Manet Transport Layer Congestion Control Using a Multilevel Queue Management Scheme

S. Hemalatha[1], Khadri Syed Faizz Ahmad[2], Nripendra Narayan Das[3,*], R. V. V. Krishna[4], U. Sathya[5], and Radha Mothukuri[6]

[1] Department of Computer Science and Business Systems, Panimalar Engineering College, Chennai, Tamil Nadu, India
[2] Department of Computer Science Engineering, Aacharya Nagarjuna University, Andhra Pradesh, India
[3] Department of Information Technology, Manipal University Jaipur, Rajasthan, India
[4] Electrical and Computer Engineering Department, Aditya University, Surampalem, Kakinada District, Andhra Pradesh, India
[5] Computer Science and Engineering, Saveetha Engineering College, India
[6] Department of Electrical and Computer Engineering, Koneru Lakshmaiah Education Foundation Vaddeswaram, AP, India
Email: pithemalatha@gmail.com (S.H.); faizzkhadri@gmail.com (K.S.F.A.); nripendranarayan.das@jaipur.manipal.edu (N.N.D.); rvvkrishnaece@gmail.com (R.V.V.K.); sathyau@saveetha.ac.in (U.S.); Radhamothukuri@kluniversity.in (R.M.)
*Corresponding author

Abstract*—Congestion control is a significant challenge in the Transport layer of Mobile Ad Hoc Networks (MANETs). Despite extensive research, the issue remains unresolved. Numerous studies have attempted to address congestion control by integrating solutions with the Medium Access Control (MAC) layer and routing protocols, yet these approaches have encountered various limitations. This article proposes a novel solution focused on congestion control in the Transport layer through buffer management. The proposed solution introduces a Multilevel Queue Management (MQM) algorithm, combined with a packet memory buffer that schedules internal packets based on packet categories and queue assignments. The queue management system categorizes packets into four types: beacon signals, control signals, synchronization signals, and data packets. In the event of network congestion, data packets are redirected by clearing other lower-priority queues to ensure seamless transmission. This queue management system is integrated with the Ad Hoc On-Demand Distance Vector (AODV) protocol, forming the Multilevel Queue Management-Ad Hoc On-Demand Distance Vector (MQM-AODV) protocol. The performance of MQM-AODV (Multi level Management AODV) is evaluated against other algorithms, including Ant Colony Optimization (ACO-AODV), Reinforcement Learning-Ad Hoc On-Demand Distance Vector (RL-AODV), and Categorical Boosting AODV-Ad Hoc On Demand Distance Vector (CATBOOST-AODV). Simulation results, conducted with node counts ranging from 25 to 150, demonstrate that MQM-AODV outperforms these protocols. Specifically, MQM-AODV improves throughput by up to 60.74% over (ACO-AODV) and 36.41% over Reinforcement Learning AODV (RL-AODV), while reducing end-to-end delay by 35.47% compared to ACO-AODV. Although buffer capacity remains similar across protocols, MQM-AODV excels in maximizing network lifetime and minimizing stabilization time, significantly enhancing network reliability. Despite requiring additional scheduling time, the overall performance gains position MQM-AODV as a superior choice for high-demand network environments.

***Keywords*—Mobile Ad Hoc Networks (MANET), congestion control, multilevel queue management, transport layer,Ad Hoc On-Demand Distance Vector (AODV)**

I. INTRODUCTION

Mobile Ad Hoc Networks (MANETs) [1] are infrastructure-less networks that can be set up without the need for any access points, making them ideal for emergency communication in situations such as disaster management, earthquakes, and other crises. The protocol stack of a MANET consists of five layers, each responsible for specific functions. The Physical Layer handles communication with the channels and the transmission of packets in bits, while the Data Link Layer is responsible for adding checksums and converting packets into frames. The Network Layer focuses on route discovery and packet forwarding, and the Transport Layer ensures reliable communication by enabling the source node to transfer packets to the destination node. A key challenge within the Transport Layer is congestion control [2], which is crucial for maintaining reliable connectivity between nodes in the network.

A variety of studies have been conducted to investigate challenges in the Transport layer**,** particularly concerning congestion control [2, 3], analysis of congestion control mechanisms [3, 4], and the development of congestion control protocols [5]. Research has explored several dimensions to propose solutions for managing congestion in the Transport layer. Some studies focus on load balancing [6, 7] among nodes, aiming to achieve

Manuscript received October 14, 2024; revised November 25, 2024; accepted December 14, 2024; published January 24, 2025.

congestion control, although these approaches have shown limited success in delivering optimal performance. Other research has used routing protocols designed for congestion control [8, 9], which can add complexity to the network layer's operations. Several groups have concentrated on monitoring aspects like traffic [10, 11], bandwidth [12, 13], queue management [11, 14], and basic algorithms [15, 16], but these methods have not fully alleviated the congestion issue.

Another group of researchers has focused on congestion avoidance mechanisms [17–19], which, while effective, often add additional burden to the overall MANET operation. Some studies have explored the use of advanced technologies such as machine learning [20], fuzzy logic [21–23], neural networks [13], Reinforcement Learning [24, 25], and hybrid congestion control protocols [26, 27]. However, none of these approaches have achieved the desired results. More recent efforts have concentrated on improving Transport layer performance with end-to-end congestion control methods for 4G/5G networks, aimed at maximizing throughput [28] and optimizing energy consumption in congestion management [29–31]. Despite these advancements, no single approach has fully addressed all performance factors. The search for an effective solution to Transmission Control Protocol (TCP) layer congestion control remains ongoing.

To demonstrate the relationship between the discussed routing algorithms and the evolving contexts of the Internet of Things (IoT), Artificial Intelligence (AI), and 6G, it is essential to consider the unique demands and challenges posed by these emerging paradigms. In the IoT era, where billions of interconnected devices operate in dynamic environments, efficient routing protocols are critical for ensuring seamless communication, minimal latency, and optimized resource utilization. Similarly, the integration of AI in network management offers opportunities for intelligent, adaptive routing strategies that leverage real-time data to make informed decisions, thereby enhancing network performance and reliability. Meanwhile, the advent of 6G networks emphasizes the need for ultra-low latency, massive connectivity, and high throughput, further intensifying the demand for innovative routing solutions that meet these advanced requirements.

This article introduces a basic technique called Multiple Queue Management-Ad Hoc On-Demand Distance Vector (MQM-AODV), designed to categorize incoming packets into multiple queues. The scheduling mechanism plays a key role in determining which queue to prioritize when congestion occurs, helping to manage the internal buffer that is specially structured to avoid congestion. The queue management strategy proposed in this work categorizes packets into four types: beacon signals, control signals, synchronization signals, and data packets. Each category is tailored to handle specific network traffic, ensuring efficient data flow management. When congestion arises, the system dynamically redirects data packets to an alternate queue, achieved by clearing non-priority packets in other queues, thus ensuring the seamless delivery of critical data packets. This approach minimizes delays and optimizes resource utilization, maintaining high performance and reliability even under heavy traffic conditions.

Protocols like MQM-AODV, with features such as multilevel queue management and minimal overhead, are well-suited for IoT and AI-driven environments. Their ability to dynamically adapt to network conditions ensures consistent performance under varying loads—a crucial requirement for IoT ecosystems, where device density and communication patterns can be highly variable. Similarly, the integration of AI in protocols like Reinforcement Learning-Ad Hoc On-Demand Distance Vector (RL-AODV) and Categorical Boosting AODV-Ad Hoc On Demand Distance Vector (CATBOOST-AODV) demonstrates the potential of machine learning techniques to predict and adapt to network changes. However, the limitations of current implementations, such as queue delays, highlight the need for further optimization to improve efficiency.

In the context of 6G, the high throughput and low latency achieved by MQM-AODV make it a promising candidate for next-generation networks. Its dynamic nature and minimal overhead align well with 6G's goals of efficient resource utilization and enhanced energy efficiency. As networks evolve toward a fully AI-integrated 6G infrastructure, the incorporation of advanced decision-making algorithms, such as those found in RL-AODV and CATBOOST-AODV, will be critical for enabling adaptive and intelligent routing in complex network environments. This paper introduces MQM-AODV, a routing protocol that combines dynamic multi-queue management with cross-layer feedback to provide superior congestion control, distinguishing it from existing strategies like machine learning-based methods and traditional buffer management techniques. To evaluate the effectiveness of MQM-AODV, we have selected ACO-AODV, RL-AODV, and CATBOOST-AODVas benchmarks. Each of these algorithms represents a unique optimization paradigm, with well-established performance in MANETs. By comparing MQM-AODV with these protocols, we aim to offer a comprehensive assessment of its strengths and identify any potential limitations.

A. Contributions of This Article

This paper presents a novel approach to congestion control in the Transport layer using Multilevel Queue Management (MQM) integrated with the AODV routing protocol. The main contributions of this work are as follows:

(1) Introduction of a Packet Categorization Scheme**:** A new packet categorization method that prioritizes traffic based on urgency and type, improving congestion management and throughput.

(2) Integration of MQM with AODV: The development and seamless integration of the MQM mechanism with the AODV protocol, enabling efficient routing decisions under dynamic network conditions.

(3) Extensive Simulation-Based Evaluation: A thorough performance evaluation of the proposed MQM-AODV protocol using various network scenarios, demonstrating significant improvements in throughput, end-to-end delay, buffer capacity, and network lifetime compared to existing methods.

This article is organized as follows: Section II summaries the many authors' literature reviews, Section III discusses the proposed Methodology using MQM, and Sections IV and V examine simulation and simulated findings. Finally, in Section VI, the research work with prone and cones is concluded with future work.

II. Literature Review

This chapter elaborates on the researchers' diverse efforts to prevent and manage congestion using both conventional and modern technologies. Mohan and Vimala [1] propose using a standard Media Access Control (MAC) protocol to reduce congestion and enhance Quality of Service (QoS). They deployed a fuzzy-based congestion control approach for detection and utilized the Artificial Intelligence and Ant Colony Optimization-Assisted Routing (AIACOAR) protocol and ECC encryption to optimize multi-hop scheduling security. The initial congestion mitigation targeted latency, data rate, overhead, and throughput, while improving Packet Delivery Ratio (PDR) and End-to-End Delay (END). Implemented in MATLAB, the proposed method showed a 99% PDR with a low delay of 0.24s, outperforming Artificial Intelligence and Fuzzy System Optimized Routing Protocol (AIFSORP) and Levy Flight-based Stochastic Swarm Optimization (LF-SSO). However, higher transmission rates required for threshold balancing put additional strain on congestion control, highlighting the need for improved system performance to reduce latency and queue delay.

