









Mitigating Malicious and Unintentional Packet Drops in Mobile Ad Hoc Networks

Arshad Ahmad Khan Mohammad¹, Poonam Verma², Kumar Babu Batta¹, Jyothi Bankapalli¹,
Mohammad Khaja Nizamuddin¹, Arif Mohammad Abdul^{1*}

¹ Department of CSE, School of Technology, GITAM Deemed to be University, Hyderabad 502329, India

² School of Computing, Graphic Era Hill University, Clement town, Dehradun, Uttarakhand 248001, India

Corresponding Author Email: aabdul@gitam.edu

Copyright: ©2025 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/ijss.150517>

ABSTRACT

Received: 5 April 2025

Revised: 10 May 2025

Accepted: 20 May 2025

Available online: 31 May 2025

Keywords:

Mobile Ad Hoc Networks, malicious activities, packet drops, vulnerable, promiscuous mode

Mobile Ad Hoc Networks' adaptability, flexibility, and autonomous characteristics make them suitable for critical applications like healthcare, military, and disaster recovery with cost-effective and time-effective deployment. However, these characteristics make them vulnerable to packet operation at the network layer, as packets get dropped due to malicious activities and resource constraints, i.e., unintentional packet drops. Packet drops directly negatively impact network performance regarding throughput, delay, and wastage of resources. Existing solutions either focus on mitigating malicious or unintentional packet drops, but fail to address both simultaneously. The work mitigates unintentional packet drops by dynamic multi-metric routing that adapts to the dynamic conditions of the network by computing the route by Current Residual Energy (CR) and Residual Buffer Space Metric (RBM). Moreover, work mitigates malicious packet drops by authenticated key agreements and cryptographic decisions. Performance results indicate that the proposed work significantly improves packet delivery and energy efficiency and reduces overhead compared to existing mechanisms.

1. INTRODUCTION

Mobile Ad Hoc Networks (MANETs) consist of autonomous wireless mobile nodes with heterogeneous constrained resources [1]. The network's adaptability, flexibility, and autonomous characteristics make the deployment cost and time effective [2]. Thus, network deployment suits critical and sensitive healthcare, military, and disaster recovery applications [3]. Despite all the advantages, MANETs face a significant challenge in enabling reliable communication due to packet drops at the network layer [4]. This packet drop occurs due to malicious threats and unintentional activities, which is vital to address as the packet drop significantly affects the network's performance [5].

The network characteristics, peer-to-peer networking, constrained resources, and lack of a centralized coordinator make this network vulnerable to packet operation at the network layer [6]. Packets get dropped due to malicious activities and resource constraints. The network enables multi-hop communication by relying on the intermediate nodes as this network is peer-to-peer. Thus, intermediate nodes become routers to facilitate communication, so they must spend their resources, such as battery and memory, on packet operations. If the intermediate node does not have sufficient resources to facilitate communication, packets get dropped, known as unintentional packet drop.

Further packets get dropped due to the voluntary participation of nodes, potential link failures, interference, and signal attenuation. The MANETs implement the routing by

assuming that intermediate nodes in the network are cooperative and coordinate for multi-hop communication [7]. Thus, malicious nodes do not follow the routing protocol consideration and create security concerns by intentionally causing packet drops to disturb network operations. All packet drops in MANETs are divided into two types: 1). Packet drops due to malicious intent and 2). Packet drops due to constrained resources and/or system faults.

Malicious nodes causing packet drops [8] exploit the MANET considerations, such as the fact that nodes in the network need to cooperate and coordinate for communication, as this network is peer-to-peer. Malicious nodes can disrupt the routing path by disseminating fraudulent messages claiming to possess the shortest path information, so diverting all packets towards themselves. Other categories of malevolent nodes first exhibit cooperation during the route discovery phase but thereafter either selectively forward or delete data during the data forwarding phase. Furthermore, some categories of malicious nodes transmit misleading reports to the source, indicating that the data has not arrived at its destination. In all three categories of malicious actions, packets fail to reach their destination and are discarded by an intermediary node. Malicious attacks like black-hole, gray-hole, and cooperative black-hole attacks intentionally drop the packets by misrouting and disrupting network operations [9].

Because of intrinsic network characteristics, including heterogeneity, mobility, resource constraints, and dynamic topology, packet drops in MANETs might happen unintentionally. Collisions, poor transmission power, low

energy, buffer overflow, or packet TTL expiration are reasons misbehaving nodes may lose packets. The intention of the unintentional packet drops node is not to disturb network operation, despite affecting overall network performance.

Intentional misbehaving nodes drop packets deliberately as part of malicious activities. In contrast, unintentional packet-dropping nodes inadvertently drop packets due to constrained resources and/or network conditions. The primary difference is that attacks cause intentional drops, whereas unintentional drops occur because of natural network limitations.

Packet drops in the network are a significant problem, and they affect overall network operations through packet drop, retransmission of packets, creating network congestion, and wasting network resources [10]. Thus, these two types of packet drops affect the overall network performance by increasing packet loss, reducing throughput, and compromising network reliability.

In literature, various mechanisms are designed to mitigate packet drop nodes from communication paths, and they are divided into three categories: 1) Monitoring to detect packet drop nodes, 2) Providing incentives for cooperative network operation, 3) Confirming the reception of the packets sent by acknowledgment [11]. These approaches mitigate the malicious packet drop nodes from the communication path but are ineffective in unintentional packet drops. Thus, the work aims to design an extensive routing mechanism that mitigates unintentional and malicious packet drop nodes from the communication path and copes with the network dynamics.

2. EXISTING WORK

In literature, many potential steps have been taken to detect and mitigate the impact of packet drops caused by nodes in MANETs on communication paths. These steps can be categorized into three types: credit-based, reputation-based, acknowledgment-based approaches, and the Secure Knowledge Algorithm.

2.1 Credit-based approaches

By introducing a credit system or a virtual currency, credit-based systems prompt the nodes to take part in the network honestly. By forwarding packets, intermediate nodes earn credits and spend these earned credits to send their packets. One such approach is described in the Terminodes Project. This project introduces the "Nuglet Counter," where each node earns credits by forwarding the packets and spending them on sending their packets to maintain the credit balance. This mechanism helps the nodes cooperate to keep a positive credit balance for transmitting their data [11].

CAODV (Credit-based Ad hoc On-demand Distance Vector) [12] is another credit-based system; CAODV broadens the concept of credit by rewarding the nodes during the route discovery and forwarding data phases. Nodes with positive credit balances are trusted to participate in the network, while those with negative credit balances are exempted from participation in communication. However, credit-based approaches require tamper-resistant hardware and central credit management systems, which limit their scalability and practicality in large-scale networks.

The prime goal of the credit-based technique is to exclude packet dropping nodes from the communication path, as detailed below: Typically, nodes receive acknowledgment for

their packet operations. The credit mechanism operates via virtual currency. Nodes represent the purchasers or vendors in the packet transaction. Nodes necessitate credits to transmit their transmissions. However, this mechanism further improved by focusing following factors. The approach lacks scalability because it relies on a centralized electronic payment mechanism. Each node requires additional hardware for tamper resistance. There are also expenses associated with safeguarding virtual currencies or electronic payment systems.

2.2 Reputation-based approaches

The reputation-based systems based on node behavior assign a reputation score and generally detect the misconducting nodes by monitoring their actions. In this approach, the nodes use the Promiscuous mode to calculate the reputation score for the neighboring nodes by monitoring their packet forwarding behavior. Nodes with less reputation scores are deemed untrustworthy and are discarded from the network. Further, this approach uses real-time observation by the neighboring nodes of the suspected node. The counters are used to track the number of packets forwarded and received. Significant inconsistencies between the counters indicate potential misconduct, leading to the node's isolation [13, 14].

However, reputation-based approaches provide a separate solution that doesn't need any additional hardware, and their usefulness can be tampered with in environments with high packet loss due to non-malicious factors, such as collisions or low transmission power.

Every node in the network collects and maintains reputation data in a reputation-based system. A node with reputation too low is denied service. The vast majority of reputation-based systems use a promiscuous monitoring approach. Each node in a network monitors its neighbors when packets are processed. It is relatively cost to apply this system; however, it is by affected by packet loss, collision and unidirectional properties. Reputation based approach is better than credit-based approach as follows: It does not require electronic payment network or virtual bank or third parties for operating in a network. It is also not in need of tamper-resistant device on the node.

This approach is more appropriate for MANETs, as it enhances network scalability through distributed implementation.

2.3 Acknowledgment-based approaches

Acknowledgment-based methods depend on transmitting the acknowledgment packets to confirm that data packets have successfully reached their destination. Two ACK: In this approach, each node transmits an acknowledgment packet to the next node, but instead of sending the acknowledgment packet back to the node, after receiving it, it sends the packet to the node that is two hops away. AACK (Adaptive Acknowledgment): AACK detects the misbehaving nodes by combining the end-to-end acknowledgments with the two-hop acknowledgments. The route is said to be compromised, and an alternative route is taken when the acknowledgment is not received within the given time. While useful, Acknowledgment-based systems may introduce significant overhead due to the additional control messages required, particularly in high-traffic scenarios [15, 16].

