Journal of Theoretical and Applied Information Technology
15th May 2024. Vol.102. No 9

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ENERGY EFFICIENT ROUTING IN QUANTUM FLYING AD HOC NETWORK (Q-FANET) USING MAMDANI FUZZY INFERENCE ENHANCED DIJKSTRA'S ALGORITHM (MFI-EDA)

S. P. GEETHA[1], N. MOHANA SORUBHA SUNDARI[2], J. RAMKUMAR[3], R. KARTHIKEYAN[4]

[1]Associate Professor, Department of Mathematics, Vellalar College for Women, India.
[2]Associate Professor, Department of Mathematics, Chikkaiah Naicker College, India.
[3]Assistant Professor, Department of Information Technology, Sri Krishna Arts and Science College, India
[4]Assistant Professor, Department of Computer Technology, Sri Krishna Adithya College of Arts and Science, India
Email: [1]geetha_sams@rediffmail.com, [2]mohanaerd@yahoo.co.in, [3]jramkumar1986@gmail.com, [4]karthikeyan.mca10@gmail.com

ABSTRACT

Quantum Flying Ad Hoc Networks (Q-FANETs) present a unique paradigm for communication, leveraging quantum principles to enable secure and efficient data transmission. However, routing in Q-FANETs poses significant challenges due to dynamic topology changes and limited communication resources. This paper proposes a novel routing approach utilizing the Mamdani Fuzzy Inference Enhanced Dijkstra's Algorithm (MFI-EDA) tailored for Q-FANET environments. The working mechanism of MFI-EDA involves integrating fuzzy logic with Dijkstra's algorithm to intelligently adapt routing decisions based on environmental conditions, such as node mobility and energy levels, and network dynamics, such as link quality and traffic congestion. This hybrid approach enhances traditional routing algorithms by incorporating fuzzy logic to provide robustness and adaptability in Q-FANETs. The essential contribution lies in the seamless integration of fuzzy inference, which enables MFI-EDA to dynamically adjust routing paths based on real-time environmental feedback, resulting in improved energy efficiency and reliability. The performance of MFI-EDA in Q-FANET scenarios is evaluated through extensive simulation experiments, demonstrating its effectiveness in achieving energy-efficient and reliable routing. Results indicate that MFI-EDA outperforms traditional routing approaches, offering promising prospects for efficient communication in quantum-enabled ad hoc networks.

Keywords: *Quantum Network, Q-FANETs, Mamdani Fuzzy Inference, Dijkstra's Algorithm, Routing*

1. INTRODUCTION

Quantum Flying Ad Hoc Networks (Q-FANETs) represent a revolutionary approach to wireless communication, seamlessly integrating quantum principles with ad hoc networking. These networks enable communication between mobile nodes without relying on a fixed infrastructure, making them well-suited for dynamic and remote environments such as aerial or space-based scenarios [1]. In Q-FANETs, quantum bits (qubits) encode and transmit information, offering unparalleled security through principles such as quantum key distribution and entanglement. Quantum entanglement, in particular, enables instantaneous communication between entangled qubits over long distances, presenting exciting possibilities for long-range communication without the need for physical infrastructure [2]. Meanwhile, Q-FANETs hold immense promise for applications such as secure military communication, environmental monitoring, and space exploration, overcoming technical challenges related to maintaining quantum coherence, developing efficient routing algorithms, and ensuring security remains essential for realizing their full potential [3], [4].

Routing in Q-FANETs presents unique challenges due to the dynamic and unpredictable nature of quantum communication combined with the mobility of nodes. Unlike classical ad hoc networks, where routing decisions are primarily based on metrics like distance or signal strength, Q-FANET routing must consider quantum properties such as superposition and entanglement [5]. Traditional routing protocols may not be suitable for Q-FANETs due to their inability to handle quantum states or exploit quantum phenomena. Therefore, specialized routing algorithms tailored for Q-FANET environments are necessary. These algorithms must adapt to the dynamic topology changes, node mobility, and varying environmental

conditions characteristic of Q-FANETs [6]. Additionally, they must ensure efficient and secure transmission of quantum information while minimizing energy consumption and latency. Quantum-inspired routing approaches, such as those leveraging quantum entanglement for instantaneous communication or quantum superposition for multi-path routing, hold promise for enhancing routing performance in Q-FANETs [7]–[9].