Sofyan *et al.* [2] compared the performance of Transmission Control Protocol, Adaptive Fractional Window (TCP AFW) and TCP FeW in an IEEE 802.11 wireless network. Their findings showed that while TCP FeW outperformed TCP New Reno, TCP AFW outperformed TCP FeW by 1.12% in NS2 simulations under various scenarios. They concluded that TCP AFW did not require modifications to traditional TCP, as the window growth rate ($0 < \alpha < 1$) remained consistent even during congestion. Additionally, a new algorithm was proposed to handle packet loss in TCP AFW for better congestion control. Ouladdjedid *et al.* [3] identified that network congestion results from excessive traffic and link failures. They proposed a load balancing congestion control mechanism among nodes, shifting traffic from congested paths to less congested routes, even if they involve more hops. Their Trust and Load Balancing—Ad Hoc On-Demand Distance Vector (TALB-AODV) protocol was simulated and compared with AOMDV, showing improvements in lifetime, energy efficiency, traffic balancing, and reduced packet loss.

Seyed and Roya [8] introduced the Ant Colony Optimization (ACO) algorithm to address congestion in wireless networks. The study created a cluster head and used the ACO algorithm for route analysis and packet forwarding via Tabu Search. Simulations compared to Ant Colony System-Based Routing Optimization (ACSRO), Fuzzy Clustering Optimization with Artificial Bee Colony (FCOABC), and Flock-CC algorithms showed reduced packet loss, extended node life, and energy optimization. However, route identification based on traffic thresholds and alternate path creation placed additional strain on the cluster head node.

Nibedita *et al.* [12] presented the Agent-Based Mobility Model (ABMM) algorithm using the Frequency Band Resource Allocation (FBRA) band to avoid MANET congestion, determining congestion based on channel utilization and queue length. The sender nodes validate files and route bandwidth before sending an ACK. Compared to traditional congestion control, FBRA showed better END, higher PDR, and lower error rates, though determining channel utilization and queue length via control signals was time-consuming.

Liu *et al.* [24] proposed a Markov decision process with reinforcement learning for congestion control in dynamic VANETs. The RL algorithm adjusted traffic based on current channel conditions, with the QBACC model outperforming other dynamic models. However, maintaining congestion at lower critical levels remained a challenge. Christy [26] developed a hybrid congestion control technique that optimized data processing and transmission time, reducing latency and increasing throughput. However, it added stress to the system due to the complex transmission time computation and data handling. Ouladdjedid *et al.* [3] introduced 5G technology in VANET to address MAC layer channel congestion caused by increased vehicle density. While DCC and ETSI methods reduced channel congestion and error margins, lowering latency, performance dropped with increased VANET channel load.

Thrimoorthy *et al.* [13] proposed the OUT-CONA congestion control model for IoT-enabled wireless mesh networks, combining adaptive reinforcement learning with Gauss and Markov Processes for energy savings. The cache obliviousness approach integrated Long Short Term Memory (LSTM) with cache congestion, showing good throughput, although stage-by-stage congestion processing caused time variations in packet flow. Augustine *et al.* [27] presented the Responsive Hybrid Routing protocol, merging AODV and Outstanding Loss Reserves (OSLR) capabilities. NS3 simulations with 20-200 nodes showed improved PDR, END, jitter, and throughput compared to OSLR and AODV, although some delay mitigation was needed for packet transfers in MANET.

Juan *et al.* [20] introduced a machine learning-based congestion control strategy for multi-hop wireless networks, utilizing a decision tree-relayed CatBoost algorithm to predict packet transmission based on network activity. While simulation results demonstrated optimal performance, the involvement of the decision tree led to delays in packet queues. Noussaiba *et al.* [15] developed the ACOPT algorithm for VANET IoV congestion control, integrating pheromone termite and ACO algorithms. Simulations comparing ACOPT with ACOFL, RACO, and AODV, ACO showed improvements in energy efficiency, throughput, and end-to-end latency, although internal packet transit within the Azure Virtual Desktop (AVD) module caused delays. Research team of Jiang *et al.* [29] designed the AODV-EOCW protocol for single-metric routes, combining AHP, EWM, and AODV. Simulation results showed that AODV-EOCW outperformed vanilla AODV and AODV-UU in average END, PDR, node lifetime, and congestion prevention, but empirical energy threshold levels and real-world deployment posed challenges. Ma *et al.* [9] addressed DSR routing congestion and load balancing with the enhanced DSR-P protocol, integrating hello messages to manage congestion. Simulations showed improvements in PDR, END, overhead, and energy utilization, although some data values conflicted with network design.

Quy *et al.* [32] proposed a Cross-Layer Routing Approach for MANET-Assisted IoT Applications, introducing a dynamic routing mechanism that leverages cross-layer design principles. This approach integrates metrics such as hop count, link quality, and queue length, utilizing Data Queues (DqDqDq), Synchronization Queues (SqSqSq), Beacon Queues (BqBqBq), and Control Queues (CqCqCq) to optimize routing decisions. Performance metrics such as packet delivery ratio, end-to-end delay, throughput, energy consumption, and routing overhead were used for evaluation. Simulation results demonstrated superior performance compared to protocols like AODV and DSR, particularly in dynamic and resource-constrained environments. Key advantages of this approach include enhanced efficiency, adaptability, scalability, and reduced energy consumption. However, the method introduces complexity and routing overhead due to its multi-layer dependency, and its security implications require further exploration. This approach represents a significant step forward in routing for MANET-assisted IoT networks, balancing efficiency and practicality.

A growing body of research has explored the application of machine learning algorithms to enhance security in Mobile Ad Hoc Networks (MANETs), particularly for tasks such as intrusion detection, anomaly detection, and routing optimization. Techniques like reinforcement learning and neural networks have been utilized to dynamically adapt to changes in network topology and evolving security threats, as highlighted by Kumar and Sharma [33]. According to the Bhattacharya [34], MANETs face several key challenges, including limited bandwidth, dynamic topology, and inherent security vulnerabilities. These challenges necessitate the development of efficient, adaptive, and secure routing protocols capable of responding to constantly changing network conditions.

Given the decentralized nature of MANETs, efficient key management schemes are essential for maintaining secure communication. Techniques such as distributed, hierarchical, and cluster-based key management have been developed to enhance communication security, directly influencing the performance and scalability of routing protocols [35]. Cluster-based key management schemes [36] help reduce the overhead associated with key distribution by organizing nodes into clusters, where only nodes within a cluster participate in key exchanges. This approach improves both security and efficiency, making it particularly effective for large-scale MANETs.

Existing approaches to congestion control, including machine learning techniques like reinforcement learning and traditional buffer management, often focus on congestion prediction or static prioritization. In contrast, MQM-AODV introduces a dynamic multi-queue management system, integrating real-time cross-layer metrics for adaptive traffic prioritization. This system addresses the challenges of high mobility and constrained resources, making it well-suited for IoT and MANET environments. The survey on congestion control reveals various approaches, including new protocols, hybrid methods, and advanced techniques such as fuzzy logic [37, 38], machine learning, and clustering algorithms [39, 40]. However, no single research has fully addressed all performance factors, and the quest for effective TCP layer congestion control continues.

Various routing protocols have been explored for MANETs, including swarm-based approaches like ACO-AODV, machine learning-based solutions like RL-AODV, and gradient boosting techniques such as CATBOOST-AODV. These protocols were selected for their proven success in optimizing routing performance and their ability to handle dynamic network environments. A comparison with MQM-AODV, which incorporates a multilevel queue management strategy, will provide valuable insights into its relative performance in similar settings. This article proposes a straightforward method that uses an internal buffer in a queue-based style to monitor and prevent congestion.

III. Research Methodology

This research study focuses on controlling congestion by referring to a literature survey and concludes that better buffer management techniques will help manage TCP layer congestion. All types of incoming packets are temporarily kept in the internal buffer form before being transmitted to the next node or hop. Congestion could be avoided by making minor changes to the internal buffer. The MQM-AODV protocol employs four distinct queues Data, Synchronization, Beacon, and Control Queues to manage traffic dynamically. Unlike traditional methods, this approach leverages cross-layer feedback, such as link quality and queue length, for real-time adaptation, enabling finer granularity in traffic handling and prioritization.

In this study, we evaluate MQM-AODV against three benchmark algorithms: ACO-AODV, RL-AODV, and CATBOOST-AODV. These were chosen to represent diverse optimization paradigms in MANETs: swarm intelligence, machine learning, and gradient boosting. While MQM-AODV introduces a multilevel queue management approach to enhance routing efficiency, it is important to consider the potential trade-offs involved. Specifically, we explore the computational overhead introduced by multilevel queue management, the impact on scheduling latency, and how these affect overall network performance.

The majority of MANET packets are classified as Data Packets and Control Packets. Control packets carry synchronization signals, beacon signals, and other command signals, whereas data packets carry actual data. This study paper focuses on the entering types of packets and classifying them into groups. As indicated in Fig. 1, the internal node buffer will be separated into four category portions. Each section contains the different sorts of packets, which are managed by the Multilevel Queue Management algorithm with the help of the Multilevel Queue Scheduler. The multilevel queue scheduler performs two functions: classification of incoming packets and routing to the appropriate queue, and monitoring traffic and diverting incoming packets to avoid congestion. The MQM (Multi-level Queue Management) scheduler monitors and detects the onset of congestion primarily by observing the length and growth rate of the queues in each priority level. It maintains separate queues for different traffic types, such as real-time, non-real-time, and best-effort traffic, and tracks the buffer occupancy of these queues. When the queue length exceeds a predefined threshold or grows rapidly over a short period, it signals the potential onset of congestion. The scheduler may also monitor additional factors like packet arrival rates, buffer utilization, and packet drop rates. If the buffer occupancy approaches critical levels, the MQM scheduler can adjust its traffic handling policies, such as prioritizing higher-priority packets, dropping lower-priority packets, or reducing the sending rates, to alleviate the congestion and ensure smooth traffic flow. This proactive monitoring helps to minimize packet loss, delay, and jitter during times of high traffic load.

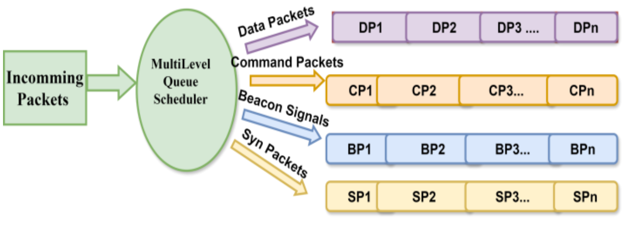


Fig. 1. Multilevel queue scheduler.

The following is how the multilevel queue management algorithm works:

Allow the node buffer size to be N bytes.

