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

Performance Evaluation of the d-Hop Clustering Algorithm in Mobile Ad Hoc Networks

CENG 797 Ad Hoc Networks
2025-2026 Fall
Term Project Report

Prepared by
Efe Kaan Fidancı
Student ID: e278620
kaan.fidanci@metu.edu.tr
Computer Engineering
7 December 2025

Table of Contents

List of Figures . . . iii
List of Tables . . . iv
1 Introduction . . . 1
1.1 The Challenge of Maintaining Stable d-Hop Clusters in MANETs . . . 1
1.2 Practical Importance of Efficient d-Hop Cluster Formation . . . 1
1.3 Objective: Assessing Max-Min d-Cluster Performance . . . 1
1.4 Hypothesis: Stability and Efficiency Through Max-Min d-Clustering . . . 1
1.5 Key Definitions for MANET Clustering Evaluation . . . 1
2 Background and Related Work . . . 2
2.1 Background . . . 2
2.2 Related Work . . . 3
3 Main Contributions . . . 4
3.1 Research Design and Methodology . . . 4
3.1.1 Measurement Tools . . . 4
3.1.2 Experimental Procedure . . . 4
3.1.3 Pilot Study and Implementation Validation . . . 5
3.1.4 Study Variables . . . 6
3.1.5 Sampling and Statistical Analysis . . . 6
3.2 Max-Min d-Cluster Algorithm Design and Implementation . . . 7
4 Results and Discussion . . . 7
4.1 Methodology . . . 7
4.1.1 Materials and Simulation Environment . . . 8
4.1.2 Data Generation and Sampling . . . 8
4.1.3 Statistical Treatment . . . 8
4.1.4 Reproducibility . . . 8

List of Figures

Figure 1 Clustered Network Topology with Gateways [1]. 3
Figure 2 3-cluster Formation in a Network of 25 Nodes [2]. 5
Figure 3 Replicated 3-cluster topology in OMNeT++ simulation. 6

List of Tables

Table 1 Max-Min Cluster Head Validation: Reference vs. OMNeT++ Simulation. 6

1 Introduction

1.1 The Challenge of Maintaining Stable d-Hop Clusters in MANETs

Mobile Ad Hoc Networks (MANETs) rely on decentralized, self-configuring nodes that communicate over wireless links without fixed infrastructure. The absence of centralized control means that each node must independently manage routing, security, and resource allocation [3]. A key challenge in MANETs is maintaining efficient and stable clustering as the network size and mobility increase. This challenge is further compounded by dynamic topologies, where nodes frequently join or leave the network, requiring adaptive clustering algorithms [4]. To address this challenge, this study focuses on implementing and evaluating the Max-Min d-Cluster Algorithm , an established heuristic that aims to balance cluster stability and scalability.

1.2 Practical Importance of Efficient d-Hop Cluster Formation

Clustering—the grouping of nodes into clusters with designated cluster heads—is essential to improving MANET performance because it provides structural organization and supports scalable network management [5, 6]. An efficient clustering algorithm facilitates the formation of desirable and well-organized MANET topologies, enhancing network efficiency, reducing routing overhead, and improving overall communication performance [7]. Conversely, failing to address the clustering problem can result in inefficient and unstable network structures, leading to excessive overhead, uneven load distribution, and frequent topology disruptions. Therefore, solving this problem is a critical step toward achieving optimized, reliable, and sustainable MANET-based communication systems [8].

However, forming and maintaining stable clusters in MANETs is challenging due to node mobility, dynamic topologies, and decentralized control [9]. Simpler approaches, such as 1-hop, often fail to adjust to variations in network density or movement patterns [10]. In other words, they lack the adaptability required for the inherently dynamic nature of MANETs.

