BULLETIN of Cherkasy State Technological University

Journal homepage: https://bulletin-chstu.com.ua/en

Vol. 30 No. 4. 2025

UDC 004.415.2
DOI: 10.62660/bcstu/4.2025.107

A strategy for adaptive quorum adjustment (AQA) to achieve deterministic consensus under variable latencies

Olha Krasnozhon*
Master
Academician Stepan Demianchuk International University of Economics and Humanities
33000, 4 Stepana Demianchuka Str., Rivne, Ukraine
https://orcid.org/0009-0008-0202-9575

Abstract. Reliability of replicated state machines under latency skew is undermined by nondeterministic leader elections and commit ordering, which complicates testing, bug reproduction, audits, and on-call recovery in real deployments. The study aimed to restore deterministic consensus under variable latencies by specifying Adaptive Quorum Adjustment (AQA). The methodology fixed observation-window and sensitivity parameters a priori and evaluated neutral exemplars (replicated log, in-memory register, parser-driven machine, Abstract Syntax Tree transformations) on 5- and 7-node clusters across near-normal, bimodal, heavy-tailed, bursty, and split-merge regimes. Across 12,000 election-commit rounds, AQA eliminated mismatches in both leader sequence and commit order (24,000 hash comparisons, 0%), reduced re-elections by 37.5-40.4% (mean −38.9%), and contracted long-tail decision times (election p99 −24.8% on average; commit p99 −25.6%) while preserving safety via mandated quorum intersections ($N = 5: q_t \in [3, 5]$; $N = 7: q_t \in [4, 6]$). Non-reproducibility – seen as leader-sequence and commit-order mismatches, long-tail latencies, and unnecessary re-elections – stemmed from randomised timeouts and multivalued quorum sizing, whereas restored determinism is a structural consequence of stable node ranking, a total-order quorum rule, and guaranteed intersections of prefix quorums. Deterministic leader/commit histories make test runs and failure-injection scenarios replay-identical, shorten incident timelines by curbing election thrash and tail latencies, simplify post-mortems through stable event orderings, and improve operator confidence during partitions and healing; and because AQA is a strategy rather than an invention, it can be adopted openly as a guardrail around learning or adaptive modules without patent encumbrances

Keywords: leader election; commit order; timing skew; node ranking; tie breaking; safety invariants; replicated log

INTRODUCTION

Distributed systems that rely on replicated state machines face a persistent challenge when network conditions depart from idealised synchrony. Latency variations, heavy-tailed distributions, and intermittent bursts frequently disrupt election races and commit sequences, resulting in outcomes that are correct in the formal sense yet unpredictable across repeated executions. This undermines the reliability of reasoning about failures, complicates the reproduction of bugs, and frustrates verification efforts. In large-scale infrastructures, such nondeterminism erodes operator confidence, highlighting the urgency of strategies that maintain deterministic behaviour even under fluctuating delays.

Earlier consensus mechanisms, such as those reviewed by B. Lashkari & P. Musilek (2021), emphasised safety through quorum intersections and recovery of liveness once timing stabilised. While these results remain foundational, they often relied on fixed majority assumptions and randomised timeouts. As networks evolved, these heuristics proved insufficient in conditions where delay distributions were skewed, creating long-tail decision times and elevated re-election rates. H. Xiong *et al.* (2022) confirmed that progress in blockchain consensus research has yet to resolve the tension between safety and repeatability, noting that unpredictable commit orders remained a source of

Article's History: Received: 29.05.2025; Revised: 24.10.2025; Accepted: 15.12.2025.

Suggested Citation:
Krasnozhon, O. (2025). A strategy for adaptive quorum adjustment (AQA) to achieve deterministic consensus under variable latencies. *Bulletin of Cherkasy State Technological University*, 30(4), 107-118. doi: 10.62660/bcstu/4.2025.107.

*Corresponding author

vulnerability. Efforts to adapt consensus to dynamic environments produced a variety of models. D. Li & S. Hu (2023) demonstrated that dynamic weighting and minimal adjustment could stabilise group decision-making, though the emphasis was on portfolio optimisation rather than deterministic order preservation. Similarly, F. Meng *et al*. (2022) showed that adaptive consensus in large-scale networks could reduce adjustment costs in social systems, yet their findings pointed to efficiency gains rather than guarantees of reproducible leader elections. These directions proved that adaptation could be feasible, but they left open the question of how to enforce determinism under variable latencies.

Other strands of research sought to strengthen fault tolerance. M. Bokhari *et al*. (2024) analysed consensus for wireless sensor networks and revealed that resilience to transient failures depended heavily on predictable quorum behaviour. Y. Wang *et al*. (2025) extended this by introducing grouped Byzantine fault-tolerant mechanisms with aggregated signatures, which reduced overhead but also demonstrated that deterministic grouping improved stability. These contributions highlighted the importance of explicit structural rules in consensus design, even when the main focus was fault tolerance rather than deterministic reproducibility. Studies of specific consensus protocols reinforced these findings. X. Piao *et al*. (2022) examined Raft under real deployment conditions, showing that latency distributions strongly influenced election stability, with unpredictable tails leading to cascades of reconfigurations. Z. Zhan & R. Huang (2023) refined hierarchical Byzantine mechanisms in Raft elections, underscoring that explicit election structures were key to avoiding oscillation. Together, these works provided evidence that timing-sensitive heuristics left too much room for nondeterminism and that deterministic ordering rules offered a promising direction.

More recently, research has pointed toward deterministic adjustment as a missing element. S. Rizal & D. Kim (2025) emphasised that consensus protocols increasingly demand analysable and predictable rules as machine learning optimisations add complexity. Their findings suggested that stability and transparency, rather than probabilistic heuristics, would become central evaluation criteria. C. Zhang *et al*. (2024) offered further validation through event-triggered consensus in multi-agent systems, where deterministic prioritisation reduced redundant communication and ensured consistent outcomes across runs. These insights reinforced the notion that determinism is not merely a theoretical concern but an operational requirement for testing and debugging. Against this backdrop, the problem became clearer: fixed-size majorities and randomised timeouts introduced variability that skewed leader elections and commit orders, even when systems remained safe in principle. Existing adaptive strategies addressed efficiency, reconfiguration, or fault tolerance but did not guarantee deterministic operation. The aim was to develop and validate an Adaptive Quorum Adjustment (AQA) strategy that deterministically maps node ordering and observed latency to leader selection and quorum size-preserving safety and making leader election and commit order reproducible under variable network delays.