Secure Knowledge Algorithm: The Secure Knowledge Algorithm assimilates promiscuous monitoring with a

comprehensive analysis of the differentiation between malicious and unintentional packet dropping nodes of packet drops.

Promiscuous Monitoring: It appears to be the same as reputation-based systems. This approach involves monitoring neighbors' packet operations. If the packet drops exceed a certain point, the algorithm will detect the cause, such as the buffer and energy status. **Reason Investigation:** The algorithm considers factors like energy depletion, buffer overflow, and TTL (Time to Live) expiration to differentiate between malicious intent and resource constraints. After detecting the malicious node, the algorithm shares this information with all network nodes to isolate the misbehaving node. Although the Secure Knowledge Algorithm provides a detailed solution by handling both types of misbehavior, it still depends on the static routing metrics, which may lead to suboptimal routing decisions in dynamic network environments.

However, the above-discussed methods mainly focus on finding and preventing intentionally misbehaving nodes while leaving unintentional misbehaving nodes to be given less attention. The secure acknowledgment algorithm uses promiscuous monitoring to tackle intentional and unintentional misbehaving nodes. In this method, every node monitors the packet operations of its neighbors. The protocol investigates the reasons for packet drops, such as energy, buffer, and Time to Live values, if a node is above the predefined threshold of packet drops. Upon inspection, if none of the mentioned factors are the cause, the node will be excluded from the other nodes' communication and is considered malicious. However, this method can cause increased network and routing overhead delays due to the limitation of the delayed calculation of packet drop reasons and minimum hop count routing metric. Thus, the work aims to design an extensive routing mechanism that mitigates malicious and unintentional packets drop nodes from the routing path and copes with the network dynamics.

2.4 Problem definition

MANETs are fundamentally vulnerable to packet drops at the network layer due to intentional or unintentional activities. Packet drops in any network are a significant problem and disturb overall network performance regarding throughput, packet loss, delay, and lifetime. Intentional packet drops activities include misrouting, not following networking protocol specifications, and disrupting network operations. Limited resources and network characteristics, such as the depletion of node battery, buffer overflow, and transmission power limitations, cause unintentional packet drops. Existing methodologies mitigate the malicious packet drop nodes from the communication path but are ineffective in unintentional packet drops due to resource constraints.

Moreover, the static nature of routing metrics in existing methodologies, disregarding the dynamic conditions of MANETs, results in selecting routes vulnerable to packet drops. Thus, the work aims to design an extensive routing protocol that mitigates malicious and unintentional packet-drop nodes from the communication path and copes with the network dynamics. The following contributions accomplish the objective of the work.

2.4.1 Develop a comprehensive misbehavior detection mechanism

Design a mechanism distinguishing between intentional and unintentional packet drop nodes. To swiftly detect and isolate

the malicious nodes while not affecting the functionality of resource-constrained nodes by ensuring real-time detection capabilities.

2.4.2 Incorporate dynamic node metrics into routing decisions

Design dynamic route computation metric based on the present conditions of the network node in terms of residual energy buffer, which will adjust to any changes in the node condition.

2.4.3 Enhance network reliability and performance

Improving the dependability of MANETs by reducing packet drop rates and ensuring that the data reaches its exact destination, even with the misbehaving nodes present. Optimize network performance metrics, including throughput, packet delivery ratio, and latency, by choosing routes that are both effective and resistant to the node's misbehavior.

By incorporating the above contributions, the work aims to improve MANETs' overall performance, reliability, and efficiency, making them more resilient and flexible in many operational environments.

3. EXTENSIVE ROUTING PROTOCOL TO MITIGATE MALICIOUS AND UNINTENTIONAL PACKET DROP NODES

The work designs an extensive routing protocol that mitigates intentional and unintentional packet drop nodes from communication in MANET. The developed protocol is dynamic and resilient to MANET's characteristics. The protocol route-finding metric is based on the network's residual status of node resources to mitigate unintentional packet drops nodes. During the packet forwarding stage, promiscuous monitoring and cryptographic decisions. The work achieves the goal by following three key contributions.

(1) **Residual Buffer Space Metric (RBM):** Determining intermediate node's buffer residual status regarding packet operations to mitigate packets drop due to buffer-overflow.

(2) **Current Residual Condition (CR):** Determining intermediate node's energy residual status regarding packet operations to minimize packet drop due to energy depletion.

(3) **Authenticated key Agreement and Cryptographic Decision** to mitigate intentional packet drops caused by malicious activities.

3.1 Residual Buffer Space Metric (RBM) to mitigate packets drop due to buffer overflow

MANET is a peer-to-peer network that enables multi-hop communication depending on intermediate nodes. Thus, intermediate nodes need to forward the packets of communicating nodes; to do this, they must spend their energy, buffer space, and other resources. Excessive traffic towards intermediate nodes causes a buffer overflow that leads to packet drops and degraded network performance. The proposed work designs a Residual Buffer Space Metric (RBM) concept that dynamically computes and verifies the buffer status regarding packet operation during communication in MANETs to mitigate packet drops caused by buffer overflow. The metric RBM used to decide whether the node should participate in communication, as it dynamically verifies the buffer residual status regarding packet operations. The decision of the node participation based on the residual buffer

status prevents congestion and packet drops. The mechanism to compute the Residual Buffer Space Metric (RBM) is based on queue size at the node buffer and is explained as follows.

The queue size at the node buffer determines the real-time buffer utilization, and it is calculated as a function of the average packet arrival and transmission rates. Each node in MANETs consists of a finite buffer space with three major parts, i.e., an input queue, an output queue, and a processing module. Consider a node $(n)_i$ of MANETs, which act as a relay node for communication between communicating entities. The node $(n)_i$ has Input Queue $(Q)_{in}$: Where packets arrive, Processing Module (PPP) : Where packets are processed, and Output Queue $(Q)_{in}$: Where processed packets are placed for transmission. The queue size of the node $(n)_i$ is computed as follows.

(1) Average Packet Arrival Rate (R) towards a relay Node $(n)_i$

$$R = \alpha(R)_c + \beta(R)_p \quad (1)$$

where, $(R)_c$ and $(R)_p$ are average packets received during current and previous time intervals respectively. And α and β are the weighted constants $(0 \leq \alpha, \beta \leq 1)$.

(2) Average Packet Transmission Rate (T) from the Node $(n)_i$

$$T = \alpha(T)_c + \beta(T)_p \quad (2)$$

where, $(T)_c$ and $(T)_p$ are average packets sent during current and previous time intervals respectively. The constants α and β in the RBM mechanism modulate the system's response dynamics: α adjusts the weight of current traffic metrics, facilitating rapid queue length recalibration, while β integrates historical data, dampening transient noise. The high value of α improves real-time sensitivity, and the high value of β stabilizes against short-term fluctuations. Dynamic adjustment of the values α and β based on residual network conditions is essential for optimal real-time buffer utilization.

(3) Queue Length Q_p the node $(n)_i$

The queue length is the cumulative effect of the difference between packet arrivals and transmissions over time. Eq. (3) determines the queue length over a specific period; the equation is:

$$Q_p = Q_{p,prev} + (R - T) \times \Delta t \quad (3)$$

where, $Q_{p,prev}$ is the previous queue length, and Δt is the time interval over which the rates are measured.

(4) Queue size in kilobytes Q_s

$$Q_s = Q_p \times \text{kbytes} \quad (4)$$

If $Q_s = B_i$ (buffer size) indicates that no space in node buffer i.e., it is full, and new packets will be dropped due to lack of buffer size.

(5) Packet Waiting Time (W_t)

$$W_t = \frac{Q_p}{R} \quad (5)$$

W_t is meaningful if $R < T$; otherwise, the system would be unstable, leading to an infinite waiting time as the arrival rate exceeds the transmission rate. If the waiting time of a packet

in the buffer exceeds its TTL, the packet should be dropped to prevent it from becoming stale or causing unnecessary congestion. i.e., $W_t \geq TTL$.

(6) Residual Buffer Space Metric (RBM)

The residual Buffer Space Metric (RBM) is defined as the ratio of the residual buffer space (RB) to the total buffer capacity (B_i) and it is calculated as follows:

$$RBM = \frac{RB}{B_i} \quad (6)$$

where, RB is the residual buffer space calculated as $RB = B_i - Q_s$ and B_i is the total buffer capacity.

The RBM values of each relay node are computed and updated periodically to ensure that the values are up-to-date and reflect current network conditions. This information is shared with neighbor nodes by exchanging control messages. Further, based on the RBM threshold value, the threshold-based approach temporarily decides whether a node can participate in routing. The threshold establishes the minimum acceptable RBM value that a node must have to participate in routing. Setting an appropriate RBM threshold value is vital to mitigate packet drops due to buffer overflow.

(7) Residual Buffer Space Metric Threshold (RBMT)

RBMT stands for the lowest residual buffer capacity, and a node has to be involved in routing to prevent buffer overflow and consequent packet losses. The threshold is not static because network traffic and buffer occupancy fluctuate over time. The threshold is not static because network traffic and buffer occupancy fluctuate over time.

Work determines the optimal $RBMT$ that dynamically adapts to network conditions based on traffic load, buffer utilization, and dynamic adaption factor. Thus, $RBMT$ ensures efficient packet forwarding in MANETs using the factors influencing RMB value.