1.3 Objective: Assessing Max-Min d-Cluster Performance

This study aims to evaluate the performance of the Max-Min d-Cluster Algorithm in MANETs under varying network topologies, mobility patterns, traffic conditions, and energy constraints. Using OMNeT++ 6.2.0 and INET 4.5.4, the study will simulate network scenarios and measure the algorithm's effectiveness across cluster quality, overhead, communication efficiency, energy/resource usage, scalability, robustness, and fairness metrics.

1.4 Hypothesis: Stability and Efficiency Through Max-Min d-Clustering

It is hypothesized that the Max-Min d-Cluster Algorithm can maintain stable and efficient clusters across a wide range of network conditions, resulting in reduced control overhead, balanced energy consumption, and improved network performance compared to unclustered or poorly clustered MANETs.

1.5 Key Definitions for MANET Clustering Evaluation

- **MANET:** A self-configuring network of mobile nodes communicating without fixed infrastructure.

- **Clustering:** Grouping nodes into clusters with a cluster head to improve network management and performance.
- **d-Hop Clustering Algorithm:** A clustering method where nodes within d hops form a cluster, with one node elected as the cluster head. This study implements the Max-Min heuristic, which utilizes a 2d-round flooding mechanism for cluster head election.
- **Cluster Head:** A node responsible for intra-cluster coordination and inter-cluster communication.
- **Metrics:** Quantitative measures of network performance, including cluster quality, communication efficiency, energy usage, scalability, and fairness.

By systematically analyzing these metrics, this study will provide a comprehensive assessment of the d-Hop Clustering Algorithm's suitability for dynamic and large-scale MANET environments.

2 Background and Related Work

2.1 Background

MANETs are decentralized, infrastructure-less systems in which mobile nodes communicate through multi-hop wireless links without relying on fixed base stations or routers. Each node acts both as a host and a router, forwarding packets for others to maintain end-to-end connectivity [3]. Due to node mobility, wireless interference, and limited energy resources, MANETs experience frequent topology changes that make routing, scalability, and energy management particularly challenging [4].

Clustering has emerged as an effective mechanism to address these challenges by partitioning the network into manageable groups of nodes, known as clusters [5, 6]. Each cluster is coordinated by a Cluster Head (CH), which is responsible for intra-cluster communication—the exchange of data and control messages *within* the cluster among its member nodes. Inter-cluster communication, on the other hand, refers to the communication *between* different clusters, typically facilitated by gateway nodes that act as bridges between cluster heads. Clustering enhances scalability, reduces routing overhead, supports efficient resource allocation, and improves overall energy efficiency by organizing communication hierarchically [7]. However, maintaining stable clusters under mobility and varying network densities remains an open research problem.

To overcome these challenges, various clustering algorithms have been proposed, including Lowest-ID, Highest-Degree, and Weighted Clustering Algorithm (WCA) methods. While these approaches improve organization and connectivity, they often struggle with scalability as node density increases [11]. To address these limitations, multi-hop clustering strategies have gained attention for enhancing communication efficiency, network longevity, and load balancing, often combined with optimization, machine learning, and trust-based mechanisms to further improve energy efficiency and stability [12, 13, 14].

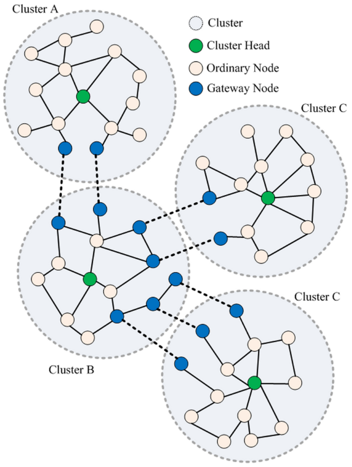

Figure 1 . Clustered Network Topology with Gateways [1].

The d-Hop Clustering Algorithm represents a significant advancement in MANET clustering. Unlike 1-hop clustering, where clusters are formed based on immediate neighbors, d-hop clustering allows nodes within d hops to form a cluster, with one node elected as the cluster head [2]. This approach balances cluster stability and scalability, making it suitable for dynamic and large-scale MANET environments.

