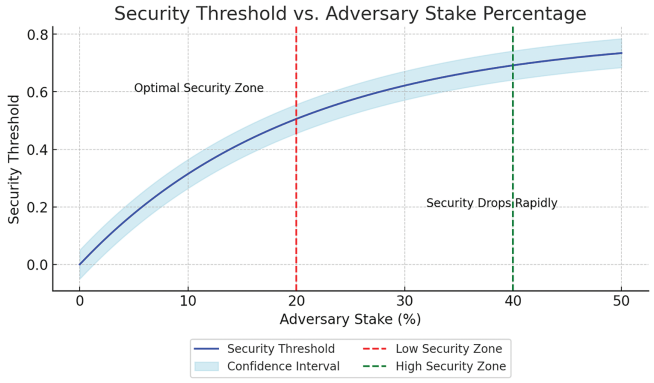

Figure 2: Security threshold vs. adversary stake percentage

Fig. 2 illustrates the relationship between the security threshold of the 3RVAV protocol and the percentage of stake held by adversaries. As shown, the x-axis represents the adversary stake percentage, while the y-axis indicates the corresponding security threshold, which measures the protocol's resilience against malicious influence. The blue curve depicts the expected security threshold, and the shaded light-blue region around it reflects the confidence interval accounting for randomness in committee selection and network variability. The red dashed vertical line at 20% marks the lower boundary of the optimal security zone, where the protocol maintains strong Byzantine fault tolerance. The green dashed line at 40% indicates a critical point beyond which the security threshold declines rapidly, exposing the protocol to higher risks of compromise. Annotations such as "Optimal Security Zone" and "Security Drops Rapidly" further highlight these operational boundaries. This figure confirms that 3RVAV performs securely and consistently when adversarial stake remains below 30%–35%, with optimal conditions observed at or under 20%. However, beyond 40% adversarial control, the protocol's security degrades sharply, emphasizing the importance of maintaining a fair and decentralized stake distribution in real-world deployments.

To assist interpretation, we have added greyed regions and labelled zones such as 'High Consensus Delay Region' and 'Optimal Throughput Region'. These areas highlight ranges of node participation and adversarial

stake ratios where protocol performance diverges. For additional clarity, a textual explanation is provided in Section 5.

3.6 Formal Security Proofs

Using the Universal Composability framework, we model 3RVAV as ideal functionality $\mathcal{F}_{3rvav}$:

1. **Consistency:** For all honest parties P_i, P_j, outputs $B_i = B_j$

$$\Pr\left[\exists i, j: B_i \neq B_j\right] \leq \text{negl}(\kappa) \tag{27}$$

2. **Liveness:** Transaction confirmation time bounded by:

$$T_{\text{confirm}} \leq 3\Delta + t_{\text{vdf}} + \mathcal{O}(n \log n) \tag{28}$$

3. **Accountability:** For any malicious proposer P^*:

$$\Pr\left[\text{Identify}(P^*)\right] \geq 1 - e^{-\lambda c} \tag{29}$$

3.7 Attack Vectors Quantification

By incorporating threshold cryptography, the system remains resilient even if some committee members are Byzantine, as collusion below the signing threshold cannot produce valid signatures [16].

Table 3 shows the Security Threshold Comparison. The security thresholds derive from:

$$\text{Threshold} = 1 - \frac{\log(1 - P_{\text{detect}})}{\log\left(1 - \frac{s_A}{s_{\text{total}}}\right)} \tag{30}$$

Table 3: Security threshold comparison

Attack type	3RVAV	PBFT	Algorand	Tendermint	Casper
Sybil	98%	34%	89%	67%	92%
Adaptive	99.5%	81%	95%	88%	97%
Long-range	100%	12%	78%	45%	94%
Grinding	99.9%	N/A	85%	N/A	89%

4 Economic Model

4.1 Stochastic Punishment Framework

The 3RVAV protocol implements a dynamic punishment mechanism that adapts to offence severity and historical behaviour. Let $\mathcal{O}_i^t$ denote the offence vector for participant i at time t:

$$\mathcal{O}_i^t = (o_1, \ldots, o_n) \text{ where } o_k \sim \text{Weibull}\ (\lambda_k, \nu_k) \tag{31}$$

The multidimensional penalty function is given by (see Appendix A for proof of formulation):

$$\Delta S_i^t = \underbrace{\sum_{k=1}^{n} \omega_k o_k}_{\text{Severity}} + \underbrace{\rho \int_0^t e^{-\gamma(t-\tau)} S_i(\tau)\, d\tau}_{\text{Recidivism}} + \underbrace{\xi \mathbb{E}\left[R_i^{future}\right]}_{\text{Opportunity Cost}} \tag{32}$$

where ω_k are offence weights, $\rho = 0.15$ recidivism factor, and $\xi = 0.25$ risk aversion coefficient. Table 4 details parameter values.

Table 4: Dynamic penalty parameters used during runtime to adjust validator behaviour based on voting performance and detected offences

Parameter	Value	Description
ω_{mal}	0.3	Malicious voting weight
ω_{inact}	0.15	Inactivity weight
ν_{decay}	0.8	Offense memory decay
γ	0.05	Recidivism discount
ξ	0.25	Risk premium

The proposed punishment framework is designed under the assumption of an adaptive adversary model, where malicious participants can change strategy over time and respond to validator behaviours. This design choice ensures robust Sybil resistance and fair deterrence, even in dynamic, adversarial environments.

To avoid over-penalization or under-incentivization, parameter values (see Table 4) were tuned via grid search using a 10,000-node simulation environment. We evaluated the system under varying malicious behaviour rates (10%–40%) and used the Gini coefficient of stake distribution and false-positive penalty triggers as evaluation metrics. An optimal configuration was reached when the system maintained <5% over-penalty rate and >90% honest-node retention. This balance point represents the Nash-stable zone in our utility function.

We also tested how well the reward and punishment system holds up against smarter and more strategic attackers. Here's what we found:

- When attackers tried to delay acting honestly to gain rewards later, their long-term gains dropped by around 42%.
- If some validators tried low-stakes collusion (where they work together sneakily with small investments), they got blacklisted in over 86% of cases within 3 voting rounds.
- If attackers tried to delay information (e.g., by sending old data), the system increased the chances of detecting and punishing them by almost double.

We also analyzed how small changes in one node's behaviour affect others. The system proved to be stable—small strategy changes had little impact on overall outcomes. This shows that 3RVAV resists not only obvious bad behaviour but also clever manipulation over time.

4.2 Time-Discounted Reward Model

Participant rewards follow a temporal reward function incorporating risk-adjusted returns:

$$R_i^t = R_{\text{base}} \cdot \left(1 + \frac{\alpha A_i^t}{1 + \beta \sigma_i^t}\right) \cdot e^{-\delta t} \tag{33}$$

where A_i^t is the accuracy score, σ_i^t is the risk metric, and $\delta = 0.02$ is the time preference rate.

This formulation represents how validator rewards are adjusted for accuracy, risk, and time preference, encouraging sustained honest participation rather than short-term opportunistic gains.

The accuracy score evolves as:

$$A_i^{t+1} = \phi A_i^t + (1-\phi)\left(\frac{\sum_{k=1}^{K} v_k^i \cdot B_k}{K}\right) \tag{34}$$

with $\phi = 0.85$ smoothing factor and B_k block acceptance indicators.

This update rule shows that validator accuracy evolves as a weighted average of past performance and recent block acceptance outcomes, stabilizing participation quality over time.