(i) Network Traffic Load (L_t) : A higher traffic load necessitates a higher threshold to ensure sufficient buffer availability. The Eq. (7) quantifies the traffic load.

$$L_t = \frac{\sum_{i=1}^N R_i}{N} \quad (7)$$

where, R_i is the packet arrival rate at node i , and N is the total number of nodes.

(ii) Buffer Utilization (U_b) : Buffer utilization indicates how much of the buffer is currently used. It is calculated using the Eq. (8).

$$U_b = \frac{Q_s}{B_i} \quad (8)$$

where, Q_s is the current queue size and B_i is the total buffer capacity of the relay node.

(iii) Dynamic Adaptation Factor (D_a) : It adjusts the threshold based on real-time network conditions and is computed based on recent buffer utilization and traffic load changes. Consider U_b^t and U_b^{t-1} be the buffer utilizations at time intervals t and $t-1$, respectively, and L_t^t and L_t^{t-1} be the traffic loads at the node buffer at time intervals t and $t-1$, respectively. Then, recent changes in buffer utilization and traffic load are computed using Eqs. (8) and (9), respectively.

$$\Delta U_b = U_b^t - U_b^{t-1} \quad (9)$$

$$\Delta L_t = L_t^t - L_t^{t-1} \quad (10)$$

Algorithm 1: Residual Buffer Space Metric (RBM) Implementation Algorithm

1. Initialize

- $R_p = 0, T_p = 0, Q_{p,prev} = 0$

2. Measure current arrival and transmission rates

- R_c , and T_c

3. Calculate Average Packet Arrival Rate (R) and Transmission Rate (T)

- $R = \alpha(R)_c + \beta(R)_p$
- $T = \alpha(T)_c + \beta(T)_p$

4. Update Queue Length (Qp), and Convert the updated queue length to kilobytes

- $Q_p = Q_{p,prev} + (R - T) \times \Delta t$
- $Q_s = Q_p \times \text{kbytes}$

5. Calculate Packet Waiting Time (Wt) and Check TTL

- $W_t = \frac{Q_p}{R}$
- If $W_t \geq TTL$; drop packets that exceed TTL

6. Calculate Residual Buffer Space (RB) and RBM

- $B = B_i - Q_s$
- $RBM = \frac{RB}{B_i}$

7. Assess Network State

- network – wide average traffic load
 - $L_t = \frac{\sum_{i=1}^N R_i}{N}$
- Buffer Current utilization
 - $U_b = \frac{Q_s}{B_i}$

8. Compute dynamic adaptation factor (Da)

- $D_a = \gamma(\Delta U_b + \Delta L_t)$

9. Update RBM threshold (RBM_{th}):

- $RBM_{th} = (1 - \frac{L_t}{L_{max}}) \times (1 - U_b) \times QoS + D_a$

10. Make Routing Decision:

- If $RBM \geq RBM_{th}$; Allow the node to participate in forwarding packets
 - Else: exclude node from routing temporarily

11. Optimize control message overhead

- If there is a significant change in buffer status : initiate update of RBM values
- limit updates to neighboring nodes

12. Update previous values for next iteration

- $R_p = R_c, T_p = T_c, Q_{p,prev} = Q_p$

13. Repeat periodically to adapt to real-time network conditions

maximum allowable traffic load, and U_b is the buffer utilization, QoS is the QoS parameter, and D_a is the dynamic adaptation factor. Desired Quality of Service (QoS) Based on application and sensitivity in MANET, QoS requirements vary. The applications with stricter QoS requirements encyst higher RBMT values to minimize packet loss. Quantifying QoS Requirements: The QoS parameter (QoS) is a dimensionless value that ranges from 0 to 1: $QoS = 0$: Represents applications with low QoS requirements, where delays and packet drops are more tolerable. $QoS = 1$: Represents applications with high QoS requirements, where minimal delays and packet drops are critical.

Control Message Overhead: Work updates the RBM values only when significant buffer changes occur in the network to minimize the overhead of control messages. Further, it limits RBM exchanges to neighboring nodes instead of broadcasting throughout the network.

3.2 Residual Buffer Space Metric (RBM) algorithm implementation

The RBM Implementation algorithm dynamically computes the RBM value of the nodes and, based on this, decides the node participation in routing by continuously monitoring packet arrival and transmission rates. The RBM value provides the buffer utilization of the nodes based on its real-time packet arrival transmission rates. Thus, it assures that a node with sufficient buffer becomes a relay node for multi-hop communication to mitigate packet drops in MANETs due to buffer overflow. Algorithm 1 shows the RBM Implementation algorithm, i.e., steps involved in calculating the RBM value of each node and making routing decisions based on real-time buffer conditions. Thus, RBM Implementation algorithm is helpful to enhance network performance, prevent congestion, and reduce packet loss.

4. CURRENT RESIDUAL CONDITION (CR) OF THE RELAY NODES BASED ON ENERGY TO MITIGATE PACKETS DROP DUE TO ENERGY DEPLETION

MANET nodes have batteries of finite power, and to charge or replace the batteries during the mission is extremely difficult. Whenever a node becomes the relay node for communication, it has to spend its energy on packet operations such as receiving, processing, and transmitting the packets of other nodes. Thus, energy constraint often leads to unintentional packet drops from intermediate nodes, affecting the network's performance. An effective routing mechanism is required to address energy constraints and ensure efficient communication.

In the literature, multiple energy-based routing mechanism have been developed and categorized as routing protocols to find 1) Reliable link route, 2) Energy efficient route, 3) Higher energy nodes route [17]. Existing work analyzed these protocols and concluded that they all make the routing decision based on a greater energy having nodes path, least energy spending path, or lowest retransmission count. Due to this, one of the relay nodes experiences heavy traffic and becomes bottlenecks [18], leading to packet loss and expiring shortly due to energy drain. Thus, we dynamically compute the residual energy status of the node regarding packet operation called Current Residual Condition (CR). The objective is to minimize unintentional packet drop due to

The dynamic adaptation factor (D_a) is computed by the Eq. (11).

$$D_a = \gamma(\Delta U_b + \Delta L_t) \quad (11)$$

where, γ is a scaling constant that controls the sensitivity of the adaptation. The RBM threshold is dynamically determined using the below equation.

$$RBM_{th} = \left(1 - \frac{L_t}{L_{max}}\right) \times (1 - U_b) \times QoS + D_a \quad (12)$$

where, L_t is the current network traffic load, L_{max} is the

energy constraints. Further, we use this metric to compute the route to ensure that nodes do not become bottlenecks due to energy depletion.

Table 1. Key parameters and their roles in computing current residual conditions

Parameter	Description
E	The total energy available in the node's battery is measured in joules.
B	The node's Buffer capacity indicates the maximum number of packets it can store.
P_i	The i -th packet requires processing by the node.
$E(P_i)$	The energy required to process the packet P_i , including: receiving, transmitting, and processing.
E_r	Energy is required to receive the packet.
E_p	Energy is required to process the packet.
E_t	Energy is required to transmit the packet.
E_{resi}	The remaining energy in the node after processing the packet P_i .
n	Total number of packets the node needs to process.

4.1 Current Residual Condition (CR)

To determine the existing residual condition (CR) of an intermediate node (n_i) in terms of energy, consider that the nodes are equipped with a battery with energy capacity E joules and needs to process the packet P_i through it. The parameters required to compute the Current Residual Condition (CR) node are shown in Table 1. The procedure for computing the Current Residual Condition is explained as follows.

(i) Energy Consumption for the packet (P_i) processing by node n_i is computed by the Eq. (13).

$$E(P_i) = (E_r + E_p + E_t) \quad (13)$$

(ii) Remaining-Energy (E_{resi}) of the node n_i after Processing the packet (P_i) is computed by the Eq. (14).

$$E_{resi} = E - E(P_i) \quad (14)$$

(iii) Maximize the number of packets processed by the node n_i without exceeding the node's energy capacity, computed using the following equation.

$$\text{Maximize } \sum_{i \in T} P_i \text{ subject to } \sum_{i \in T} E(P_i) \leq E \quad (15)$$

(iv) Dynamic Programming Approach [19] to maximize the packet processing capacity of the node regarding energy is computed by constructing a 2D array $K[i, E]$ to represent the extreme number of packets that can be processed, given the first i packets and a specific energy level E .

(1) Initialization: For zero packets or zero energy, the processed packets count is zero:

$$K[0, E(P_i)] = 0 \forall 0 \leq E(P_i) \leq E \quad (16)$$

(2) Filling the Table: For each packet P_i and each energy level $E(P_i)$.

$$K[i, E(P_i)] = \max(K[i-1, E(P_i)], K_i + K[i-1, E - E(P_i)]) \quad (17)$$

where,

$K[i-1, E(P_i)]$ represents the case where the current packet is not processed.

$K_i + K[i-1, E - E(P_i)]$ represents the case where the current packet is processed, and the remaining energy adjusts to the previous maximum.

