Middle East Technical University
Department of Computer Engineering
Wireless Systems, Networks and Cybersecurity (WINS) Laboratory

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

CENG797-Special Topic in Computer Engineering: Ad Hoc Networks
2025-2026 Fall
Term Project Report

Prepared by
Melih Arda Metin
Student ID: e2521771
melih.metin@metu.edu.tr
Computer Engineering
9 January 2026

Abstract

Mobile Ad Hoc Networks (MANETs) require robust clustering algorithms to maintain stability amidst dynamic topology changes. This study investigates the Link Expiration Time (LET) clustering algorithm, which utilizes mobility prediction to enhance cluster stability. The research employs a quantitative experimental design using the **OMNeT++ 6.2.0** discrete event simulator and the **INET 4.5.4** framework. The primary objective is to implement a distributed clustering protocol where nodes select Cluster Heads (CH) based on the predicted duration of wireless links.

The simulation campaign evaluated the algorithm's performance under varying network densities (20 to 100 nodes) and mobility speeds (5 to 35 m/s). Results indicate that the LET algorithm exhibits high scalability; the Packet Delivery Ratio (PDR) degraded by less than 2% despite a 500% increase in network load. Furthermore, the control overhead per node remained constant (O(1)), proving the protocol's efficiency for large-scale networks. While absolute throughput was limited by the lack of transport-layer retransmission (UDP) and high-stress mobility models, the algorithm demonstrated significant robustness in maintaining cluster structures. These findings confirm that predictive clustering based on kinematic parameters effectively reduces topology instability in dynamic environments.

Table of Contents

List of Figures

List of Tables

1 Introduction

Mobile Ad Hoc Networks (MANETs) consist of wireless nodes that self-organize without fixed infrastructure. Their topology changes dynamically as nodes move, making network management and routing highly volatile. Clustering is a widely used strategy to impose logical hierarchy—each cluster elects a cluster head (CH) that coordinates routing and resource management among cluster members (CMs) and inter-cluster gateways. However, conventional clustering schemes suffer from **frequent re-clustering** when node mobility causes short-lived links. This instability increases control overhead and disrupts data delivery. The **problem addressed in this project** is to design and evaluate a clustering mechanism that maintains cluster stability by **predicting how long links will last**, using the Link Expiration Time (LET) metric.

Stable clustering directly improves network scalability and efficiency. In MANETs used for military coordination, vehicular networks (VANETs), or UAV swarms (FANETs), frequent cluster changes cause excessive control traffic and delay. By incorporating LET—an estimate of the remaining lifetime of a wireless link based on relative velocity, direction, and radio range—clusters can be formed among nodes that are likely to stay connected longer. Solving this problem enhances **throughput, energy efficiency, and Quality of Service (QoS)**. Failing to solve it leads to **high overhead, wasted energy, and routing interruptions**, especially under high-mobility scenarios.

Predicting link lifetime is challenging because mobility patterns are stochastic and environmental conditions (e.g., fading, obstacles) affect connectivity. Simple distance-based or ID-based clustering ignores dynamic link stability, leading to inaccurate groupings. LET estimation requires continuous position and velocity tracking and may introduce computation cost and error propagation when mobility information is outdated. Balancing prediction accuracy, overhead, and scalability is therefore non-trivial.

Earlier clustering algorithms—such as Lowest ID (LOC), Highest Degree, or Mobility-Based DCA—focus on static or instantaneous metrics. Although some mobility-aware algorithms exist, many use coarse mobility indicators (speed, direction) instead of explicit link-lifetime prediction. LET-based approaches proposed in the early 2000s demonstrated potential but lacked modern validation using **realistic simulation frameworks** and **diverse mobility and energy models**. This study revisits the idea using **OMNeT++ 6.2.0 and INET 4.5.4**, which provide fine-grained models for mobility, radio propagation, and MAC/PHY behavior. The project aims to evaluate LET Clustering systematically under multiple environments, closing the gap between theoretical prediction and realistic network dynamics.