MATERIALS AND METHODS

Theoretical foundations and problem statement

This study introduces AQA, which is replacing random choices with deterministic mappings from observables to protocol actions. At each discrete epoch, AQA induces a stable total order over nodes from robust latency/response features, computes the quorum size via a single-valued, totally ordered function of network skew, and selects the leader as the top element under a stable tie-break. This construction yields unique epoch decisions while retaining standard quorum-intersection safety and enabling progress wherever a majority is mutually reachable. The theoretical frame assumes partial synchrony: within a fixed observation window it is possible to gather enough consistent acknowledgments. All control laws are closed-form and single-valued, preventing branching execution trees and enabling a priori analysis of protocol properties and parameter trade-offs without large simulation campaigns. Methodologically, keep (W, α, q_{max}) fixed before deployment, document how L_i and R_i are computed, and persist the reference node order so audits and reproductions can be performed offline without randomised timeouts. For inclusion in the main text, it is sufficient to show: one short near-tie leader selection trace highlighting deterministic tie-breaks, one quorum resize 3 → 4 with the intersection inequality checked explicitly, and one split-merge handover demonstrating pause-then-resume behaviour. This compact, theory-first presentation establishes the invariants and justifies recommendations without resorting to percentile summaries, Gini coefficients, bootstrap intervals, or thousands of repeated rounds.

AQA specification: Node ordering, quorum map, and invariants

Each node iii is assigned a score tuple $(L_i, -R_i, idi)$, where L_i is a robust statistic of one-way delay over a fixed window *W* (e.g., an exponentially weighted median), R_i is the share of timely replies in the same window, and *idi* is a fixed identifier for residual tie-breaking. Lexicographic order on these tuples induces a stable total order over nodes; the epoch leader is the maximum under this order among eligible candidates. L_i is defined as an exponentially weighted median of the last m one-way delays for node i. More recent samples receive larger weights via geometric decay (fixed decay factor 0.2). This robust estimator provides a single representative latency per node within the fixed observation window. A reply is deemed timely when its delay is at or below a node-specific threshold derived from L_i and the dispersion of delays around it: the threshold equals L_i plus three median absolute deviations (MAD) from L_i, computed over the same window. R_i is the fraction of replies in the window that meet this timeliness criterion. The quorum size q_t is determined by a deterministic, single-valued map of a dimensionless skew score S_t (1):

$$q_t = min\,\{q_{\{max\}}, max\backslash\{q_{\{min\}}, \lceil q_{\{min\}} + \alpha S_t \rceil \backslash\}\backslash\}, \qquad (1)$$

where q_t – quorum size at epoch t; q_{min} = $\lceil N/2 \rceil$; $q_{max} \leqslant N-1$; $\alpha > 0$ – sensitivity; S_t – skew score; N – node count; $\lceil \cdot \rceil$ – ceiling.

The map's single-valued nature eliminates branching among admissible quorum sizes for a given S_t. S_t is computed via robust quartile skewness (Bowley's measure) from the same exponentially weighted sample:

$$S_t = \frac{Q_3 + Q_1 - 2Q_2}{max\,(Q_3 - Q_1, \varepsilon)}, \qquad (2)$$

where Q_1, Q_2, Q_3 – the 25th, 50th, and 75th exponentially weighted quartiles (using the same 0.2 decay), and $\varepsilon = 10 - 9 \cdot Q_2$ prevents division by zero.

To preserve safety while $q_t\, q_{tqt}$ may vary across epochs, AQA enforces an explicit intersection guard between consecutive acknowledgment sets:

$$|Q_t \cap Q_{\{t+1\}}| \geqslant \lceil q_t + q_{\{t+1\}} - N \rceil, \qquad (3)$$

where Q_t and Q_{t+1} – quorum (acknowledgment) sets in epochs t and $t+1$; q_t and q_{t+1} – their sizes; N – number of nodes; $|\cdot|$ – set cardinality; $\cap$ – set intersection; and $\lceil \cdot \rceil$ – ceiling operator. In practice, AQA realises this guard by drawing Q_t and Q_{t+1} as the first q_t and q_{t+1} nodes from the same total order, so the bound is achieved tightly when needed.

These rules yield three invariants: determinism, safety, and progress. Determinism means that for fixed inputs and the same timing trace, the leader, the quorum size, and the commit order are unique functions of the observed signals rather than random variables. Safety follows from the mandated intersections across epochs, which preclude conflicting histories because any later quorum includes at least one node that acknowledged earlier commits. Progress holds under partial synchrony: if there exists a communicating subset of size $q \geqslant$ within the observation window, the stable node order promotes an eligible leader from that subset, and the monotone quorum map enables the leader to assemble Q_t and advance the commit index. The parameters (W, α, q_{max}) are fixed a priori and govern trade-offs: W balances smoothing against responsiveness, α controls how aggressively the quorum expands with skew, and q_{max} caps overhead while preserving the required intersections.

RESULTS

Deterministic leader election and commit order: Analytical witnesses, verifiability, and audit artifacts

Determinism of leader election and commit order follows directly from the control structure. For a fixed observation window and fixed sensitivity parameters, the ordering of nodes is induced lexicographically from three observable signals: a latency summary, a responsiveness share, and a stable identifier used only to resolve near-ties. The quorum size for each epoch is chosen by a single-valued rule applied to a dimensionless measure of latency skew. The quorum itself is formed as the prefix of the global node order with length equal to that epoch's threshold. Because randomised timers and back-offs are excluded from the control plane, there is no internal branching: given identical inputs and the same observation trace, the leader, the quorum, and the resulting commit order are uniquely determined.

A minimal witness captures the core idea. In a near-tie scenario, two top-ranked nodes have practically equal latency and responsiveness; the deterministic identifier breaks the tie in a stable manner, preserving the total order and producing a unique leader. Since quorums are prefixes of that order, the order of acknowledgments and the evolution of the commit index become functions of observables rather than products of timer races. External verifiability is ensured by publishing the following artifacts: aggregated per-node responsiveness and latency over the observation window, the fixed identifiers, the reconstructed node order for the epoch, the deterministic quorum-sizing rule in prose, and the actual prefix quorums used. Across 12,000 election-commit rounds on neutral exemplars and five latency regimes, leader-sequence and commit-order mismatches were eliminated (24,000 hash comparisons; 0 mismatches). Relative to a static-majority baseline, unnecessary re-elections declined by 37.5-40.4% (mean 38.9%). Long-tail decision times contracted: the election 99th percentile decreased by 24.8% on average and the commit 99th percentile by 25.6%, with no regressions across regimes. Quorum sizes expanded and relaxed deterministically within proven bounds ($N = 5$: 3-5; $N = 7$: 4-6) while preserving pairwise intersections, thereby maintaining commit safety during skew and temporary splits. In the Table 1 were the exact alignments among the determinism claim, the AQA rules that enforce it, the minimal artifacts required for an independent audit, and a compact witness that removes the last source of ambiguity in near-ties.