(v) Tracking Packet Processing: With the help of an auxiliary Boolean array, $Keep[i, E(P_i)]$ to track which packets are processed:

$$\begin{aligned} Keep[i, E(P_i)] &= 1 \text{ if packet } P_i \text{ is processed.} \\ Keep[i, E(P_i)] &= 0 \text{ otherwise.} \end{aligned}$$

(vi) Deriving the Current Residual (CR) Conditions

CR reflects the extreme number of packets processed without exceeding the node's energy capacity. Compared to dynamically computed thresholds CR_{max} , CR_{min} . CR thresholds define energy levels, determining active, backup, or excluded node roles to prevent depletion.

CR thresholds (CR_{min} , CR_{max}) represent energy capacity levels guiding node participation: Nodes with $CR \geq CR_{max}$ have sufficient energy for active routing. Nodes with $CR_{min} < CR < CR_{max}$ serve as backups to balance load. Nodes with $CR < CR_{min}$ are excluded to prevent energy depletion and network failure.

$$\begin{cases} CR \geq CR_{max} & \text{Node participates in routing} \\ CR_{min} < CR < CR_{max} & \text{Node acts as a backup node} \\ CR < CR_{min} & \text{Node excluded from routing} \end{cases} \quad (18)$$

Algorithm 2: Compute Current Residual Condition (CR)

1. Initialize the dynamic programming table $K[i, E]$ with base conditions:
for $E(P_i) = 0$ to E :
 $K[0, E(P_i)] = 0$
2. Iterate through each packet i from 1 to n :
for each energy level $E(P_i)$ from 0 to E :
if $(E(P_i) \leq E)$ and $(K[i-1, E(P_i)] < K[i-1, E - E(P_i)] + P_i)$:
 $K[i, E(P_i)] = K[i-1, E - E(P_i)] + P_i$
 $Keep[i, E(P_i)] = 1$
else:
 $K[i, E(P_i)] = K[i-1, E(P_i)]$
 $Keep[i, E(P_i)] = 0$
3. Derive the CR based on the maximum number of packets that can be processed within the node's energy limit:
 $CR = \max(K[n, E(P_i)])$
4. Compare the CR against threshold values to decide the node's role in routing:
if $CR \geq CR_{max}$:
Node participates in routing
else if $CR_{min} < CR < CR_{max}$:
Node acts as a backup node
else:
Node excluded from routing

These thresholds dynamically adjust based on network density and traffic to balance energy use. This ensures efficient load distribution, prolongs network lifetime, and maintains routing reliability.

It is essential to dynamically monitor each node's energy levels to minimize energy depletion and ensure reliable network performance in MANET. The metric Current

Residual Condition (*CR*) determines the packet processing competence of the node based on its residual energy and helps to determine the node involvement in routing. Algorithm 2 shows the steps to calculate the *CR* value using a dynamic programming approach to maximize the number of packets processed within the node's energy limit and determine the node's role in routing based on dynamically computed threshold values. Dynamic adjustment prevents node failures and optimizes overall network performance.

(vii) Dynamic Threshold Computation

Thresholds CR_{max} CR_{min} are dynamically computed based on network conditions such as Node Density, Traffic Load, and Energy Depletion Rates. Higher node density (ρ) typically requires a lower threshold value, as more devices support more support for packet operation. High traffic load (λ) requires a higher threshold value, as heavy traffic in the network. Rapid energy depletion (η) requires more conservative thresholds, as nodes lose energy quickly. Dynamic threshold values are computed as follows.

Maximum Threshold CR_{max}

$$CR_{max} = \alpha' \times \frac{1}{N} \sum_{i=1}^N E(p_i) \quad (19)$$

where, α' is a scaling factor defined as:

$$\alpha' = \frac{C_\rho \cdot \rho + C_\lambda \cdot \lambda + C_\eta \cdot \eta}{C_\rho + C_\lambda + C_\eta} \quad (20)$$

Minimum Threshold CR_{min}

$$CR_{min} = \alpha' \times \frac{1}{N} \sum_{i=1}^M E_{resi} \quad (21)$$

where, $C_\rho, C_\lambda, C_\eta$ are weighted constants for node density, traffic load, and energy depletion rate. ρ, λ, η represent node density, traffic load, and energy depletion rates. N is the total number of packets, and $\sum_{i=1}^N E(p_i)$ is the sum of energy required to process ' N ' packets and $\sum_{i=1}^M E_{resi}$ is the sum of residual energies of all nodes.

Algorithm 2 shows the steps to calculate the *CR* value of the nodes using a dynamic programming approach. Then, it decides the node participation in routing by comparing the computed *CR* value with dynamically computed threshold values. Thus, this algorithm ensures energy-efficient routing decisions by dynamically computing the '*CR*' of the nodes in the network and comparing it against threshold values. This approach significantly reduces packet drops and enhances overall network performance.

5. AUTHENTICATED KEY AGREEMENT AND CRYPTOGRAPHIC DECISION TO MITIGATE INTENTIONAL PACKET DROPS CAUSED BY MALICIOUS ACTIVITIES

The work proposes a mechanism to mitigate intentional packet drops nodes based on Authenticated Key Agreement and Cryptographic decisions. Authenticated Key Agreement ensures the secure session key agreement between communicating parties. Thus, it allows only authorized nodes to participate in communication. The Cryptographic decisions

included in the proposed work are digital signature and session-based digested acknowledgment. The Cryptographic decisions mechanism provides packet integrity and confirms the packet communication, i.e., sent packets are received at the destination. These mechanisms collectively mitigate the intentional packet drops nodes and enable tamper-resistant communication.

5.1 Authenticated key agreement

The authenticated key agreement mechanism in MANETs ensures that only authorized nodes participate by securely exchanging session keys. The proposed authenticated key agreement is computationally efficient and suitable for MANET constraints compared to existing RSA and ECC-based authenticated key agreements, as it is based on chaotic maps [20]. Performance comparisons between chaotic maps in authenticated key agreement and conventional techniques such as RSA and ECC support their benefit. Chaotic map operations such as Chebyshev polynomials computation require roughly 0.021 seconds in a controlled environment (dual-core 2.33 GHz processor, 2 GB RAM), significantly less than RSA modular exponentiation at 0.093 seconds. Cutting delay, energy consumption, and buffer use reduces computational overhead and improves network performance, therefore proving a strong efficiency advantage of chaotic maps over conventional encryption techniques.

Further, the agreed session key is used to create a digested acknowledgment regarding packet reception, so malicious nodes cannot send the forgery acknowledgment to the source node regarding packet reception. Thus, digested acknowledgment is vital to enhance security and helps to mitigate the malicious packet drops nodes from the communication path in MANETs. The authenticated key arrangement between two communicating entities *A* and *B* is explained as follows.

During initialization, both communicating entities, *A* and *B*, have prior knowledge of a shared low-entropy password '*PW*'. This password is the basis for mutual authentication. Further, they established and agreed on the public parameters, such as a large prime '*P*', an elliptic curve ' $E(Fp)$ ', a generator point '*G*', and a hash function '*H*'. Each party also generates its elliptic curve key pair and standard value '*x*' for chaotic maps computation.

1. Chaotic Sequence Generation and Public Value Computation at Node *A*:

Chaotic Sequence Computation: node *A* chooses a random integer '*a*' and computes a chaotic sequence $\Psi_a(x)$ using Chebyshev polynomials, as shown in Eq. (22).

$$\Psi_a(x) = (T_a(x) \oplus H(T_a(x))) \bmod P \quad (22)$$

where, $T_a(x)$ refers to the Chebyshev polynomial of degree *a* evaluated at *x*.

Public Value Calculation: *A* then computes a public value C_A by hashing the chaotic sequence combined with the password and a session-specific salt *s*, as shown in Eq. (23).

$$C_A = H(\Psi_a(x) \parallel PW \parallel s) \bmod p \quad (23)$$

Salt (*s*) is computed using the Eq. (24).

$$s = H(PW \parallel \text{session}_{ID}) \bmod p \quad (24)$$

Message to B: A sends C_A to B

2. Chaotic Sequence Generation and Public Value Computation at Node B:

Chaotic Sequence Computation: node B chooses a random integer b and computes a chaotic sequence $\Psi_b(x)$ using Chebyshev polynomials:

$$\Psi_b(x) = \left(T_b(x) \oplus H(T_b(x)) \right) \text{mod } P \quad (25)$$

Public Value Calculation: node B then computes a public value C_B by hashing the chaotic sequence combined with the password and a session-specific salt

$$C_B = H(\Psi_b(x) \parallel \text{PW} \parallel s) \text{mod } p \quad (26)$$

Message to A: B sends C_B to A

3. Mutual Authentication and Verification

Verification by Node A:

Public Value Verification: Upon receiving C_B , A recomputes the expected value using $\Psi_b(x)$, the shared password PW, and the salt s

$$C'_B = H(\Psi_b(x) \parallel \text{PW} \parallel s) \text{mod } p \quad (27)$$

If computed C'_B matches C_B , A is assured of B's knowledge of the password.

Verification by Node B:

Public Value Verification: Upon receiving C_A , node B recomputes the expected value using $\Psi_a(x)$, the shared password PW, and the salt s .

$$C'_A = H(\Psi_a(x) \parallel \text{PW} \parallel s) \text{mod } p \quad (28)$$

If computed value C'_A matches C_A , B is assured of A's knowledge of the password.

