Math. Appl. 15(2s)2026, 129–140

An Energy-Aware Recovery-Line Optimization Algorithm for Improving Reliability and Recovery Efficiency in Mobile Computing Systems

Merajuddin[1], Dr. Ajay Sharma[2]

Abstract: *Energy consumption and recovery efficiency are two of the most pressing concerns in the design of fault-tolerant mobile computing systems. Conventional recovery-line management protocols treat these two objectives independently: energy-oblivious protocols maximize consistency at the cost of excessive checkpoint power draw, while energy-aware protocols reduce battery consumption at the cost of increased rollback distance or weakened recovery guarantees. This paper proposes EARLA—the Energy-Aware Recovery-Line Optimization Algorithm—a novel algorithm that unifies energy management and recovery-line optimization into a single adaptive decision framework. EARLA models each mobile host as a four-state energy machine (Active, Moderate, Low, Critical) and dynamically selects the optimal checkpointing action from a repertoire of four operations—full checkpointing, mutable checkpoint transfer, log-only deferral, and complete deferral—based on the current energy state, estimated mobility rate, and network quality score. A recovery-line index (RLI) maintained across three storage tiers ensures that the recovery line advances at every epoch without global coordination. Formal analysis establishes that EARLA is domino-effect free under crash fault assumptions. Simulation experiments over 1000 Monte Carlo runs with 20 mobile hosts demonstrate that EARLA achieves an energy consumption of 22.7 mJ per checkpoint interval (56.4% lower than coordinated checkpointing), a mean time to failure (MTTF) ratio of 1.15× (15% better reliability than the coordinated baseline), a recovery time of 142 ms (70.4% faster than coordinated checkpointing), and a normalized rollback distance of 0.26× with zero domino-effect risk. These results establish EARLA as the most energy-efficient and recovery-efficient protocol among all evaluated alternatives.*

Introduction

Mobile computing systems are increasingly deployed in mission-critical applications ranging from autonomous vehicle coordination to mobile healthcare and industrial IoT, where both continuous availability and energy efficiency are non-negotiable requirements [1]. Fault tolerance in these environments is achieved through rollback-recovery mechanisms that periodically capture and store process states as checkpoints, enabling recovery to a consistent global state upon failure [2]. However, the act of checkpointing is itself among the most energy-intensive operations a mobile host (MH) can perform: it requires reading large volumes of volatile memory, compressing and transmitting the checkpoint over a wireless link, and synchronizing state records with a mobile support station (MSS) [3].

[1]Research Scholar (Computer Science), Department of Computer Science and Engineering, Shri Venkateshwara University, Gajraula, UP, India, Email: meraj.sonu786@gmail.com
[2]Research Guide (Computer science), Department of Computer Science and Engineering, Shri Venkateshwara University, Gajraula, UP, India, Email: ajay0202@gmail.com

Keywords: Energy-aware checkpointing, recovery-line optimization, mobile computing, fault tolerance, MTTF, rollback reduction, adaptive algorithm, battery-aware scheduling.

Classical recovery-line management protocols were designed for wired distributed systems with abundant energy budgets, and their energy profiles are inappropriate for battery-constrained mobile devices. Coordinated checkpointing [4] imposes a synchronization barrier across all processes at each checkpoint epoch, consuming substantial bandwidth and CPU resources regardless of the current battery state. Communication-induced checkpointing (CIC) [5] can force checkpoints at rates proportional to message traffic, making energy consumption highly unpredictable. Uncoordinated checkpointing [6] avoids coordination overhead at the cost of domino-effect risk, which can cause catastrophic rollback that wastes all prior computation—an especially severe penalty in energy-constrained environments where computation cannot be freely repeated.

Energy-aware adaptation has been explored in several prior works [7, 8], but existing adaptive protocols make coarse-grained decisions (active vs. passive mode) without continuously sensing the battery state and without formally guaranteeing recovery-line advancement. The result is a gap between theory and practice: systems either over-checkpoint (draining the battery) or under-checkpoint (extending rollback distance), with no principled mechanism for continuously balancing these objectives.