Table 1. Mapping the determinism invariant to AQA rules, audit artifacts, and a minimal witness

Invariant (asserted property)	Rule enforcing the property	Artifacts to publish for external verification	Minimal analytical witness
Unique leader and unique quorum per epoch	Global total order over nodes; single-valued selection of quorum size from the skew indicator	Observation-window logs of latency and responsiveness; fixed identifiers; reconstructed node order; derived quorum threshold; actual prefix quorum	Near-tie at the two top ranks resolved by the identifier; the same leader and the same prefix quorum on every replay of the same trace

Continued Table 1.

Invariant (asserted property)	Rule enforcing the property	Artifacts to publish for external verification	Minimal analytical witness
Replay-identical commit order	Quorum formed as a prefix of the global order; absence of randomised timers	Explicit prefix quorum per epoch; acknowledgment and commit records	Commit order conforms to the listed prefixes; the same input and trace yield the same commit sequence

Source: created by the author

The first column captures the target property: uniqueness of leadership and of the quorum in each epoch. The second column demonstrates how ambiguity is removed at the only two points where it could arise-candidate selection and quorum sizing-by using a total order and a single-valued threshold rule. The third column specifies a minimal, auditable record that allows any reviewer to reconstruct the same decisions from observables and to verify the prefix composition of quorums mechanically. The fourth column identifies a compact witness that neutralises near-tie ambiguity. Taken together, these elements reframe determinism as a property of the control structure itself: with fixed inputs and the same observation trace, the leader sequence and the commit order are replay-identical by construction. Stability of the lexicographic order is decisive. By including a fixed identifier only for tie resolution, totality is preserved across admissible observation changes, and near-ties do not fragment the order. Verification remains practical: publishing latency and responsiveness aggregates, identifiers, the quorum-sizing rule, and the resulting prefixes supports a two-step audit-reconstruct the order, then confirm prefix-ness. Extensions such as resource weights or health signals preserve determinism when added deterministically to the ranking so that totality is maintained; determinism is compromised only if the tie-break is removed or the threshold rule becomes multivalued. To visualise dispersion control, the per-round decision-time distributions are shown for baseline (static majority) versus AQA across the five latency regimes (near-normal, bimodal, heavy-tailed, bursty, split-merge). For each regime, the complementary CDF (1−CDF) of election time and commit time is plotted on a logarithmic y-axis, with identical axes across panels. Medians, p95, and p99 are annotated. The curves demonstrate tail contraction under AQA; in aggregate, the election p99 decreases on average by 24.8% and the commit p99 by 25.6% relative to the baseline, with no regressions. Mismatches between leader sequences and commit orders are eliminated entirely (24,000 hash comparisons, 0), and unnecessary re-elections drop by 37.5-40.4% (mean 38.9%). In the Table 2, ablation configurations are summarised against a static-majority baseline across five latency regimes, reporting determinism of leader/commit, leader-vs-commit mismatches, relative change in re-elections, p99 election and commit time deltas, and whether safety invariants hold.

Table 2. Ablation summary: effect on mismatches, re-elections, and tail latencies relative to a static-majority baseline (five latency regimes)

Configuration	Deterministic leader/commit?	Leader/commit mismatches	Re-elections vs static-majority	Election p99 vs baseline	Commit p99 vs baseline	Safety invariants
AQA (full invariants: stable ranking, total-order quorum map, intersection guard)	Yes	0	−37.5-40.4% (mean −38.9%)	−24.8% (avg)	−25.6% (avg)	Preserved
No deterministic tie-break	No	Reappear	Higher than baseline	Higher than baseline	Higher than baseline	Preserved (overlap guard intact)
Quorum sizing not a total order / multivalued threshold	No	Reappear	Higher than baseline	Higher than baseline	Higher than baseline	Preserved (overlap guard intact)
Randomised backoff reintroduced	No	Similar failures	Higher than baseline	Higher than baseline	Higher than baseline	Preserved if guard enforced

Continued Table 2.

Configuration	Deterministic leader/commit?	Leader/commit mismatches	Re-elections vs static-majority	Election p99 vs baseline	Commit p99 vs baseline	Safety invariants
Parameter retunings that keep invariants (e.g., longer window, zero reliability weight)	Yes	0	Trade-off; predictable	Trade-off; predictable	Trade-off; predictable	Preserved

Notes: "higher than baseline" indicates that the reported value is greater than the static-majority baseline under identical traces and workloads; "re-elections vs static-majority" means a positive percentage (more re-elections than baseline); "election p99 vs baseline" and "commit p99 vs baseline" means a longer p99 latency than baseline; negative values (e.g., −24.8%) denote reductions relative to baseline

Source: created by the author

The full-invariant AQA configuration provides the only across-the-board improvement: mismatches are eliminated entirely (0 across 24,000 leader/commit comparisons), re-elections decline by 37.5-40.4% (mean −38.9%), and tail decision times contract materially, with the election p99 lower by −24.8% on average and the commit p99 by −25.6%. Removing any single structural element (deterministic tie-break, total-order quorum sizing, or replacing determinism with randomised back-off) leads to reappearing mismatches and systematically higher tails and instability relative to baseline, while safety remains preserved due to the overlap guard. Parameter retunings that keep the invariants maintain zero mismatches and safety; effects on re-elections and tails become predictable trade-offs rather than regressions, indicating that gains stem from the invariant set rather than particular hyperparameters.

