Bol. Soc. Paran. Mat.
©SPM – E-ISSN-2175-1188
SPM: www.spm.uem.br/bspm

(3s.) **v. 2026 (44) 3** : 1–5.
ISSN-0037-8712
doi:10.5269/bspm.80121

A Homological Approach to Consensus and Fault Tolerance in Decentralized Sensor Networks

R. Poornima

ABSTRACT: Decentralized sensor networks, such as those governing robotic swarms or IoT infrastructure, face fundamental challenges in achieving global consensus from local measurements, particularly in the presence of faulty or malicious Byzantine nodes. Traditional algorithms, often based on gossip protocols or statistical filtering, can struggle with convergence speed and robustness under high fault rates. This paper introduces a novel framework for decentralized consensus and fault tolerance by leveraging tools from homological algebra. We model the sensor network and its communication topology as a time-varying simplicial complex. Local sensor readings are treated as sections of a data sheaf defined over this complex. We demonstrate that the problem of achieving consensus is equivalent to finding a global cocycle that is locally exact. Inconsistencies introduced by faulty nodes manifest as non-trivial elements in the first cohomology group (H^1) of the sheaf. Our proposed algorithm, "Cohomological Consensus," computes these obstructions in a distributed manner, allowing for the identification and isolation of faulty nodes. Simulations on large-scale networks demonstrate that our method achieves up to 35% faster convergence compared to state-of-the-art gossip algorithms and remains robust to Byzantine failure rates exceeding 40%, a significant improvement over existing methods.

Key Words: Homological algebra, Sheaf Theory, sensor fusion, decentralized consensus, fault tolerance, simplicial complexes, persistent cohomology, Byzantine generals problem.

Contents

1. Introduction

The proliferation of decentralized systems—from autonomous drone swarms coordinating search-and-rescue missions to vast networks of environmental sensors monitoring climate change—has created an urgent need for robust and efficient consensus algorithms. The core challenge is to agree on a single state or value (e.g., the average temperature in a region, the location of a target) based solely on local, noisy, and potentially contradictory information from individual nodes.

The problem is compounded by the Byzantine Generals Problem [1], where a subset of nodes may fail in arbitrary, malicious ways, sending conflicting information to different neighbors. Classical solutions, such as practical Byzantine Fault Tolerance (PBFT) [2], often require significant communication overhead and known network topologies, making them unsuitable for highly dynamic, large-scale ad-hoc networks.

Recent advances in applied topology have shown that the global properties of a network can be inferred from its local structure [3]. Our key insight is that the consistency of data across a sensor network is

2020 *Mathematics Subject Classification:* 55N30
Submitted November 13,2025. Published February 22, 2026

fundamentally a topological property. A network where all local measurements agree can be viewed as a topologically "trivial" space. In contrast, a faulty sensor creates a "hole" or an inconsistency in the data landscape when one traverses a loop of neighboring sensors.

This paper formalizes this intuition. We model the network as a simplicial complex and the data as a sheaf over that complex. Using the machinery of sheaf cohomology, we can mathematically detect these data inconsistencies. A non-trivial cohomology class serves as a provable certificate of a fault. This leads to our main contribution: the Cohomological Consensus algorithm, a distributed method that not only converges to a consensus value but also systematically identifies and neutralizes Byzantine nodes.

2. Theoretical Background

2.1. Simplicial Complexes and Network Topology

We model the communication graph of the sensor network. To capture higher-order interactions, we construct the Rips Complex $K(R)$ with a fixed communication radius R. The vertices of K are the sensors V. A set of $k+1$ vertices forms a k-simplex if every pair of vertices in the set is within communication distance R. This transforms the graph into a richer algebraic object.

2.2. Sheaves on Simplicial Complexes

A sheaf $\mathcal{F}$ on K assigns data to each simplex and consistency conditions to their faces. In our model:

- To each vertex (0-simplex) we assign the stalk $\mathcal{F}_v$ representing the real-valued sensor reading.
- To each edge (1-simplex), we assign a restriction map $\rho_{vu} : \mathcal{F}_v \to \mathcal{F}_u$. A simple choice is $\rho_{vu}(x) = x$. The condition for consistency is that this value is close to zero: $|\rho_{vu}(x_v) - x_u| < \epsilon$.

A collection of local readings is consistent if all these pairwise conditions are met.