The reviewed studies highlight continuous efforts to enhance the stability and scalability of d-hop clustering through mobility-awareness and adaptive cluster sizing.

2.2 Related Work

1. **Max-Min D-Cluster (foundational formal model):** Introduced by Amis et al. at INFOCOM 2000, this algorithm is a widely cited, efficient heuristic for forming d-hop clusters in distributed networks. It establishes a formal framework for clusterhead election, ensuring all nodes are within d hops of a clusterhead while balancing clusterhead selection. The approach serves as a baseline for both theoretical extensions and practical comparisons in d-hop clustering research [2].
2. **Mobility-aware metrics and their adoption:** Basu et al. introduced a relative mobility metric derived from successive received signal strength (RSS) measurements between neighboring nodes. This metric quantifies pairwise mobility and was later integrated into mobility-based d-hop clustering algorithms. The core principle is to prioritize nodes with lower relative mobility as clusterheads, enhancing overall cluster stability. This mobility-centric approach became a foundational element in subsequent mobility-sensitive clustering designs and analyses [15].
3. **MobDHop: adaptive d-hop clustering based on mobility:** Er and Seah introduced MobDHop, an algorithm that integrates d-hop clustering with Basu's mobility metric to create variable-diameter clusters. By adapting cluster formation to local mobility conditions, MobDHop enhances stability and minimizes reconfiguration overhead. This work highlights the potential of dynamically adjusting the d-hop diameter parameter, moving beyond static configurations. MobDHop stands as a key example of mobility-aware multi-hop clustering in MANET research [16].
4. **Performance studies and tradeoffs:** Subsequent research (e.g., Er 2006) examines the tradeoffs associated with increasing d: while a larger d reduces the number

of clusterheads and simplifies inter-cluster routing, it can also increase intra-cluster control overhead and make clusters more vulnerable to topological changes. Empirical and simulation-based evaluations—such as those of MobDHop and related algorithms—assess the balance between stability improvements and overhead costs. These studies also compare mobility-aware d-hop schemes with traditional 1-hop clustering approaches, providing critical insights for selecting optimal values of d and clusterhead election strategies based on mobility and traffic patterns [17].

5. **Recent distributed multi-hop methods:** Recent advancements, such as DC2HC (Distributed 2-Hop Clustering), revisit multi-hop intra-clustering to address modern challenges. These approaches prioritize fully distributed operation, reduced maintenance overhead, and seamless integration with connectivity and routing decisions. They demonstrate the enduring relevance of the d-hop concept, which has been refined to meet contemporary demands—such as scalability, reliability, and energy efficiency—in both MANET and VANET environments [18].

3 Main Contributions

3.1 Research Design and Methodology

This study employs a quantitative, simulation-based research methodology to evaluate the performance of the Max-Min d-Cluster algorithm in MANETs. The research is structured as an implementation and performance evaluation study. We are implementing a well-established, non-proprietary heuristic within a modern, standardized simulation environment to analyze its performance under a comprehensive set of conditions. This is an original implementation, but it replicates the logic proposed by Amis et al. (2000).

3.1.1 Measurement Tools

The primary tools for this study are OMNeT++ version 6.2.0 and the INET framework version 4.5.4.

- **OMNeT++** serves as the core discrete-event simulation engine, managing the simulation clock, message passing, and event scheduling.
- **INET** provides the extensive library of pre-built, validated models necessary for realistic MANETs, including network hosts, mobility models, various radio (PHY) and MAC protocols (like IEEE 802.11), and IP-level routing protocols (like AODV). Our Max-Min algorithm will be implemented as a new agent module that interacts with these INET components.

3.1.2 Experimental Procedure

The overall experimental procedure is structured into five distinct phases:

- **Implementation:** The Max-Min *d*-Cluster algorithm is implemented as a new simple module in INET, based on the logic described by Amis et al. .
- **Validation (Pilot Study):** The implementation's correctness is verified by replicating the paper's illustrative example.
- **Scenario Definition:** A comprehensive set of simulation campaigns is defined in `.ini` configuration files by varying the independent variables (e.g., node count, mobility model, `d` parameter).