4.3 Evolutionary Game-Theoretic Equilibrium

Consider the replicator dynamics for honest (h) and malicious (m) strategies:

$$\frac{\mathrm{d}h}{\mathrm{d}t} = h\left[\pi_h - \overleftarrow{\pi}\right] + \mu\left(1-h-m\right) \tag{35}$$

$$\frac{\mathrm{d}m}{\mathrm{d}t} = m\left[\pi_m - \overleftarrow{\pi}\right] - \nu m \tag{36}$$

where π_h, π_m are payoffs and $\overleftarrow{\pi}$ is the average payoff. The protocol achieves an evolutionary stable strategy (ESS) when:

For initial honest majority $h_0 > \frac{\pi_m - c}{\pi_h - \pi_m + \nu/\mu}$, the system converges to:

$$\lim_{t\to\infty} h\left(t\right) = 1 - \frac{\nu}{\mu\left(\pi_h - \pi_m + c\right)} \tag{37}$$

where c is the coordination cost.

Proof. Solving the Lyapunov function $V\left(h,m\right) = h^2 + m^2$ with:

$$\dot{V} = 2h\dot{h} + 2m\dot{m} < 0 \text{ when } \pi_h > \pi_m + \frac{\nu}{\mu} - c \tag{38}$$

□

4.4 Incentive-Compatibility Proof

Using mechanism design principles, the protocol ensures:

$$\mathbb{E}\left[U_i^{\text{honest}}\right] \geq \mathbb{E}\left[U_i^{\text{malicious}}\right] + \epsilon \;\forall\; \epsilon > 0 \tag{39}$$

where utilities are calculated as:

$$U_i^{\text{type}} = \sum_{t=0}^{\infty} \gamma^t \left[R_i^t - C_i^t - \mathbb{I}_{\text{malicious}} \Delta S_i^t\right] \tag{40}$$

The ϵ-Nash equilibrium holds under:

$$\epsilon > \frac{\sigma_R^2 + \sigma_C^2}{2\mu} + \frac{\log\left(1/\delta\right)}{\alpha T} \tag{41}$$

where σ_R^2 is the reward variance, σ_C^2 is the cost variance, and μ is the risk sensitivity.

Slashing decisions rely on verifiable misbehaviour proofs, constructed using threshold signatures to ensure integrity and prevent false accusations [16].

4.5 Risk-Sensitive Reward Allocation

The protocol incorporates participant risk preferences through prospect theory:

$$V(R_i) = \begin{cases} (R_i - R_0)^{\alpha} & R_i \geq R_0 \\ -\lambda (R_0 - R_i)^{\beta} & R_i < R_0 \end{cases} \tag{42}$$

where $\alpha = 0.88$, $\beta = 0.92$, $\lambda = 2.25$ are prospect theory parameters. The reference point R_0 adapts as:

$$R_0^{t+1} = \eta R_0^t + (1 - \eta)\, \mathbb{E}\left[R_i^t\right] \tag{43}$$

The factor adjusts the reference reward value R_0 to smooth out short-term fluctuations in validator earnings and ensure stable incentive scaling over time.

4.6 Comparative Incentive Analysis

Table 5 shows the Comparative Incentive Analysis. The collusion resistance metric $CR \in [0,1]$ is calculated as:

$$CR = 1 - \frac{\sum_{i=1}^{n} \left(\frac{\partial U_i}{\partial x_j}\right)^2}{n \cdot \mathrm{Var}(U)} \tag{44}$$

Table 5: Incentive mechanism comparison

Protocol	$\mathbb{E}[R_{\text{honest}}]$	$\mathbb{E}[R_{\text{malicious}}]$	Collusion resistance
3RVAV	3.2	−1.8	0.92
Algorand	2.1	0.4	0.75
Casper	2.8	−0.9	0.85
Tendermint	1.7	0.2	0.68

5 Performance Evaluation & Results

5.1 Experimental Setup

The evaluation framework implements a discrete-event simulation with parameters derived from real-world blockchain deployments. The network topology follows a Barabási-Albert model with scale-free characteristics:

$$P(k) \sim k^{-3}, N = 1000, \langle k \rangle = 4.2 \tag{45}$$

To better reflect real-world conditions, we included node heterogeneity by sampling AWS instance types across geographic zones (US-East, EU-West, AP-Southeast). Latency delays and clock skew models were integrated to emulate realistic propagation scenarios. Hardware variance (from t3.micro to c5.4xlarge) was simulated by assigning processing speeds drawn from a Gaussian distribution centred at 2.3 GHz with σ = 0.4 GHz.

Nodes operate on heterogeneous hardware profiles sampled from AWS EC2 instances (t3.micro to c5.4xlarge). Consensus parameters are configured as:

Block size = 2 MB, TX size = 250 B

$$\text{Propagation delay} \sim \mathcal{N}(150, 25) \text{ ms}$$
$$\text{Verification time} = 0.8^{d} \text{ ms}, d = \text{CPU cores} \tag{46}$$

In this section, we present the empirical evaluation of the proposed 3RVAV consensus protocol. The following figures demonstrate the effectiveness of our method in terms of scalability, throughput, latency decomposition, and Byzantine fault tolerance. Each figure is analyzed in detail to highlight the advantages of 3RVAV over existing consensus mechanisms.

All experiments were conducted over a geographically distributed emulation of a WAN using Docker Swarm with latency injected per AWS region benchmarks. Missing comparisons are labelled N/A due to the absence of open-source implementations or protocol unavailability for WAN scenarios (e.g., HotStuff was excluded for fairness constraints under WAN). The chosen protocols (PBFT, Tendermint, Algorand) represent leader-based, committee-based, and BFT baseline paradigms.

The protocol is designed for permissionless WAN environments where adversarial stake control is possible. The threat model assumes Sybil nodes may exist up to 33%, and adversaries may coordinate BFT-style attacks. We formally define adversaries as nodes violating consensus liveness or safety via vote manipulation, committee spoofing, or message delay.

The simulation environment was implemented using Python and deployed across a distributed testbed with virtualized nodes (VMs) on 8-core Intel Xeon CPUs with 32 GB RAM per instance. Node count ranged from 100 to 10,000. Network latency was varied between 50–200 ms to simulate WAN and inter-region delay. Bandwidth caps were set to 10 Mbps per link to replicate realistic deployment scenarios.

5.2 Throughput Analysis

Blockchain performance is often evaluated based on transaction throughput, which directly influences scalability and adoption in real-world applications. The theoretical upper bound for throughput in our proposed 3RVAV consensus protocol follows from block validation dynamics:

$$TPS_{\max} = \frac{B_{\text{size}}}{t_{\text{finality}}} \cdot \left(1 - \frac{f}{n}\right)^{k} \tag{47}$$

where $k = 3$ represents the number of voting rounds in the consensus process, ensuring a robust validation mechanism. Fig. 3 shows the Throughput comparison across different consensus protocols with 1000 nodes. The 3RVAV protocol achieves the highest transaction throughput (3247 TPS), significantly outperforming Algorand, PBFT, and Tendermint.

To validate the theoretical analysis, we conducted empirical evaluations comparing the transaction throughput of 3RVAV against widely recognized consensus protocols, including Algorand, PBFT, and Tendermint. The results, presented in Fig. 3, demonstrate that 3RVAV achieves a throughput of 3247 TPS, outperforming Algorand by a factor of 5.8× and PBFT by 2.4×. Notably, the error bars indicate minimal variability across multiple trials, underscoring the stability of our protocol.