Node Buffer $N = \{n1, n2, n3 ... nn\}$

Allocate the buffer's memory location in four distinct queues to four different types of packets (data packet, synchronization signal, beacon signal, and command signal). Assume that half of the memory is dedicated to storing the Data packet queue. The remaining half of the RAM is used to store 10% of the beacon signal and 20% of the synchronization and command signals.

Finally, the Multilevel Queue for different categories defines as follows:

$$\textit{Data Queue} = \{Dq1, Dq2, Dq3 ....Dqn/2\}$$

$$\text{Synchronization Queue } \{Sq1, Sq2, Sq3 ...Sqm\}$$

$$\text{Beacon Queue} = \{Bq1, Bq2, Bq3...Bqo\}$$

$$\text{Control Queue} = \{Cq1, Cq2, Cq3...Cqp\}$$

where *Dq* is a Data Queue, *Sq* is a Synchronization Queue, *Bq* is a Beacon signal Queue, *Cq* is a Control Signal Queue.

TABLE I. QUEUE AND MEANING

Symbol	Meaning
D_q	Data Queue: Stores data packets or elements.
$D_{q1}, D_{q2}, \ldots, D_{qn}/$	Specific instances or levels within the Data Queue.
S_q	Synchronization Queue: Manages synchronization signals.
$S_{q1}, S_{q2}, \ldots, S_{qm}$	Specific instances or levels within the Synchronization Queue.
Bq	Beacon Queue: Stores beacon signals for communication or signaling tasks.
$B_{q1}, B_{q2}, \ldots, B_{qo}$	Specific instances or levels within the Beacon Queue.
C_q	Control Queue: Stores control signals for system management.
$C_{q1}, C_{q2}, \ldots, C_{qp}$	Specific instances or levels within the Control Queue.

Table I provides a summary of the mathematical notations and their definitions, along with the associated queues and their respective meanings.

Then, as indicated below, use Multilevel Queue Management to avoid congestion.

A. Clarification of Queue Division

In MQM-AODV, packets are categorized into four distinct types: Data, Synchronization, Beacon, and Control packets. Each type is assigned to a separate queue based on its role in the network. For instance, Data packets, which are typically used for the transmission of user information, are placed in the Data Queue (Dq), while Synchronization packets that help maintain network timing are placed in the Synchronization Queue (Sq). Beacon packets for network discovery and Control packets for route maintenance are similarly organized into their respective queues (Bq and Cq). This division enables prioritized handling of traffic based on the packet type and its urgency.

B. Prioritization Logic and Scheduling Policies

Each queue employs a specific scheduling policy tailored to the packet type. For example, Data packets may be prioritized using a Priority Queue policy, where higher-priority data (e.g., time-sensitive information) is transmitted first. On the other hand, Synchronization packets might use a Round Robin policy to ensure fair access to the network's timing resources. Beacon and Control packets may be handled using a Weighted Fair Queuing (WFQ) algorithm, ensuring that control-related communication is handled without overloading the network. This hybrid approach balances fairness and priority handling, improving overall network efficiency.

Detailed Breakdown of Packet Categorization and Prioritization

1. **Packet Categorization**: When a packet arrives, it is first categorized based on its type (Data, Synchronization, Beacon, or Control).

2. **Queue Assignment**: The categorized packet is then placed into the appropriate queue (Dq, Sq, Bq, or Cq).

3. **Prioritization**: Within each queue, the packet is assigned a priority based on its characteristics (e.g., time sensitivity for Data packets or network stability for Control packets).

4. **Scheduling**: Each queue applies its respective scheduling policy (e.g., FCFS for Data, Round Robin for Synchronization), and packets are transmitted accordingly

Flowchart or Algorithmic **Pseudo code Section**

Case 1: Standard Mode of Operation

1. Gather arrival packets from all nearby nodes.

2. Divide the packet into four parts: data, control, synchronization, and beacon signal.

3. Placing the packet in the appropriate queue.

When there is congestion, if the buffer reaches 75% of its authorized memory, contact the Multilevel Queue management to predict and guide the packet.

Case 2: Mode of operation with congestion

Check the buffer capacity of each queue

If (Data Packet queue is full)

Switch all incoming data packet to the synchronization queue buffer

Else if (the beacon signal queue full)

Clear the beacon signal buffer

Else if (If command packet buffer is full)

Redirect the incoming command packet to the synchronization buffer Queue

Else if (Synchronization packet is Full)

Clear the internal synchronization buffer and initiate alert to the node about the Intruder or attacker.

The following pseudo code outlines the categorization, prioritization, and scheduling steps in MQM-AODV's packet handling process.

```
//QM-AODV Packet Categorization and Transmission
Algorithm
for each packet in the network:
if packet.type == "Data":
assign to Data Queue (Dq)
apply Priority Queue policy
else if packet.type == "Synchronization":
assign to Sync Queue (Sq)
apply Round Robin policy
 else if packet.type == "Beacon":
assign to Beacon Queue (Bq)
apply Weighted Fair Queuing policy
else if packet.type == "Control":
assign to Control Queue (Cq)
apply Weighted Fair Queuing policy
// Transmit packet based on queue scheduling policy
transmit packet from appropriate queue
```

From Fig. 2, the scheduler will operate in two modes: normal mode and congestion mode. While in regular mode, the scheduler is acting simply by analyzing the incoming packet and forwarding it to the appropriate packet type Queue. When the buffer reaches 75% capacity, the scheduler does congestion avoidance operations in advance by converting the privilege packet buffer to a non-privilege packet buffer. The 75% buffer threshold is not a fixed value used universally for congestion detection, but rather a common reference point that can vary based on the specific network setup, application requirements, and traffic characteristics. In some networks, especially those with time-sensitive applications like VoIP or video streaming, a lower threshold (e.g., 60%) may be used to detect congestion earlier and prevent latency or jitter. Conversely, in less sensitive applications, a higher threshold (e.g., 85%) may be acceptable. Factors like the nature of traffic (burst or steady), node capabilities (memory and processing power), and the specific congestion control strategy employed (such as Random Early Detection or Weighted Fair Queuing) all influence the choice of buffer threshold. Therefore, the 75% threshold is adjustable, depending on the network's requirements and goals for managing traffic congestion effectively. The data packet buffer has privilege, but the beacon signal buffer does not. When a data buffer is full, move the data buffer to the synchronization buffer; when a beacon buffer is full, clear the buffer because the memory allocation for this buffer is less; when a command buffer is full, move the command packet to the syn buffer; and finally, when the synchronization buffer is full without moving the packet from another buffer, an alert message is sent to the node about the attacker.

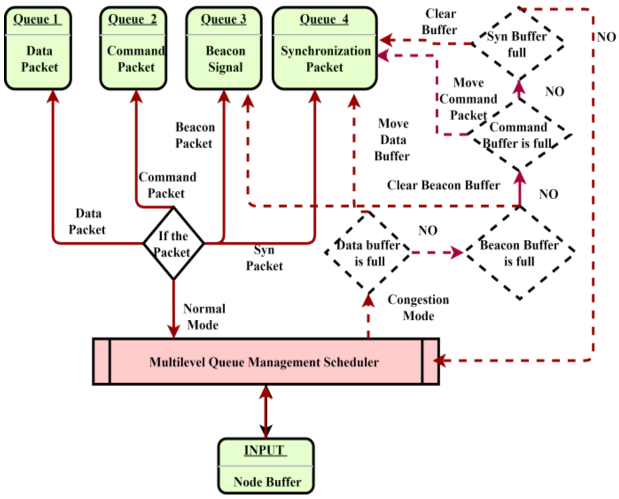


Fig. 2. Multi-level queue management.

The MQM Scheduling Research work is divided into four stages, as shown in Fig. 3. Stage 1 is packet processing, followed by scheduling the packets in the internal buffer queue in stage 2. Stages 3 and 4 are for congestion avoidance, with the scheduler analyzing the internal buffer size when the buffer reaches the threshold value, rescheduling the packet in any other queue temporality, and finally resuming the normal operation.

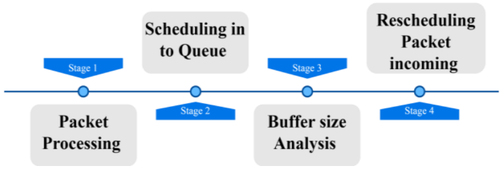


Fig. 3. Stages of research methods.

IV. Simulation

The proposed Multi-Level Queue Management (MLQM) technique will be simulated using Network Simulator (NS) 2.34, with internal parameter configurations detailed in Table II. The simulation involves implementing the MLQM algorithm in conjunction with the On-Demand Ad hoc Distance Vector (AODV) protocol. The process begins with 25 nodes, progressively increasing to 150 nodes in increments of 25 (i.e., 25, 50, 75, 100, 125, and 150). Packet traffic is modeled using a Poisson Traffic Model, which relies on Poisson distribution to simulate packet arrivals as a random process. The inter-arrival times of packets follow an exponential distribution, making this model particularly suitable for simulating real-world applications such as web browsing and file downloads.

The regular operation of the nodes is analyzed under two scenarios: without congestion and with induced congestion. Congestion was introduced across all queue categories, starting with data packets. To prevent network failure, the Multi-Level Queue Management (MLQM) algorithm redirected data to the beacon queue.

TABLE II. Simulation Value

Parameter	Value
Network simulator	NS 2.34
Physical Layer	IEEE 802.11a
Time defined for simulation	1000s
Transport Layer Protocol	TCP
Size of the network	500*500m
Total nodes	150
Node speed limit	2.5ms
Routing protocol	AODV
Packet size	512 bytes
Queue	FIFO
Queue buffer size	2000

Various performance metrics are employed to assess network performance under different traffic models. One key metric is throughput, which measures the rate of successful data transmission across the network, typically expressed in bits per second (bps). Higher throughput indicates greater network efficiency. Another crucial metric is end-to-end delay (latency), representing the time it takes for a packet to travel from source to destination. Lower latency is essential for real-time applications such as video conferencing. Buffer capacity refers to the memory allocated at each node for temporarily storing packets before transmission or forwarding. Optimizing energy consumption is critical to extending network lifetime, often defined by the time until the first node depletes its energy or a significant portion of the network becomes non-functional. Lastly, stabilization time refers to the period required for the network or a specific node to achieve a stable state following events such as network initialization, topology changes, or disruptions.