- **Simulation Execution:** Each scenario is run multiple times with different random number seeds to ensure statistical validity.
- **Data Collection & Analysis:** Dependent variables (metrics) are recorded using OMNeT++'s signal and scalar mechanisms. The results are then processed to derive averages and 95% confidence intervals, as required by the project.

3.1.3 Pilot Study and Implementation Validation

To validate the correctness of our implementation, a pilot study was conducted by replicating the 25-node, $d = 3$ hop scenario described in the original Max-Min formulation. The simulation environment was configured with 25 stationary nodes placed at coordinates mimicking the topology shown in Figure 2. The simulation logs were captured and compared directly against the reference results provided by Amis et al.

The pilot study confirmed the correctness of the implementation. The algorithm successfully converged after $2d$ rounds, and the elected Cluster Heads matched the reference exactly. This validates that the floodmax propagation, floodmin revocation, and the three-rule election logic are functioning correctly within the OMNeT++ environment. Notably, the selected Cluster Heads for non-head nodes also aligned with those proposed in the original paper, even though the visual representation in the OMNeT++ simulation does not explicitly highlight them.

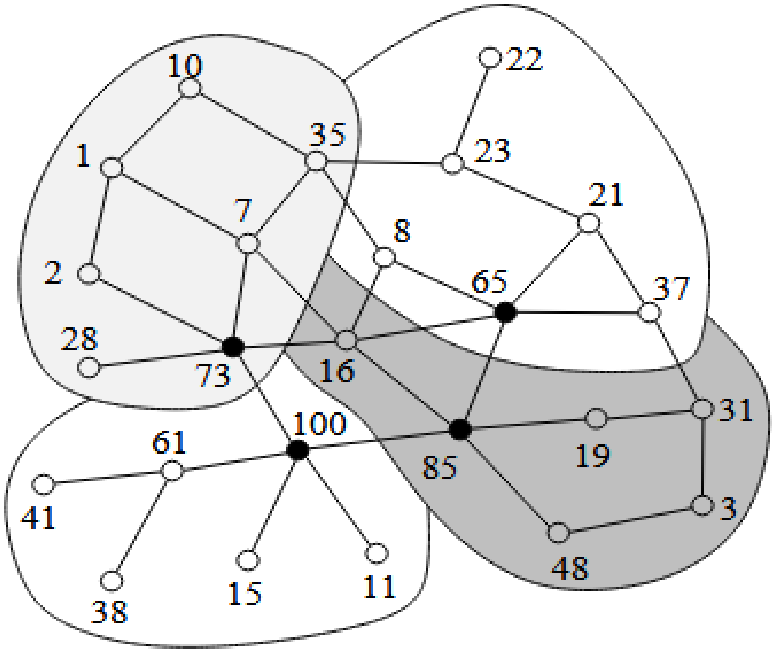

Figure 2 . 3-cluster Formation in a Network of 25 Nodes [2].

The simulation logs demonstrated a precise match with the reference table from the paper: nodes 65, 73, 85, and 100 were correctly elected as Cluster Heads.

