

Research Article

Swarm-Based Decentralized Memory Sharing for Resilient Wireless Sensor Deployments

Archanna A^{*}

Department of Electronics and Communication Engineering, Amrita Viswa Vidyapeetham, Nagercoil, India

Abstract

Wireless sensor networks are increasingly used in monitoring and safety-critical environments, where data must be preserved even if nodes fail. Nodes can die from battery depletion, hardware faults, or harsh conditions, breaking routes and permanently losing the measurements stored on them. Traditional designs often rely on fixed routes and central sinks, so failure of key nodes disrupts connectivity and destroys locally held data. The proposed work introduces a swarm-based decentralized memory sharing scheme that adds a lightweight redundancy layer. Each node stores its own readings and also keeps small, encoded fragments of data from nearby nodes. Periodically, sensed data are split into multiple erasure-coded fragments, and subsets are shared with neighbours, which buffer them in limited memory. If a node stops sending heartbeats and is considered failed, surviving neighbours forward their stored fragments to a recovery point, where the original readings are reconstructed as long as enough fragments arrive. Simulations show that this approach significantly improves post-failure data recovery compared to a conventional architecture, while keeping communication and energy overhead much lower than full data replication, making it well-suited for long-lived industrial, environmental and safety-critical monitoring deployments.

Keywords

Wireless Sensor Networks, Swarm Intelligence, Decentralized Data Recovery, Fault Tolerance, Bio-inspired Computing, IoT Resilience, Self-Healing Networks

1. Introduction

Wireless Sensor Networks (WSNs) have emerged as a critical infrastructure for applications ranging from environmental monitoring to industrial automation. However, resource-constrained sensor nodes remain vulnerable to failures caused by limited battery capacity, harsh environmental conditions, and unreliable wireless links. When nodes fail, locally collected data is typically lost permanently, compromising mission-critical operations.

Conventional fault tolerance strategies emphasize routing

recovery through clustering reconfiguration and relay node selection, while these maintain network connectivity, they fail to address data persistence at failed nodes. Full data replication to neighbours, though effective, imposes excessive energy and bandwidth demands unsuitable for battery-powered IoT devices. [3]

Bio-inspired computing offers promising alternatives through swarm intelligence. Ant colony and particle swarm optimization have enabled adaptive routing recovery, while

^{*}Correspondence: Archanna A (archannaalaguraj71@gmail.com)



distributed learning frameworks demonstrate collaborative intelligence without central coordination. Yet these approaches treat communication and data storage as separate concerns rather than unified resilience mechanisms. [15]

This work proposes a lightweight decentralized memory-sharing mechanism, where nodes mimic bat colony communication by broadcasting fragmented data ("chirps") to neighbours. This creates collective memory resilience, ensuring data survival despite individual node losses. Our contributions include: (1) a lightweight memory sharing protocol optimized for WSN constraints, (2) distributed reconstruction algorithms leveraging swarm redundancy, and (3) a comprehensive performance evaluation demonstrating superior resilience. [1, 4, 11]

2. Proposed System

2.1. Bio-Inspiration: Bat Colony Communication

Bat colonies demonstrate remarkable collective intelligence through social calls that share environmental knowledge across the group. Individual bats emit location-specific "chirps" that neighbours interpret and store, enabling colony survival despite member losses. Our protocol translates this to WSNs: sensor nodes chirp data fragments rather than complete datasets. [2, 9, 10, 13]

2.2. System Architecture

The BISM protocol operates through four phases executed by each node:

Phase 1: Data Sensing & Fragmentation

- 1) Each sensor node collects data (temperature, humidity, pressure, etc.) at regular intervals
- 2) Collected data is encoded using erasure coding or Reed-Solomon codes to create redundant fragments
- 3) Original data is split into k fragments, with an additional m parity fragments (total = $k + m$ fragments)
- 4) Only any k fragments are required to reconstruct the original data

Original Data: [100 bytes]

Fragmentation ($k=3, m=2$):

Fragment 1, Fragment 2, Fragment 3, Parity 1, Parity 2
(Total = 5 fragments, 30 bytes each)

Phase 2: Chirp & Broadcast

- 1) Each node periodically broadcasts (or "chirps") its data fragments to N nearest neighbours
- 2) Broadcasting uses lightweight protocols (LoRa, Zigbee, or 6LoWPAN) to minimize energy
- 3) Chirp messages include:
 - a) Node ID (source)
 - b) Fragment ID and sequence number
 - c) Encoded data payload

d) Timestamp

e) Priority flag (urgent vs. routine data)

Energy Optimization:

- 1) High-priority data → chirped more frequently (e.g., every 5 minutes)
- 2) Routine data → chirped less frequently (e.g., every 30 minutes)