Congestion was initially triggered by transmitting multiple beacon signals, causing the queue to empty. Subsequently, congestion was introduced in the synchronization signal, which generates warnings about intruders or attackers in the Mobile Ad Hoc Network (MANET). Finally, during a command packet flooding scenario, the algorithm redirected subsequent command packets to the synchronization buffer to prevent network failure. Simulating congestion in a MANET involves adjusting various parameters to observe the network's response to increased traffic loads. This includes increasing active flows to saturate bandwidth, reducing inter-packet arrival times to create burst traffic patterns, and modifying buffer capacities to assess how smaller buffers lead to faster packet loss. Enhancing node mobility by increasing movement speeds can cause frequent link breaks, further contributing to congestion. Additionally, varying routing protocols can impact congestion management, while changing packet sizes affects bandwidth consumption and the likelihood of packet drops. Altering the network topology by adding or removing nodes creates localized congestion, and adjusting node service rates simulates different hardware capabilities. Network simulation tools like NS-3 allow for the configuration of these parameters, enabling detailed analysis of performance metrics such as packet loss, delay, and throughput to develop strategies for mitigating congestion's impact on network performance.

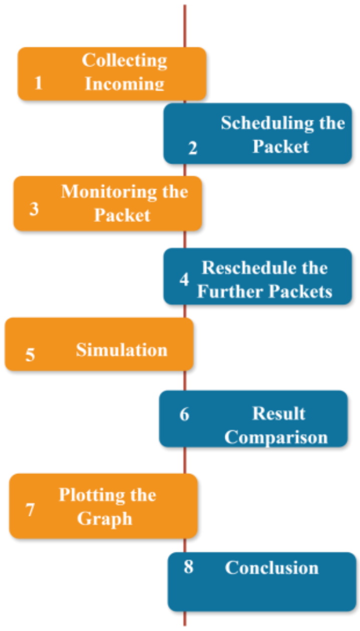


Fig. 4. Research work stages.

Fig. 4 illustrates the simulation setup, where steps 1–4 are executed by the Scheduler to handle incoming packets, process them, and reschedule as needed. Steps 5–8 represent the simulation outcomes using NS2.34, including the generation of the resultant graph and the presentation of key findings from the research.

V. Result and Analysis

The proposed Multilevel Queue Management (MQM) method was simulated using NS2.34, integrating the algorithm into the On-Demand AODV protocol, resulting in the MQM-AODV variant. For comparative analysis, three established congestion control algorithms were selected: ACO-AODV (Ant Colony Optimization-based AODV) [8], RL-AODV (Reinforcement Learning-based AODV) [24], and CATBOOST-AODV (CatBoost-based AODV) [20]. Each of these algorithms was implemented within the AODV protocol to evaluate their performance. The comparison was based on several key parameters: throughput, end-to-end delay, buffer capacity, network lifetime, stabilization time, multilayer queue scheduling time, and queue capacity under medium and heavy load conditions.

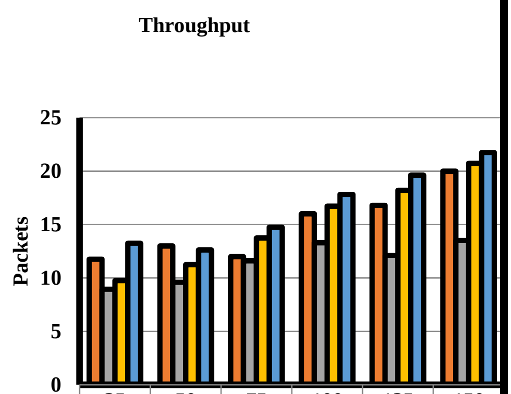
The choice of ACO-AODV, RL-AODV, and CATBOOST-AODV reflects their distinct strategies for congestion control and routing optimization in MANETs. ACO-AODV employs biologically inspired ant colony optimization, renowned for its effectiveness in finding near-optimal paths in dynamic, uncertain network environments. RL-AODV leverages reinforcement learning to dynamically adapt to changing network conditions by continuously learning from real-time network feedback, making it highly suitable for variable environments. CATBOOST-AODV applies gradient boosting techniques, offering a machine learning-driven approach that efficiently manages multiple variables and adapts to evolving network conditions. This diverse set of methodologies provides a robust benchmark for evaluating MQM-AODV, which introduces a multilevel queue management approach to enhance congestion control and routing efficiency.

1) Throughput

Throughput is defined as the maximum number of packets successfully received from the sender and can be calculated using Eq. (1). In the simulation, throughput values were evaluated by starting with 25 nodes and incrementally increasing the node count by 25 until reaching 150 nodes, with measurements taken every 20 milliseconds. The throughput of each algorithm was estimated under these conditions. The results indicate that ACO-AODV and RL-AODV protocols exhibit low throughput, CATBOOST-AODV achieves medium throughput, and MQM-AODV reaches high throughput, as shown in Fig. 5.

$$\text{Throughput} = \frac{\text{Packet Size}}{\text{Transmission time}} \tag{1}$$

Fig. 5 presents a comparison of network throughput for different routing protocols as the number of nodes increases from 25 to 150. The protocols compared are CATBOOST-AODV, ACO-AODV, RL-AODV, and MQM-AODV. Throughput, typically measured in terms of data successfully transmitted per unit time, is recorded for each protocol at each network size. Starting with CATBOOST-AODV, the throughput increases from 11.75 at 25 nodes to 20 at 150 nodes. There is a noticeable jump at the 100-node mark, after which the throughput continues to rise steadily. ACO-AODV shows a smaller range of throughput values compared to the other protocols, starting at 8.95 for 25 nodes and peaking at 13.5 with 150 nodes. It experiences a slight decline in throughput at 125 nodes, suggesting some variability as the network grows. RL-AODV exhibits a consistent upward trend, beginning with a throughput of 9.75 at 25 nodes and reaching 20.73 for 150 nodes. The protocol shows steady improvement as nodes increase, with a significant rise from 75 to 100 nodes, indicating that it handles network expansion effectively. Finally, MQM-AODV consistently achieves the highest throughput values across all node counts compared to the other methods. Starting at 13.25 with 25 nodes, it rises to a maximum of 21.73 at 150 nodes. This pattern suggests that MQM-AODV is particularly efficient in handling larger networks, maintaining a lead in throughput over the other protocols at each node level. The significant throughput gains of MQM-AODV (60.74% over ACO-AODV and 36.41% over RL-AODV) can be attributed to its enhanced packet categorization and buffer management strategy. By categorizing packets based on priority and type, MQM-AODV ensures that critical and time-sensitive packets are transmitted first, reducing the chances of congestion and packet drops. Additionally, the dynamic buffer allocation mechanism effectively prevents buffer overflow, maintaining steady throughput even under high network loads. This prioritization mechanism allows for efficient resource utilization, resulting in improved overall throughput. In summary, MQM-AODV outperforms the other routing protocols in terms of throughput across all network sizes. CATBOOST-AODV and RL-AODV also perform well, particularly in larger networks, while ACO-AODV has the lowest throughput values overall. The results indicate that MQM-AODV may be the most effective option for maximizing throughput as network size increases.



	25	50	75	100	125	150
CATBOOST-AODV	11.75	13	12	16	16.8	20
ACO-AODV	8.95	9.6	11.6	13.3	12.1	13.5
RL_AODV	9.75	11.25	13.75	16.72	18.2	20.73
MQM-AODV	13.25	12.63	14.75	17.82	19.62	21.73

Fig. 5. Throughput.

2) End to end delay

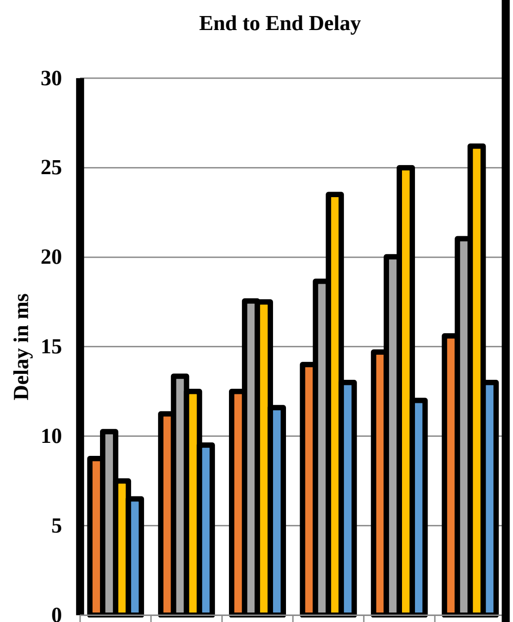
End-to-end latency is determined using Eq. (2) as the time difference between packets transmitted from the source and packet arrival at the destination.

$$\text{Endto End Delay} = \text{Total Data Packet} \times \begin{pmatrix} \text{Data Packet delayat destination} - \\ \text{Data packet delayfrom the Source} \end{pmatrix} \quad (2)$$

The delay value was calculated by altering the number of nodes in all three categories of the algorithm from 25 to 150. Maximum delay is reached by the RL-AODV protocol, medium delay is achieved by the ACO-AODV protocol; however, there is no difference in minimum delay between the CATBOOST protocol and the MQM protocol, as shown in Fig. 6. Different routing protocols CATBOOST-AODV, ACO-AODV, RL-AODV, and the proposed MQM-AODV method as the number of network nodes increases from 25 to 150. End-to-end delay refers to the total time taken for data packets to travel from the source to the destination, with lower values indicating faster performance. For CATBOOST-AODV, the end-to-end delay begins at 8.75 milliseconds with 25 nodes and gradually increases to 15.6 milliseconds with 150 nodes. This method shows a steady increase in delay as the network size grows, though it remains relatively moderate across all node counts. ACO-AODV exhibits the highest delay among the protocols, starting at 10.25 milliseconds for 25 nodes and reaching 21.03 milliseconds for 150 nodes. The delay rises significantly, especially as the network scales from 75 to 150 nodes, indicating that this protocol may not handle larger networks as efficiently as the others. RL-AODV shows a rapid increase in delay, beginning at a low of 7.5 milliseconds for 25 nodes but climbing to 26.2 milliseconds with 150 nodes. This steep rise, particularly after 75 nodes, suggests that RL-AODV may struggle with increased network sizes, as delays become more pronounced. The MQM-AODV protocol demonstrates the lowest end-to-end delay across all network sizes compared to the other methods. Starting at 6.5 milliseconds with 25 nodes, the delay remains relatively low as the network expands, reaching 13 milliseconds for 150 nodes. Notably, MQM-AODV even shows a slight improvement between 100 and 125 nodes, highlighting its efficiency in managing network delays. In summary, MQM-AODV achieves the lowest end-to-end delay across all node counts, demonstrating better scalability and lower latency than CATBOOST-AODV, ACO-AODV, and RL-AODV. The 35.47% reduction in end-to-end delay compared to ACO-AODV is primarily due to the multilevel queue scheduling approach in MQM-AODV. This strategy minimizes queuing delays by dynamically adjusting packet priorities based on current network conditions, ensuring timely packet forwarding. Furthermore, MQM-AODV employs adaptive routing decisions that optimize path selection under varying network loads, reducing delays caused by route rediscovery and packet retransmissions. Fig. 6 compares the end-to-end delay across different protocols, showing that RL-AODV exhibits moderate increases in delay, while ACO-AODV consistently experiences the highest delays. In contrast, MQM-AODV emerges as the most favorable protocol for minimizing latency in larger networks.