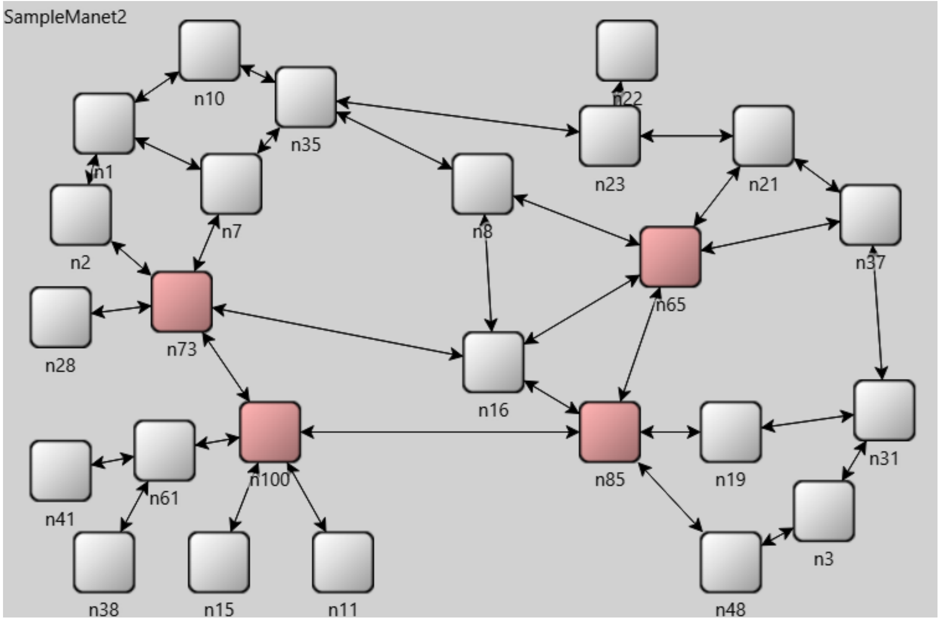

Figure 3 . Replicated 3-cluster topology in OMNeT++ simulation.

Node	10	1	2	7	35	8	23	22	21	65	37	31	19	85	16	100	73	28	41	61	11	48	3	15	38
Paper	73	73	73	73	73	65	65	65	65	65	65	85	100	85	100	100	73	100	100	100	100	100	100	100	100
Simulation	73	73	73	73	73	65	65	65	65	65	65	85	100	85	100	100	73	100	100	100	100	100	100	100	100

Table 1. Max-Min Cluster Head Validation: Reference vs. OMNeT++ Simulation.

3.1.4 Study Variables

The variables for this study are broken into two categories: independent (what we change) and dependent (what we measure).

These are the parameters we will systematically vary in the `.ini` files to observe their effect on the network.

- **Independent Variables:** Mobility models, network size (number of nodes), routing protocols (AODV, DSDV), MAC layer protocols, and the *d* parameter of the algorithm (e.g., 2, 3).
- **Dependent Variables:**
  - Cluster Quality (Number of Clusters, Cluster Head Lifetime),
  - Overhead (Control Messages),
  - Communication Efficiency (Throughput, PDR),
  - Energy (Network Lifetime).

3.1.5 Sampling and Statistical Analysis

Sampling is performed by repeating each simulation scenario multiple times (e.g., 30-50 runs) using different, independent random number seeds. This method ensures that the results are statistically significant and allows for the calculation of 95% confidence intervals, as required by the project.

3.2 Max-Min d-Cluster Algorithm Design and Implementation

The algorithm implemented in this project is the Max-Min d-Cluster heuristic proposed by Amis et al. (2020). This algorithm is designed to asynchronously form d-hop clusters, where every node is guaranteed to be at most d hops from a cluster head. The heuristic's goals include stability (favoring re-election of existing cluster heads) and fairness (distributing the cluster head load).

The algorithm's core logic is divided into three primary stages and runs for a total of $2d$ communication rounds. Each node maintains two arrays of size $2d$: `WINNER` and `SENDER`.

1. **Stage 1: Floodmax (d rounds)**
   - Initially, each node sets its `WINNER` to its own node ID.
   - For d rounds, each node broadcasts its current `WINNER` value to its 1-hop neighbors.
   - After receiving messages from its neighbors in a round, the node selects the largest value among its own `WINNER` and all received `WINNER`s as its new `WINNER`.
   - The `WINNER` for each round is logged in the `WINNER` array.
2. **Stage 2: Floodmin (d rounds)**
   - This stage immediately follows Floodmax and also lasts for d rounds.
   - The process is identical to Floodmax, but the node chooses the smallest value as its new `WINNER`.
   - The `WINNER` for each of these rounds is also logged.
3. **Stage 3: Cluster Head Selection (Rules)**
   - After $2d$ rounds are complete, each node evaluates its `WINNER` log to determine its cluster head. The decision is made by applying the following rules in order:

     (a) **Rule 1:** A node checks if its own ID appears in the log for the second d rounds (the Floodmin stage). If it does, the node declares itself a cluster head.