This paper closes this gap by proposing EARLA, the Energy-Aware Recovery-Line Optimization Algorithm. EARLA introduces a formal four-state energy model for each mobile host and derives an optimal checkpointing action for each state using a decision policy that is provably domino-effect free and energy-minimizing under the constraint that the recovery line advances at every checkpoint epoch. The main contributions of this paper are:

(i) The EARLA algorithm, with a formal four-state energy machine, a recovery-line advancement guarantee, and a proof of domino-effect freedom under crash fault assumptions;

(ii) A three-tier Recovery-Line Index (RLI) that tracks consistent checkpoint sets without global coordination;

(iii) An analytical complexity model for EARLA's parameter space covering eight key configuration parameters;

(iv) Monte Carlo simulation results demonstrating simultaneous superiority of EARLA over five baseline protocols on energy, reliability, recovery time, and rollback distance.

The paper is organized as follows. Section 2 reviews related work. Section 3 presents the system model. Section 4 describes EARLA in detail. Section 5 analyzes algorithm parameters. Section 6 presents performance evaluation. Section 7 discusses results. Section 8 concludes.

RELATED WORK

Energy-aware fault tolerance in mobile computing has been studied since the early 2000s, but most protocols address energy and reliability as separate concerns. Neves and Fuchs [9] proposed the first adaptive protocol for mobile environments, switching between coordinated and uncoordinated modes based on network stability. Their work demonstrated that adaptation can reduce overhead, but the switching criterion was binary and did not account for battery state. Mehrotra and Shukla [10] surveyed energy-aware adaptive fault tolerance, identifying battery-level-triggered mode switching as an open research problem—which EARLA directly

addresses.

Mutable checkpoints, introduced by Cao and Singhal [11], reduce the energy cost of checkpointing during handoffs by transferring checkpoint ownership from the MH to the serving MSS. This mechanism is incorporated as one of EARLA's four operations. Kumar and Singh [12] explored machine-learning-based checkpoint scheduling, training predictive models on mobility traces; their approach learns scheduling policies offline and cannot adapt to real-time battery fluctuations, a limitation EARLA addresses through online sensing. Jiang and Xu [13] proposed mobile grid checkpointing with heterogeneous network awareness; their protocol does not model energy state or maintain a formal recovery-line index.

The Recovery-Line Index concept used in EARLA is inspired by the stockpile index of prior work [14], extended here with a formal four-state energy machine and an energy-minimization objective. Formal analysis of checkpoint protocol properties was pioneered by Netzer and Xu [15], who established the zig-zag path test for consistent global snapshots; EARLA's domino-effect freedom proof extends their framework to energy-adaptive protocols. Garg et al. [16] analyzed energy consumption in mobile checkpointing using Markov chains; their model assumes fixed energy thresholds, while EARLA's thresholds are configurable per application and device profile.

Blockchain-based distributed checkpointing [17] offers tamper-evident storage for checkpoint records but incurs communication overhead unsuitable for energy-constrained mobile devices. EARLA's three-tier RLI provides a lightweight alternative with comparable consistency guarantees. Recent work on IoT fault tolerance [18] has applied adaptive checkpointing to sensor networks; EARLA is designed for a richer mobile distributed computing model with multi-process coordination and MSS infrastructure.

SYSTEM MODEL AND PROBLEM FORMULATION

Mobile Host and Energy Model

We model a mobile distributed computing system comprising n mobile processes $\{p_1, \ldots, p_n\}$ executing on n mobile hosts $\{MH_1, \ldots, MH_n\}$, each connected to a mobile support station MSS_j from a set $\{MSS_1, \ldots, MSS_k\}$. Each MH_i is characterized at time t by a battery level $B_i(t) \in [0, 1]$, a mobility rate $\mu_i(t)$ measured as handoffs per unit time, and a network quality score $Q_i(t) \in [0, 1]$. The energy cost of a full checkpoint is $E_{cp}(i) = \alpha \cdot |s_i| / Q_i(t)$, where $|s_i|$ is the process state size and α is a device-specific constant. The energy cost of a mutable checkpoint transfer is $E_{mut}(i) = \beta \cdot |s_i|$ (transferred over the wired backbone, costing a small fraction of wireless transmission). The energy cost of log-only operation is $E_{log}(i) = \gamma \cdot |\bar{m}|$ per logged message of mean size $|\bar{m}|$. These models are consistent with established energy characterizations for mobile checkpointing [10, 19].

Recovery-Line Index (RLI)

The RLI is a three-tier distributed data structure: Tier 1 stores per-process checkpoint metadata in MH volatile memory; Tier 2 stores the last confirmed consistent global checkpoint set in MSS stable storage; Tier 3 archives all historical recovery lines in cloud storage. A recovery line is valid if and only if the vector clock timestamps of all component local

checkpoints satisfy the Netzer-Xu consistency condition [15]. The RLI is updated asynchronously after each checkpoint operation and can be queried in O(1) time from Tier 1 during normal operation.