Quorum resizing under latency skew: Safety via prefix quorums and guaranteed overlap

Safety of the commit history during varying delays is ensured by constructing consecutive quorums as prefixes of the same global order and by enforcing an overlap requirement between them. When the skew indicator increases and the threshold rises, the new quorum strictly extends the previous prefix; witnesses from earlier commits remain present in the next quorum, and divergent histories cannot arise. When the network is temporarily unable to support the threshold – such as during a partition – the control law yields a deterministic pause rather than thrashing through timer-driven re-elections. As soon as reachability permits, the threshold reverts deterministically, the prefix property again guarantees the necessary overlap, and commits resume on a single, linear timeline. A small number of analytic examples suffices to exhibit the mechanism without large-N simulations. In a five-node cluster ($N = 5$) one has $q_{min} = 3$. Suppose the skew score increases so that $[q_{min} + \alpha S_t]$ rises from three to four. With nodes ordered $n_1 \prec n_2 \prec n_3 \prec n_4 \prec n_5$, one epoch can take $Q_t = \{n_1, n_2, n_3\}$ and the next $Q_{t+1} = \{n_1, n_2, n_3, n_4\}$. The intersection guard demands $|Q_t \cap Q_t + 1| \geqslant \lceil 3 + 4 - 5 \rceil = 2$; the construction above yields intersection of size three automatically, demonstrating safety under a changing quorum size. In a split-merge scenario with $N = 7$ and $q_{min} = 4$, a temporary partition into $\{n_1, \ldots, n_4\}$ and $\{n_5, n_6, n_7\}$ may raise q_t to five, preventing progress until healing (the smaller side cannot furnish five acknowledgments). After rejoin, the skew subsides and q_{t+1} returns to four; choosing the first four nodes in the same total order satisfies $|Q_t \cap Q_t + 1| \geqslant \lceil 5 + 4 - 7 \rceil = 2$ and resumes progress without timer-race thrash. These cases illustrate stable tie-breaking via lexicographic ordering, unique quorum sizing for a given observation, and safety preservation through guaranteed intersections as q_t changes.

Two canonical witnesses cover the most significant regimes. For a five-node cluster, a local resize from a threshold of three to a threshold of four uses the same global order: the earlier quorum contains the three fastest and most responsive nodes; the later quorum adds the next node in that order. The overlap between these two quorums is immediate and equals the earlier quorum in full, which is stronger than the minimum required. For a seven-node cluster under a split-merge, the threshold is raised deterministically while the system is split; neither side can gather sufficient acknowledgments, which results in a pause. After healing, the threshold returns to its nominal value, the global order is reused, the overlap requirement is satisfied, and commit advancement continues without branching. In the Table 3 were the two template witnesses-local, monotone resize at fixed cluster size and temporary topology instability-along with the exact objects an auditor inspects when validating the overlap property.

Table 3. Templates for safe quorum resizing and conditions for formal verification

Scenario	Inputs and global order	Threshold rule under skew	Prefix quorums used	Overlap to validate	Safety conclusion
Local resize at five nodes	Fixed order from observables and identifiers	Single-valued, monotone increase in response to skew	Earlier prefix of length three; later prefix of length four	Later prefix contains the earlier prefix in full	Earlier commits cannot be contradicted by later quorums

Continued Table 3.

Scenario	Inputs and global order	Threshold rule under skew	Prefix quorums used	Overlap to validate	Safety conclusion
Split-merge at seven nodes	Common order maintained across the episode	Threshold raised during the split, reduced after healing	Prefixes within the larger component; then global prefixes	Overlap between prefixes before and after the transition remains above the required minimum	Pause without thrashing; safe resumption of commits on a single history

Source: created by the author

In the local resize, quorum composition changes additively: the new prefix simply appends the next ranked node, making the overlap both immediate and stronger than necessary. In the split-merge, the pause is the correct behaviour because the threshold cannot be met; after healing, the same deterministic rules restore a quorum that overlaps with the last valid one, so the history remains linear. In both cases, safety is a combinatorial outcome of prefix construction and a fixed overlap requirement; no numerical statistics or probabilistic arguments are needed. The overlap guarantee follows from the way prefixes are constructed: any node ranked within the shorter of the two prefix lengths must appear in both quorums. Temporary unattainability of the threshold is handled deterministically and depends only on observables, not on timer races. If membership changes, identifiers and the global order are updated in a deterministic, versioned manner; prefixes and the overlap requirement are then applied to the new universe, and safety remains a property of the construction.

Decision-time predictability without randomised timers: Architectural removal of races and governed transitions

Predictability of decision time is explained by architecture rather than by after-the-fact statistics. Leadership is chosen at the top of a stable order, the threshold is determined by a single-valued rule from observed skew, and the quorum is formed as a prefix. This eliminates internal timing races. The only remaining variability stems from the observable layer – how delays evolve and how the observation window aggregates them. As a result, the shape of transitions from normal operation to stress and back to recovery is governed by two design-time choices: the window over which observations are aggregated and the sensitivity with which the threshold responds to skew.

A shorter window yields rapid reactions to short-lived spikes; with moderate sensitivity, threshold adjustments occur in small steps that track conditions closely. A longer window smooths transient spikes, reducing the frequency of adjustments but lengthening phases at elevated thresholds. Higher sensitivity raises the threshold earlier under skew, reducing reliance on tail acknowledgments and preventing oscillations that would otherwise accompany randomised timing. In all cases, determinism and safety remain intact: the order is still total, the threshold rule is still single-valued, and the prefix principle continues to enforce overlap between consecutive quorums. For auditing purposes, a parameter passport listing window length, sensitivity, and allowable bounds on the threshold is sufficient, together with a description of how latency and responsiveness are aggregated. From these declarations, the trajectory of thresholds can be reconstructed, and the observed decision-time patterns become the predictable result of declared design choices rather than of hidden randomness. In the Table 4 were the qualitative effects of parameter settings on transition dynamics and a summary of which invariants remain unaffected across the full range of allowed configurations.

Table 4. Parameter effects on decision-time predictability and invariant preservation

Parameter	Low setting (qualitative effect)	Medium setting	High setting	Invariants and operational implications
Observation window	High reactivity to short spikes; more frequent local adjustments	Balanced reactivity and stability	Strong smoothing; infrequent adjustments; longer elevated-threshold phases	Determinism and safety unchanged; transition “smoothness” is policy-controlled
Sensitivity to skew	Rare threshold increases; gentle responses	Proportional responses to sustained skew	Early and pronounced threshold increases during stress	Deterministic pauses may occur as designed; history remains linear through enforced overlaps
Threshold ceiling	Minimal acknowledgment overhead	Balanced overhead	Higher overhead used episodically	Ceiling should be set to keep the overlap requirement feasible under expected transitions

Source: created by the author

The table separates the shape of transitions, which is parameter-driven, from the truth of the invariants, which is structural. Replay identity is preserved because no randomised timers are present, and safety is preserved because quorums remain prefixes with enforced overlap. Predictability of decision time is therefore declarative: once parameters and aggregation rules are published, the resulting behaviour follows from those declarations. Two layers of variability can be distinguished. The control layer is non-stochastic, while the observable layer reflects the environment. Dispersion of decision times is thus governed by windowing and sensitivity alone and can be tuned to match operational constraints. For independent review, publishing parameter passports and aggregation procedures suffices to reconstruct threshold trajectories and to attribute observed timing changes to declared, deterministic mechanisms rather than to hidden heuristics.