     (b) **Rule 2:** If Rule 1 is false, the node identifies ‘‘node pairs’’---any node ID that appears at least once in both the Floodmax log and the Floodmin log. The node then selects the minimum node ID from this set of pairs as its cluster head.

     (c) **Rule 3:** If Rule 1 is false and no node pairs exist (Rule 2 fails), the node selects the `WINNER` from the final round of the Floodmax stage (i.e., the value at round d) as its cluster head.

The Max-Min d-Cluster heuristic is implemented within a custom OMNeT++ simple module, MaxMinNode, which extends cSimpleModule. The core algorithmic rules follow the theoretical design described in Section 3.1.

4 Results and Discussion

4.1 Methodology

The methodology for this study is designed to ensure that the evaluation of the Max–Min d-Cluster Algorithm can be independently reproduced, verified, and critically examined. To achieve this, the research is conducted within a controlled and well-defined simulation environment, and each step—ranging from data generation to statistical analysis—is documented with precision.

The experiments are carried out entirely in OMNeT++ 6.2.0 using the INET 4.5.4 framework, which together provide the computational infrastructure, mobility models, wireless channel models, and protocol stacks required for realistic MANET simulations. The Max–Min algorithm is implemented as a custom simple module that integrates directly into INET's node architecture. This controlled environment ensures that all experimental conditions, parameters, and events can be reproduced faithfully by other researchers.

4.1.1 Materials and Simulation Environment

The "materials" of this study consist of:

- The OMNeT++ discrete-event simulator
- The INET framework (mobility, PHY/MAC, routing modules)
- The custom MaxMinNode C++ implementation
- Simulation configuration files `.ini` that define network size, mobility behaviors, protocol settings, and random number seeds

4.1.2 Data Generation and Sampling

Data is generated through simulation campaigns. Each scenario defined in the `.ini` files is executed multiple times with different random seeds. These seeds affect node placement, mobility patterns, wireless channel randomness, and message timing.

Repeating scenarios under randomized conditions provides a diverse dataset that captures the stochastic nature of MANETs and guards against biased single-run outcomes.

4.1.3 Statistical Treatment

To ensure that the results reflect typical system behavior rather than noise introduced by random mobility or wireless variability, all metrics are aggregated across multiple independent runs. For each metric, the study computes:

- The sample mean
- The sample variance
- The 95% confidence interval

4.1.4 Reproducibility

Every scenario can be fully reconstructed by:

1. Using the same OMNeT++ and INET versions,
2. Applying the same `.ini` configuration files,
3. Using the same initial seed values (if replication is desired instead of randomization),
4. Compiling and running the same MaxMinNode module.

References

[1] I. Ullah, T. Hussain, A. Khan, I. Ali, F. Ali, and C. Choi, ‘‘Retracted article: Analyzing the impacts of node density and speed on routing protocol performance in firefighting applications,’’ *Fire Ecology*, vol. 19, no. 1, p. 62, Oct 2023. [Online]. Available: https://doi.org/10.1186/s42408-023-00220-4

[2] A. Amis, R. Prakash, T. Vuong, and D. Huynh, ‘‘Max-min d-cluster formation in wireless ad hoc networks,’’ in *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064)*, vol. 1, 2000, pp. 32--41 vol.1.

[3] I. Chlamtac, M. Conti, and J. J.-N. Liu, ‘‘Mobile ad hoc networking: imperatives and challenges,’’ *Ad Hoc Networks*, vol. 1, no. 1, pp. 13--64, 2003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1570870503000131

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

[5] T. Rahman, I. Ullah, A. Rehman, and R. Naqvi, ‘‘Clustering schemes in manets: Performance evaluation, open challenges, and proposed solutions,’’ *IEEE Access*, vol. PP, pp. 1--1, 01 2020.