4. Session Key Derivation

Shared Secret Calculation: Both parties now compute a shared elliptic curve point based on their chaotic sequences and public values [21].

$$P_{ab} = a \times P_a = b \times P_b \quad (29)$$

P_a and P_b are derived from $\Psi_a(x)$ and $\Psi_b(x)$, respectively, by multiplying the generator point G on the elliptic curve $E(E_p)$ with the chaotic sequence $\Psi_a(x)$ and $\Psi_b(x)$, respectively.

Session Key Derivation: The session key K_s is derived by both parties using the shared secret P_{ab} , the password PW, and the bilinear pairing.

$$K_s = \text{KDF}(P_{ab} \parallel \text{PW} \parallel \hat{e}(P_a, P_b) \parallel s) \text{mod } p \quad (30)$$

This key K_s will be identical for both parties and used to secure subsequent communications.

5.2 Cryptographic decision mechanism for misbehaving nodes in MANETs

Mobile Ad Hoc Networks (MANETs) are decentralized, self-organizing networks where nodes communicate directly. The characteristics, such as the absence of a fixed infrastructure, dynamic and unpredictable topology, and resource-constrained environment, introduce significant

vulnerabilities and can be leading to various attacks, such as packet drops, data tampering, and false reporting. These attacks directly impact network performance in terms of network reliability.

We design malicious packet drop node mitigation mechanisms by integrating cryptographic verification, session-based acknowledgment, and real-time monitoring that effectively mitigate malicious packet drops, data tampering, and false reporting attacks. The mechanism consists of 3 significant components: 1). Digital Signatures for Packet Integrity, 2). Session Key-Based Counter-Enforced Digested Acknowledgment, and 3). Promiscuous Monitoring for Anomaly Detection. Each component is critical to mitigate the threats and ensure secure and efficient communication within the network.

1. Digital Signatures for Packet Integrity

The goal is to ensure that transmitted packets are actual and haven't been changed by attackers, thus providing the authenticity and integrity of transmitted packets. The digital signature between the source node n_i and destination node n_j is explained as follows.

Source nodes say n_i which transmitting packet P_k generates the digital signature σ_{N_i} using the agreed session key k_{s_i} , and is shown in the Eq. (31).

$$\sigma_{N_i} = \text{Sign}(k_{s_i}, H(P_k)) \quad (31)$$

where, $H(P_k)$ is the digest value of the packet. The digital signature is appended to the original packet before transmission. Upon receiving the packet P_k , the destination node n_j verifies the packet's integrity by using the session key k_{s_j} .

$$\text{Verify}(k_{s_j}, H(P_k), \sigma_{N_i}) \quad (32)$$

If the verification is successful, the packet is accepted at the destination, as it is an authentic and untampered packet. Otherwise, the packet is discarded. The security of the digital signatures is grounded in the Elliptic Curve Digital Signature Algorithm (ECDSA). The signature generation and verification processes are mathematically represented as follows.

$$\sigma_{N_i} = \text{Sign}(P_r n_i, H(Pk)) \quad (33)$$

$$\text{Verify}(P_u n_i, H(Pk), \sigma_{N_i}) = \begin{cases} \text{True} & \text{if } \sigma_{N_i} \text{ is valid} \\ \text{False} & \text{if } \sigma_{N_i} \text{ is invalid} \end{cases} \quad (34)$$

The security of this process relies on the difficulty of solving the Elliptic Curve Discrete Logarithm Problem (ECDLP), which ensures that the signatures cannot be forged without the private key $P_r n_i$.

2. Session Key-Based Counter-Enforced Digested Acknowledgment

It is used to verify the reception of the transmitted packet at the destination over a specific time interval T , to prevent packet loss and tampering. The time interval T for generation acknowledgment must be dynamically optimized to balance security and network performance. This interval is calculated as:

$$T = \alpha'' \cdot \left(\frac{1}{\lambda} \cdot \mu + \frac{1}{M} \cdot S + \delta \right) \quad (35)$$

where, λ : Packet arrival rate (packets per second), μ : Maximum tolerable packet drop rate, M : Node mobility factor, representing the frequency of topological changes. S : Security sensitivity factor, reflecting the required security level, δ : Processing delay per packet. And α'' Proportionality constant, adjusted according to network conditions. Destination node n_j accumulates and computes a cumulative digest D_{n_j} of the received packets over a specific time interval T , and the digest is defined as:

$$D_{N_j} = H \left(\sum_{k=1}^n Pk \parallel Ks \parallel C_{n_j} \right) \quad (36)$$

where, Pk are the packets received during time interval T , and Ks is the session key agreed between the source n_i and destination n_j , and C_{n_j} is the acknowledgment counter. The digest D_{N_j} is computed using a collision-resistant cryptographic hash function H , which is crucial for ensuring the integrity of the acknowledgment. By including the session key Ks and acknowledgment counter C_{n_j} , the digest is bound to a specific communication session, thereby preventing replay attacks and ensuring session integrity. Then, the destination constructs the acknowledgment packet and sends it to the source; the acknowledgment packet is as follows.

$$A_{N_j} = \{D_{N_j}, C_{N_j}\} \quad (37)$$

Upon receiving the acknowledgment packet, the source node n_i calculates the expected digest D_{N_i} using the packets it transmitted as follows.

$$D_{N_i} = H \left(\sum_{k=1}^n Pk \parallel Ks \parallel C_{N_i} \right) \quad (38)$$

It compares computed D_{N_i} with received D_{N_j} , if both match, then the acknowledgment counter is consistent, and the integrity of the communication is confirmed. Discrepancies indicate potential misbehavior, prompting further investigation into node N_j .

3. Promiscuous Monitoring for Malicious Packet Drops Detection

This mechanism detects deviations in packet forwarding behavior, particularly identifying nodes that drop packets.

Monitoring Mechanism:

(i) Promiscuous Mode Activation: Each Node in the network operates in promiscuous mode, which allows one to observe the packets received and forwarded by their neighboring nodes. A monitoring node N_k records the number of packets N_j receives and forwards.

(ii) Discrepancy Calculation: The monitoring node N_k calculates the difference (ΔN_j) between the packets received by the nodes that should be forwarded and those it forwards, as follows:

$$\Delta N_j(t) = \text{ReceivedPackets to be forwarded}(N_j) - \text{ForwardedPackets}(N_j) \quad (39)$$

Threshold-Based Detection: The discrepancy ΔN_j is compared to a predefined threshold θ . If ΔN_j exceeds θ ($\Delta N_j(t) > \theta(t)$), node ΔN_j is flagged as a potentially misbehaving node. This triggers a network-wide alert, leading to the exclusion of N_j from route paths, thus preserving the integrity of the network.

$$\Theta(t) = \mu(t) + \alpha_p \cdot \Delta N_j(t) \quad (40)$$

$\mu(t)$ is the average difference between the number of packets a node should forward and the actual number it forwards during a recent time window, i.e., N packets.

$$\mu(t) = \frac{1}{N} \sum_{i=t-N+1}^t \Delta N_j(i) \quad (41)$$

$\Delta N_j(i)$ represents the packet discrepancy at observation i . This average gives a historical view of the node's behavior over the last N packets and α_p is a sensitivity factor that adjusts how much weight is given to the current discrepancy $\Delta(t)$.

The proposed detection mechanism for MANETs integrates cryptographic assurance, session-based acknowledgment, and continuous monitoring to form a comprehensive defense against node misbehavior. The mechanism ensures the network's security, reliability, and resilience by employing mathematically rigorous methods, such as ECDSA-based signatures, session-specific digests, and threshold-based anomaly detection.

5.3 Extensive routing

The proposed routing protocol is an extension of the existing reactive AODV [22] routing protocol, where instead of hop count as a metric to compute the route, the proposed protocol includes Residual Buffer Metric and Current Residual Energy metrics to calculate the route to mitigate unintentional packet drops. Furthermore, the protocol includes cryptographic Decision-making and monitoring during the data forwarding phase to mitigate malicious packet drops.

The routing algorithm considers that each node in the network can determine the remaining buffer space and compute Residual Buffer Metric (RBM), and cable of determining the remaining energy and computing Current Residual Energy (CR). The protocol also considers that nodes that do not meet the energy threshold (CR_{Th}) are excluded from the routing process; otherwise, it adds its RBM to a cumulative value (RBM_{RBM}) in the Route Request (RREQ) packet. The nodes also monitor neighboring nodes' packet forwarding behavior.

5.4 Routing process and data transmission

When a source node wants to send data to a destination node, it generates an RREQ packet similar to an AODV RREQ packet, but instead of a hop count field in an AODV RREQ packet, it includes cumulative Residual Buffer Metric (RBM_{cum}). Initially, the value of RBM_{cum} in the RREQ packet is the source node's RBM value. Then, the source node Broadcasts the RREQ packets in the network. The intermediate node, which receives RREQ, computes its Current Residual Energy (CR); if it is more than the threshold value, then it participates in routing by calculating its RBM

and adding this value to the cumulative RBM_{cum} in the RREQ packet and rebroadcast; otherwise, it drops the RREQ packet. Furthermore, the intermediate nodes that satisfy the Current Residual Energy (CR) conditions rebroadcast the updated RREQ to its neighboring nodes, forwarding the cumulative RBM along the route. When the destination node receives multiple RREQs from different routes, it selects the path with the highest cumulative RBM_{cum} value. Thus, the computed path consists of nodes with higher buffer capacity and remaining energy and mitigates unintentional packet drops. The destination node unicast Route Reply (RREP) packet back to the source node through the computed path. Each node in the route maintains route information to enable efficient data transmission.