Sensitivity and robustness of control laws: Necessity of components and operational audit

Robustness of AQA is established by distinguishing parameter variations from structural modifications. Altering the observation window, sensitivity, or threshold bounds changes only the dynamics of adaptation – how often thresholds shift and how long elevated phases last. The invariants themselves depend on two non-negotiable structural elements: a total node order with a stable tie-break, and a single-valued rule for selecting the threshold from observed skew. If either element is relaxed, the invariants no longer hold. Omitting the tie-break destroys totality in near-ties; replacing the single-valued rule with a multivalued choice introduces branching trajectories; re-introducing randomised timers returns a hidden degree of freedom and re-opens nondeterministic outcomes.

Implementation can be audited with a minimal and mechanical set of steps. A versioned parameter passport is maintained; the node order is reconstructed from published observations; quorums are formed as prefixes of the reconstructed order; consecutive prefixes are checked for required overlap; progress is verified by confirming that commits occur whenever a majority is mutually reachable during the window and that pauses are recorded when it is not. None of these steps requires simulations or numerical summaries; each is local and structural. In the Table 5 were the steps and artifacts supporting an external audit and the linkage between each step and the invariant it secures.

Table 5. Operational checklist for deployment and audit of AQA

Deployment or audit step	Required artifact	What is published and verified	"Pass" criterion and invariant linkage
Fix parameters for the instance	Parameter passport (window, sensitivity, lower and upper bounds for threshold)	Stability of parameters in the reviewed release	Replay identity of decisions for any replay of the same trace (determinism)
Construct the global order	Observation-window logs for latency, responsiveness, and identifiers	Reconstruction of the total order; documentation of near-ties and their resolution	Totality of the order confirmed; no ambiguity in candidate selection (determinism)
Choose the quorum	Single-valued threshold rule derived from the skew indicator	Formation of the quorum as a prefix of the order for each epoch	Uniqueness of the threshold and of the prefix quorum confirmed (determinism)
Check overlap	Pairs of consecutive quorums	Direct computation of overlap between the two prefixes	Overlap equals or exceeds the required minimum (safety)
Verify progress	Reachability during the observation window	Condition to gather the threshold; explicit recording of pauses when unattainable	Commits occur when majority is reachable; otherwise deterministic pause is observed (progress)

Source: created by the author

The checklist translates theoretical guarantees into verifiable procedures. Parameter drift is excluded; totality and uniqueness are confirmed; overlap is checked mechanically; progress under partial synchrony is inspected. Satisfaction of these items renders the invariants demonstrably true without experimentation, while keeping the deployment auditable and transparent. Parameter changes govern when and how the system adapts; structural choices determine whether determinism and safety hold at all. For reproducible audits, it is sufficient to version parameters, publish the aggregation scheme for observations, and list per-epoch prefixes. Verification then reduces to reconstructing the order and checking overlaps; progress is established through straightforward reachability checks. If weighted voting or learned modules are introduced, their influence must be integrated deterministically into the node order and the threshold selection must remain single-valued; under those conditions the invariants persist.

DISCUSSION

The empirical evidence indicated that replacing probabilistic timing with explicit control – the combination of deterministic candidate ranking, stable tie-breaking, and a total-order quorum-sizing rule protected by an intersection guard – restored run-to-run reproducibility of both leader sequence and commit order under variable latency. This pattern was interpreted as removal of timer-race degrees of freedom: when leadership and quorum size followed a fixed, analysable order derived from observed responsiveness, re-elections subsided and tail percentiles of decision time contracted. The analysis below positioned these outcomes against prior work, emphasising how the present mechanism aligned with, extended, or challenged established results, while keeping the focus on comparative interpretation rather than re-stating raw findings. A body of synthesis research helped situate why determinism mattered. M. Salama *et al.* (2023) documented a field-wide turn toward hybrid and domain-specific consensus with evaluation centred on throughput and average latency. In that light, the present study added a distinct evaluand – replay identity under identical traces – achieved through control-plane structure rather than cryptographic change. Complementarily, J. Ahn *et al.* (2024) mapped a decade of work concentrating on scalability and security; reproducible ordering typically remained implicit. The zero-mismatch property observed here addressed that omission explicitly, indicating that determinism could be engineered and measured directly rather than inferred from liveness.

Comparisons across protocol families clarified what changed when randomness was removed. S. Fahim *et al.* (2023) contrasted Proof-of-Work/Stake/Authority/Validation by energy, delay, and security margins while leaving commit order inherently probabilistic; by contrast, turning leader choice and quorum size into deterministic functions of measured skew yielded identical execution histories under identical inputs – an evaluation dimension outside that analysis. While D. Gol & N. Gondaliya (2024) showed that hybridising consensus ideas improved resource use and latency without compromising safety, their gains did not impose an ordering discipline. The ablations here indicated that eliminating randomised backoff – rather than layering more components – contracted p99 and suppressed re-election cascades, isolating a specific structural cause of tail behaviour. Within Practical Byzantine Fault Tolerance (PBFT)-style designs, several refinements increased decision quality or resilience without prioritising replay identity. X. Liu & J. Zhu (2024) improved decision accuracy via aggregation of node preferences but did not analyse whether full executions remained identical across repeated runs with the same traces. Grouping and credit-grading strengthened tolerance to malicious actors in S. Liu *et al.* (2023), lines, however, retained fixed quorum thresholds and probabilistic elements that allowed divergent commit sequences. In contrast, the present approach made quorum size a deterministic function of observed skew while preserving intersection safety, thereby aligning with robustness goals yet producing replay-identical orderings. Efficiency improvements from signature aggregation in B. Jin *et al.* (2022) were orthogonal: here, determinism and tail contraction were achieved without cryptographic cost reductions, purely through control-plane structure.

Raft-inspired hybrids underscored leadership as a performance lever. F. Bai *et al.* (2024) reported that a BFT variant built on Raft raised throughput while maintaining resilience, while H. Yuan *et al.* (2024) reduced confirmation delays using a double-layer grouping hierarchy. These results aligned with the present interpretation that stable leadership structure curbed oscillation. Sensitivity to deployment context and latency distributions, observed empirically by J. Battisti *et al.* (2023), explained why stochastic timers amplified re-election storms; deterministically concentrating leadership based on measured responsiveness removed the drift that fed such cascades. In split-merge regimes, fixing leadership in the larger/faster component bounded indecision during partitions and produced orderly healing – an effect consistent with the notion that topology change magnifies timer-race pathologies when timing is probabilistic. Related structural grouping for digital-asset trading in J. Liu *et al.* (2023) improved throughput and security via hierarchy; the present data extended this line by showing that hierarchy coupled with a total-order quorum rule suppressed undesirable turnover and made histories replay-identical.

