Implementation and Performance Analysis of the Link Expiration Time (LET) Clustering Algorithm

CENG797 Term Project

Melih Arda Metin

Middle East Technical University
Dept. of Computer Engineering

January 2, 2026

Problem Definition

- MANETs lack fixed infrastructure; nodes move constantly.
- Traditional algorithms (e.g., Lowest-ID) are **Reactive**.
- They wait for links to break before selecting a new leader.

Consequences

- Frequent Link Breakages
- Packet Loss & High Latency
- "Broadcast Storms" due to excessive control messages

Core Idea: Change from Reactive to **Proactive**.

The **Link Expiration Time (LET)** metric predicts the future:

- Instead of random selection, we calculate the **connection duration**.
- We choose the leader that will stay connected the longest.

Inputs for Prediction:

- Relative Velocity (v)
- Relative Position (x, y)
- Transmission Range (r)

The exact link duration (t) is calculated using kinematic parameters:

$$\text{LET} = \frac{-(ab + cd) + \sqrt{(a^2 + c^2)r^2 - (ad - bc)^2}}{a^2 + c^2} \tag{1}$$

Where:

- **a, c**: Relative Velocity vector components
- **b, d**: Relative Distance vector components

This formula gives us the precise time remaining until disconnection.

Distributed Election

Each node calculates a **Stability Weight (W_i)**:

$$W_i = \sum_{j \in Neighbors} \text{LET}_{ij}$$

Score Mechanism (Loop Prevention)

To break ties and prevent loops, we use a composite score:

$$\text{Score}_i = W_i + \frac{1}{\text{ID}_i}$$

The node with the strictly highest score becomes the **Cluster Head**.

Environment:

- **Simulator:** OMNeT++ 6.2.0
- **Framework:** INET 4.5.4
- **Area:** $1000m \times 1000m$

Scenarios:

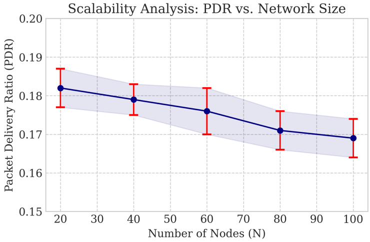
1. Scalability: $20 \rightarrow 100$ nodes
2. Mobility: $5 \rightarrow 35$ m/s

Parameters:

- **Mobility:** Random Waypoint (High Stress)
- **MAC:** IEEE 802.11
- **Traffic:** UDP Applications

Live Demo

Switching to Simulation Environment...



(Visualization of Cluster Head selection under 35 m/s mobility)

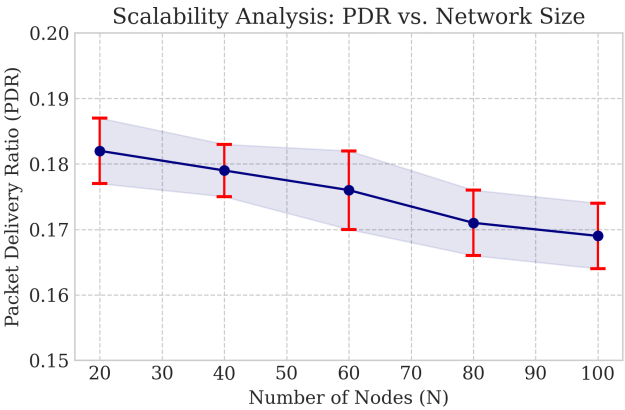


Figure: PDR vs. Network Size. Degradation is less than 2%.

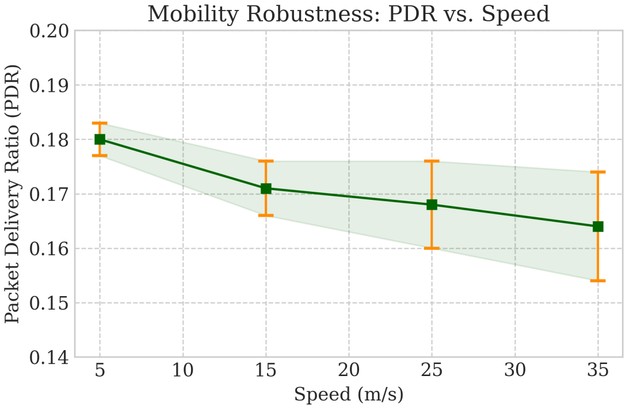


Figure: PDR vs. Speed. Connectivity maintained at highway speeds.

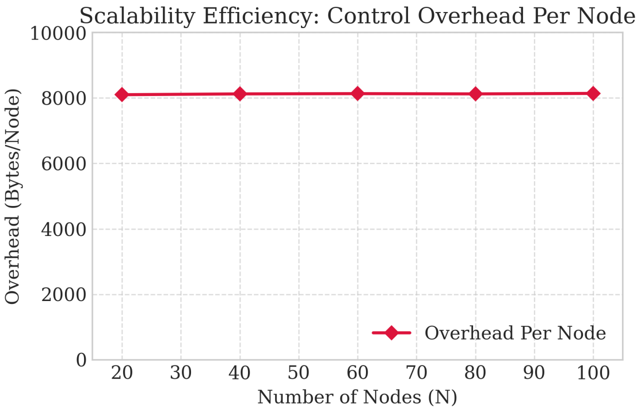


Figure: Control Overhead per Node. **O(1)** complexity demonstrated.

- **Success:** LET algorithm successfully predicts link breaks.
- **Scalability:** Proven O(1) overhead ensures scalability.
- **Robustness:** Works effectively under high-mobility.
- **Future Work:**
  - Integrate TCP for reliability.
  - Implement Energy-Aware Cluster Head rotation.

Thank You