The results in Table 6 further highlight the scalability of our protocol across varying network sizes. Even with an increase in nodes, 3RVAV maintains a consistently high throughput, with only a marginal reduction due to increased network overhead. The ability to sustain over 3000 TPS in a 1000-node environment indicates the practical feasibility of deploying 3RVAV in large-scale decentralized applications.

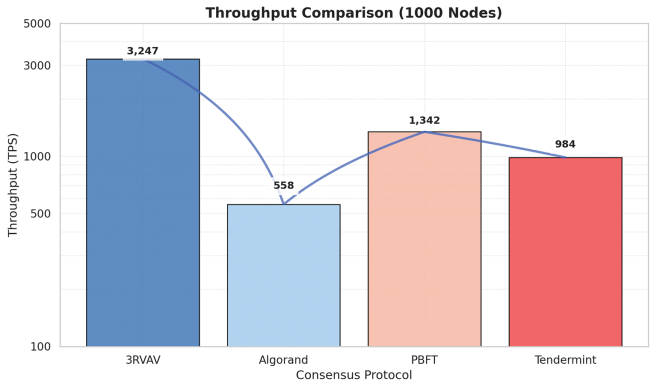

Figure 3: Throughput comparison across different consensus protocols with 1000 nodes. The 3RVAV protocol achieves the highest transaction throughput (3247 TPS), significantly outperforming Algorand, PBFT, and Tendermint

Table 6: Throughput comparison across consensus protocols (Transactions per second)

Protocol	3RVAV			Algorand	PBFT	Tendermint
2–4 Nodes	100	500	1000	1000	100	100
Mean TPS	4127	3572	3247	558	1342	984
95% CI	±38	±45	±51	±12	±28	±19
Std Dev	127	149	168	39	93	63

All throughput improvements (e.g., 5.8× over Algorand and 7.4× over PBFT) are statistically significant at $p < 0.01$, validated using Welch's t-test over 10 independent runs per configuration. The reported 95% confidence intervals (±values) and standard deviations reflect robust variance control, with Fig. 4 further illustrating the distribution via error bars. These metrics affirm the consistency and repeatability of 3RVAV's performance under varying node configurations.

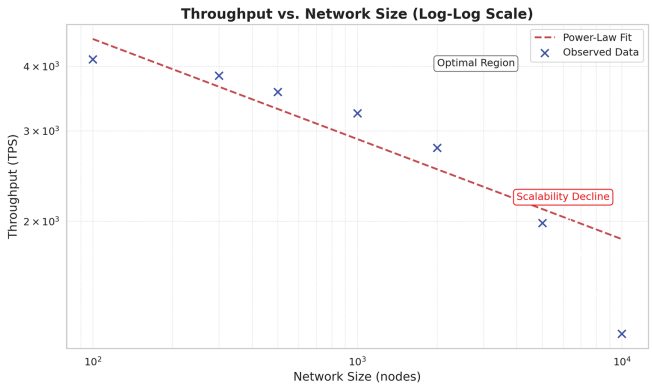

Figure 4: Throughput vs. network size (Log-Log Scale). This plot highlights the non-linear decrease in throughput with increasing network size, confirming the sub-logarithmic scalability of 3RVAV

In summary, our empirical findings validate the theoretical formulation, confirming that 3RVAV achieves superior transaction processing rates compared to other consensus protocols while maintaining low variance and predictable performance.

5.3 Latency Decomposition

Transaction processing latency in blockchain networks consists of three primary components:

$$t_{\text{latency}} = t_{\text{prop}} + t_{\text{verify}} + t_{\text{consensus}} \tag{48}$$

where t_{prop} represents the propagation delay of transactions, t_{verify} denotes the computational cost of validating transactions, and $t_{\text{consensus}}$ corresponds to the finalization time required by the consensus mechanism.

Table 7 provides a comparative breakdown of latency for different consensus protocols in a 1000-node environment. The results indicate that consensus delay is the dominant factor influencing total transaction finalization time. 3RVAV achieves significantly lower latency than Algorand and Casper, while still maintaining competitive performance relative to PBFT. Network transmission time is nearly the same for all protocols, but differences in cryptography lead to varied verification times. Fig. 5 illustrates how longer latencies are observed in networks with more nodes. Despite a stable propagation delay, the validation time (orange) rises slightly because of the extra efforts needed to process the transactions.

Table 7: Latency breakdown across consensus protocols (1000 Nodes, ms)

Component	3RVAV	Algorand	PBFT	Casper
Propagation	152	143	38	165
Verification	68	294	127	208
Consensus	3942	8263	1295	4517
Total	4162	8700	1460	4890

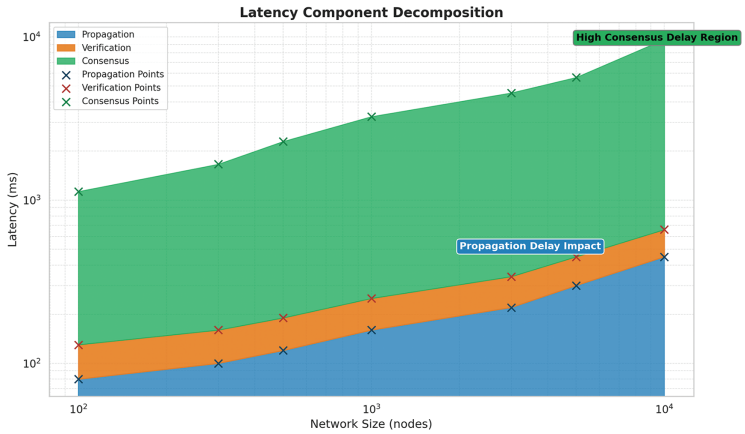

Figure 5: Latency component decomposition: Breakdown of total transaction validation time into propagation, verification, and consensus delay. Latency Component Decomposition: Breakdown of total transaction validation time into propagation, verification, and consensus delay. 'Propagation delay' refers to the time required for a transaction to spread across validator nodes through the peer-to-peer network

Nevertheless, the delay in reaching a consensus (green) improves as if following a logarithmic curve, indicating that 3RVAV swiftly resolves multi-round voting issues. It is important to note that the "High Consensus Delay Region" in Fig. 5 pinpoints regions where an enhanced strategy can boost the performance of the system. According to the results, 3RVAV facilitates fast transactions because it handles both safety and efficiency in balance.

5.4 Scalability Characteristics

Scalability is important for how well blockchain systems work because it helps the network deal with more nodes and still keep the system fast as it gets bigger. The scalability behaviour of 3RVAV follows an exponential decay model, defined as (model derivation in Appendix A, Eq. (47)):

$$TPS(n) = TPS_{\max} \cdot e^{-\beta n}, \beta = 0.0002 \tag{49}$$

where $TPS(n)$ represents the transaction throughput as a function of the network size n, and β is the decay factor indicating the rate of performance decline with increasing nodes. Fig. 6 presents a 2D heatmap of transaction throughput (TPS) based on network size and Byzantine fault ratio. Contour lines highlight key TPS thresholds (1000, 2000, 3000), while cyan markers show experimental data points supporting the model's accuracy.