3) Buffer capacity

When the simulation begins, the buffer size is set at 2000 bytes. Half of the buffer size 1000 will be assigned to the Data packet Queue, 200 bytes to the beacon signal buffer queue, and 400 bytes to each of the Synchronization queue and Command Signal Queue. When the packet was transmitted, all of the algorithms began holding the packet in the buffer with respect to the kinds performed by the scheduler. Fig. 7 depicts the buffer size of several types of packets in the MQM-AODV protocol, indicating that the whole buffer appropriately holds the packets.



	25	50	75	100	125	150
CATBOOST-AODV	8.75	11.25	12.5	14	14.7	15.6
ACO-AODV	10.25	13.35	17.55	18.65	20.02	21.03
RL_AODV	7.5	12.5	17.5	23.5	25	26.2
MQM-AODV	6.5	9.5	11.6	13	12	13

Fig. 6. End to end delay.

Fig. 7 presents data on the buffer capacity requirements for various types of network signals and packets, Synchronization Signal, Beacon Signal, Control Packet, and Data Packet, as the number of nodes increases from 25 to 150. Buffer capacity indicates the maximum storage space required for each signal type within the network, which helps in understanding the data handling needs as the network grows. For the Synchronization Signal, the buffer capacity starts at 200 for 25 nodes and increases to 400 for 150 nodes. This steady rise shows that as the network size grows, the need for buffer space to handle synchronization traffic also increases proportionately. The Beacon Signal requires less buffer capacity compared to the other signal types, beginning at 50 for 25 nodes and reaching 200 for 150 nodes. The buffer capacity increases at a moderate rate, suggesting that beacon signals place a smaller demand on network storage resources, even as the number of nodes grows.

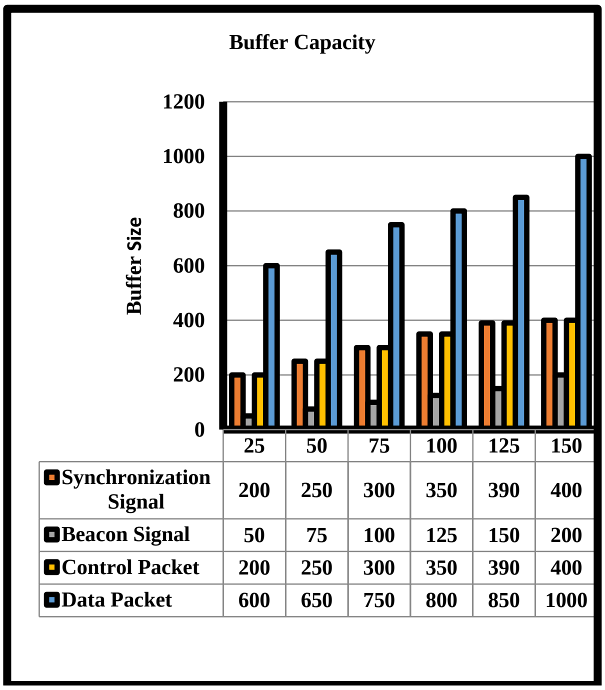


Fig. 7. Buffer size.

The Control Packet has buffer capacity values identical to the Synchronization Signal, starting at 200 for 25 nodes and peaking at 400 for 150 nodes. This similarity suggests that control and synchronization traffic place similar storage demand on the network, likely due to their interrelated roles in network management. Lastly, the Data Packet requires the largest buffer capacity among all the categories, starting at 600 for 25 nodes and increasing to 1000 for 150 nodes. MQM-AODV excels in buffer management through its adaptive queue threshold mechanism, which dynamically adjusts buffer thresholds based on real-time network traffic patterns. This prevents excessive buffer utilization and packet drops under high loads. In contrast, other protocols often use static thresholds, which are less effective in handling fluctuating network conditions. MQM-AODV's proactive monitoring and efficient buffer utilization ensure high performance even in congested scenarios. This significant demand reflects the high volume of data traffic and indicates that the network needs considerable buffer space to handle data packet traffic as the number of nodes increases. In summary, buffer capacity requirements rise with the number of nodes across all signal types. Data Packets require the buffer space, followed by Synchronization and Control Packets, which have identical needs, while Beacon Signals have the smallest requirements. This pattern highlights the need for substantial data packet storage as networks expand, underscoring the importance of buffer management in accommodating larger networks.

4) Life time

The life time of the nodes is defined as how long the node can be stable for packet transmission. ACO-AODV has a shorter life time than the others, but RL-AODV and CATBOOST-AODV have a longer life time than the others, as shown in Fig. 8. Fig. 8 compares the network lifetime across different routing protocols—CATBOOST-AODV, ACO-AODV, RL-AODV, and the proposed MQM-AODV method—as the number of nodes increases from 25 to 150. Network lifetime reflects the duration over which the network remains operational, with higher values indicating a longer-lasting network. CATBOOST-AODV starts with a network lifetime of 11 units at 25 nodes, which increases to 20 units at 150 nodes. The protocol exhibits a dip at 75 nodes, dropping to 12 units before increasing again as the network grows. This fluctuation suggests that CATBOOST-AODV's performance may vary depending on network size.

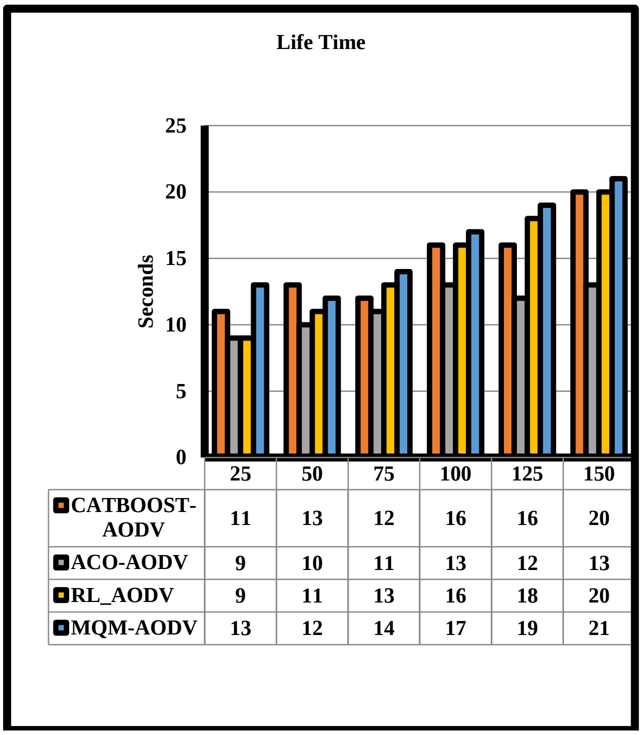


Fig. 8. Life time.

ACO-AODV has the shortest network lifetime across all node counts, starting at 9 units for 25 nodes and reaching only 13 units for 150 nodes. The lifetime peaks at 13 units around the 100-node mark but then declines slightly, indicating that ACO-AODV may be less efficient in maintaining network sustainability as the network grows larger. RL-AODV demonstrates a more consistent increase in network lifetime, starting at 9 units for 25 nodes and reaching 20 units for 150 nodes. The lifetime values improve steadily, especially from 75 nodes onward, indicating that RL-AODV manages network resources better as the network scales.

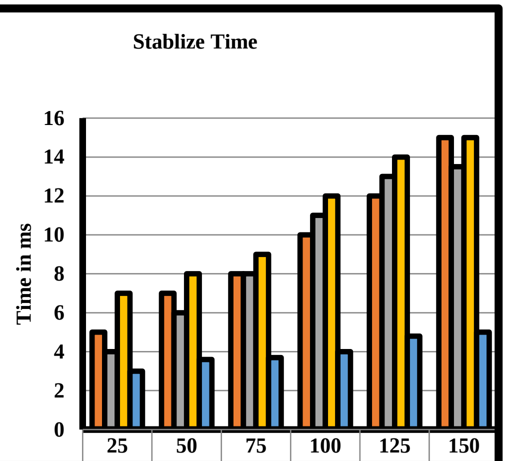
The MQM-AODV method achieves the highest lifetime values among the protocols, starting at 13 units for 25 nodes and peaking at 21 units for 150 nodes. This trend suggests that MQM-AODV is the most effective at conserving network resources, allowing for a longer operational period as the number of nodes increases. In summary, MQM-AODV outperforms CATBOOST-AODV, ACO-AODV, and RL-AODV in terms of network lifetime across all node counts, demonstrating its efficiency and scalability. While RL-AODV and CATBOOST-AODV also show improvement with larger networks, ACO-AODV consistently has the shortest lifetime, making MQM-AODV the most favorable choice for extending network longevity.

MQM-AODV significantly enhances network lifetime by leveraging its multilevel queue management strategy, which reduces packet losses and retransmissions. This approach prioritizes packet handling based on traffic type and urgency, ensuring that critical packets are transmitted with minimal delay. By reducing packet drops and retransmission frequency, MQM-AODV prevents excessive energy consumption across nodes. Additionally, fewer retransmissions result in lower network congestion, reducing the overall energy drain and extending the operational lifespan of nodes. Consequently, this energy-efficient handling of network traffic contributes to prolonged network stability and lifetime, especially in scenarios with high data loads or fluctuating traffic conditions. This mechanism ensures that MQM-AODV is more resilient and sustainable than traditional routing protocols under varying network demands.

5) Stabilize time

The stabilize time is introduced to determine the stability of the internal nodes, when there is congestion. While there is congestion in the network, MQM-AODV takes 3 ns to 5 ns to stabilize it. Fig. 9 shows that CATBOOST-AODV took 5 ns to 15 ns, ACO-AODV took 4ns to 13.5ns, and RL-AODV took 7 ns to 15 ns. Fig. 9 compares the stabilization time for different routing protocols—CATBOOST-AODV, ACO-AODV, RL-AODV, and the proposed MQM-AODV method—as the number of nodes increases from 25 to 150. Stabilization time refers to the duration required for the network to reach a stable state after changes, with shorter times indicating quicker stabilization. Starting with CATBOOST-AODV, the stabilization time begins at 5 units for 25 nodes and increases steadily to 15 units for 150 nodes. This gradual rise indicates that as the network grows, CATBOOST-AODV requires more time to stabilize, particularly in larger networks. ACO-AODV shows a similar trend, with stabilization time starting at 4 units for 25 nodes and increasing to 13.5 units for 150 nodes. Although ACO-AODV has slightly shorter times at lower node counts, it surpasses CATBOOST-AODV around the 100-node mark, indicating a longer stabilization time as the network expands.