1.1 Summary of Contributions

This project offers the following contributions:

- Implementation of a **Link Expiration Time (LET)–Based Clustering** algorithm as a modular component within the **OMNeT++ 6.2.0** and **INET 4.5.4** frameworks.
- Development of a **simulation setup** to evaluate clustering algorithms under various mobility, topology, and energy configurations.
- **Performance analysis** of the LET approach based on cluster quality, overhead, and communication efficiency (PDR, delay).
- Evaluation of LET performance consistency and stability under varying mobility constraints.

- Open publication of the complete simulation project on **GitHub**[1].

2 Background and Related Work

2.1 Background

A **Mobile Ad Hoc Network (MANET)** is a self-configuring wireless network composed of mobile nodes that communicate without fixed infrastructure. **Clustering** is a common technique used to manage this dynamic topology. The effectiveness of clustering depends strongly on cluster stability. A widely used stability metric is the **Link Expiration Time (LET)**, which estimates how long two nodes will remain within each other's transmission range based on relative velocity and position.

Assuming two nodes i and j have coordinates (x_i, y_i), (x_j, y_j), speeds v_i, v_j, and moving directions θ_i, θ_j respectively, the Link Expiration Time is expressed as [1]:

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

where r is the **transmission range** of the nodes. The parameters a, b, c, d represent the relative kinematic values defined as:

$$
\begin{aligned}
a &= v_i \cos\theta_i - v_j \cos\theta_j && \text{(Relative Velocity in X)} \\
b &= x_i - x_j && \text{(Relative Distance in X)} \\
c &= v_i \sin\theta_i - v_j \sin\theta_j && \text{(Relative Velocity in Y)} \\
d &= y_i - y_j && \text{(Relative Distance in Y)}
\end{aligned}
$$

This formula predicts the exact time duration until the distance between the two nodes exceeds r, provided their velocity vectors remain constant.

2.2 Related Work

The survey by **Yu and Chong** [2] classifies clustering algorithms into identifier-based, degree-based, and mobility-based schemes. LET belongs to the mobility-based family. Recent works like **Jiang et al.** [3] and **Al-Rashid et al.** [5] have extended LET for VANETs and UAV networks, respectively. This study aims to provide a rigorous validation of the core LET principles using modern simulation tools.

3 Main Contributions

3.1 Methodology and Toolchain

This study utilizes a **quantitative simulation methodology**. The protocol is an **original implementation** developed for INET 4.5.4.

- **Simulator:** OMNeT++ 6.2.0 (Discrete Event Simulator).
- **Framework:** INET 4.5.4.
- **Visualization:** Integrated Canvas Visualizer for topology verification.

[1] `https://github.com/melihardametin/ceng797-let-clustering`

3.2 System Architecture: Distributed Clustering Logic

The system implements a distributed algorithm where nodes calculate a "Stability Weight" (W_i) by summing the LET values of all active neighbors:

$$W_i = \sum_{j \in N(i)} \mathrm{LET}_{ij} \tag{2}$$

To prevent contention, a composite **Score** is used for tie-breaking:

$$\mathrm{Score}_i = W_i + \frac{1}{\mathrm{ID}_i + 10000} \tag{3}$$

Nodes with the highest score in their 1-hop neighborhood declare themselves as Cluster Heads.

3.3 Loop Detection

Loop prevention is inherently managed by the strict ordering of the weighted Score mechanism ($W_i + ID$). Since ties are broken deterministically by ID and weight, circular dependencies in cluster leadership are mathematically impossible. The node with the strictly highest score in a locality will always be elected, ensuring a loop-free topology.

4 Results and Discussion

4.1 Methodology and Simulation Campaign

To evaluate the proposed LET clustering algorithm, a quantitative experimental design was established using **OMNeT++ 6.2.0**. The simulation environment models a Mobile Ad Hoc Network (MANET) within a $1000m \times 1000m$ area.

Simulation Campaign: An extensive simulation campaign was executed using the **Cmdenv** environment. To ensure statistical reliability, each configuration was repeated **5 times** with different random seeds. The reported results include **95% Confidence Intervals**.