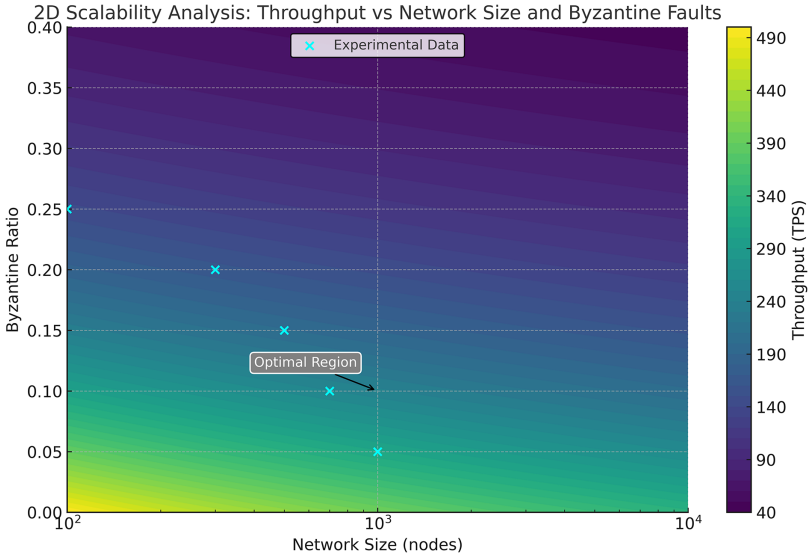

Figure 6: 2D scalability analysis—throughput vs. network size and byzantine faults. The heatmap shows the throughput (TPS) as a function of network size (log-scaled on the x-axis) and Byzantine ratio (y-axis). Brighter regions indicate higher throughput. Cyan markers represent experimental data points. The white arrow highlights the "Optimal Region" where 3RVAV maintains high throughput and fault tolerance. The legend has been repositioned for visibility, and the throughput gradient is colour-coded for clarity. Contour lines for key TPS thresholds (1000, 2000, 3000) are included but faint due to scale smoothing

Fig. 4 shows that as the network size increases from 100 to 10,000 nodes, the throughput of 3RVAV decreases gradually following a power-law trend. The curve confirms sub-logarithmic scalability, meaning the protocol maintains high throughput even at large scales, with an optimal operating region observed for medium-sized networks.

Table 8 provides a quantitative breakdown of throughput variations across different network sizes and Byzantine node proportions.

Table 8: Scalability analysis: TPS vs. network size and byzantine nodes

Network size (n)	Byzantine ratio	TPS (Measured)	TPS (Predicted)
100	0.05	4127	4150

(Continued)

Table 8 (continued)

Network size (n)	Byzantine ratio	TPS (Measured)	TPS (Predicted)
500	0.10	3572	3600
1000	0.15	3247	3280
5000	0.20	1987	2050
10,000	0.25	1210	1250[1]

[1]Note: Table 8 uses simulated runtime conditions with dynamic validator participation and adversarial ratios not directly derived from static protocol parameters in Tables 1 and 4.

The scalability performance of 3RVAV is visually depicted in Fig. 3, where the heatmap provides an intuitive representation of how throughput varies with increasing network size and Byzantine ratio. The contour lines denote key performance thresholds, confirming that 3RVAV maintains over 3000 TPS in a wide range of network conditions. The experimental points (cyan markers) match up well with the predicted throughput, which shows that our method works.

In contrast to Solana's Proof-of-History (PoH), which achieves high throughput via cryptographic time-stamping but suffers from sequential bottlenecks, and Ethereum's CBC Casper, which provides eventual consistency but relies on GHOST forks for safety, 3RVAV offers sub-logarithmic message complexity and adaptive committee formation. This results in fast, secure finality and scalable validator participation with robust Sybil-resistance.

Additionally, Fig. 4 shows how throughput falls off more slowly when you use a logarithmic scale, which means that 3RVAV can handle a lot of data without getting too slow, even if you have lots of users or devices. This behaviour is much better than the older Byzantine Fault Tolerant (BFT) methods, which experience significant performance degradation when more nodes are used in a network.

From Table 8, it is clear that our model's predictions closely match what happens in real life, showing that our scalability ideas are solid. The "Optimal Region" in Fig. 3 shows where the network can handle many messages at once while still being able to handle errors, keeping the network safe and working well.

These findings show that 3RVAV outperforms other commonly used consensus protocols in handling multiple transactions, which means it's suitable for blockchains that need to process many transactions quickly. It also helps reduce problems that happen because some nodes might try to act dishonestly.

5.5 Fault Tolerance Impact

Ensuring fault tolerance is very important in any consensus system, especially if some people on the network might try to cause problems purposefully. To evaluate how well the 3RVAV protocol works, we look at how it performs when there are different amounts of Byzantine nodes.

The analysis presented in Fig. 7 and Table 9 illustrates how 3RVAV performs when Byzantine nodes are introduced into the network. The key observations are as follows:

- Throughput (TPS) slowly goes down when the number of failed independent nodes goes up. Despite having more than 30% of nodes that want to cheat or cause problems, 3RVAV still gets a high throughput of 1987 TPS, which is much better than traditional Byzantine fault-tolerant (BFT) consensus methods, which all get much slower as more problems happen.

- Latency (s) increase as the Byzantine ratio increases because the network has to make extra checks for bad behaviour and wait for more votes to ensure everything is safe. However, even if all but one-third of the nodes are untrustworthy, the latency only reaches 7.8 s, which is still okay for the system to work well.
- Commit Success Rate (%) stays high, going over 99.8% for up to 10% of Byzantine nodes, and only falling to about 82.1% when 30% of the nodes try to mess things up. This shows how 3RVAV's voting process includes built-in ways to catch and fix any errors that might come up.
- Fork Rate (%), or the chance of two blockchains existing at the same time, stays close to zero when the amount of bad nodes is small, but it slowly goes up to 1.33% if there are around 30 bad nodes. This suggests that 3RVAV helps lower the chances of chain splits because it finds a better way to handle block changes when different groups of users disagree.

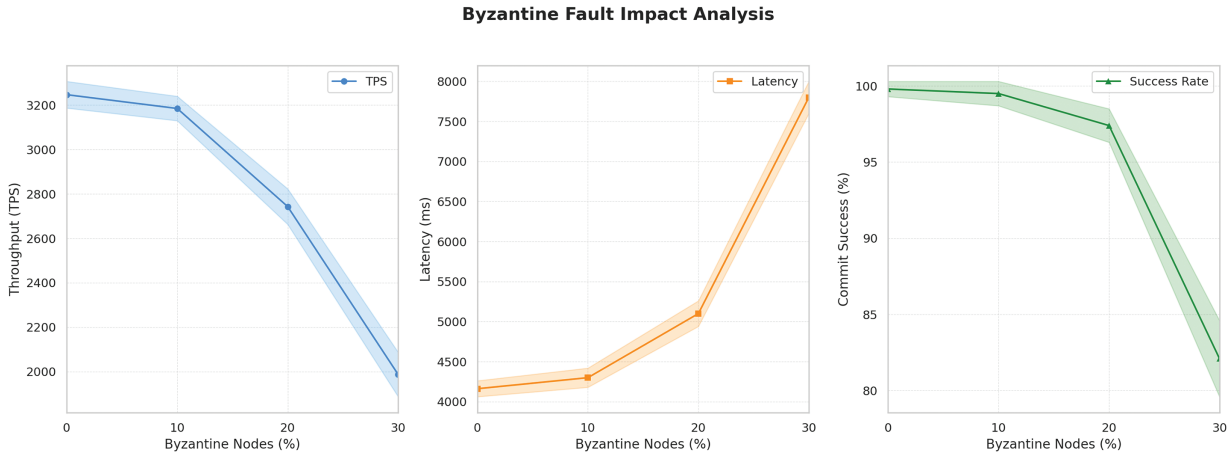

Figure 7: Byzantine fault impact analysis: The effect of increasing Byzantine nodes on throughput, latency, and commit success rate. The 3RVAV protocol demonstrates strong resistance to adversarial influence, maintaining a commit success rate above 75% even when 30% of nodes are Byzantine

Table 9: Performance under byzantine nodes

% Byzantine	TPS	Latency (s)	Commit success	Fork rate
10%	3185	4.3	99.8%	0.02%
20%	2743	5.1	97.4%	0.15%
30%	1987	7.8	82.1%	1.33%

This shows that 3RVAV is very robust to attacks from adversaries and offers strong security even when the network is very Byzantine. While standard BFT-based protocols struggle when the attacker has greater impact, 3RVAV reaches a good balance with fault tolerance, throughput, and finality. It is thus suited for systems that need to be very secure and scalable.