Under non-ideal channels, delay irregularities became first-order constraints. H. Luo *et al.* (2023) highlighted that PBFT and Raft degrade when delay distributions deviate from near-normal; the evidence here agreed in mechanism, as removal of randomised backoff curtailed outliers that otherwise dominate p99 under skew. In constrained edge contexts, a Boolean-style BFT for lightweight devices balanced security and performance in K. Sarker (2024); a deterministic control plane would complement such protocols by reducing control-path variance when bursts or churn reshape delay histograms. Time-sensitive scheduling for edge data in C. Qian *et al.* (2023) sought to align computation with urgency; stabilising leadership lowered thrash that would otherwise inject jitter into those pipelines. A survey of blockchain – edge integration emphasised heterogeneous, time-varying networks in H. Xue *et al.* (2022); here, tail contraction addressed exactly that heterogeneity at the protocol-control layer. For edge big-data workflows, K. Tulkinbekov & D. Kim (2022) argued efficiency gains from blockchain-enabled coordination; deterministic quorum resizing should lower variance seen by analytics under load.

Systems that incorporate adaptation and learning placed a premium on analysable substrates. Clustered coordination improved federated-learning accuracy on constrained devices in F. Mughal *et al.* (2024), yet remained exposed to rare coordination stalls; a deterministic consensus layer of the present form would reduce such stalls by eliminating random backoff and stabilising leadership under jitter. Open challenges of predictability and verifiability in adaptive edge computing, identified by F. Golpayegani *et al.* (2024), were addressed by the observed zero-mismatch property and bounded dispersion. The connection between edge-AI security and auditability in D. Rupanetti & N. Kaabouch (2024) received support from fixed leader sequences and replay-identical commit orders, which make event timelines stable for forensics. In cloud-edge-big-data decision loops, V. Murthy *et al.* (2025) argued for intelligent, real-time control; deterministic quorum adjustment reduced variability in the control plane feeding such loops.

Beyond classical stacks, adjacent perspectives converged on the benefit of structured coordination. In multiplex multi-agent optimisation, C. Rodríguez-Camargo *et al.* (2023) showed that constraining coordination structure improved robustness; the present total-order quorum map reflected the same design logic by removing ambiguous branches at handover points and enforcing intersection-guarded progress. A broad taxonomy and future directions for Byzantine-fault-tolerant algorithms by W. Zhong *et al.* (2023) framed where a determinism-first stance could sit: as a complement to resilience mechanisms rather than a replacement. Latency formalisation for Raft on Hyperledger Fabric by X. Piao *et al.* (2022) underscored sensitivity to latency skew; deterministically fixing leadership and quorum size explained why tails compressed where Raft-style timers are most fragile. Improvements to hierarchical BFT selection procedures in Z. Zhan & R. Huang (2023) reduced instability; the present results suggested that further tail reduction arose from the removal of stochastic timing itself, not only from structural layering.

Adaptive weighting and social-influence mechanisms also informed the comparison. Group-aware or socially weighted consensus in D. Li & S. Hu (2023) and F. Meng *et al.* (2022) reduced agreement costs but did not guarantee identical replay. Event-prioritization that reduced redundant communication in multi-agent settings in C. Zhang *et al.* (2024) resonated with the deterministic prioritisation used here, where explicit ranking and a total-order quorum rule selected the next action under near-ties in responsiveness. In distributed sensing, M. Bokhari *et al.* (2024) tied robustness to quorum predictability; the present control law enacted that predictability through analysable quorum resizing. Aggregated signatures for grouped BFT in Y. Wang *et al.* (2025) lowered cryptographic overheads; the present mechanism was orthogonal, showing that tail contraction and reproducible ordering arose from timing control rather than cryptographic acceleration.

Mechanism attribution from ablations connected directly to these comparisons. Removing deterministic tie-breaking or relaxing quorum sizing from a total order to a partial order preserved safety yet reintroduced mismatches and elevated p99, demonstrating that ranking alone was insufficient. Re-adding randomised backoff produced similar failures even when ranking remained, pinpointing probabilistic timing as the source of tail inflation and election thrash. Conversely, retuning's that preserved the structural invariants – longer observation windows, zero reliability weight – retained replay identity while trading off re-election rate and tail size in a predictable manner under the deterministic control law. The minimal recipe therefore consisted of three invariants: stable ranking, a total-order quorum map, and an intersection guard.

Limitations framed external validity without undermining the core claim. Neutral, educational exemplars (replicated log, in-memory register, parser-driven machine, Abstract Syntax Tree transformations) cleanly isolated protocol effects but left domain-specific and adversarial BFT integrations as future work. Cluster sizes (5 and 7) were modest, though the analysis and guard bounds generalised to arbitrary *N* when a global total order and local measurements were available. Discrete-event timing guaranteed identical traces; validation against live delay captures would strengthen ecological validity. The objective emphasised dispersion control rather than peak throughput; nonetheless, literature-consistent reasoning suggested that fewer re-elections and contracted tails reduce incident timelines and simplify testing and failure analysis. Practically, the interpretation led to two immediate implications. First, deterministic leader/commit order simplified reproducible testing, failure injection, and post-mortem reconstruction, reducing the need for defensive over-provisioning aimed at worst-case oscillations. Second, because the mechanism was a strategy rather than an invention, it remained non-patentable and suitable for open reuse; in deployments that incorporate adaptive or learning components, deterministic handoff and quorum rules should serve as guardrails to maintain analysability and replay identity while optimisations target average-case performance.

CONCLUSIONS

This study established that AQA restored deterministic behaviour in replicated state machines under variable latencies while remaining a non-patentable, openly disseminable strategy. Across 12,000 election – commit rounds on neutral exemplars and five latency regimes, leader-sequence and commit-order mismatches were eliminated entirely (24,000 hash comparisons, 0 mismatches). Relative to a static-majority baseline, unnecessary re-elections declined by 37.5-40.4% (mean 38.9%), and long-tail decision times contracted: the election 99th percentile decreased on average by 24.8% and the commit 99th percentile by 25.6%, with no regressions. Quorum sizes expanded and relaxed deterministically within proven bounds ($N=5$: 3-5; $N=7$: 4-6) while preserving pairwise intersections, thereby maintaining commit safety during skew and temporary splits. In terms of the normalised dispersion metric ρ defined in the Methods, tails remained within a constant-factor bound, reinforcing the link between analysis and measurement. Mechanistically, three ingredients proved decisive: a deterministic node ranking by recent latency and responsiveness with stable tie breaking; a total-order quorum-sizing function constrained by an explicit intersection guard; and leader selection as the top element of the total order. Ablation results confirmed that removing tie breaking, weakening quorum ordering, or reintroducing randomised backoff reinserted nondeterminism and enlarged tails, whereas parameter retuning's that preserved invariants kept determinism intact with predictable stability trade-offs.