The source node sends the information to the destination node through the computed path in terms of data packets. Then, all the nodes in the route compute and update their residual energy and buffer status. The destination node sends an Acknowledgment (ACK) packet to the source node after a predefined amount of time (T) to verify successful packet delivery. This ACK packet contains a digest of received packets during the time (T) to ensure data integrity. The source node considers the route compromised whenever it does not receive ACK packets in a predefined amount of time and notices a discrepancy in the ACK packet digest value. Thus, the protocol successfully mitigates packet drops caused by malicious activities.

Further, the protocol uses promiscuous monitoring, in which every node monitors their neighbor regarding packet operations to determine the intentional drop. The monitoring node determines the difference between actual packets received to be forwarded and packets forwarded by its neighboring node, which will give the packet drop caused by the nodes. The node is marked as a malicious packet drops node if this value exceeds a certain level. Then, the monitoring node broadcasts the same information so that other communicating nodes avoid including this node. By utilizing essential techniques, the protocol successfully mitigates packet drops caused by both intentional and unintentional. The steps to compute route and data forward for communication are shown in Algorithm 3.

Algorithm 3. Dynamic and resilient routing protocol for MANET

1. Initialization: Initialize each node n_i with $RBM_i = 0$, $CR_i = 0$, session keys K_s , buffer capacity B_i , and energy capacity E_i .
 2. Packet Rate Calculation: For each node n_i Compute packet arrival and transmission rates. $R_i(t) = \alpha R_c(t) + \alpha R_p(t)$ and $T_i(t) = \alpha T_c(t) + \beta T_p(t)$ by adjusting α and β based on real-time traffic and historical data.
 3. Queue and Buffer Size Update: Update queue length and buffer utilization of each node by computing $Q_{p,i}(t) = Q_{p,prev}(t) + (R_i(t) - T_i(t)) \cdot \Delta t$ and $Q_{s,i}(t) = Q_{p,i}(t)$. kbytes
 4. Residual Buffer Space Metric (RBM) Calculation: Compute residual buffer space and RBM of each node $RB_i(t) = B_i - Q_{s,i}(t)$ and $RBM_i(t) = \frac{RB_i(t)}{B_i}$. Exclude the nodes from the routing if the node $RBM_i(t) \leq RBM_{th}(t)$, by adjusting the threshold value
-

5. Current Residual Energy (CR) Calculation: Calculate the energy consumption for each node $E_i(t) = (E_r(t) + E_p(t) + E_t(t))$ and $E_{resi,i}(t) = E_i - E(P_i)$ and compute the Current Residual Energy (CR) by condition Maximize $\sum_{P_i} E(P_i) \forall \sum_{P_i} E(P_i) \leq E_i$. Exclude the nodes from routing if $CR_i(t) < CR_{min}(t)$, by adjusting the CR threshold
6. Authenticated Key Agreement (AKA): Execute mutual authentication between communicating entities $\Psi_a(x) = (T_a(x) \oplus H(T_a(x))) \bmod P$ and $C_A = H(\Psi_a(x) \parallel PW \parallel s) \bmod p$
7. Digital Signature for Packet Integrity: Generate and verify signatures by $\sigma_{N_i} = \text{Sign}(k_{s_i}, H(P_k))$ and $\text{Verify}(k_{s_j}, H(P_k), \sigma_{N_i})$
8. Session Key-Based Acknowledgment: Compute the digests of the acknowledgment packet $D_{N_j} = H(\sum_{k=1}^n P_k \parallel K_s \parallel C_{n_j})$ and $D_{N_i} = H(\sum_{k=1}^n P_k \parallel K_s \parallel C_{N_i})$
9. Promiscuous Monitoring: Monitor neighboring nodes for packet operation by $\Delta N_j(t) = \text{ReceivedPackets to be forwarded}(N_j) - \text{ForwardedPackets}(N_j)$ and if $\Delta N_j(t) > \Theta(t)$ flag node as malicious
10. Route Discovery with RBM and CR: Broadcast Route Request (RREQ) with cumulative RBM and CR values:

$$RREQ_{cum} = \sum_{i=1}^k RBM_i(t) + \sum_{i=1}^k CR_i(t)$$

11. Route Selection: Select the route with the highest cumulative RBM $\text{Select Route} = \text{argmax}(RREQ_{cum})$ subject to at each node $CR_i(t) > CR_{min}(t)$
 12. Data Transmission and Integrity Check: transmit data and verify acknowledgment integrity $D_{n_j(t)} \neq D_{n_i(t)}$ then re-route and flag the compromised route.
-

6. PERFORMANCE ANALYSIS

The performance evaluation of the proposed work is conducted using an NS2 simulator and compared with existing reactive routing [22], resource-aware routing [23, 24], and secure knowledge algorithms [9] under identical conditions. The performance evaluation metrics are packet delivery, energy efficiency, and overhead. The simulation uses Constant Bit Rate (CBR) traffic at the application layer and UDP at the transport layer, as MANET is a connectionless, highly dynamic, and unreliable network. The network layer includes the protocol based on the specific one evaluated in NS2 for performance. The data link layer utilizes 802.11, allowing nodes to monitor all traffic within their radio range when necessary.

A variable number of nodes is considered with a random waypoint mobility model and a 20 m/s pause time. Each node is equipped with a 100-joule battery, a fixed radio transmission range of 250m, and an IEEE 802.11 MAC card with a data rate of 2 Mbps. Power consumption is set to 300mW for receiving and 600mW for transmission. Source nodes generate CBR traffic with a packet size of 512 bytes at a rate of 4 packets per

second, and the simulation duration is 1000 seconds. Nodes are categorized into three types: (i) reputed nodes that follow protocol specifications, (ii) 20% intentional misbehaving nodes that drop packets due to malicious activities, and (iii) unintentional misbehaving nodes that drop packets due to buffer overflow or constrained energy.

Each node is configured with a buffer size of 50 packets, simulating resource-constrained devices. The simulation occurs in a 1000m x 1000m area, reflecting the unpredictable topology changes typical in real-world MANET scenarios. The Proposed Protocol combines Promiscuous Monitoring with an Authenticated Key Agreement mechanism to secure the network against malicious activities while making energy-efficient and buffer-optimal routing decisions. The simulation results are shown in Figures 1 to 10.

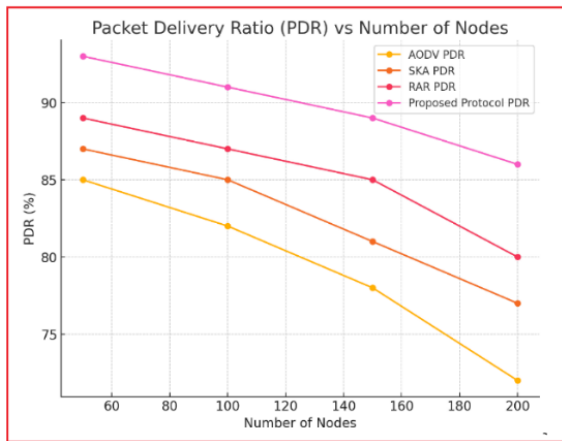


Figure 1. Packet delivery ratio comparison

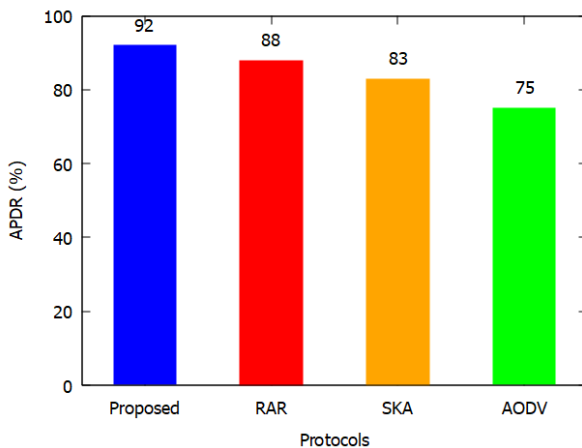


Figure 2. Average packet delivery ratio comparison

6.1 Discussion of results

The section discusses the results of the proposed work and compares it with existing AODV, SKA, and RAR protocols in terms of Packet Delivery Ratio (PDR), Energy Consumption, Delay, and Routing Overhead. The evaluation is conducted under varying conditions, such as the number of nodes, simulation time, and packet drops scenarios.

The Packet Delivery Ratio (PDR) measures the percentage of data packets that successfully reach their destination compared to the total number of packets sent. The Proposed Protocol achieves the highest Packet Delivery Ratio (PDR) across varying node counts, leveraging Residual Buffer Space

Metric (RBM) and Current Residual Energy (CR) for dynamic route adjustments. While AODV, SKA, and RAR show improvements, they lack the adaptive mechanisms that enhance PDR in the Proposed Protocol, shown in Figure 1 and 2.