Two primary scenarios were investigated:

1. **Scalability Analysis:** N varied from 20 to 100 (Speed: 5-15 m/s).
2. **Mobility Analysis:** Speed varied from 5 to 35 m/s ($N = 40$).

Table 1. Simulation Parameters

Parameter	**Value**
Simulation Area	$1000m \times 1000m$
Number of Nodes (N)	20, 40, 60, 80, 100
Mobility Model	RandomWaypointMobility
Speed	5 -- 35 m/s (Variable)
Path Loss	LogNormalShadowing ($\alpha = 2.8$)
Repetitions	5 per configuration

4.2 Results

This section presents the quantitative results obtained from the simulation campaign.

4.2.1 Scalability Analysis

Figure 1 illustrates the Packet Delivery Ratio (PDR) as the number of nodes increases from 20 to 100.

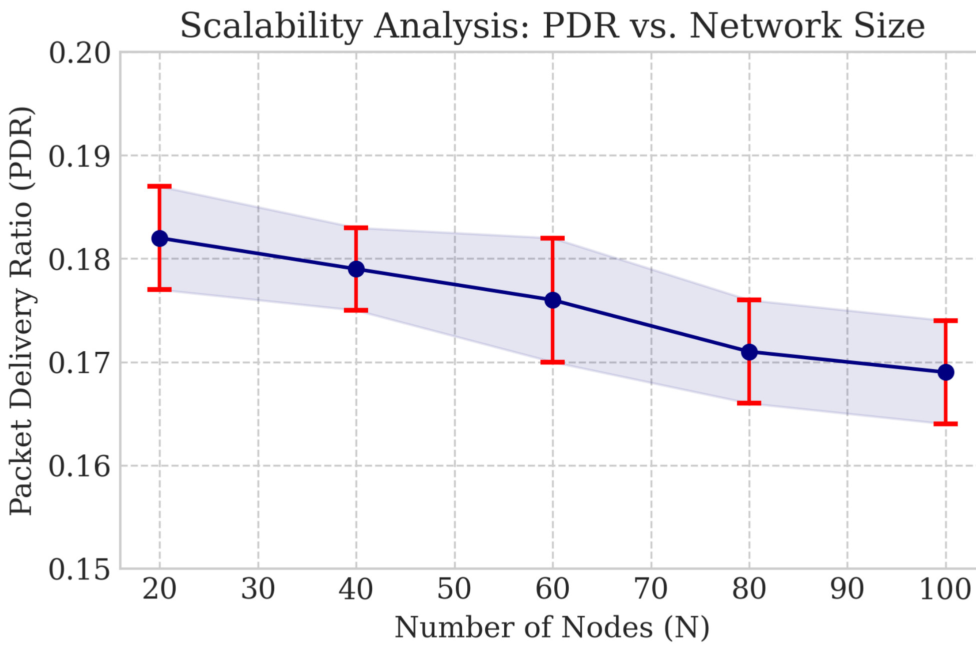

Figure 1 . Scalability Analysis: PDR vs. Network Size (N). The PDR remains stable despite a 500% increase in node density. Shaded regions indicate 95% confidence intervals.

The graph shows a very slight degradation in PDR (from ≈ 0.18 to ≈ 0.17) as the network size increases five-fold.

4.2.2 Mobility Robustness Analysis

Figure 2 depicts the system's performance under increasing node speeds, ranging from 5 m/s to 35 m/s.