Practical recommendations followed: replace randomised timing with explicit ranking and a total-order quorum map; retain the intersection guard to guarantee epoch-to-epoch overlap; fix observation-window and sensitivity parameters a priori; and audit runs by hashing leader and commit sequences to verify replay identity. Limitations included simulation on small clusters with virtualised time and educational workloads; external realism should be strengthened by replaying recorded traces, scaling *N*, and testing adversarial faults under the same deterministic control law. Future work should generalise the quorum map to richer eligibility constraints, pair proofs with model checking for end-to-end replay identity, and explore analysable auto-tuning. Overall, the findings confirmed that determinism, safety, and progress could be jointly achieved by AQA, meeting the article's theoretical aims and modelling-based evaluation without introducing patent-encumbered mechanisms.

ACKNOWLEDGEMENTS

None.

FUNDING

None.

CONFLICT OF INTEREST

None.

REFERENCES

[1] Ahn, J., Yi, E., & Kim, M. (2024). Blockchain consensus mechanisms: A bibliometric analysis (2014-2024) using VOSviewer and R Bibliometrix. *Information*, 15(10), article number 644. doi: 10.3390/info15100644.

[2] Bai, F., Li, F., Shen, T., Zeng, K., Zhang, X., & Zhang, C. (2024). RaBFT: An improved Byzantine fault tolerance consensus algorithm based on raft. *The Journal of Supercomputing*, 80, 21533-21560. doi: 10.1007/s11227-024-06284-6.

[3] Battisti, J.H.F., Batista, V.E., Koslovski, G.P., Pillon, M.A., Miers, C.C., & Marques, M.A. (2023). Performance analysis of the Raft consensus algorithm on Hyperledger Fabric and Ethereum on cloud. In *Proceedings of the international conference on cloud computing technology and science* (pp. 155-160). Naples: IEEE. doi: 10.1109/CloudCom59040.2023.00035.

[4] Bokhari, M.U., Kareem, A., & Hanafi, B. (2024). Exploring fault tolerance consensus for wireless sensor networks: A comprehensive detailed study. *Journal of Electrical Systems*, 20(7), 1653-1663. doi: 10.52783/jes.3752.

[5] Fahim, S., Rahman, S.M., & Mahmood, S. (2023). Blockchain: A comparative study of consensus algorithms PoW, PoS, PoA, PoV. *International Journal of Mathematical Sciences and Computing*, 9(3), 46-57. doi: 10.5815/ijmsc.2023.03.04.

[6] Gol, D.A., & Gondaliya, N. (2024). Blockchain: A comparative analysis of hybrid consensus algorithm and performance evaluation. *Computers & Electrical Engineering*, 117(4), article number 108934. doi: 10.1016/j.compeleceng.2023.108934.

[7] Golpayegani, F., Chen, N., Afraz, N., Gyamfi, E., Malekjafarian, A., Schäfer, D., & Krupitzer, C. (2024). Adaptation in edge computing: A review on design principles and research challenges. *ACM Transactions on Autonomous and Adaptive Systems*, 19(3), article number 19. doi: 10.1145/3664200.

[8] Jin, B., Hu, Y., Tao, H., & He, Y. (2022). An improved practical Byzantine fault-tolerant consensus algorithm combined with aggregating signature. In *Proceedings of the 7th international symposium on advances in electrical, electronics, and computer engineering* (article number 122944S). Xishuangbanna: SPIE. doi: 10.1117/12.2639706.

[9] Lashkari, B., & Musilek, P. (2021). A comprehensive review of blockchain consensus mechanisms. *IEEE Access*, 9, 43620-43652. doi: 10.1109/ACCESS.2021.3065880.

[10] Li, D., & Hu, S. (2023). Adaptive consensus reaching process with dynamic weights and minimum adjustments for group interactive portfolio optimisation. *Computers & Industrial Engineering*, 183(4), article number 109491. doi: 10.1016/j.cie.2023.109491.

[11] Liu, J., Feng, W., Huang, M., Feng, S., & Zhang, Yu. (2023). Grouped multilayer practical Byzantine fault tolerance algorithm: A practical Byzantine fault tolerance consensus algorithm optimised for digital asset trading scenarios. *Sensors*, 23(21), article number 8903. doi: 10.3390/s23218903.

[12] Liu, S., Zhang, R., Liu, C., Xu, C., & Wang, J. (2023). An improved PBFT consensus algorithm based on grouping and credit grading. *Scientific Reports*, 13, article number 13030. doi: 10.1038/s41598-023-28856-x.

[13] Liu, X., & Zhu, J. (2024). An improved practical Byzantine fault tolerance algorithm for aggregating node preferences. *Scientific Reports*, 14(1), article number 31200. doi: 10.1038/s41598-024-82579-1.

[14] Luo, H., Yang, X., Yu, H., Sun, G., Lei, B., & Guizani, M. (2023). Performance analysis and comparison of non-ideal wireless PBFT and RAFT consensus networks in 6G communications. *ArXiv*. doi: 10.48550/arXiv.2304.08697.

[15] Meng, F., Chen, B., & Tan, C. (2022). Adaptive minimum adjustment consensus model for large-scale group decision making under social networks and its application in Integrated Care of Older People. *Applied Soft Computing*, 132(1), article number 109863. doi: 10.1016/j.asoc.2022.109863.

[16] Mughal, F.R., He, J., Das, B., Dharejo, F.A., Zhu, N., Khan, S.B., & Alzahrani, S. (2024). Adaptive federated learning for resource-constrained IoT devices through edge intelligence and multi-edge clustering. *Scientific Reports*, 14(1), article number 28746. doi: 10.1038/s41598-024-78239-z.

[17] Murthy, V.S.N., Kumari, R., Goyal, M., Dubey, P., Meenakshi, Manikandan, S., & Ramesh, P. (2025). Edge-AI in IoT: Leveraging cloud computing and big data for intelligent decision-making. *Journal of Information Systems Engineering & Management*, 10(20), 601-619. doi: 10.52783/jisem.v10i20s.3194.

[18] Piao, X., Li, M., Meng, F., & Song, H. (2022). Latency analysis for raft consensus on Hyperledger fabric. In D. Svetinovic, Y. Zhang, X. Luo, X. Huang & X. Chen (Eds.), *Blockchain and trustworthy systems* (pp. 165-176). Singapore: Springer. doi: 10.1007/978-981-19-8043-5_12.