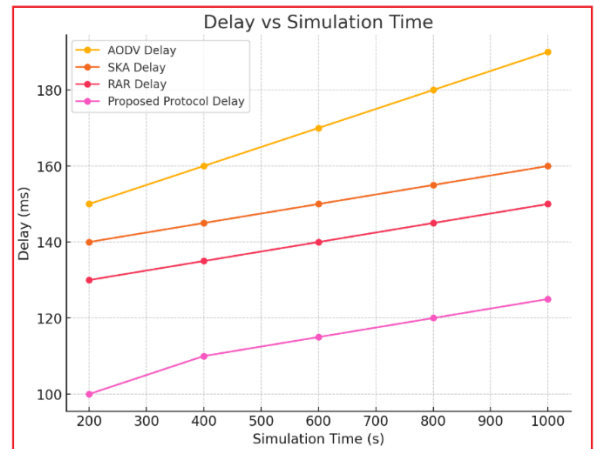


Figure 3. End-to-end delay comparison

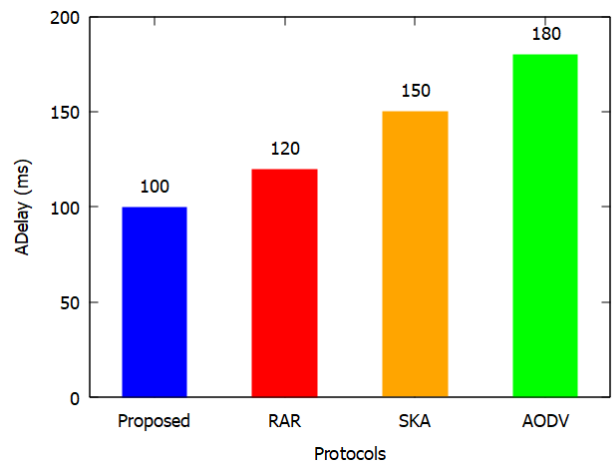


Figure 4. Average end-to-end delay comparison

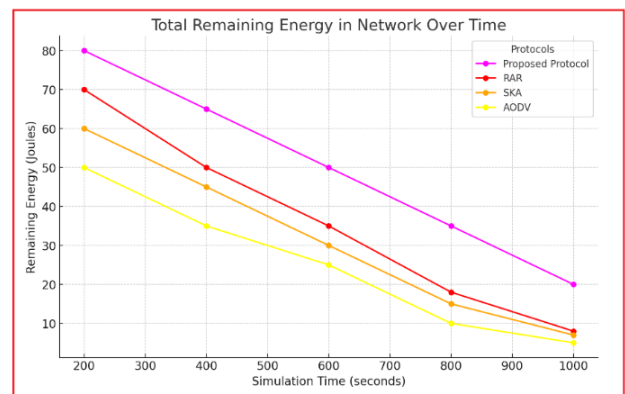


Figure 5. Energy efficiency comparison

End-to-end delay measures the time a data packet travels from the source to the destination. It is critical in time-sensitive applications where delays need to be minimized. The Proposed Protocol consistently achieves the lowest delay by dynamically avoiding congested nodes with low buffer and energy, outperforming AODV, SKA, and RAR. While AODV

suffers from higher delays due to frequent route rediscovery, SKA and RAR improve delays but lack the adaptive routing metrics of the Proposed Protocol, shown in Figures 3 and 4.

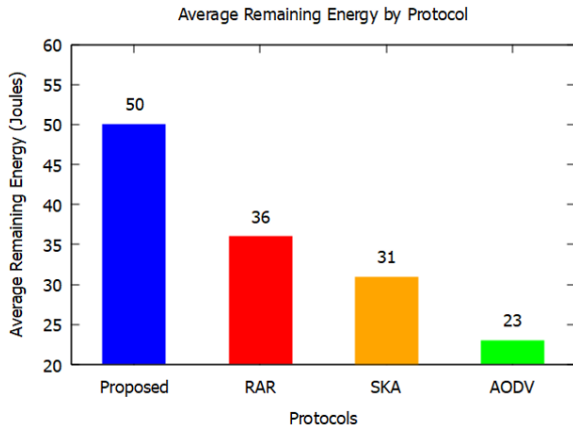


Figure 6. Average energy efficiency comparison

Energy consumption measures the average remaining energy per node after a certain simulation time. It is crucial in networks where nodes operate on limited battery power. Due to its dynamic routing decisions, the Proposed Protocol conserves energy most effectively, maintaining 80J after 200 seconds and 20J after 1000 seconds. AODV, by contrast, consumes the most energy, with only 50J remaining after 200 seconds and 5J after 1000 seconds due to inefficient routing, shown in Figures 5 and 6.

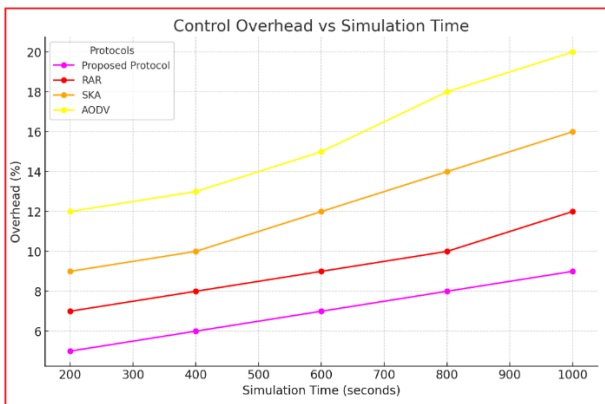


Figure 7. Overhead comparisons

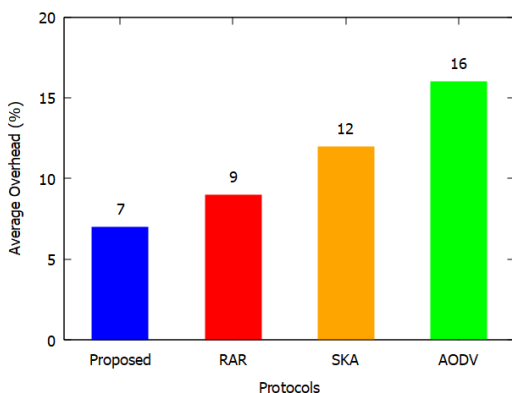


Figure 8. Average overhead comparisons

Routing overhead is the ratio of control packets (used for route discovery and maintenance) to the total number of data packets. High overhead indicates that a protocol uses more resources for control rather than actual data transmission. The Proposed Protocol incurs the lowest overhead, starting at 5% and rising to 9% due to efficient, dynamic routing. AODV, with the highest overhead at 12%-20%, suffers from frequent route discoveries, while RAR and SKA have moderate overhead due to resource-awareness and cryptographic processes, respectively, shown in Figures 7 and 8. All the performance, i.e., PDR, energy consumption, delay, and overhead comparisons are shown in Figure 9.

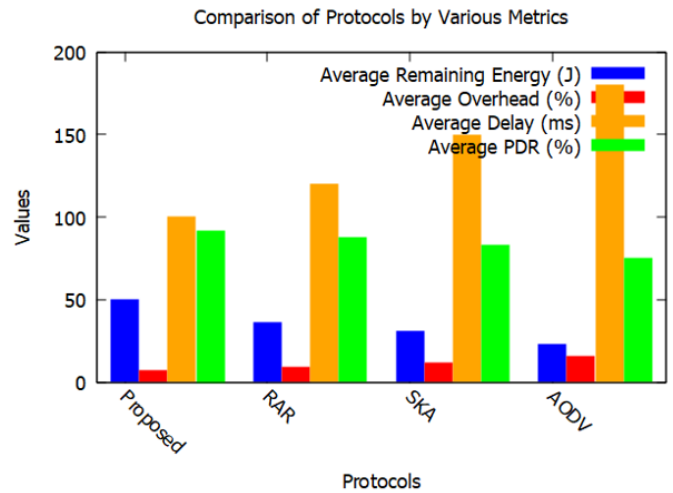


Figure 9. PDR, energy consumption, delay, and overhead comparison

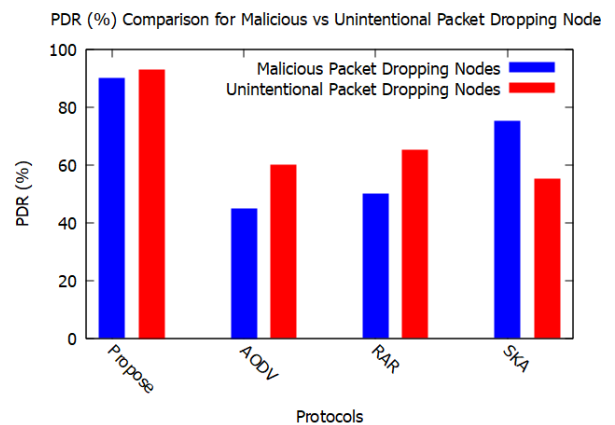


Figure 10. PDR (%) Comparison under malicious and unintentional packet drops nodes

Figure 10 contrasts, under two distinct packet drops scenarios, the Packet Distribution Ratio (PDR) performance of every protocol: aiming to interrupt network connection by refusing to relay any acquired packets, malicious packet drop nodes purposefully discard packets as part of a black-hole attack. Unintentional Packet Drops Nodes: Network resource limitations, especially buffer overflow brought on by congestion and energy depletion by restricted battery power in mobile nodes, cause packet drops.