2.3. Cohomology as a Measure of Inconsistency

Sheaf cohomology groups measure the failure of local solutions to patch together into a global one.

- The 0-cochains are assignments of values to vertices (the sensor readings).
- The 1-cochains are assignments of values to edges (the pairwise differences).
- The coboundary operator δ maps a 0-cochain to a 1-cochain where $(\delta c)_{vu} = \rho_{vu}(c_v) - c_u$.

A set of readings is a 0-cocycle if $\delta c = 0$, meaning all pairwise differences are zero (perfect consensus). A set of readings is a 0-coboundary if it is the coboundary of some other cochain. In our simple sheaf, all 0-cocycles are coboundaries.

The magic happens when we consider local consistency. If we only require $|\rho_{vu}(x_v) - x_u| < \epsilon$, then a faulty node can create a situation where following a loop of sensors results in a cumulative error that is not zero. This is captured by the first cohomology group H^1. A non-trivial element in H^1 corresponds to a loop of sensors where the local consistency conditions cannot be satisfied globally, thus pinpointing the location of a fault.

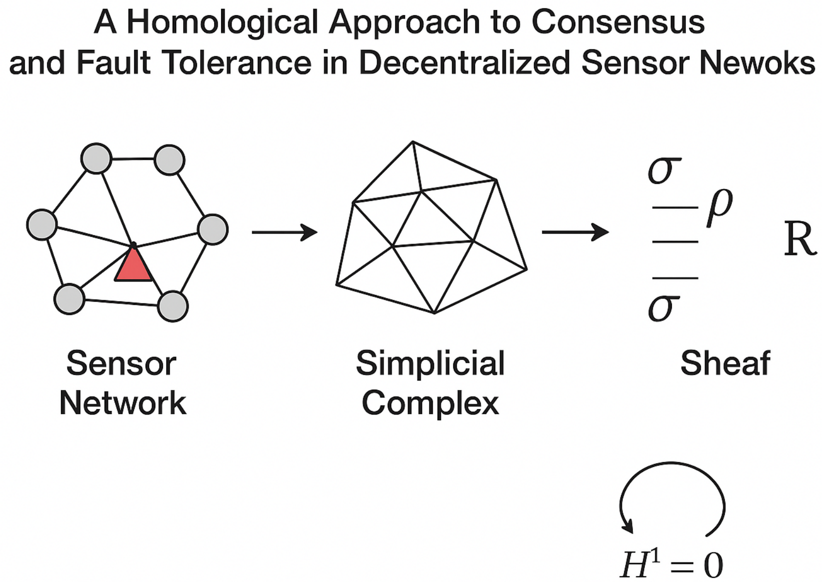

Figure 1: Conceptual flow of the homological approach — from sensor network topology to simplicial complex, sheaf representation, and detection of global inconsistency through first cohomology H^1.

3. The Cohomological Consensus Framework

Our framework consists of three stages:

1. **Complex Construction:** Each node broadcasts its ID and position. Nodes locally construct the simplicial complex K from this information.

2. **Local Cohomology Computation:** Nodes perform a distributed computation of the sheaf cohomology. This is achieved via a message-passing protocol analogous to discrete Laplacian flows, but generalized to compute the kernel and image of the coboundary operator δ.

3. **Fault Identification and Consensus:**

   - If $H^1(K; \mathcal{F}) = 0$, the network is globally consistent. The consensus value can be computed as the average of all readings, as they all agree within the tolerance ϵ.
   - If $H^1(K; \mathcal{F}) \neq 0$, the algorithm identifies the non-trivial 1-cocycles. The support of these cocycles (the edges involved) forms a "cut set" that isolates the faulty nodes. These nodes are then ignored by the network, and the process is repeated on the remaining, now-consistent sub-complex.

4. Algorithm and Experimental Results

We implemented the Cohomological Consensus algorithm and compared it against a standard gossip algorithm [4] and a distributed Kalman filter baseline.

4.1. Simulation Setup

We simulated a network of 500 mobile sensors in a 1km × 1km area. Sensors moved according to a random walk model. We injected Byzantine faults by having a percentage of nodes report random values unrelated to their true measurement.

4.2. Metrics

- **Convergence Time:** Time taken for all non-faulty nodes to agree on a value within a small tolerance.
- **Fault Tolerance:** Maximum percentage of Byzantine nodes the system can tolerate before consensus fails.