[6] N. Khatoon, P. Pranav, S. Roy, and Amritanjali, ‘‘Fq-mec: Fuzzy-based q-learning approach for mobility-aware energy-efficient clustering in manet,’’ *Wireless Communications and Mobile Computing*, vol. 2021, no. 1, p. 8874632, 2021. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1155/2021/8874632

[7] G. Abdulsaheb, O. Khalaf, N. Sulaiman, H. Zmezm, and H. Zmezm, ‘‘Improving ad hoc network performance by using an efficient cluster based routing algorithm,’’ *Indian Journal of Science and Technology*, vol. 8, 12 2015.

[8] I. Shayeb, A. Hamzah, and A. Nassuora, ‘‘A survey of clustering schemes for mobile ad-hoc network (manet),’’ vol. 20, pp. 135--151, 07 2011.

[9] S. Pal and S. P. Singh, ‘‘Survey on mobility based clustering algorithms in manet,’’ *International Journal of Engineering Research Technology (IJERT)*, vol. 1, no. 10, Dec 2012. [Online]. Available: https://www.ijert.org/research/survey-on-mobility-based-clustering-algorithms-in-manet-IJERTV1IS10201.pdf

[10] S. K. Rath, ‘‘A survey on one-hop clustering algorithms in mobile ad hoc networks,’’ *Journal of Network and Systems Management*, 2009.

[11] M. Chatterjee, S. K. Das, and D. Turgut, ‘‘Wca: A weighted clustering algorithm for mobile ad hoc networks,’’ *Cluster Computing*, vol. 5, no. 2, pp. 193--204, Apr 2002. [Online]. Available: https://doi.org/10.1023/A:1013941929408

[12] S. Zhang, X. Liu, and M. Trik, ‘‘Energy efficient multi hop clustering using artificial bee colony metaheuristic in wsn,’’ *Scientific Reports*, vol. 15, no. 1, p. 26803, Jul 2025. [Online]. Available: https://doi.org/10.1038/s41598-025-12321-y

[13] A. elBanna, B. ElHalawany, A. Bayomy, J. Huang, and K. Wu, ‘‘Machine learning-based multi-layer multi-hop transmission scheme for dense networks,’’ *IEEE Communications Letters*, vol. PP, pp. 1--1, 09 2019.

[14] N. Nathiya, R. Chinnasamy, and K. Geetha, ‘‘A hybrid optimization and machine learning based energy-efficient clustering algorithm with self-diagnosis data fault detection and prediction for wsn-iot application,’’ *Peer-to-Peer Networking and Applications*, vol. 18, 01 2025.

[15] P. Basu, N. Khan, and T. D. C. Little, ‘‘A mobility based metric for clustering in mobile ad hoc networks,’’ *Proceedings 21st International Conference on Distributed Computing Systems Workshops*, pp. 413--418, 2001. [Online]. Available: https://api.semanticscholar.org/CorpusID:1395019

[16] I. Er and W. Seah, ‘‘Mobility-based d-hop clustering algorithm for mobile ad hoc networks,’’ in *2004 IEEE Wireless Communications and Networking Conference (IEEE Cat. No.04TH8733)*, vol. 4, 2004, pp. 2359--2364 Vol.4.

[17] ------, ‘‘Performance analysis of mobility-based d-hop (mobdhop) clustering algorithm for mobile ad hoc networks,’’ *Computer Networks*, vol. 50, pp. 3375--3399, 12 2006.

[18] M. S. Batta, H. Mabed, Z. Aliouat, and S. Harous, ‘‘A distributed multi-hop intra-clustering approach based on neighbors two-hop connectivity for iot networks,’’ *Sensors*, vol. 21, no. 3, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/3/873