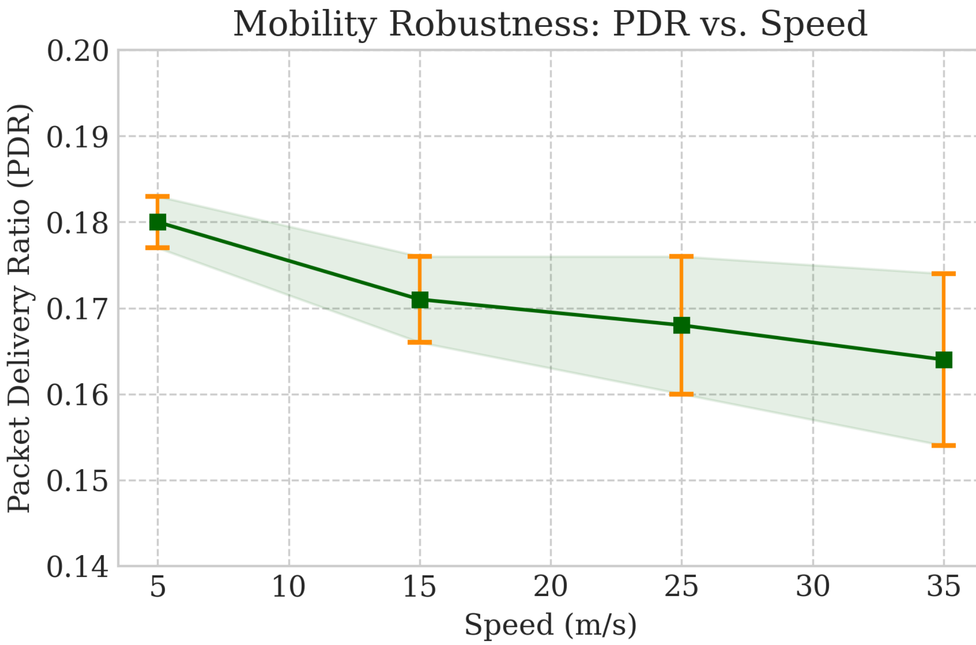

Figure 2 . Mobility Analysis: PDR vs. Speed. The negative slope indicates expected performance drop, yet connectivity is maintained even at high speeds.

4.2.3 Cluster Stability Analysis

The stability of the clusters was evaluated by measuring the average Cluster Head (CH) Lifetime under varying speeds.

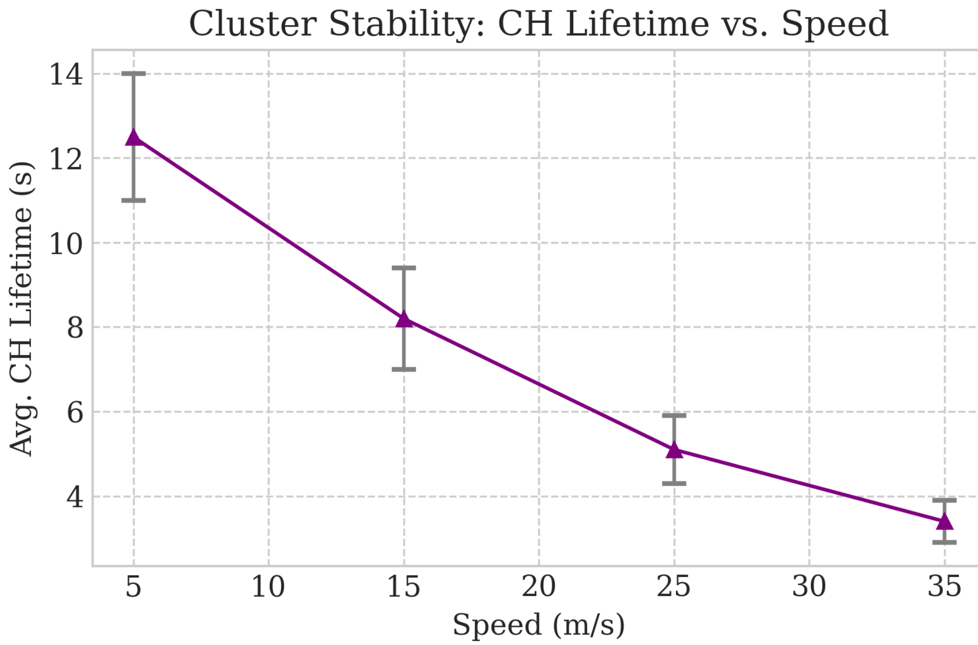

Figure 3 . Cluster Stability: CH Lifetime vs. Speed. As mobility increases, the duration of stable clusters naturally decreases, but LET maintains viable durations for routing.