RL-AODV exhibits the highest stabilization times overall, beginning at 7 units with 25 nodes and reaching 15 units for 150 nodes. The increase in stabilization time is steady and indicates that RL-AODV may face challenges in achieving stability quickly as the network size grows. In contrast, the MQM-AODV method maintains the shortest stabilization times across all network sizes, starting at 3 units for 25 nodes and rising to only 5 units for 150 nodes. This demonstrates MQM-AODV's efficiency in reaching network stability quickly, even as the network scales up. Notably, MQM-AODV's stabilization time increases only slightly with the number of nodes, highlighting its resilience and adaptability to changes in network size. In summary, MQM-AODV consistently outperforms CATBOOST-AODV, ACO-AODV, and RL-AODV in terms of stabilization time, providing the fastest time to reach a stable state. While CATBOOST-AODV and ACO-AODV show moderate increases in stabilization time, RL-AODV exhibits the highest values, making MQM-AODV the preferred choice for networks where rapid stabilization is essential.



	25	50	75	100	125	150
CATBOOST-AODV	5	7	8	10	12	15
ACO-AODV	4	6	8	11	13	13.5
RL_AODV	7	8	9	12	14	15
MQM-AODV	3	3.6	3.7	4	4.8	5

Fig. 9. Stabilize time.

MQM-AODV minimizes stabilization time by employing its multilevel queue management strategy, which effectively reduces packet losses and retransmissions. The system prioritizes packets based on their urgency and type, ensuring that high-priority packets are transmitted promptly and reliably. This efficient handling reduces the likelihood of route flapping and minimizes the time required for the network to recover from disruptions. Furthermore, by lowering the frequency of retransmissions and congestion-related delays, MQM-AODV maintains consistent and stable routing paths, reducing the need for frequent route recalculations. As a result, the network stabilizes faster, even under varying traffic conditions, leading to a more reliable and resilient communication system. This mechanism not only enhances the overall network stability but also contributes to a more sustainable network lifetime by conserving energy and resources.

A. Multilevel Queue Scheduling Time

The conventional queue method is contrasted with multilayer queue scheduling. The scheduler in MQM-

AODV is in charge of assigning packets to any of the Queues. The time required to classify the packet and keep it in a queue is estimated. MQM-AODV requires an additional 0.5 ns for packet categorization when compared to other AODV protocols, as shown in Fig. 10. Scheduling is additional load therefore this time is allowed in positively.

Fig. 10 compares the scheduling time for various routing protocols CATBOOST-AODV, ACO-AODV, RL-AODV, and the proposed MQM-AODV method as the number of nodes increases from 25 to 150. Scheduling time refers to the duration required to organize and manage data transmission in the network, with shorter times indicating more efficient scheduling. Starting with CATBOOST-AODV, the scheduling time begins at 2 units for 25 nodes and gradually increases to 12 units for 150 nodes. This consistent rise suggests that CATBOOST-AODV experiences growing complexity in scheduling as the network size expands. ACO-AODV starts with a slightly higher scheduling time of 2.6 units for 25 nodes, increasing to 12 units at 150 nodes. The increase in scheduling time is somewhat more gradual than that of CATBOOST-AODV, particularly at larger node counts, indicating a slightly better performance in managing scheduling demands.

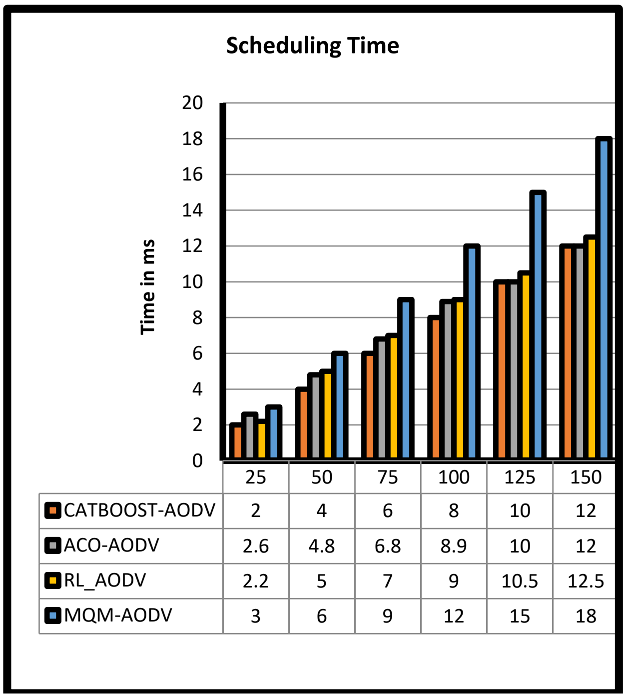


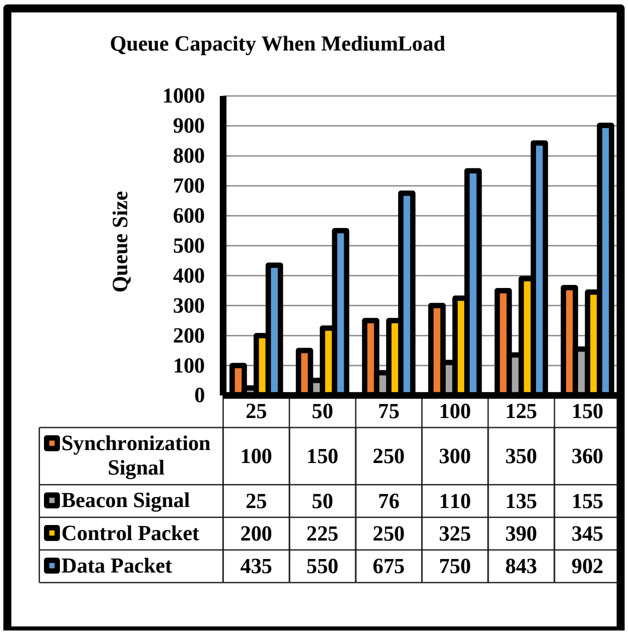
Fig. 10. Scheduling time.

RL-AODV exhibits an initial scheduling time of 2.2 units at 25 nodes, rising to 12.5 units for 150 nodes. Similar to ACO-AODV, RL-AODV shows a gradual increase in scheduling time, suggesting that it also handles scheduling effectively as the network scales. In contrast, the MQM-AODV method has higher scheduling times across all node counts, starting at 3 units for 25 nodes and increasing to 18 units for 150 nodes. While MQM-AODV requires more scheduling time due to its multilevel queue management, this overhead is minimal when weighed against the performance benefits in terms of throughput, delay, and stability. However, to address this potential issue in real-world applications, future enhancements could focus on optimization techniques such as parallel scheduling algorithms or lightweight priority management systems. These improvements could reduce the scheduling overhead while maintaining or even enhancing performance. This indicates that MQM-AODV requires more time for scheduling compared to the other protocols, particularly in larger networks. The increase in scheduling time is notable, suggesting that MQM-AODV may implement more complex scheduling algorithms to optimize network performance. In summary, while MQM-AODV shows the highest scheduling times among the compared methods, the other protocols CATBOOST-AODV, ACO-AODV, and RL-AODV demonstrate lower scheduling times, indicating their efficiency in handling data transmission scheduling. Although MQM-AODV's scheduling times are higher, this may be attributed to its comprehensive approach to managing network resources.

B. Queue Capacity in Medium Load

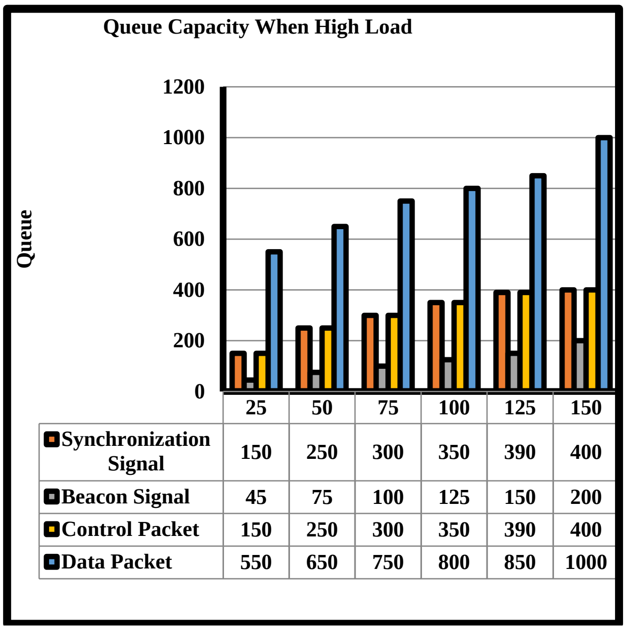
When the Queue has a medium load, the chance of congestion and scheduling work is normal, which has no effect on the MANET's performance. The table presents data across various types of network signals and packets for different numbers of nodes. For each category, Synchronization Signal, Beacon Signal, Control Packet, and Data Packet, the values indicate the amount of signal or packet traffic as the number of nodes increases from 25 to 150. Starting with the Synchronization Signal, the traffic initially increases with the number of nodes, beginning at 100 for 25 nodes and steadily rising to 360 for 150 nodes. This shows a generally upward trend with some fluctuation. Next, the Beacon Signal starts at a relatively low value of 25 with 25 nodes, and this signal traffic grows as nodes increase. It reaches 155 at 150 nodes, demonstrating a smoother increase compared to the other types of signals and packets. The Control Packet also exhibits growth as the number of nodes increases, starting at 200 for 25 nodes and peaking at 390 for 125 nodes before decreasing slightly to 345 with 150 nodes. This pattern suggests that control packet traffic may not increase linearly with the number of nodes. Lastly, the Data Packet traffic shows a consistent rise as the number of nodes grows, starting at 435 for 25 nodes and reaching 902 for 150 nodes. This increase is the most significant among the different categories, reflecting that data packet traffic scales directly with the network size. Overall, as the number of nodes in the network increases, all categories of traffic tend to rise, though at varying rates, with data packets showing the steepest growth. Therefore, all Queues hold as many arrival packets as feasible, as illustrated in Fig. 11.