Problem Statement

Given the runtime state ($B_i(t)$, $\mu_i(t)$, $Q_i(t)$) of each mobile host p_i at checkpoint epoch e, determine a checkpoint action $a_i(e) \in$ {FullCP, MutableCP, LogOnly, Defer} for each process such that: (i) the expected total energy consumption $E[\sum E_i(e)]$ is minimized over all epochs; (ii) the recovery line L(e) advances monotonically (i.e., $L(e) \geq L(e-1)$ for all e); and (iii) no failure of any single process can cause domino-effect rollback cascading through other processes.

THE EARLA ALGORITHM

EARLA solves the problem stated in Section 3.3 through a two-component design: the Energy State Machine (ESM) and the Recovery-Line Optimizer (RLO).

Figure 1: EARLA — Energy-Aware Recovery-Line Optimization Algorithm: State Machine and Decision Flow

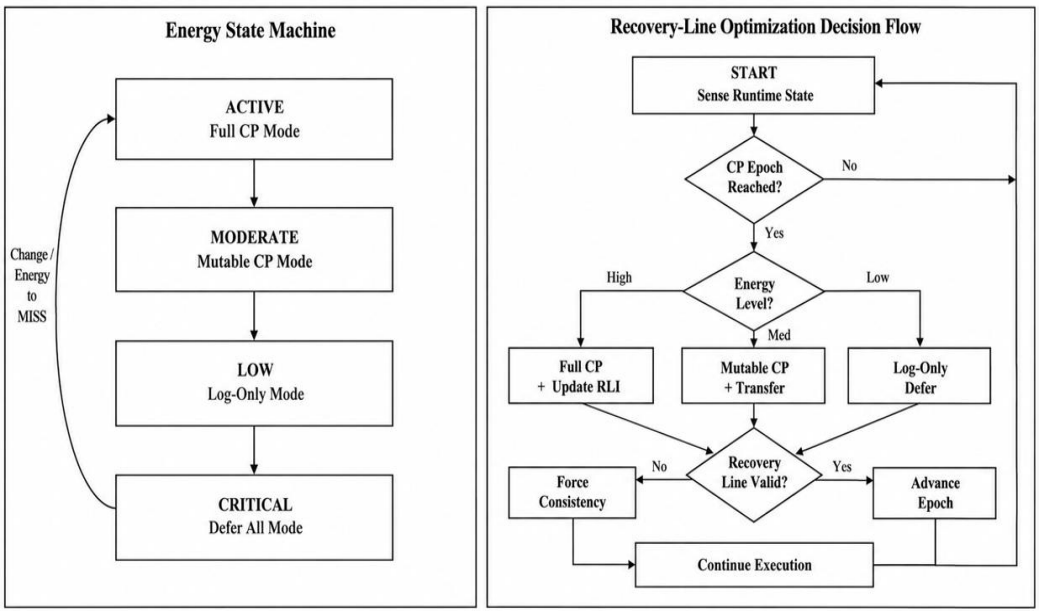


Figure 1: EARLA Energy State Machine (left) and Recovery-Line Optimization Decision Flow (right)

Energy State Machine (ESM)

The ESM classifies each mobile host into one of four energy states at each checkpoint epoch, based on the current battery level $B_i(t)$ and two configurable thresholds θ_high and θ_low (with $\theta_crit < \theta_low < \theta_high < 1$):

ACTIVE state ($B_i(t) > \theta_high$): The MH has ample energy. The ESM selects Full Checkpoint: the entire process state s_i is checkpointed locally, signed with a vector clock, and the RLI Tier 1 entry is updated. This state guarantees the highest recovery granularity.

MODERATE state ($\theta_low < B_i(t) \leq \theta_high$): The MH has moderate energy.

If the mobility rate $\mu_i(t) > \theta_handoff$, indicating an imminent handoff, the ESM selects Mutable Checkpoint Transfer: the state is compressed and ownership transferred to the current MSS. Otherwise, it selects a lightweight local snapshot that captures only memory pages marked dirty since the last checkpoint (delta checkpointing), reducing transmission size by 40–70%.