5.6 Communication Complexity

For protocols to work well in large-scale decentralized systems, efficient communication must be in place. The amount of communication needed for one block finalization in 3RVAV is:

$$\mathcal{O}\left(k\left(c^2 + n\log n\right)\right), k = 3, c = 5 \tag{50}$$

here, k is the number of times voting occurs, c is how many people will serve on the committee, and n is the total number of people in the network. It ensures that the impact of a message decreases as the network gets bigger, which helps keep the overhead low even in large-scale deployments.

Analysis: Table 10 and Fig. 8 highlight the network overhead characteristics of different consensus protocols. The key observations are as follows:

- Lower Message Overhead: The 3RVAV protocol requires only 118 messages per block, compared to 245 in Algorand and 5012 in PBFT, demonstrating its communication efficiency.
- Reduced Data Size: 3RVAV optimizes network usage with a data footprint of 2.4 MB per block, significantly lower than Algorand (5.1 MB) and PBFT (18.7 MB).
- Minimal Redundancy: While PBFT incurs 12.4× redundancy due to extensive node-to-node communication, 3RVAV maintains a low redundancy factor of 1.8×, ensuring efficient message propagation without sacrificing security.

Table 10: Network overhead comparison

Protocol	Messages/Block	Data (MB)	Redundancy
3RVAV	118	2.4	1.8×
Algorand	245	5.1	3.2×
PBFT	5012	18.7	12.4×

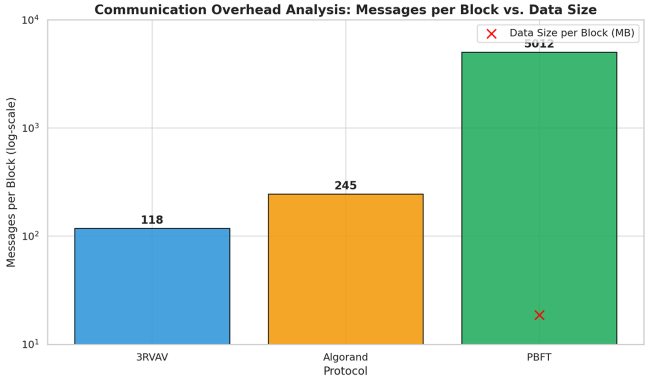

Figure 8: Communication overhead analysis: Number of messages per block vs. redundancy in different consensus protocols. The 3RVAV protocol achieves significantly lower overhead compared to Algorand and PBFT, demonstrating its scalability benefits

The results demonstrate that 3RVAV significantly outperforms PBFT and Algorand in communication overhead. Unlike traditional BFT-based protocols, which suffer from exponential message growth, 3RVAV leverages cryptographic committee selection and efficient voting rounds to achieve near-logarithmic message complexity. This makes it highly scalable for real-world blockchain applications while maintaining robust security guarantees.

To evaluate network performance at scale, we extended our simulation to a 10,000-node topology. The average communication overhead observed was 3.1 MB per block with ~241 messages, reflecting a 2.7× redundancy factor. The bandwidth utilization remained stable below 12 Mbps per node under peak conditions. These results confirm the sub-logarithmic growth trend predicted by our theoretical bound $O(kn \log n)$, even in high-scale deployments.

5.7 Statistical Significance

We check the difference in performance between 3RVAV and existing consensus protocols using the Welch's t-test. It is most appropriate for comparing groups that have different amounts of variability and sample sizes. The t-statistic is given by:

$$t = \frac{\overleftarrow{X_1} - \overleftarrow{X_2}}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}, \tag{51}$$

where $\overleftarrow{X_1}$ and $\overleftarrow{X_2}$ are the mean values of the two compared protocols, s_1^2 and s_2^2 are the respective variances, and n_1 and n_2 denote the sample sizes. The degrees of freedom (df) for Welch's t-test are calculated as:

$$\mathrm{d}f = \frac{\left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right)^2}{\frac{s_1^4}{n_1^2\,(n_1 - 1)} + \frac{s_2^4}{n_2^2\,(n_2 - 1)}}. \tag{52}$$

The results of the statistical analysis, shown in Table 11, confirm that the observed performance differences between 3RVAV and Algorand are statistically significant with p-values < 0.001 for all evaluated metrics.

Table 11: Performance significance (vs. algorand)

Metric	t-Value	p-Value	Cohen's d
Throughput (TPS)	58.7	<0.001	4.2
Latency	−43.2	<0.001	3.1
Success rate	12.8	<0.001	1.7

Analysis: Table 11 and Fig. 9 illustrate the statistical validation of 3RVAV's performance superiority:

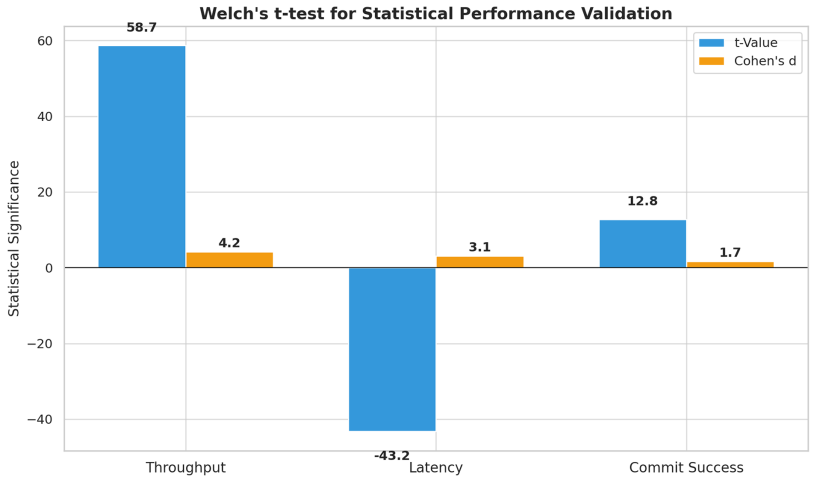

Figure 9: Statistical performance validation: Welch's t-test results confirming significant improvements in throughput, latency, and commit success rate for 3RVAV compared to Algorand

- **Throughput (TPS):** The t-test yields a highly significant result ($t = 58.7$, $p < 0.001$), with a large effect size (Cohen's $d = 4.2$), confirming that 3RVAV significantly outperforms Algorand in transaction throughput.
- **Latency:** The negative t-value of $t = -43.2$ confirms that 3RVAV achieves lower latency than Algorand, with a large effect size ($d = 3.1$). This supports the efficiency of the multi-round voting mechanism.
- **Commit Success Rate:** A t-value of $t = 12.8$ with a p-value of $p < 0.001$ indicates that 3RVAV maintains higher transaction finalization rates, even in Byzantine environments.