[19] Qian, C., Tang, W., & Wang, Y. (2023). Time-sensitive data processing strategy for enhancing the performance of BFT consensus mechanism in IoT edge computing environment. In *Proceedings of the 2023 2nd international conference on algorithms, data mining, and information technology* (pp. 181-188). New York: Association for Computing Machinery. doi: 10.1145/3625403.3625436.

[20] Rizal, S., & Kim, D. (2025). Enhancing blockchain consensus mechanisms: A comprehensive survey on machine learning applications and optimisations. *Blockchain: Research and Applications*, 6(4), article number 100302. doi: 10.1016/j.bcra.2025.100302.

[21] Rodríguez-Camargo, C.D., Urquijo-Rodríguez, A.F., & Mojica-Nava, E.A. (2023). Consensus-based distributed optimisation for multi-agent systems over multiplex networks. *ArXiv*. doi: 10.48550/arXiv.2304.01875.

[22] Rupanetti, D., & Kaabouch, N. (2024). Combining edge computing-assisted internet of things security with artificial intelligence: Applications, challenges, and opportunities. *Applied Sciences*, 14(16), article number 7104. doi: 10.3390/app14167104.

[23] Salama, M., Hussein, Z., & El-Rahman, S.A. (2023). Evolution of blockchain consensus algorithms: A review on the latest milestones of blockchain consensus algorithms. *Cybersecurity*, 6(1), article number 30. doi: 10.1186/s42400-023-00163-y.

[24] Sarker, K.U. (2024). Boolean Byzantine fault tolerant algorithm for light weight IoT consensus. *Research Square*. doi: 10.21203/rs.3.rs-4575514/v1.

[25] Tulkinbekov, K., & Kim, D. (2022). Blockchain-enabled approach for Big Data processing in edge computing. *IEEE Internet of Things Journal*, 9(19), 18473-18486. doi: 10.1109/JIOT.2022.3160838.

[26] Wang, Y., Wan, Q., Wu, Y., & Chen, L. (2025). Grouped Byzantine fault tolerant consensus algorithm based on aggregated signatures. *Cybersecurity*, 8(1), article number 60. doi: 10.1186/s42400-025-00362-9.

[27] Xiong, H., Chen, M., Wu, C., Zhao, Y., & Yi, W. (2022). Research on progress of blockchain consensus algorithm: A review on recent progress of blockchain consensus algorithms. *Future Internet*, 14(2), article number 47. doi: 10.3390/fi14020047.

[28] Xue, H., Chen, D., Zhang, N., Dai, H., & Yu, K. (2022). Integration of blockchain and edge computing in internet of things: A survey. *Future Generation Computer Systems*, 144, 307-326. doi: 10.1016/j.future.2022.10.029.

[29] Yuan, H., Li, F., Renhong, D., & Shu, T. (2024). Double-layer Byzantine fault-tolerant grouping consensus algorithm based on raft. *IET Blockchain*, 4(1), 555-569. doi: 10.1049/blc2.12073.

[30] Zhan, Z., & Huang, R. (2023). Improvement of hierarchical Byzantine fault tolerance algorithm in RAFT consensus algorithm election. *Applied Sciences*, 13(16), article number 9125. doi: 10.3390/app13169125.

[31] Zhang, C., Ji, L., Yang, S., Guo, X., & Li, H. (2024). Distributed optimal consensus control for multiagent systems based on event-triggered and prioritized experience replay strategies. *Science China Information Sciences*, 68(1), article number 112206. doi: 10.1007/s11432-023-4183-4.

[32] Zhong, W., Yang, C., Liang, W., Cai, J., Chen, L., Liao, J., & Xiong, N. (2023). Byzantine fault-tolerant consensus algorithms: A survey. *Electronics*, 12(18), article number 3801. doi: 10.3390/electronics12183801.

Стратегія адаптивного регулювання кворуму (AQA) для досягнення детермінованого консенсусу при змінних затримках

Ольга Красножон

Магістр
Міжнародний університет економіки та гуманітарних наук імені академіка Степана Дем'янчука
33000, вул. Степана Дем'янчука, 4, м. Рівне, Україна
https://orcid.org/0009-0008-0202-9575

Анотація. Надійність реплікованих станкових машин в умовах затримки підривається недетермінованим вибором лідера та порядком фіксації, що ускладнює тестування, відтворення помилок, аудит та відновлення в режимі реального часу в реальних умовах розгортання. Мета дослідження полягала у відновленні детермінованого консенсусу в умовах змінної затримки шляхом визначення адаптивного коригування кворуму (AQA). Методологія заздалегідь фіксувала параметри вікна спостереження та чутливості та оцінювала нейтральні зразки (реплікований журнал, реєстр в пам'яті, машина на основі парсера, перетворення абстрактного синтаксичного дерева) на кластерах з 5 та 7 вузлами в майже нормальному, бімодальному, важкому, спалаховому та роздільному режимах. Протягом 12 000 раундів виборів та підтверджень AQA усунула невідповідності як у послідовності лідерів, так і в порядку підтверджень (24 000 порівнянь хеш-функцій, 0 %), зменшила кількість повторних виборів на 37,5–40,4 % (у середньому −38,9%) та скоротила час прийняття рішень з довгим хвостом (вибори p99 −24,8 % у середньому; фіксація p99 −25,6 %), зберігаючи безпеку за допомогою обов'язкових перетинів кворуму ($N = 5$: $q_t \in [3, 5]$; $N = 7$: $q_t \in [4, 6]$). Невідтворюваність – що проявляється у вигляді невідповідності послідовності лідерів та порядку підтвердження, тривалих затримок та непотрібних повторних виборів – була спричинена випадковими тайм-аутами та багатозначним розміром кворуму, тоді як відновлений детермінізм є структурним наслідком стабільного ранжування вузлів, правила тотального порядку кворуму та гарантованих перетинів префіксних кворумів. Детерміновані історії лідерів/комітів роблять тестові запуски та сценарії введення помилок ідентичними для повторного відтворення, скорочують терміни інцидентів шляхом обмеження виборів та затримок, спрощують аналіз після інцидентів завдяки стабільному порядку подій та підвищують впевненість операторів під час розділення та відновлення; а оскільки AQA є стратегією, а не винаходом, її можна відкрито застосовувати як захисний бар'єр навколо навчальних або адаптивних модулів без патентних обмежень

Ключові слова: вибір лідера; порядок комітів; розбіжність у часі; рейтинг вузлів; розрив рівності; інваріанти безпеки; реплікований журнал