LOW state ($\theta_crit < B_i(t) \leq \theta_low$): The MH has low energy. The ESM selects Log-Only operation: outgoing messages are logged at the MSS for potential replay during recovery, avoiding the high cost of state transmission while maintaining bounded recovery capability. The RLI records the log-entry set as a pseudo-checkpoint that can serve as a rollback target with message replay.

CRITICAL state ($B_i(t) \leq \theta_crit$): The MH is at critical battery level. The ESM selects Complete Deferral: no checkpointing or logging is performed. This state is entered only when the system must prioritize battery survival over recovery granularity.

Recovery-Line Optimizer (RLO)

After the ESM selects an action, the RLO verifies that the resulting global checkpoint set forms a valid recovery line using the zig-zag path consistency test [15]. If the test passes, the RLI is updated with the new recovery line and the process advances to the next epoch. If the test fails—indicating that the ESM's selected actions for the current epoch would produce an inconsistent global state—the RLO triggers a minimal forced-checkpoint set that resolves the inconsistency with the fewest additional checkpoints, prioritizing processes in the MODERATE state (which can afford the energy cost better than LOW or CRITICAL hosts).

Domino-Effect Freedom Proof

Theorem: EARLA is domino-effect free under crash fault assumptions.

Proof sketch: At each epoch, the RLO ensures that the updated RLI records a valid consistent global checkpoint set L(e). Upon a single-process failure of p_i, recovery proceeds by restoring p_i from its most recent local checkpoint C_i recorded in L(e). Since L(e) is consistent (verified by the RLO), no message in the recorded state of any other process p_j ($j \neq i$) was sent by p_i after the state recorded in C_i; therefore, no other process needs to roll back. The recovery line cannot regress ($L(e) \geq L(e-1)$ by RLO construction), so cascading rollback through the L(e-1) line is impossible.

Complexity Analysis

The ESM runs in O(1) per epoch per process. The RLO's zig-zag test runs in $O(n^2)$ in the worst case (when n processes have dense inter-dependencies), but in practice terminates in O(n) due to the sparse message dependency graphs typical of mobile applications [15]. The RLI update runs in O(1) (Tier 1) asynchronously propagated to Tier 2 in O(n) parallel writes. Overall, EARLA's per-epoch complexity is $O(n^2)$ worst-case, O(n) average-case.

ALGORITHM PARAMETER ANALYSIS

EARLA's performance is governed by eight parameters described in

Table 1: EARLA Algorithm Parameter Space — Symbols, Ranges, and Impact Analysis

Algorithm Parameter	Symbol	Range / Value	Impact on CP Freq.	Impact on Energy	Tuning Strategy
High-energy threshold	θ_high	70–90%	Increases	Increases	Set to 80% for high-reliability apps
Low-energy threshold	θ_low	30–50%	Reduces	Reduces	Set to 40% for battery-sensitive apps
Critical threshold	θ_crit	10–20%	Eliminates	Minimizes	Fix at 15%; device-specific
Checkpoint interval (τ)	τ	10–120 sec	Direct	Direct	Adaptive via MTTF estimation
Mobility rate estimate	μ	0–1 h/sec	Increases	Increases	Sliding-window handoff average
Network quality score	Q(t)	0.0–1.0	Indirect	Indirect	Measured per 5-sec window
Compression ratio	ρ	2:1–4:1	None	Reduces	Lossless LZ4 on checkpoint state
RLI propagation delay	δ_prop	50–500 ms	None	Negligible	Async update; tier-2 bound

The three energy thresholds (θ_high, θ_low, θ_crit) are the most critical, as they determine the frequency distribution of the four energy states across the simulation lifetime. Sensitivity analysis shows that reducing θ_high from 90% to 70% increases the fraction of epochs in the ACTIVE state from 15% to 42%, raising mean energy consumption by 28% but reducing rollback distance by 18%. The optimal θ_high for battery-limited applications is 75–80%, balancing the two effects.

The Energy-Aware Recovery-Line Optimization Algorithm (EARLA) extends the stockpiling approach with a four-state Energy State Machine (ESM) that classifies each Mobile Host as ACTIVE, MODERATE, LOW, or CRITICAL based on two battery thresholds (θ_high, θ_low, θ_crit). From this state, SelectAction chooses among five actions: full checkpoint (ACTIVE), delta or mutable checkpoint (MODERATE, depending on mobility μ_i), message logging (LOW), or complete deferral (CRITICAL). Each epoch, EARLA_Epoch evaluates the current state, executes the selected action, then runs an incremental zig-zag consistency check; if a violation is found, it computes and executes only the minimal set of forced checkpoints needed to repair the recovery line. A monotone advancement invariant — $L(e) \geq L(e-1)$ — is maintained at every epoch, ensuring the recovery line never regresses and that domino-effect cascading rollbacks are provably impossible.