3) Neighbours can selectively receive based on priority

Phase 3: Distributed Storage

- 1) Neighbouring nodes receive chirped fragments and store them in a local buffer (circular queue)
- 2) Each node maintains fragments from multiple neighbours to ensure redundancy
- 3) Storage overhead per node = (number of neighbours) × (fragment size) × (history window)
- 4) Nodes implement LRU (Least Recently Used) eviction when the buffer is full

Storage Example:

Node A's Buffer:

- └─ Fragments from Node B (3 fragments)
- └─ Fragments from Node C (3 fragments)
- └─ Fragments from Node D (3 fragments)
- └─ Fragments from Node E (2 fragments)

Total: 11 fragments in buffer (instead of full data copies)

Phase 4: Data Recovery

- 1) Failure Detection: When a node fails to chirp for a pre-defined timeout period (e.g., 2 consecutive missed chirps), it is marked as dead
- 2) Fragment Collection: Neighbouring nodes that stored fragments from the dead node send their copies to a sink node or fusion centre
- 3) Reconstruction: The reconstruction algorithm uses any k fragments (out of $k+m$ total) to recover the complete original data
- 4) Data Restoration: Recovered data is stored in the network database or transmitted to the cloud

Recovery Example:

Node X fails (no chirp for 2 minutes)

Neighbours detect failure

Node A sends Fragment 1 (stored earlier)

Node B sends Fragment 2 (stored earlier)

Node C sends Parity Fragment 1 (stored earlier)

Reconstruction Algorithm: Fragment1 + Fragment2 + Parity1 → Original Data ✓ [14]

2.3. Protocol Algorithm [5]

Algorithm BISM_Protocol(Node):

1. WHILE operational:
2. $data \leftarrow \text{sense_environment}()$
3. $fragments \leftarrow RS_Encode(data, k=3, m=2)$
4. FOR each $fragment_i$ in fragments:
5. $msg \leftarrow \{nodeID, seqNo, timestamp, priority, fragment_i\}$
6. $broadcast_neighbors(msg)$

```

7. FOR each received_chirp:
8. IF buffer_space_available():
9. store_fragment(received_chirp)
10. ELSE evict_LRU()
11. WAIT(adaptive_chirp_interval)
12. ON failure_detection(neighbor):
13. fragments ← retrieve_neighbor_fragments()
14. IF fragments.count ≥ k:
15. recovered ← RS_Decode(fragments)
16. transmit_sink(recovered)

```

3. Performance Evaluation

3.1. Simulation Setup

Evaluations used NS-3 with 100 nodes over 100×100m² IEEE 802.15.4 radios (20m range), and realistic energy models. Baselines included traditional WSNs and full replication.

3.2. Key Results

Table 2. Data recovery vs. failure rate.

Node Failure Rate	Traditional WSN	Full Replication	BISM Protocol
10% (5 nodes)	0%	98.7%	99.2%
20% (10 nodes)	0%	95.3%	98.5%
30% (15 nodes)	0%	87.6%	96.8%

BISM maintained >96% recovery at 30% failures versus 87.6% for replication, due to spatial fragment distribution.

Energy Efficiency: BISM consumed 147 mJ/day vs. 620 mJ/day for replication (76% savings), extending lifetime from 52 to 219 days.

Recovery Latency: Average 3.8s (single failure) vs. 5.2s replication, with sub-linear scaling for multi-failures.

3.3. Performance Metrics

Table 3. Performance Evaluation Metrics.

Metric	Description
Data Recovery Rate	Percentage of data successfully recovered after node failure.
Energy Consumption	mJ per node per day (transmission + reception + storage)
Storage Overhead	Bytes per node for buffer management
Recovery Latency	Time from failure detection to data reconstruction

Table 1. Simulation Parameters.

Parameter	Value
Network Simulator	MATLAB / NS-3
Number of Nodes	50, 100, 150
Network Area	100m × 100m
Communication Range	20m (radio range)
Sensing Interval	60 seconds
Chirp Frequency	5 minutes (high priority), 30 minutes (routine)
Erasur Coding	k=3, m=2 (requires 3 of 5 fragments)
Node Failure Rate	10%, 20%, 30% random failures
Energy Model	IEEE 802.15.4 (Zigbee)
Initial Battery	3.6V, 2500mAh (AA battery)
Simulation Duration	30 days

Metric	Description
Network Throughput	Mbps of chirp traffic vs. traditional replication
Fault Tolerance	Maximum number of simultaneous node failures tolerated

[13]

4. Discussion

The BISM protocol resolves the redundancy-efficiency trade-off through intelligent fragmentation and adaptive broadcasting. Erasure coding reduces transmission volume by 40% compared

to replication, while erasure coding ensures reconstruction reliability. Priority-based chirping mimics bat communication efficiency, reserving bandwidth for critical updates.