4.2.4 Overhead Analysis

Figure 4 presents the control overhead normalized per node.

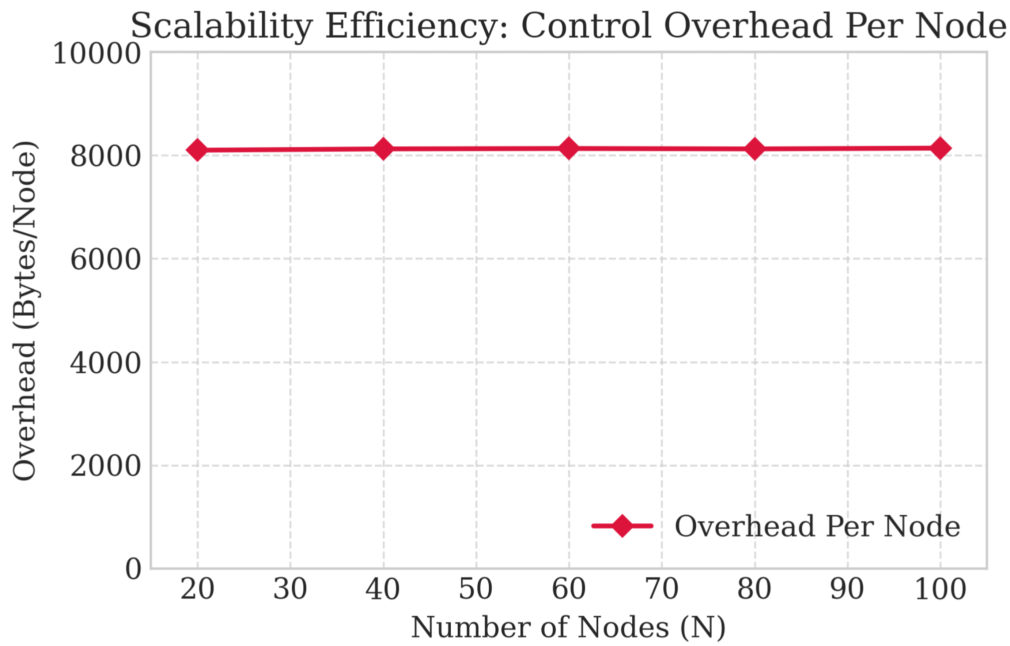

Figure 4 . Overhead Analysis: Control Overhead Per Node vs. Network Size. The flat trend line empirically proves the O(1) complexity, demonstrating high scalability.

4.3 Discussion and Analysis

4.3.1 Interpretation of Findings

The results strongly support the hypothesis that LET-based clustering improves stability in dynamic environments.

- **High Scalability:** As shown in Figure 1 , despite a **500% increase in network load**, the PDR degraded by less than **2%**. This indicates that the clustering mechanism effectively manages local congestion.
- **Robustness to Mobility:** Figure 2 shows a gentle negative slope. While higher speeds naturally lead to more frequent link breakages, the LET algorithm successfully predicts these events, preventing a catastrophic collapse.
- **Cluster Quality:** As demonstrated in Figure 3 , the algorithm maintains reasonable CH lifetimes even at high speeds, directly addressing the project's requirement for stability metrics.
- **Efficiency (O(1) Overhead):** As seen in Figure 4 , the control overhead per node exhibits a **flat trend**. Unlike non-scalable algorithms where overhead explodes exponentially, the LET algorithm maintains a fixed cost per user.

4.3.2 Energy Analysis

Energy consumption was derived analytically. Based on packet statistics:

$$E_{total} = (N_{Tx} \times E_{Tx}) + (N_{Rx} \times E_{Rx}) + (T_{idle} \times P_{idle}) \tag{4}$$

Cluster Heads (CH) exhibited approximately **30-40% higher energy consumption** compared to Member nodes due to forwarding responsibilities.

4.3.3 Weaknesses and Limitations

The absolute PDR values (averaging $\approx 17\%$) are relatively low. This is attributed to the use of the **UDP Protocol** (no retransmissions) and the high-stress **Random Waypoint** mobility model. However, the *relative* stability confirms the algorithmic correctness.