Figure 10 compares how well each protocol PDR performs under malicious conditions (black hole) and unintentional drops (due to buffer overflow and energy depletion). The Proposed Protocol outperforms others, maintaining a high

PDR of 90% with malicious nodes and 93% with unintentional packet drops. The Proposed Protocol consistently outperforms the other protocols across all metrics—PDR, energy consumption, delay, and overhead. Its dynamic, resource-aware routing allows it to deliver more packets, conserve energy, minimize delay, and reduce overhead. AODV, while simple and widely used, suffers from poor performance due to its lack of resource awareness and attack mitigation mechanisms. RAR improves on AODV by incorporating resource awareness, but it still lacks the adaptive routing features of the Proposed Protocol. SKA provides moderate security through cryptographic mechanisms but at the cost of increased delay and overhead. Overall, the Proposed Protocol offers the best performance and efficiency balance in malicious and unintentional packet drop scenarios. The Proposed Protocol outperforms the other protocols across all metrics, demonstrating its adaptability, energy efficiency, and resilience under malicious and unintentional packet-drop scenarios. Its use of Residual Buffer Space (RBM) and Current Residual Energy (CR) allows it to adjust routing decisions dynamically, maximizing packet delivery and minimizing delay, overhead, and energy consumption.

7. CONCLUSION

The paper presents an extensive routing protocol to mitigate malicious and unintentional packet drops in MANETs. The routing decision is based on Residual Buffer Space Metric (RBM) and Current Residual Energy (CR), which ensure that only nodes with sufficient energy and buffer participate in routing, thereby mitigating unintentional packet drops. The authenticated key agreement, cryptographic mechanisms, and promiscuous monitoring mitigate intentional packet drops during data forwarding. Simulation results confirm that the proposed protocol outperforms existing mechanisms regarding packet delivery, energy consumption, delay, and overhead. The ability to adapt network dynamics and mitigate intentional and unintentional packet drops makes this algorithm suitable for the MANET environment to enhance security, reliability, and performance. Although the protocol reduces packet drops, the computational expense of updating RBM and CR metrics in real-time may cause scalability difficulties when node density is raised. Future research will optimize these computations using distributed or hierarchical routing strategies to preserve effective and scalable routing in crowded MANET settings. Further, Future work also focuses on integrating machine learning techniques for real-time detection of misbehaving nodes.

REFERENCES

[1] Fatima, M., Khurshed, A. (2022). Heterogeneous Ad-Hoc Network management: An overview. *Cloud Computing Enabled Big-Data Analytics in Wireless Ad-hoc Networks*, pp. 103-123. <https://doi.org/10.1201/9781003206453-7>

[2] Safari, F., Savic, I., Kunze, H., Ernst, J., Gillis, D. (2023). The diverse technology of MANETs: A survey of applications and challenges. *International Journal of Future Computer and Communication*, 12(2): 37-48. <https://doi.org/10.18178/ijfcc.2023.12.2.601>

[3] Chitra, P., Ranganayaki, T. (2020). A study on Manet:

Applications, challenges and issues. *International Journal of Engineering Research & Technology (IJERT)*, 8(3): 1-4.

[4] Malik, K., Bhasin, A. (2022). A survey of mitigation techniques of packet drop attacks in MANET. In *Proceedings of the International Conference on Innovative Computing & Communication (ICICC)*, Delhi, India, pp. 1-5.

[5] Mohammad, A.A.K., Mahmood, A.M., Vemuru, S. (2019). Intentional and unintentional misbehaving node detection and prevention in mobile ad hoc network. *International Journal of Hybrid Intelligence*, 1(2-3): 239-267. <https://doi.org/10.1504/IJHI.2019.103580>

[6] Banerjee, S. (2008). Detection/Removal of cooperative black and gray hole attack in mobile ad-hoc networks. In *Proceedings of the World Congress on Engineering and Computer Science (Vol. 2008)*, London, U.K.

[7] Shakshuki, E.M., Kang, N., Sheltami, T.R. (2012). EAACK—A secure intrusion-detection system for MANETs. *IEEE Transactions on Industrial Electronics*, 60(3): 1089-1098. <https://doi.org/10.1109/TIE.2012.2196010>

[8] Vijayalakshmi, S., Bose, S., Logeswari, G., Anitha, T.J.C.S. (2023). Hybrid defense mechanism against malicious packet dropping attack for MANET using game theory. *Cyber Security and Applications*, 1: 100011. <https://doi.org/10.1016/j.csa.2022.100011>

[9] Siddiqua, A., Sridevi, K., Mohammed, A.A.K. (2015). Preventing black hole attacks in MANETs using secure knowledge algorithm. In *2015 International Conference on Signal Processing and Communication Engineering Systems*, Guntur, India, pp. 421-425. <https://doi.org/10.1109/SPACES.2015.7058298>

[10] Djahel, S., Nait-Abdesselam, F., Zhang, Z. (2010). Mitigating packet dropping problem in mobile ad hoc networks: Proposals and challenges. *IEEE Communications Surveys & Tutorials*, 13(4): 658-672. <https://doi.org/10.1109/SURV.2011.072210.00026>

[11] Buttyán, L., Hubaux, J.P. (2003). Stimulating cooperation in self-organizing mobile ad hoc networks. *Mobile Networks and Applications*, 8: 579-592. <https://doi.org/10.1023/A:1025146013151>

[12] Saetang, W., Charoenpanyasak, S. (2012). Caodv free blackhole attack in ad hoc networks. In *International Conference on Computer Networks and Communication Systems (CNCS 2012)*, Kuala Lumpur, Malaysia, pp. 63-58.

[13] Thachil, F., Shet, K.C. (2012). A trust based approach for AODV protocol to mitigate black hole attack in MANET. In *2012 International Conference on Computing Sciences*, Phagwara, India, pp. 281-285. <https://doi.org/10.1109/ICCS.2012.7>

[14] Kshirsagar, D., Patil, A. (2013). Blackhole attack detection and prevention by real time monitoring. In *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, Tiruchengode, India, pp. 1-5. <https://doi.org/10.1109/ICCCNT.2013.6726597>

[15] Ukey, A.S.A., Chawla, M. (2010). Detection of packet dropping attack using improved acknowledgement based scheme in MANET. *IJCSI International Journal of Computer Science Issues*, 7(4): 12-17.

[16] Al-Roubaiey, A., Sheltami, T., Mahmoud, A., Shakshuki, E., Mouftah, H. (2010). AACK: Adaptive

- acknowledgment intrusion detection for MANET with node detection enhancement. In 2010 24th IEEE International Conference on Advanced Information Networking and Applications, Perth, WA, Australia, pp. 634-640. <https://doi.org/10.1109/AINA.2010.136>
- [17] Vazifehdan, J., Prasad, R.V., Niemegeers, I. (2013). Energy-efficient reliable routing considering residual energy in wireless ad hoc networks. In IEEE Transactions on Mobile Computing, 13(2): 434-447. <https://doi.org/10.1109/TMC.2013.7>
- [18] De Rango, F., Fazio, P., Scarcello, F., Conte, F. (2014). A new distributed application and network layer protocol for VoIP in mobile ad hoc networks. IEEE Transactions on Mobile Computing, 13(10): 2185-2198. <https://doi.org/10.1109/TMC.2014.2307315>
- [19] Ahmad, S.J., Reddy, V.S.K., Damodaram, A., Krishna, P.R. (2015). Delay optimization using Knapsack algorithm for multimedia traffic over MANETs. Expert Systems with Applications, 42(20): 6819-6827. <https://doi.org/10.1016/j.eswa.2015.04.027>
- [20] Srinivas, J., Das, A.K., Wazid, M., Kumar, N. (2018). Anonymous lightweight chaotic map-based authenticated key agreement protocol for industrial Internet of Things. IEEE Transactions on Dependable and Secure Computing, 17(6): 1133-1146. <https://doi.org/10.1109/TDSC.2018.2857811>
- [21] Sowjanya, K., Dasgupta, M., Ray, S. (2021). Elliptic curve cryptography based authentication scheme for internet of medical things. Journal of Information Security and Applications, 58: 102761. <https://doi.org/10.1016/j.jisa.2021.102761>
- [22] Chakeres, I.D., Belding-Royer, E.M. (2004). AODV routing protocol implementation design. In 24th International Conference on Distributed Computing Systems Workshops, 2004, Proceedings, Tokyo, Japan, pp. 698-703. <https://doi.org/10.1109/ICDCSW.2004.1284108>
- [23] Nizamuddin, M.K., Mohammad, A.A.K., Hashmi, S.S., HariKrishna, D., Anusha, M. (2024). Efficient routing in MANETs by optimizing packet loss. Ingenierie des Systemes d'Information, 29(3): 961-968. <https://doi.org/10.18280/isi.290316>
- [24] Sastry, M.K., Mohammad, A.A.K., Arif, M.A. (2021). Optimized energy-efficient load balance routing protocol for wireless mesh networks. International Journal of Advanced Computer Science and Applications, 12(8): 605-610.