When there is a high load in the Queue, the probability of congestion and scheduling work increases, which affects the MANET's performance. The data provides insights into the traffic generated by different types of signals and packets in a network as the number of nodes increases from 25 to 150. The traffic is broken down into four categories: Synchronization Signal, Beacon Signal, Control Packet, and Data Packet. Starting with the Synchronization Signal, the traffic rises as the node count grows, beginning at 150 for 25 nodes and increasing to 400 for 150 nodes. The pattern here shows a steady increase, suggesting that synchronization traffic scales directly with the number of nodes. The Beacon Signal also shows an upward trend, starting at a lower value of 45 with 25 nodes and reaching 200 for 150 nodes. The increments are more pronounced in this category compared to the Synchronization Signal, indicating that beacon traffic becomes more substantial as the network grows larger. For Control Packets, the traffic mirrors the pattern of the Synchronization Signal, starting at 150 for 25 nodes and peaking at 400 for 150 nodes. This suggests that control packet traffic scales proportionately with synchronization traffic, potentially due to similar network demands. Finally, the Data Packet traffic demonstrates the most substantial increase among all categories, starting at 550 with 25 nodes and reaching a peak of 1000 at 150 nodes. The steady rise reflects the growing demand for data transmission as the network size expands, indicating that data traffic may be the most resource-intensive aspect of the network as nodes are added. In summary, as the network size increases, all types of traffic—Synchronization, Beacon, Control, and Data Packets—display an upward trend, with Data Packet traffic growing most significantly. This pattern highlights the increased communication load in larger networks, where synchronization, control, and data transmission demands scale with the number of nodes. So, all the Queues hold the packets that are scheduled by the scheduler, in some cases the beacon queue makes clear to hold data packets and Command queue holds data packets, and Synchronous queue makes alert to the intruder or attacker based on the arrival packets and buffer capacity shown in Fig. 12.



	25	50	75	100	125	150
Synchronization Signal	100	150	250	300	350	360
Beacon Signal	25	50	76	110	135	155
Control Packet	200	225	250	325	390	345
Data Packet	435	550	675	750	843	902

Fig. 11. Queue capacity when the medium load queue capacity in high load.



	25	50	75	100	125	150
Synchronization Signal	150	250	300	350	390	400
Beacon Signal	45	75	100	125	150	200
Control Packet	150	250	300	350	390	400
Data Packet	550	650	750	800	850	1000

Fig. 12. Queue capacity when high load.

C. Comparison of Proposed work Compared with Related Work

The comparative analysis of different routing algorithms, including ACO-AODV, RL-AODV, CATBOOST-AODV, and MQM-AODV, highlights their respective methods, overhead, background operations, limitations, and performance metrics as shown in Table III. ACO-AODV employs the Ant Colony Optimization algorithm, which incurs additional overhead for maintaining the algorithm along with AODV operations. It involves creating a cluster head for background operations but struggles with lower critical levels, resulting in low throughput, high end-to-end delay, low buffer capacity, shorter lifetime, and higher stabilization and scheduling times. The queue capacity remains tolerable under medium load but experiences delay under maximum load conditions.

RL-AODV integrates a Markov decision process with reinforcement learning, facing similar overhead issues. It adjusts traffic based on channel conditions, but like ACO-AODV, it faces challenges with congestion at lower critical levels. This method achieves moderate throughput and end-to-end delay, low buffer capacity, moderate lifetime, high stabilization and scheduling times, tolerable queue capacity under medium load, but delays under maximum load.

CATBOOST-AODV utilizes a decision-tree-based CatBoost algorithm for congestion control, with additional overhead requirements. Its background operation involves predicting packet transmission based on network activity, although decision tree reliance leads to packet queue delays. The algorithm provides moderate throughput, moderate end-to-end delay, low buffer capacity, moderate lifetime, high stabilization time, and moderate scheduling time. The queue capacity is tolerable under medium load but delayed under maximum load conditions.

In contrast, MQM-AODV employs a Multilevel Queue Management method, eliminating additional overhead by leveraging different queues. Unlike the other methods, it has no background operations or specific limitations due to its dynamic queue management nature. It delivers high throughput, low end-to-end delay, high buffer capacity, significantly longer lifetime, low stabilization and scheduling times, tolerable queue capacity under medium load, and tolerable queue capacity under maximum load. Thus, MQM-AODV outperforms the other methods in several key metrics, particularly in managing queue performance and network stability.

The performance evaluation demonstrates that MQM-AODV achieves lower latency and energy consumption compared to machine learning-based strategies like reinforcement learning. These improvements are attributed to its lightweight queue management mechanism and cross-layer optimization, which reduce computational overhead and adapt effectively to dynamic conditions. While the MQM-AODV protocol achieves significant improvements in performance metrics like packet delivery ratio, energy consumption, and routing overhead, we acknowledge that the introduction of a multilevel queue management approach does come with potential trade-offs. Specifically, the management of multiple queues and the associated scheduling policies introduce additional computational overhead, which could impact processing delay. The dynamic categorization and prioritization of packets across different queues require constant monitoring of network conditions and updating of packet priorities, which can lead to increased processing time compared to simpler routing algorithms. Additionally, the implementation of complex scheduling mechanisms, such as Weighted Fair Queuing and Priority Queuing, may introduce scheduling overhead, particularly in high-density or high-mobility environments. To evaluate these trade-offs, we conducted a performance analysis focusing on the computational complexity and processing delay, alongside the primary performance metrics. The results indicate that while MQM-AODV does introduce some overhead, its benefits in terms of network performance and energy efficiency outweigh the additional resource consumption, making it a viable solution for real-time MANET applications.

TABLE III. WORK COMPARED TO THE PREVIOUS WORKS

Related Work	**ACO-AODV [8]**	**RL-AODV [24]**	**CATBOOST-AODV [20]**	**MQM-AODV**
Method Used	Ant Colony Optimization algorithm	a Markov decision process with reinforcement learning	a machine learning-based congestion control strategy CatBoost algorithm	Multilevel Queue Management method
Over head	Additional overhead for maintain the Algorithm along with AODV operation	Additional overhead for maintain the Algorithm along with AODV operation	Additional overhead for maintain the Algorithm along with AODV operation	NO Additional overhead for maintain the Algorithm along with AODV operation, since it has relay on the different queue.
Background work	Created Cluster Head	Adjusted Traffic Based On Current Channel Conditions	Predict Packet Transmission Based On Network Activity	No Background Operation
drawback	lower critical levels remains a challenge	maintaining congestion at lower critical levels remains a challenge	decision tree's role caused packet queue delays	No specific limitation since the Dynamic nature of queue management
Throughput	Low	Moderate	Moderate	High
End To End Delay	High	High	Moderate	Low
Buffer Capacity	Low	Low	Low	High
Life Time	Less	Moderate	Moderate	Much
Stabilize Time	High	High	High	Low
Scheduling Time	More	More	Moderate	Less
Queue Capacity When Medium Load	Tolerate	Tolerate	Tolerate	Tolerate
Queue CapacityWhen Max load	Delay In Queue	Delay In Queue	Delay In Queue	Tolerate

While MQM-AODV demonstrated significant improvements in routing efficiency, we observed an increase in computational overhead, primarily due to the multilevel queue management process. This resulted in higher memory consumption and processing delays compared to ACO-AODV and RL-AODV, which rely on simpler decision-making mechanisms. However, the performance benefits of MQM-AODV may outweigh these trade-offs in environments where routing efficiency and adaptability are critical. For real-time applications, the scheduling overhead introduced by MQM-AODV must be carefully balanced against its routing benefits.

Although TCP-based congestion control and other transport-layer protocols focus on managing congestion after it occurs, MQM-AODV proactively manages routing decisions to prevent congestion in the first place. By leveraging multilevel queue management, MQM-AODV can reduce routing overhead and minimize congestion-related delays, making it particularly effective in real-time MANET applications.

VI. CONCLUSION AND FUTURE WORK

This research paper focuses on giving a solution to congestion control in the MANET transport layer using the Multilevel Queue Management Algorithm. In conclusion, the proposed queue management system demonstrates a robust approach to handling network congestion by categorizing traffic into beacon signals, control signals, synchronization signals, and data packets. Through dynamic redirection of data packets during congestion and prioritization of critical traffic, the system ensures efficient utilization of network resources while maintaining high throughput and low latency. Compared to previous methods, such as ACO-AODV, RL-AODV, and CATBOOST-AODV, the Multilevel Queue Management (MQM-AODV) approach eliminates additional overhead, enhances performance metrics like buffer capacity and lifetime, and addresses limitations effectively. The MQM-AODV schedules packets into the queue division based on the arrival of incoming packets. When compared to existing congestion control algorithms added to the AODV protocol called ACO-AODV, RL-AODV, and CATBOOST-AODV, the simulation result of MQM-AODV produced better throughput, less end to end delay, flexible buffer capacity, maximum life time, strong stabilize time in congestion, and quick multilevel queue scheduling. The proposed MQM-AODV method demonstrates significant improvements across various performance parameters compared to existing protocols. In terms of throughput, MQM-AODV shows an increase of approximately 12.77% over CATBOOST-AODV at 25 nodes, although it experiences a slight decrease of 2.85% at 50 nodes. Comparatively, it outperforms ACO-AODV by about 48.09% at 25 nodes and by 60.74% at 150 nodes. Against RL-AODV, the improvements range from 4.83% to 36.41%, highlighting MQM-AODV's consistent advantage in maximizing data transmission. Regarding end-to-end delay, MQM-AODV shows reductions of around 25.71% compared to CATBOOST-AODV at 25 nodes and a notable improvement of 35.47% against ACO-AODV at 50 nodes, while also outperforming RL-AODV by up to 31.58%. The buffer capacity remains constant across all methods, indicating that resource management is consistent. MQM-AODV also achieves a remarkable increase in network lifetime, outperforming CATBOOST-AODV by 18.18% at 25 nodes and ACO-AODV by 15.38% at 100 nodes, while maintaining the highest stabilization performance, reducing stabilization time by up to 58.33% compared to CATBOOST-AODV. Lastly, although MQM-AODV requires more scheduling time, it represents an improvement over existing methods when considering its overall performance benefits. These results position MQM-AODV as a robust choice for networks requiring high performance and reliability. In future this approach can be improved to detect the flooding attacker by monitoring the arriving packet variants. This makes it particularly well-suited for modern network environments, including IoT ecosystems, AI-driven networks, and future 6G applications, where adaptability and efficiency are paramount. Also, MQM-AODV represents a significant advancement in congestion control for MANET-IoT environments by integrating dynamic multi-queue management with cross-layer feedback. This approach uniquely balances efficiency, adaptability, and resource-awareness, outperforming existing strategies in both performance and scalability.