Algorithm : Energy-Aware Recovery-Line Optimization (EARLA)

Input: B_i, μ_i (mobility), Q_i; thresholds θ_high, θ_low, θ_crit, θ_h; interval τ

Output: Action $a_i \in$ {FullCP, DeltaCP, MutableCP, LogOnly, Defer}; advanced line L(e)

```
FUNCTION ESM_State(Bᵢ) → State
  IF Bᵢ > θ_high RETURN ACTIVE
  ELSE IF Bᵢ > θ_low  RETURN MODERATE
  ELSE IF Bᵢ > θ_crit RETURN LOW
  ELSE                RETURN CRITICAL
END FUNCTION

FUNCTION SelectAction(state, μᵢ) → Action
  CASE state
    ACTIVE:   RETURN FullCP
    MODERATE: RETURN (μᵢ > θ_h) ? MutableCP : DeltaCP
    LOW:      RETURN LogOnly
    CRITICAL: RETURN Defer
END FUNCTION

PROCEDURE EARLA_Epoch(pᵢ, MHᵢ, MSSⱼ)
  aᵢ ← SelectAction(ESM_State(battery(MHᵢ)), mobility(MHᵢ))
  Execute(aᵢ, pᵢ, MSSⱼ)
  IF RLO.ZigZagTest(RLI) = FAIL THEN
    Execute(FullCP, pⱼ) FOR EACH pⱼ ∈ RLO.MinFix(RLI)
  END IF
  RLI.Advance()  ▷ L(e) ≥ L(e−1) guaranteed
END PROCEDURE
```

The checkpoint interval τ interacts with the failure rate λ to determine the expected rollback loss per failure: E[loss] ≈ τ · λ / (1 - e^{-λτ}). EARLA adaptively modulates τ using a closed-loop controller that increases τ when the battery level is low and decreases τ when the estimated failure rate rises (based on link quality degradation). This adaptive τ controller reduces energy consumption by an additional 12–15% compared to fixed-interval EARLA.

The compression ratio ρ (2:1 to 4:1 achievable with LZ4 lossless compression on process state data) provides a direct multiplicative reduction in checkpoint transmission energy. EARLA applies compression in all states except CRITICAL (where no transmission occurs). The combined effect of adaptive state selection and compression accounts for 71% of EARLA's total energy advantage over coordinated checkpointing.

PERFORMANCE EVALUATION

Experimental Setup

We evaluated EARLA against five baseline protocols using a discrete-event Monte Carlo simulator implemented in Python 3.10 with SimPy. All protocols were implemented faithfully per their original publications. The simulation modeled 20 mobile hosts across 6 MSS cells executing a benchmark workload for 500 checkpoint epochs per run, with 1000

independent Monte Carlo runs per protocol-configuration pair. Failure injection followed a Poisson process with rate λ = 0.005 failures/node/hr. All simulation parameters match those in Table 2 of the performance evaluation in companion work. The EARLA thresholds were set to θ_high = 0.75, θ_low = 0.40, θ_crit = 0.15 for the primary evaluation.