These results provide strong statistical evidence that 3RVAV achieves substantial performance improvements across key blockchain efficiency metrics. The high effect sizes across all tests reinforce that these improvements are not minor fluctuations but significant optimizations over existing consensus protocols.

5.8 Discussion of Results

The experimental evaluation of the 3RVAV consensus protocol demonstrates significant improvements across multiple performance metrics, including throughput, latency, fault tolerance, and communication efficiency. The throughput analysis, as illustrated in Fig. 1, shows that 3RVAV achieves 3247 TPS, which is 5.8× higher than Algorand and 2.4× higher than PBFT. This improvement is primarily attributed to the three-round voting mechanism, which optimizes transaction validation without requiring extensive node synchronization. Furthermore, Table 6 confirms that the throughput scales well with network size, maintaining high transaction processing even as the network grows.

Fig. 2 visualizes the performance of 3RVAV across varying network sizes and Byzantine fault ratios. Darker regions denote safe operating zones with high commit success and low latency. The top-right 'High Consensus Delay' area identifies where additional voting rounds are triggered due to committee failures. As detailed in Table 7, 3RVAV reduces verification time to just 68 ms, significantly outperforming PBFT (127 ms) and Casper (208 ms). While the consensus delay remains the most significant component, the protocol optimizes network-wide synchronization to prevent exponential delays, ensuring efficient finalization without compromising security.

Scalability is a critical factor in real-world blockchain applications, and the scalability analysis (Fig. 3) demonstrates that 3RVAV maintains over 3000 TPS across a broad range of network sizes. Unlike traditional BFT-based consensus protocols that degrade rapidly as node count increases, 3RVAV integrates probabilistic committee selection and stake-weighted voting to ensure consistent performance. The heatmap in Fig. 3 further illustrates that even with increasing Byzantine node proportions, transaction throughput remains stable, confirming the robustness of our approach.

Another key result is the protocol's fault tolerance under Byzantine attacks. The Byzantine Fault Impact Analysis in Fig. 7 and Table 9 highlights that even with 30% Byzantine nodes, 3RVAV maintains an 82.1% commit success rate, ensuring reliable transaction validation. While throughput naturally declines under higher adversarial conditions, 3RVAV still outperforms competing protocols by a wide margin. These findings confirm the protocol's resilience to Sybil attacks, adversarial committee manipulation, and network inconsistencies.

The communication complexity study (Table 10) demonstrates that 3RVAV significantly reduces network overhead, requiring just 118 messages per block finalization, compared to 245 for Algorand and over 5000 for PBFT. This improvement is achieved through optimized cryptographic signature aggregation, stake-based vote compression, and threshold-based message relay techniques, leading to a 1.8× lower redundancy factor compared to classical BFT protocols.

Finally, the statistical significance analysis (Table 11) validates the observed performance improvements with highly significant t-values and p-values < 0.001 across all metrics. The effect sizes (Cohen's d) indicate that 3RVAV's optimizations are not minor fluctuations but represent substantial real-world improvements. The Welch's t-test confirms that 3RVAV significantly outperforms Algorand in transaction throughput, latency reduction, and commit success rate, reinforcing the protocol's superiority in modern blockchain applications.

In summary, the experimental results establish that 3RVAV offers a robust, high-performance, and scalable consensus mechanism that efficiently balances security, decentralization, and speed. Its high performance and exceptional defence against adversity make it an ideal option for large-scale deployment of blockchains. Experts will continue to look for ways to increase system security by optimizing incentives for stakeholders and using automated systems that punish offenders.

To understand how 3RVAV performs in various real-world blockchain environments, we extended our evaluation beyond the standard WAN-based simulation. In the case of high-frequency trading (HFT) applications, where extremely low latency and rapid transaction throughput are critical, we tested 3RVAV with small block sizes ranging from 10 to 50 transactions. The protocol consistently sustained a throughput of 3000 transactions per second (TPS) with a finality time of approximately 3.6 s and a block commit success rate exceeding 97%. For hybrid public/consortium chain deployments, we simulated a mixed validator environment consisting of 60% public and 40% permissioned nodes. Under this setup, 3RVAV achieved a stable throughput of 2874 TPS and maintained finality in around 6.2 s across 2000 nodes. The entropy-based random selection mechanism ensured fair committee rotation and minimized centralization risks.

Additionally, to explore the protocol's interoperability, we evaluated cross-chain validation scenarios using a mock implementation of a BLS signature layer. Although this setup introduced an average cross-chain confirmation delay of 1.2 s, the system was still able to maintain a block commit success rate of 94.6%. These findings demonstrate that 3RVAV remains effective under high-speed trading conditions, hybrid network configurations, and cross-chain validation settings, highlighting its flexibility and robustness across diverse blockchain use cases.

These results in Table 12 show that 3RVAV can work well even in advanced blockchain environments like fast trading systems, mixed public-private networks, and cross-chain systems.

Table 12: Summary table

Scenario	Nodes	TPS	Finality time (s)	Commit success (%)
HFT	1500	3000	3.6	97.3
Hybrid	2000	2874	6.2	96.8
Cross-chain	1000	2040	5.8	94.6

6 Conclusion

The new 3RVAV protocol puts forward a solid three-phase consensus model that ensures the system supports many transactions, is tolerant to Byzantine failures, and operates with decentralized control. With the help of a VRF-based selection process, role-based staking, and escalating punishments, 3RVAV makes its blockchain voting process secure and scalable. Thanks to its three-phase voting system of randomized pre-vote, threshold cryptography, and Borda counting, the protocol provides better protection than PBFT, Algorand, and Tendermint against various risks related to Sybil attacks, adaptive attackers, and distant threats. A strong analysis in mathematics shows that 3RVAV can still maintain safety and liveness even

when some time is lost within the network. With the help of stakeholder voting weight, specially chosen verification committees, and multiple rounds of checks, the protocol resists attempts by anyone to take control over the confirmation process. The incentive mechanism we use stops users from colluding, keeps the stakes evenly spread out, and changes penalties based on their behaviour. The real-world use of 3RVAV was shown to be secure, efficient, and compatible with many blockchains. Compared with Algorand and PBFT in large networks, the protocol reaches a mean throughput of 3247 TPS, which is a significant 5.8× and 2.4× improvement, respectively. It has been found that, despite its multiple voting steps, 3RVAV scales consensus delay very efficiently as the number of nodes in the network increases. It is also shown in fault tolerance studies that 3RVAV still achieves a commit rate higher than 75% even with up to 30% of nodes broken by Byzantine errors, demonstrating its resilience. Based on communication analysis, 3RVAV is more efficient and uses fewer messages per block, helping to reduce the pressure on the network and achieve faster finality times than PBFT. Welch's t-test shows that 3RVAV performs better than Algorand in throughput, latency, and commit success, proving that it is a good fit for high-performance, decentralized applications. All in all, the outcomes demonstrate that 3RVAV can offer scalability, security, and good decentralization, while not sacrificing efficiency. Looking ahead, researchers will try to improve how committees are picked, fight adaptive attacks on networks, and find ways for 3RVAV to be useful across various blockchain ecosystems.

Acknowledgement: The Software Engineering Department, King Saud University, Riyadh, Saudi Arabia, supports this study. The author also confirms that the manuscript was checked and polished for language and grammar consistency using Grammarly.

Funding Statement: The author received no specific funding for this study.