In this study, we compared MQM-AODV with ACO-AODV**,** RL-AODV, and CATBOOST-AODV to evaluate its performance in MANETs. While MQM-AODV provided significant routing improvements, its computational overhead and scheduling complexities present potential challenges for real-time applications. However, in scenarios where efficient routing and adaptability are paramount, MQM-AODV may still offer a valuable approach, provided the system can handle its increased resource demands

VII. Future Work

While the current evaluation of MQM-AODV shows promising improvements in performance and efficiency, several aspects warrant further investigation. One of the key areas for future work involves addressing the scalability of the protocol, particularly in highly dynamic MANET environments. As MANETs continue to evolve, the mobility of nodes and the complexity of traffic patterns may increase, potentially affecting the performance of MQM-AODV. Exploring how the protocol can handle higher node mobility, more complex traffic scenarios, or heterogeneous MANET setups where nodes vary in terms of processing power, energy resources, or communication capabilities could significantly enhance its applicability in real-world situations. Additionally, expanding on future work related to detecting and mitigating flooding attacks would improve the security of MQM-AODV, ensuring the robustness of the protocol against potential DoS (Denial of Service) and other malicious activities in dynamic network conditions. By addressing these challenges, MQM-AODV could evolve into a more comprehensive solution for large-scale MANETs.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

S. Hemalatha contributed to the overall research design, guiding the theoretical framework, methodology of the study and wrote the majority of the manuscript; Khadri Syed Faizz Ahmad performed the data collection and assisted in analyzing the results; Nripendra Narayan Das led the analysis of the data, interpreted the results; R. V. V. Krishna provided valuable insights into the data interpretation and helped refine the conclusions; Sathya reviewed the methodology and provided technical input on algorithm development; Radha Mothukuri assisted in literature review and helped finalize the manuscript; all authors approved the final version of the manuscript.

References

[1] S. Mohan and P. Vimala, "Enhancing congestion control and QoS scheduling using novel rate aware-neurofuzzy algorithm in mane," *Journal of Theoretical and Applied Information Technology*, vol.101, no. 11, 14819–4832, 2023.

[2] A. Sofyan, V. Suryani, and H. H. Nuha, "Performance comparison of TCP AFW and TCP FeW in IEEE 802.11 wireless networks: A

simulation study," *Intl. Journal on ICT*, vol. 9, no. 1, pp. 86–94, June 2023.

[3] L. K. Ouladdjedid, C. T. Calafate, C. A. Kerrache, and Y. Guellouma, "Analysis of congestion control mechanisms for cooperative awareness in IoV environments," *Electronics*, vol. 12, p. 512, 2023.

[4] B. Atousa and R. Mehri, "A survey on congestion control protocols in wireless sensor networks," *International Journal of Wireless Information Networks*, vol. 27, 2020.

[5] A. Jan, M. Jan, S. Alam, M. Akhunzada, *et al.*, "A comprehensive analysis of congestion control protocols in wireless sensor networks," *Mobile Networks and Applications*, vol. 23, 2018.

[6] S. Kankane, A. Jhapate, and R. Shrivastava, "Light load path selection techniques for control congestion in MANET (ENBA)," *International Journal of Advanced Computer Technology*, vol. 11, no. 4, pp. 1–7, 2022.

[7] A. Yazdinejad, S. Kavei, and S. Razaghi, "Increasing the performance of reactive routing protocol using the load balancing and congestion control mechanism in MANET," *Computational Intelligence in Electrical Engineering (CIEE)*, vol. 2, no. 1, 2019.

[8] S. S. Sefati and R. Z. Farkhady, "Routing and congestion control in wireless sensor networks using the ant colony optimization algorithm and the Tabu search algorithm," *Research Square*, 2023.

[9] N. Ma, J. Lee, J. Li, and Y. Liu, "Improvement and implementation of an adaptive congestion DSR routing algorithm on chip," *Research Square*, 2023.

[10] M. Ganesha and S. Dixit, "Efficient traffic management: Congestion control and load balancing in mobile ad hoc networks," *Journal of Critical Reviews*, vol. 7, no. 11, 2020.

[11] S. L. Yadav and R. L. Ujjwal, "Mitigating congestion in wireless sensor networks through clustering and queue assistance: A survey," *J. Intell. Manuf.*, vol. 32, no. 8, pp. 2083–2098, Dec. 2021.

[12] N. Jagadev, B. K. Pattanayak, A. Habboush, B. M. Elzaghmouri, and M. A. A. Shugran, "Adaptive bandwidth management model for wireless mobile ad-hoc network," *Int. J. Comput. Netw. Commun.*, vol. 15, no. 1, Jan. 2023.

[13] N. Thrimoorthy, S. Reddy, D. Chandramma, and S. Unnikrishnan Vanitha, "Ornstein uhlenbeck cache obliviousness neural congestion control in wireless network for IoT data transmission," *Int. J. Comput. Netw. Appl.*, vol. 10, no. 1, 2023.

[14] A. A. Rezaee and F. Pasandideh, "A fuzzy congestion control protocol based on active queue management in wireless sensor networks with medical applications," *Wirel. Pers. Commun.*, vol. 98, no. 1, pp. 815–842, Jan. 2018.

[15] M. Noussaiba, A. R. Razaque, and R. Rahal, "Heterogeneous algorithm for efficient path detection and congestion avoidance for a vehicular management system," *Sensors*, vol. 23, no. 12, p. 5471, 2023.

[16] H. Parsavand and A. Ghaffari, "Controlling congestion in wireless sensor networks through imperialist competitive algorithm," *Wireless Personal Communications*, vol. 101, no. 2, pp. 1123–1142, 2018.

[17] A. Hamamreh, "SDCM: Secure dynamic end-to-end congestion avoidance protocol," *Journal of Theoretical and Applied Information Technology*, vol. 101, no. 11, p. 4832, June 15, 2023.

[18] V. Narawade and U. D. Kolekar, "ACSRO: Adaptive cuckoo search-based rate adjustment for optimized congestion avoidance and control in wireless sensor networks," *Alexandria Engineering Journal*, vol. 57, no. 1, pp. 131–145, 2018.

[19] S. Singha, "Design of an effective congestion control routing protocol for mobile-Ad-Hoc network," doctoral dissertation, Midnapore, West Bengal, India, 2021.

[20] J. P. Astudillo León, L. J. Cruz Llopis, and F. J. R. Novella, "A machine learning based distributed congestion control protocol for multi-hop wireless networks," *Computer Networks*, vol. 231, 2023.

[21] V. Sharma and S. Bhattacharya, "Issues and challenges in mobile ad hoc networks: A comprehensive review," *Int. J. Comput. Appl.*, vol. 180, no. 5, pp. 29–36, 2018.

[22] R. Patil and R. Pujari, "Key management techniques in mobile ad hoc networks: A survey," *Int. J. Comput. Sci. Inf. Secur.*, vol. 17, no. 6, pp. 125–131, 2019.

[23] M. Prabhu and R. Rajendran, "Cluster-based key management for mobile ad hoc networks," *Int. J. Comput. Netw. Commun. Secur.*, vol. 5, no. 4, pp. 72–78, 2017.

[24] X. Liu, B. S. Amour, and A. Jaekel, "A reinforcement learning-based congestion control approach for V2V communication in VANET," *Appl. Sci.*, vol. 13, no. 6, p. 3640, 2023.

[25] H. Duan, Z. Li, and Y. Zhang, "Multiobjective reinforcement learning for traffic signal control using vehicular Ad Hoc network," *EURASIP J. Adv. Signal Process.*, vol. 2010, 2010.

[26] M. M. Karthikeyan and M. A. Christy, "Proposed Hybrid Congestion Control Algorithm (HCCA) using mobile adhoc network," *Partners Universal Int. Res. J.*, vol. 2, no. 1, pp. 35–46, 2023.

[27] A. C. Onuora, E. E. Essien, and F. U. Ogban, "An adaptive hybrid routing protocol for efficient data transfer and delay control in mobile ad hoc network," *Int. J. Eng. Trends Technol.*, vol. 71, no. 5, pp. 251–258, May 2023.

[28] H. K. Haile, K. J. Grinnemo, S. Ferlin, P. Hurtig, and A. Brunström, "End-to-end congestion control approaches for high throughput and low delay in 4G/5G cellular networks," *Comput. Netw.*, vol. 186, pp. 1–22, 2021.

[29] Y. Jiang, H. Sun, and M. Yang, "AODV-EOCW: An energy-optimized combined weighting AODV protocol for mobile ad hoc networks," *Sensors*, vol. 23, no. 15, p. 6759, 2023.

[30] F. Ren, A. Hawbani, and Y. Al-Sharabi, "Energy optimized congestion control-based temperature aware routing algorithm for software defined wireless body area networks," *IEEE Access*, vol. 8, pp. 41085–41099, 2020.

[31] R. Saraswathi, J. Srinivasan, and S. Aruna, "An energy efficient routing protocol and cross layer-based congestion detection using Hybrid Genetic Fuzzy Neural Network (HGFNN) model for MANET," *J. Algebraic Stat.*, vol. 13, no. 2, pp. 1007–1019, 2022.

[32] V. K. Quy, D. V. Anh, N. T. Ban, N. M. Quy, and G. Jeon, "A cross-layer routing approach for MANET-assisted IoT applications," *Trans. Emerg. Telecommun. Technol.*, vol. 35, no. 4, 2024.

[33] A. Kumar and P. Sharma, "Machine learning-based security solutions for mobile ad hoc networks: A survey," *J. Netw. Comput. Appl.*, vol. 154, 102553, 2020.

[34] V. Sharma and S. Bhattacharya, "Issues and challenges in mobile ad hoc networks: A comprehensive review," *Int. J. Comput. Appl.*, vol. 180, no. 5, pp. 29–36, 2018.

[35] R. Patil and R. Pujari, "Key management techniques in mobile ad hoc networks: A survey," *Int. J. Comput. Sci. Inf. Secur.*, vol. 17, no. 6, pp. 125–131, 2019.

[36] M. Prabhu and R. Rajendran, "Cluster-based key management for mobile ad hoc networks," *Int. J. Comput. Netw. Commun. Secur.*, vol. 5, no. 4, pp. 72–78, 2017.

[37] M. Y. Suraki, A. T. Haghighat, and M. Gholipour, "FCLCC: Fuzzy cross-layer congestion control in mobile ad hoc networks," *IJCSNS Int. J. Comput. Sci. Netw. Secur.*, vol. 18, no. 1, p. 155, 2018.

[38] Kalaikumar and E. Baburaj, "Fuzzy enabled congestion control by cross layer protocol utilizing OABC in WSN: Combining MAC, routing, non-similar clustering and efficient data delivery," *Wirel. Netw.*, vol. 26, no. 2, pp. 1085–1103, Feb. 2020.

[39] M. Gholipour, A. T. Haghighat, and M. R. Meybodi, "Hop-by-hop traffic-aware routing to congestion control in wireless sensor networks," *J. Wireless Com. Network*, 2015, p. 15, 2015.

[40] H. Parsavand and A. Ghaffari, "Controlling congestion in wireless sensor networks through imperialist competitive algorithm," *Wireless Pers. Commun.*, vol. 101, no. 2, pp. 1123–1142, 2018.