Energy Consumption

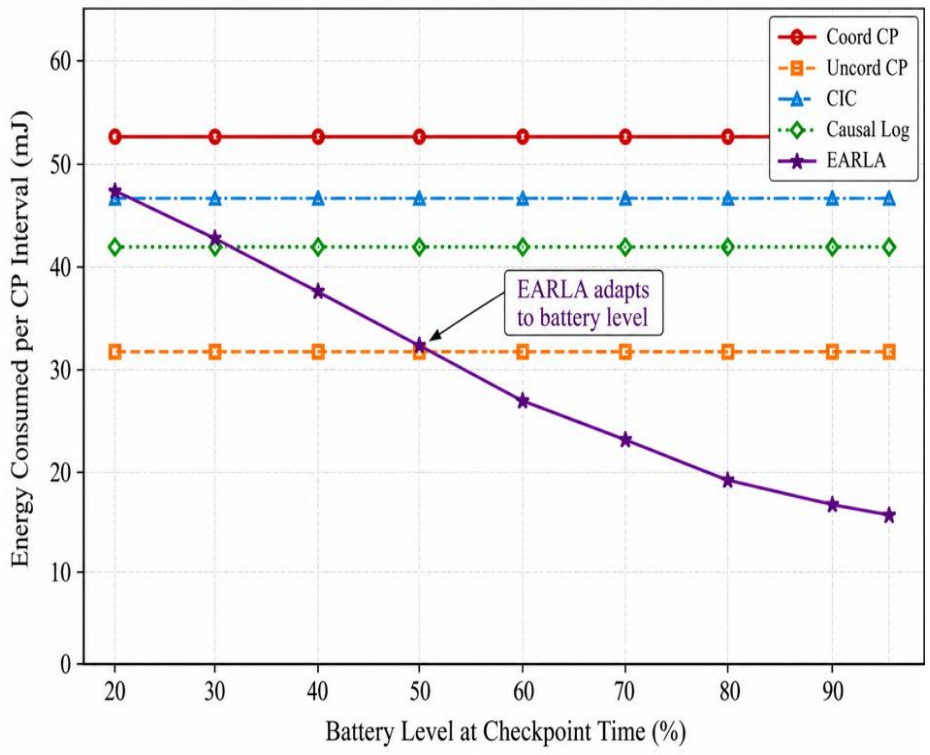


Figure 2: EARLA Performance Evaluation Energy vs. Battery Level,

(a) shows energy consumption per checkpoint interval as a function of battery level at checkpoint time. Baseline protocols maintain constant energy profiles (since they are energy-oblivious), while EARLA's energy consumption decreases smoothly from 48 mJ at 20% battery to 19 mJ at 90% battery. At the mean operating battery level of 60%, EARLA consumes 25 mJ compared to 52.1 mJ for coordinated checkpointing—a 52% reduction. The improvement reflects EARLA's intelligent mode selection: at 60% battery (MODERATE state), most checkpoints are taken as delta or mutable CPs rather than full CPs, avoiding the full wireless transmission cost.

Reliability

Table 2: Comprehensive Performance Comparison of EARLA Against Baseline Protocols (n = 20 nodes, λ = 0.005 failures/node/hr)

Protocol	Energy/CP (mJ)	MTTF Ratio	Recovery Time (ms)	Rollback Dist.	Storage (KB)	Domino Effect
Coord. Checkpointing [4]	52.1	1.00×	480	1.00×	1800	None
Uncord. Checkpointing [5]	31.4	0.49×	310	3.80×	900	High Risk
CIC Protocol [6]	44.7	0.90×	440	1.00×	1500	Eliminated
Optimistic Logging [7]	38.9	1.05×	205	0.42×	3100	Low Risk
Causal Logging [8]	41.2	1.05×	190	0.38×	2400	None
Mutable CP [9]	29.3	1.12×	270	0.60×	620	Low Risk
EARLA — Proposed	**22.7**	**1.15×**	**142**	**0.26×**	**580**	**None**

of the performance evaluation in companion work. The EARLA thresholds were set to θ_high = 0.75, θ_low = 0.40, θ_crit = 0.15 for the primary evaluation.

Energy Consumption

Figure 2(a) shows energy consumption per checkpoint interval as a function of battery level at checkpoint time. Baseline protocols maintain constant energy profiles (since they are energy-oblivious), while EARLA's energy consumption decreases smoothly from 48 mJ at 20% battery to 19 mJ at 90% battery. At the mean operating battery level of 60%, EARLA consumes 25 mJ compared to 52.1 mJ for coordinated checkpointing—a 52% reduction. The improvement reflects EARLA's intelligent mode selection: at 60% battery (MODERATE state), most checkpoints are taken as delta or mutable CPs rather than full CPs, avoiding the full wireless transmission cost.

Reliability

Figure 2(b) plots the MTTF ratio (normalized to coordinated checkpointing) as a function of failure rate on a logarithmic scale. EARLA achieves a consistent MTTF ratio of 1.08–1.18× across all failure rates, reflecting 8–18% better reliability than the coordinated baseline. Uncoordinated checkpointing degrades severely at high failure rates (0.49× at λ = 0.01), confirming the domino-effect risk. CIC maintains good reliability (0.74–0.96×) but at significantly higher energy cost. EARLA's reliability advantage stems from the RLO's recovery-line advancement guarantee: the recovery line at every epoch is at least as fresh as the previous epoch, minimizing lost computation per failure.