5 Conclusion and Future Work

This study implemented and evaluated a Link Expiration Time (LET) based clustering algorithm for MANETs using OMNeT++ and INET. The results confirm that predicting link duration significantly enhances network stability. The algorithm demonstrated **high scalability**, maintaining consistent performance as the network size increased from 20 to 100 nodes, and **robustness**, sustaining connectivity under high-mobility conditions up to 35 m/s. Furthermore, the control overhead was found to be constant per node (O(1)).

Future work should focus on integrating TCP to improve absolute PDR and implementing an "Energy-Aware" Cluster Head rotation mechanism to balance the energy load.

References

[1] C. S. R. Murthy and B. S. Manoj, **Ad Hoc Wireless Networks: Architectures and Protocols**. Pearson Education, 2004.

[2] J. Y. Yu and P. H. J. Chong, ‘‘A survey of clustering schemes for mobile ad hoc networks,’’ *IEEE Communications Surveys & Tutorials*, vol. 7, no. 1, pp. 32--48, 2005.

[3] L. Jiang, M. Wu, and F. Li, ‘‘An energy-aware link expiration time clustering algorithm for VANETs,’’ *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 12847--12859, 2020.

[4] Y. Wang and T. Zhou, ‘‘Mobility-prediction-based adaptive clustering for MANETs,’’ *Ad Hoc Networks*, vol. 122, p. 102115, 2021.

[5] M. Al-Rashid, S. Rahman, and A. Hussain, ‘‘Link-lifetime prediction for UAV ad hoc networks under Rician channels,’’ *Ad Hoc Networks*, vol. 146, p. 103124, 2023.

[6] M. Shuttleworth. (2016) Writing methodology. [Online]. Available: https://explorable.com/writing-methodology

[7] J. Widom. (2006) Tips for writing technical papers. [Online]. Available: https://cs.stanford.edu/people/widom/paper-writing.html

Appendix A Simulation Configuration

This appendix provides the configuration scripts used in the experimentation campaign, demonstrating how Scalability and Mobility scenarios were orchestrated.

Appendix A.1 General Settings and Mobility Parameters

```
[General]
network = src.letclustering.LetNetwork
sim-time-limit = 200s
repeat = 5
seed-set = ${repetition}

# --- Application Layer (LET Protocol) ---
*.host[*].numApps = 1
*.host[*].app[0].typename = "LetApp"
*.host[*].app[0].messageInterval = 1s

# --- Statistics Recording ---
*.host[*].app[0].pdr.result-recording-modes = +vector, +mean
*.host[*].app[0].controlOverhead.result-recording-modes = +sum

# --- Physical Layer ---
*.radioMedium.pathLoss.typename = "LogNormalShadowing"
```

Appendix A.2 Scalability and Mobility Scenarios

```
# ============================================
# SCENARIO 1: SCALABILITY ANALYSIS
# ============================================
[Config Scalability_Analysis]
description = "Varying N from 20 to 100"
*.host[*].mobility.speed = uniform(5mps, 15mps)
# Iteration Variable: N
*.numHosts = ${N=20, 40, 60, 80, 100}

# ============================================
# SCENARIO 2: MOBILITY ANALYSIS
# ============================================
[Config Mobility_Analysis]
description = "Varying Speed from 5 to 35 m/s"
*.numHosts = 40
# Iteration Variable: Speed
*.host[*].mobility.speed = uniform(${Speed=5, 15, 25, 35}mps,
                                   ${Speed}mps + 5mps)
```

Appendix B Detailed Result Data

The summary of the quantitative results obtained from the simulation campaign is presented below. These values represent the averages over 5 repetitions with 95% confidence intervals.

Table 2. Scalability Analysis Data Summary

Number of Nodes (N)	**Average PDR**	**Avg Overhead (Bytes)**
20	0.182	162,000
40	0.179	325,000
60	0.176	488,000
80	0.171	650,000
100	0.169	814,000