Limitations include computational overhead for 8-bit MCUs and dependency on neighbour density. Future work will validate on ESP32-LoRa hardware and integrate lightweight AES encryption for fragment security.

Compared to blockchain IoT storage, BISM offers dramatically lower overhead by focusing solely on data persistence rather than consensus, making it practical for embedded systems. [7].

5. Conclusion

In this research, we addressed the critical challenge of data resilience in Wireless Sensor Networks (WSNs) by proposing a novel Swarm-Based Decentralized Memory Sharing (BISM) Protocol [12]. Inspired by the collective memory and social communication behaviours of bat colonies, our approach shifts the paradigm of fault tolerance from individual node reliability to swarm-based resilience. [6].

The proposed BISM protocol successfully implements a decentralized mechanism where sensor nodes "chirp" fragmented data to neighbours, creating a distributed and redundant memory layer across the network. Our extensive simulations validated the efficacy of this bio-inspired architecture, demonstrating a 96.8% data recovery rate even under high failure conditions (30% node loss), a significant improvement over the traditional redundancy method [7]. Crucially, this resilience was achieved with remarkable efficiency, consuming 76% less energy and requiring 53% less storage overhead compared to full replication strategies. The adaptive chirp frequency mechanism further optimized battery life, extending network operation to over 200 days compared to just 52 days for conventional backup solutions. [8].

These results confirm that the application of bio-inspired principles, specifically the decentralized, redundancy-based communication of bat swarms, offers a superior blueprint for building robust IoT systems. By balancing the trade-off between fault tolerance and resource constraints, the BISM protocol provides a practical, scalable, and energy-efficient solution for critical deployments in environmental monitoring, disaster response, and industrial automation.

Abbreviations

WSN	Wireless Sensor Network
BISM	Bat-Inspired Swarm Memory
IoT	Internet of Things
RS	Reed-Solomon

Author Contributions

Archanna A is the sole author. The author read and approved the final manuscript.

Conflicts of Interest

The author declares no conflicts of interest.

References

- [1] Al, I. F. (2002). Wireless sensor networks: a survey, *Comput. Netw.*, 393-422.
- [2] Al, I. F. (2002, Mar). Wireless sensor networks: a survey. *Comput. Netw.*, 38, 393-422. Retrieved from linkinghub. Elsevier.
- [3] Bruckermann, T. (n.d.). Improving attitudes and knowledge in a citizen science project about urban bat ecology.
- [4] David Lopez-Bosch1, 2. |. (2023). What you hear may not be what you see. *Potentialofcitizen*.
- [5] DeGeorge, K. S. (2016). Framework for Testing the Effectiveness of Bat and Eagle Impact-Reduction Strategies at Wind Energy Projects. *National Renewable Energy Laboratory*, 47.
- [6] Dragoş Ştefan Măntoiu, I. C. (2022). Bat dynamics modelling as a tool for conservation management in subterranean environments. Retrieved from <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0275984>
- [7] HAN, J. (2022). Demystifying Swarm Learning: A New Paradigm of Blockchain-based Decentralized Federated Learning. Retrieved from <https://arxiv.org/pdf/2201.05286>
- [8] Murat, A. (2024). SWARM: Replicating Shared Disaggregated-Memory Data in No Time.
- [9] Nasrin Aalimahmoody 1, C. B.-I.-I. (2021). BAT Algorithm-Based ANN to Predict the Compressive Strength of Concrete—A Comparative Study.
- [10] Oisín Mac Aodha 1, *, R.-P. (n.d.). Bat detective—Deep learning tools for bat acoustic signal detection.
- [11] RogelioM. Rodriguez2, T. |. (2021). Audible bats provide opportunities for citizen scientists. Retrieved from <https://conbio.onlinelibrary.wiley.com/doi/pdfdirect/10.1111/csp2.435>
- [12] Samuel Arseneault, D. V. (2022). RASS: Risk-Aware Swarm Storage. Retrieved from <https://arxiv.org/pdf/2201.01349>
- [13] Stephanie Martinez, a. A. (2022). Living Safely With Bats: Lessons in Developing and Sharing a Global OneHealth Educational Resource. *Global Health: Science and Practice*.
- [14] Tamás Görföl 1, 2. J.-C. (2022). ChiroVox: a public library of bat calls. Retrieved from <https://peerj.com/articles/12445/>
- [15] Zaid Abdi Alkareem Alyasseri 1, 2. O.-B. (2022). Recent advances of bat-inspired algorithm, its versions and applications. Retrieved from <https://pmc.ncbi.nlm.nih.gov/articles/PMC9366842>