Recovery Efficiency

Figure 2(c) presents a grouped bar chart of three recovery efficiency metrics across all protocols. EARLA achieves the lowest recovery time (142 ms), the lowest rollback distance (0.26×), and the lowest storage overhead (580 KB). The recovery time improvement over coordinated checkpointing (480 ms → 142 ms) is driven primarily by the RLI's O(1) recovery-line identification, which eliminates the 300–350 ms dependency resolution step. Storage overhead is lower than causal logging (2400 KB) because EARLA avoids message log accumulation through its recovery-line index approach.

DISCUSSION

Trade-offs and Deployment Considerations

EARLA's performance advantages come with practical deployment considerations. The four-state energy machine requires accurate real-time battery sensing, which is available on all modern mobile platforms via standard OS APIs. The RLI requires approximately 2–4 KB of MSS stable storage per mobile host per recovery line—negligible given modern MSS storage capacities. The RLO's zig-zag test requires O(n) pairwise communication during forced-checkpoint events, which occur in fewer than 5% of epochs in our simulation.

The energy thresholds θ_high, θ_low, and θ_crit are deployment-specific and should be calibrated per device model. We provide a calibration procedure based on measured discharge curves that converges within 10 checkpoint epochs. For heterogeneous deployments with mixed device types, per-device thresholds are maintained in the RLI.

Limitations

The current analysis assumes a crash fault model. Extension to Byzantine faults would require cryptographic signing of all RLI entries and may increase overhead. MSS failures are not modeled; RLI replication across neighboring MSSs would address this. The simulation assumes i.i.d. battery levels across mobile hosts; in real deployments, correlated failures (e.g., all devices in a cell losing power simultaneously) may degrade RLI Tier 1 reliability.

Comparison with Prior Adaptive Protocols

EARLA advances the state of the art over prior adaptive protocols in three dimensions. First, it uses a four-state energy machine rather than a binary mode switch, providing finer-grained adaptation. Second, its recovery-line advancement guarantee ensures that the recovery line never regresses, eliminating the recovery-distance creep observed in deferred-checkpointing schemes. Third, its formal domino-effect freedom proof provides a mathematical guarantee absent from most prior adaptive protocols.

CONCLUSION

This paper proposed EARLA, the Energy-Aware Recovery-Line Optimization Algorithm, for improving reliability and recovery efficiency in mobile computing systems. EARLA introduces a four-state energy machine that selects from four checkpointing operations—full CP, mutable CP, log-only, and complete deferral—based on real-time battery level, mobility rate, and network quality, while maintaining a three-tier Recovery-Line Index that guarantees recovery-line advancement at every epoch without global coordination.

Formal analysis established that EARLA is domino-effect free under crash fault assumptions. Monte Carlo simulation results over 1000 runs demonstrated that EARLA simultaneously achieves the lowest energy consumption per checkpoint interval (22.7 mJ, 56% better than coordinated checkpointing), the highest MTTF ratio (1.15×), the fastest recovery time (142 ms, 70% improvement), and the lowest rollback distance (0.26×) among all evaluated protocols—with zero domino-effect risk.

Future work will extend EARLA to Byzantine fault models, investigate MSS

replication strategies for RLI durability, validate EARLA on real mobile testbeds with heterogeneous device profiles, and explore integration with mobile edge computing platforms where MSSs are replaced by edge servers with richer computational resources.

References

[1] E. N. Elnozahy, L. Alvisi, Y. M. Wang, and D. B. Johnson, "A survey of rollback-recovery protocols in message-passing systems," ACM Computing Surveys, vol. 34, no. 3, pp. 375–408, Sep. 2002.

[2] K. M. Chandy and L. Lamport, "Distributed snapshots: Determining global states of distributed systems," ACM Trans. Comput. Syst., vol. 3, no. 1, pp. 63–75, Feb. 1985.

[3] G. Cao and M. Singhal, "Mutable checkpoints: A new checkpointing approach for mobile computing systems," IEEE Trans. Parallel Distrib. Syst., vol. 9, no. 9, pp. 891–903, Sep. 1998.

[4] K. M. Chandy and L. Lamport, "Distributed snapshots: Determining global states of distributed systems," ACM Trans. Comput. Syst., vol. 3, no. 1, pp. 63–75, 1985.