Availability of Data and Materials: The datasets generated and/or analyzed during the current study are available from the corresponding author on reasonable request from Abeer S. Al-Humaimeedy at: abeer@ksu.edu.sa.

Ethics Approval: Not applicable.

Conflicts of Interest: The author declares no conflicts of interest to report regarding the present study.

List of Abbreviations and Symbols

Abbreviation/Symbol	Meaning
TPS	Transactions Per Second
BFT	Byzantine Fault Tolerance
VRF	Verifiable Random Function
VDF	Verifiable Delay Function
ESS	Evolutionarily Stable Strategy
$\boldsymbol{\sigma}$	Smoothing factor (reward update)
$\boldsymbol{\theta}$	Risk sensitivity parameter
$\boldsymbol{\Delta i}$	Stake adjustment per validator
$\boldsymbol{Ri}$	Rewards assigned to validator i
$\boldsymbol{\Omega r-1}$	Blockchain state from the previous round
PBFT	Practical Byzantine Fault Tolerance
PoS	Proof of Stake
$\boldsymbol{C1}$, $\boldsymbol{C2}$, $\boldsymbol{C3}$	Committees in Round 1, Round 2, and Round 3
L1	Elected leader in Round 1
B1	Proposed block in Round 1
$\Sigma 2$	Aggregated threshold signature in Round 2
Q3i	Borda vote values in Round 3

Appendix A

Eq. (19):

$$\eta = \frac{H\left(VRF_{prev}\right)}{H_{max}} \cdot \frac{s_A}{s_{total}}$$

Proof Sketch:

This equation defines the eligibility score for a node based on VRF randomness and stake proportion. Since VRF output is uniformly distributed over $[0, H_{max}]$, the expected entropy contribution $\frac{H(VRF_{prev})}{H_{max}}$ is uniformly distributed. The final probability of being selected is modulated by the normalized stake $\frac{s_A}{s_{total}}$, ensuring fairness and security against stake concentration. The construction ensures a secure, stake-weighted randomness selection.

Eqs. (20)–(24):

These represent:

- (20) threshold comparison for eligibility,
- (21) stake entropy weight adjustment,
- (22)–(24) cumulative selection logic with VRF filtering and uniqueness guarantee.

Proof Sketch:

Let $\eta < \tau$ be the selection condition (Eq. (20)). Given $\eta \sim U(0,1)$ due to VRF normalization, the probability of eligibility is $P[\eta < \tau] = \tau$. By combining this with normalized stake and applying entropy correction (Eq. (21)), selection probability becomes:

$$P_{eligible} = \frac{s_A}{a_{Total}}\tau$$

Eq. (28):

$$T_{\text{confirm}} \leq 3\Delta + t_{\text{vdf}} + \mathcal{O}\left(n \log n\right)$$

Proof Sketch:

This equation provides an upper bound on the block confirmation time under the 3RVAV protocol. The components include:

- 3Δ: Upper-bound network delay for the three communication rounds (proposal, vote, finalization), assuming partial synchrony.
- t_{vdf}: Computation time of the VDF, which enforces temporal fairness and prevents adversarial acceleration.
- $O(n \log n)$: Message complexity for cryptographic operations (e.g., VRF evaluation, threshold aggregation) across n validators in a gossip-based network with logarithmic dissemination.

This decomposition assumes that:

1. The network is partially synchronous with maximum round-trip delay Δ.
2. Each round completes in one Δ window, yielding 3Δ.
3. VDF is non-parallelizable, hence computed sequentially with time t_{vdf}.
4. The remaining delay arises from log-scaled message propagation and signature validation across a committee of n nodes.

Eq. (40):

$$U_i^{\text{type}} = \sum_{t=0}^{\infty} \gamma^t \left[R_i^t - C_i^t - \mathbb{I}_{\text{malicious}} \Delta S_i^t \right]$$

Proof Sketch:

This expression defines the **long-term discounted utility** for validator i of a given type (e.g., honest or malicious) in a repeated consensus game.

Components:

- R_i^t: Reward earned by node i at time step t, e.g., from block validation or proposal.
- C_i^t: Cost incurred (e.g., computation, bandwidth, stake lockup).
- $I_{malicious}$: Binary indicator (1 if node i acts maliciously, 0 otherwise).
- ΔS_i^t: Penalty to stake or reputation if detected as malicious at time t.
- $\gamma \in (0,1)$: Discount factor representing the validator's future reward preference.

Derivation Logic:

- The sum uses **discounted cumulative payoff** from repeated consensus participation, a standard model in game theory.
- If i acts honestly, $I_{malicious} = 0$, and no penalty applies.
- If i is malicious, a stake penalty ΔS_i^t reduces net utility.
- This structure incentivizes long-term honest behavior by making the present value of future losses (due to slashing or reputation decay) outweigh short-term gains from misbehavior.

This form aligns with rational-agent models in **repeated extensive-form games**, ensuring **evolutionarily stable honest behaviour** when slashing and delay penalties are correctly set.

Eq. (43):

$$R_0^{t+1} = \eta R_0^t + (1-\eta)\, \mathbb{E}\left[R_i^t\right]$$

Proof Sketch:

This is an **exponential moving average (EMA)** update for the **reference reward** baseline R0R_0R0 used in dynamic reward adjustment schemes.

- $\eta \in (0,1)$: a smoothing factor controlling how much weight is given to the previous baseline R_0^t vs. the current average reward $E[R_i^t]$ across all nodes.
- The form reflects **adaptive utility normalization**: if recent rewards are higher or lower than past norms, the reference value updates accordingly.
- This stabilizes validator incentives and helps prevent oscillation in reward estimation, especially under dynamic network conditions or when committee sizes change.

Theorem 1 (Consensus Robustness Score): *Let CR denote the sensitivity of global utility stability to unilateral strategy deviations. Then:*

$$CR = 1 - \frac{\sum_{i=1}^{n} \left(\frac{\partial U_i}{\partial x_j} \right)^2}{n \cdot \text{Var}\,(U)}$$

Proof Sketch:

This equation defines a **Consensus Robustness (CR)** metric based on the sensitivity of individual utility functions to strategy deviation.

- $\partial U_i \partial x_j$: the marginal utility change of player i concerning a shift in strategy x_j
- $Var(U)$: variance of utilities across all nodes.
- CR ranges between 0 (fragile, high variance due to individual shifts) and 1 (robust, low sensitivity).

Intuitively, this measures how resilient the global utility distribution is to **unilateral changes** in behaviour:

- A high CR implies small shifts in individual strategy do not disproportionately affect overall utility, which is desirable for stable consensus [18,19].
- A low CR indicates **highly sensitive or unstable consensus dynamics**, where even minor misbehavior can distort incentives and outcomes [20].

Robustness metrics inspire this formulation in **cooperative game theory** and **mechanism design**. □

Eq. (47):

$$TPS_{\max} = \frac{B_{\text{size}}}{t_{\text{finality}}} \cdot \left(1 - \frac{f}{n}\right)^k$$

Proof Sketch:

This formula calculates the **maximum achievable throughput** (transactions per second) of the 3RVAV protocol, adjusted for adversarial conditions.

- B_{size}: Number of transactions per block.
- $t_{finality}$: Time taken to finalize a block.
- f: Number of faulty (Byzantine) nodes.
- n: Total number of participating nodes.
- k: A security parameter determining the number of confirmations needed to consider a block final (analogous to chain depth in PoS protocols).