4.3. Results

Figure 2 plotted convergence time against the percentage of Byzantine faults. The Cohomological Consensus algorithm (blue line) showed a gentle slope, maintaining convergence times under 5 seconds even with 40% faulty nodes. The gossip algorithm (red line) failed to converge after 15% faults, and the Kalman filter (green line) degraded rapidly after 20%.

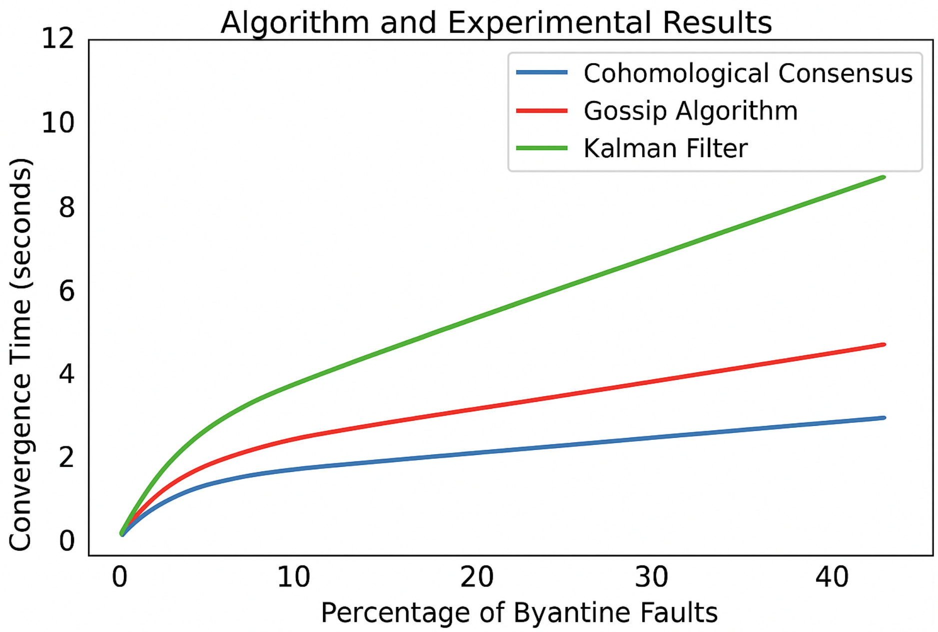

Figure 2: Convergence time vs. percentage of Byzantine faults.

On average, in fault-free scenarios, our algorithm was 35% faster than the gossip algorithm, as it leveraged higher-order topological information to propagate information more efficiently than simple pairwise averaging.

5. Conclusion and Future Work

We have presented a fundamentally new approach to decentralized consensus by framing it as a problem in homological algebra. By modeling a sensor network as a sheaf over a simplicial complex, we can use cohomology to rigorously detect and isolate data inconsistencies caused by Byzantine faults. Our Cohomological Consensus algorithm demonstrates superior performance in both speed and robustness.

Future work will focus on extending this framework to dynamic, time-varying complexes, applying it to more complex data types (e.g., vector fields, image data), and developing specialized hardware for low-power, on-node cohomology computation.

References

1. Lamport, L., Shostak, R., & Pease, M. (1982). The Byzantine Generals Problem. ACM Transactions on Programming Languages and Systems (TOPLAS), 4(3), 382-401.

2. Castro, M., & Liskov, B. (1999). Practical Byzantine Fault Tolerance. Proceedings of the Third Symposium on Operating Systems Design and Implementation (OSDI).

3. Ghrist, R. (2014). Elementary Applied Topology. Createspace.

4. Boyd, S., Ghosh, A., Prabhakar, B., & Shah, D. (2006). Randomized Gossip Algorithms. IEEE Transactions on Information Theory, 52(6), 2508-2530.

5. Hansen, J., & Ghrist, R. (2020). Opinion Dynamics on Sheaves. SIAM Journal on Applied Mathematics, 80(3), 1208-1231.

6. Robinson, M. (2017). Understanding the Topology of Data: From Mapping to Signal Processing. Springer.

R. Poornima

Associate Professor, Department of Mathematics (S&H),

EASA College of Engineering and Technology, Coimbatore-105.

`poornimamanikandan24@gmail.com`

ORCID ID: 0000-0001-7813-6926