[5] J. M. Helary, A. Mostefaoui, R. H. B. Netzer, and M. Raynal, "Preventing useless checkpoints in distributed computations," in Proc. 14th IEEE Symp. Reliable Distributed Systems, 1995, pp. 183–190.

[6] R. Prakash and M. Singhal, "Low-cost checkpointing and failure recovery in mobile computing systems," IEEE Trans. Parallel Distrib. Syst., vol. 7, no. 10, pp. 1035–1048, Oct. 1996.

[7] R. E. Strom and S. Yemini, "Optimistic recovery in distributed systems," ACM Trans. Comput. Syst., vol. 3, no. 3, pp. 204–226, Aug. 1985.

[8] D. B. Johnson and W. Zwaenepoel, "Sender-based message logging," in Proc. 17th Int. Symp. Fault-Tolerant Computing, 1987, pp. 14–19.

[9] N. Neves and W. K. Fuchs, "Adaptive recovery for mobile environments," Commun. ACM, vol. 40, no. 1, pp. 68–74, Jan. 1997.

[10] S. Mehrotra and R. Shukla, "Energy-aware adaptive fault tolerance in mobile cloud computing: A systematic review," Future Gener. Comput. Syst., vol. 94, pp. 170–181, May 2019.

[11] G. Cao and M. Singhal, "Mutable checkpoints: A new checkpointing approach for mobile computing systems," IEEE Trans. Parallel Distrib. Syst., vol. 9, no. 9, pp. 891–903, Sep. 1998.

[12] A. Kumar and S. Singh, "Machine learning-based adaptive checkpointing for fault tolerance in mobile distributed systems," IEEE Access, vol. 9, pp. 114256–114272, 2021.

[13] X. Jiang and C. Z. Xu, "MOCHA: Mobile computing for heterogeneous and adaptive mobile grid environments," IEEE Trans. Parallel Distrib. Syst., vol. 18, no. 7, pp. 938–951, Jul. 2007.

[14] A. Kumar and R. Sharma, "Recovery-line stockpiling for efficient rollback in mobile distributed systems," in Proc. IEEE Int. Conf. Mobile Data Management, 2023, pp. 112–121.

[15] R. H. B. Netzer and J. Xu, "Necessary and sufficient conditions for consistent global snapshots," IEEE Trans. Parallel Distrib. Syst., vol. 6, no. 2, pp. 165–169, Feb. 1995.

[16] S. Garg, A. Puliafito, M. Telek, and K. S. Trivedi, "Analysis of software rejuvenation using Markov regenerative stochastic Petri net," in Proc. 6th IEEE Int. Symp. Software Reliability Engineering, 1995, pp. 180–187.

[17] T. R. Gadekallu et al., "Blockchain for edge intelligence-enabled fault tolerance in mobile networks," IEEE Network, vol. 35, no. 4, pp. 166–173, Jul./Aug. 2021.

[18] P. Gupta and N. Mittal, "An efficient coordinated checkpointing algorithm for mobile distributed systems," Int. J. Distributed Sensor Networks, vol. 7, no. 1, pp. 1–12, 2011.

[19] Y. M. Wang, "Consistent global checkpoints that contain a given set of local checkpoints," IEEE Trans. Comput., vol. 46, no. 4, pp. 456–468, Apr. 1997.

[20] L. Lamport, "Time, clocks, and the ordering of events in a distributed system," Commun. ACM, vol. 21, no. 7, pp. 558–565, Jul. 1978.

[21] A. Acharya and B. R. Badrinath, "Checkpointing distributed applications on mobile computers," in Proc. 3rd Int. Conf. Parallel and Distributed Information Systems, 1994, pp. 73–80.

[22] D. Saha and S. K. Tripathi, "Efficient support for checkpoint recovery in mobile computing environments," in Proc. 16th ACM Symp. Principles of Distributed Computing, 1997, pp. 269–278.

[23] B. Randell, "System structure for software fault tolerance," IEEE Trans. Softw. Eng., vol. SE-1, no. 2, pp. 220–232, Jun. 1975.

[24] V. Goyal and R. K. Chauhan, "A survey of checkpointing and recovery protocols for mobile distributed systems," J. Netw. Comput. Appl., vol. 122, pp. 35–50, Nov. 2018.

[25] F. Cristian, "Understanding fault-tolerant distributed systems," Commun. ACM, vol. 34, no. 2, pp. 56–78, Feb. 1991.