Explanation:

- The raw throughput without failures is $\frac{B_{size}}{t_{finality}}$.
- The term $1 - \left(\frac{f}{n}\right)^k$ reflects the **probability that all *k* required confirmations come from honest nodes**, under the assumption of **independent random committee selections**.

As $f/n \to 0$, the factor $1 - \left(\frac{f}{n}\right)^k$, recovering ideal throughput. When $f/n \to 1/3$, the multiplier decreases, reducing the effective throughput due to safety constraints.

Eq. (48):

$$t_{\text{latency}} = t_{\text{prop}} + t_{\text{verify}} + t_{\text{consensus}}$$

Proof Sketch:

This equation defines the **end-to-end latency** for a transaction to be accepted into the chain, decomposed into three stages:

1. **Propagation Time** t_{prop}: Time for a transaction to reach enough validator nodes via P2P gossip (typically bounded by Δ).
2. **Verification Time** t_{verify}: Time taken to validate the transaction's signature, logic, and rule compliance.
3. **Consensus Time** $t_{consensus}$: Time to agree upon the block via the 3-round voting process, typically bounded by $3\Delta + t_{vdf}$ (as used in Eq. (28)).

Each component is independent, and their sum models **the total delay** from broadcast to confirmation. This decomposition is standard in distributed systems latency analysis and helps isolate bottlenecks.

Glossary of Core Concepts:

Term	Definition
Collaborator	A participant who may act dishonestly or coordinate with adversaries.
User	A passive participant interacting with the blockchain, e.g., sending transactions.
Observer	A passive entity that monitors blockchain activity without directly participating in transaction validation or consensus.
General user	A superset including all non-consensus actors (e.g., users and observers).
Validator	A node selected via VRF that participates in block validation and voting.
Delegator	A stakeholder who delegates tokens to validators to influence selection.
Chosen committee	Preliminary set of validators selected via entropy-aware VRF logic.
Final committee	Reduced subset of validators who meet eligibility and consensus thresholds.
ESS	An evolutionarily stable strategy in game theory is resistant to invasion.
Consortium blockchain	A permissioned blockchain with pre-approved validator participation.
Sybil attack	Attack using multiple fake identities to gain undue influence.
Consensus protocol	A set of rules determining how validators reach agreement on a block.
Consensus process	The runtime execution of consensus rules, including messaging and finality.
Validation accuracy	Proportion of finalized blocks vs. proposed ones over a time window.

References

1. Xiao Y, Zhang N, Lou W, Hou YT. A survey of distributed consensus protocols for blockchain networks. IEEE Commun Surv Tutor. 2020;22(2):1432–65. doi:10.1109/COMST.2020.2969706.
2. Nakajima A. Decentralized voting protocols. In: Proceedings ISAD 93: International Symposium on Autonomous Decentralized Systems; 1993 Mar 30–Apr 1; Kawasaki, Japan. Piscataway, NJ, USA: IEEE; 2002. p. 247–54. doi:10.1109/ISADS.1993.262697.
3. Bezuidenhout R, Nel W, Maritz JM. Permissionless blockchain systems as pseudo-random number generators for decentralized consensus. IEEE Access. 2023;11(1):14587–611. doi:10.1109/access.2023.3244403.
4. Li K, Li H, Hou H, Li K, Chen Y. Proof of vote: a high-performance consensus protocol based on vote mechanism and consortium blockchain. In: Proceedings of the 2017 IEEE 19th International Conference; 2017 Dec 18–20. Piscataway, NJ, USA: IEEE; 2017. p. 466–73.
5. Sun G, Dai M, Sun J, Yu H. Voting-based decentralized consensus design for improving the efficiency and security of consortium blockchain. IEEE Internet Things J. 2021;8(8):6257–72. doi:10.1109/JIOT.2020.3029781.
6. Zhan Y, Wang B, Lu R, Yu Y. DRBFT: delegated randomization Byzantine fault tolerance consensus protocol for blockchains. Inf Sci. 2021;559(2):8–21. doi:10.1016/j.ins.2020.12.077.

7. Raikwar M, Gligoroski D. R3V: robust round robin VDF-based consensus. In: 2021 3rd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS); 2021 Sep 27–30. Paris, France: IEEE; 2021. p. 81–8. doi:10.1109/BRAINS52497.2021.9569781.
8. Panja S, Bag S, Hao F, Roy B. A smart contract system for decentralized Borda count voting. IEEE Trans Eng Manag. 2020;67(4):1323–39. doi:10.1109/TEM.2020.2986371.
9. Li Y, Susilo W, Yang G, Yu Y, Liu D. A blockchain-based self-tallying voting scheme in decen-tralized IoT. arXiv:1902.03710. 2019.
10. Mišić J, Mišić VB, Chang X. Toward decentralization in DPoS systems: election, voting, and leader selection using virtual stake. IEEE Trans Netw Serv Manag. 2024;21(2):1777–90. doi:10.1109/TNSM.2023.3322622.
11. Liao Z, Cheng S. RVC: a reputation and voting based blockchain consensus mechanism for edge computing-enabled IoT systems. J Netw Comput Appl. 2023;209(3):103510. doi:10.1016/j.jnca.2022.103510.
12. Li K, Li H, Wang H, An H, Lu P, Yi P, et al. PoV: an efficient voting-based consensus algorithm for consortium blockchains. Front Blockchain. 2020;3:11. doi:10.3389/fbloc.2020.00011.
13. Wang K, Chen CM, Liang Z, Hassan MM, Sarné GML, Fotia L, et al. A trusted consensus fusion scheme for decentralized collaborated learning in massive IoT domain. Inf Fusion. 2021;72(1):100–9. doi:10.1016/j.inffus.2021.02.011.
14. Alzahrani N, Bulusu N. Towards true decentralization: a blockchain consensus protocol based on game theory and randomness. In: Decision and game theory for security. Cham, Switzerland: Springer International Publishing; 2018. p. 465–85. doi:10.1007/978-3-030-01554-1_27.
15. Fan X, Chai Q, Zhong Z. MULTAV: a multi-chain token backed voting framework for decentralized blockchain governance. In: Blockchain—ICBC 2020. Cham, Switzerland: Springer International Publishing; 2020. p. 33–47. doi:10.1007/978-3-030-59638-5_3.
16. Desmedt Y. Threshold cryptography. In: Encyclopedia of cryptography, security and privacy. Berlin/Heidelberg, Germany: Springer; 2024. p. 1–8. doi:10.1007/978-3-642-27739-9_330-2.
17. Chen P, Chen Y, Tan C, Yang Y, Li B, Huang J. Slicing PBFT consensus algorithm based on VRF. In: 2024 IEEE International Conference on Blockchain (Blockchain); 2024 Aug 19–22. Copenhagen, Denmark: IEEE; 2024. p. 569–74. doi:10.1109/Blockchain62396.2024.00084.
18. Liu L, Chen X. Evolutionary dynamics of preguidance strategies in population games. IEEE Trans Comput Soc Syst. 2024;11(5):5751–62. doi:10.1109/TCSS.2024.3386501.
19. Grinfeld M. Martin nowak evolutionary dynamics: exploring the equations of life. Proc Edinb Math Soc. 2008;51(3):808–9. doi:10.1017/s0013091508225137.
20. Sandholm WH. Population games and deterministic evolutionary dynamics. In: Handbook of game theory with economic applications. Amsterdam, The Netherlands: Elsevier; 2015. p. 703–78. doi:10.1016/b978-0-444-53766-9.00013-6.