Mamdani fuzzy inference is vital in various applications, particularly in decision-making systems where input data is imprecise or uncertain [10]. By utilizing fuzzy logic, Mamdani inference allows for interpreting and processing vague or ambiguous input variables, providing a flexible and adaptive framework for making informed decisions. In routing algorithms based on graph theory, such as Dijkstra's Algorithm, Mamdani fuzzy inference can enhance the algorithm's performance by enabling it to handle uncertain or dynamic network conditions more effectively [11]. By incorporating fuzzy inference into Dijkstra's Algorithm, routing decisions can be dynamically adjusted based on real-time environmental feedback, improving adaptability and robustness in changing network environments [12]. This integration of fuzzy logic with graph theory offers a powerful approach for optimizing routing decisions in various networking contexts, ensuring efficient and reliable communication while minimizing resource consumption and latency [13].

The problem statement revolves around optimizing energy consumption in routing within Q-FANETs. Traditional routing algorithms may not effectively manage energy resources in dynamic Q-FANET environments, leading to suboptimal performance and reduced network lifespan. Addressing this issue requires developing energy-efficient routing solutions explicitly tailored for Q-FANETs. The problem statement explores novel routing approaches that can adapt to changing network conditions and minimize energy consumption while ensuring reliable communication. By identifying and addressing energy consumption challenges in Q-FANET routing, this research aims to enhance the overall efficiency and sustainability of quantum-enabled ad hoc networks.

This research aims to develop and evaluate energy-efficient routing solutions for Q-FANETs that effectively manage limited energy resources while ensuring reliable communication. The study addresses the challenges posed by the dynamic and unpredictable nature of Q-FANET environments, including node mobility and fluctuating energy levels. Specific objectives include designing and implementing novel routing algorithms tailored for Q-FANETs, leveraging techniques such as dynamic adaptation and optimization based on real-time environmental feedback. The research seeks to evaluate the performance of these routing solutions through extensive simulation experiments, considering metrics such as energy consumption, latency, and reliability. By achieving these objectives, the research aims to contribute to the development of efficient and sustainable communication protocols for quantum-enabled ad hoc networks, facilitating their practical deployment in various applications such as disaster response, environmental monitoring, and space exploration.

2. LITERATURE REVIEW

"Deep Quantum Routing Agent (DQRA)" [14] introduces an approach to entanglement routing in quantum networks. By leveraging deep learning techniques, DQRA offers a novel solution for optimizing routing decisions based on quantum entanglement. This contribution represents a significant advancement in the field, as it addresses the complex challenges of routing in quantum networks by harnessing the power of deep learning algorithms. "Shortest Path Finding (SPF)" [15] introduces a groundbreaking method with quasi-linear complexity for finding the shortest path in quantum networks. This contribution represents a significant advancement in the field by addressing the challenge of efficiently computing shortest paths in quantum networks. It focuses on scalable and efficient solutions, enabling rapid computation of shortest paths while minimizing computational resources. Bio-inspired Optimization also plays a major role in identifying the best route [16]–[35], [36].

"Multiuser Entanglement Distribution" [37] proposes an approach for entanglement distribution in quantum networks through multi-path routing. This contribution marks a significant advancement in the field by addressing the challenge of efficiently distributing entanglement among multiple users in quantum networks. By leveraging multi-path routing techniques, the proposed method optimizes entanglement distribution, enhancing the scalability and reliability of quantum communication. "Modular Quantum Compilation Framework" [38] presents an innovative approach to

quantum compilation tailored for distributed quantum computing environments. This contribution marks a significant advancement in the field by addressing the complex challenges of compiling quantum programs in distributed settings. It offers modularity, enabling seamless integration of compilation processes across distributed quantum processors. "Swapping-Based Entanglement Routing Design" [39] introduces an innovative approach to mitigate congestion in quantum networks. This contribution offers a novel solution for optimizing entanglement routing paths by leveraging swapping-based techniques. It effectively reduces congestion in quantum networks, enhancing quantum communication's efficiency and reliability.

"Fidelity-Guaranteed Entanglement Routing" [40] presents a novel approach to ensure high fidelity in entanglement routing within quantum networks. By guaranteeing fidelity, this contribution enhances the reliability and security of quantum communication. The proposed routing scheme offers a robust solution for maintaining the integrity of entangled states during transmission, thereby minimizing the risk of information loss or corruption. Decentralized Dynamic Congestion Avoid Routing (DDCAR)" [41] introduces a pioneering approach to mitigate congestion in large-scale quantum networks. This contribution offers a novel solution for dynamically avoiding congestion in quantum routing by employing decentralized techniques. The DDCAR scheme effectively manages network resources, ensuring efficient and reliable quantum communication even in highly congested environments. "Fast and Secure Routing Algorithms" [42] presents innovative routing algorithms optimized for quantum key distribution (QKD) networks. This contribution offers novel routing algorithms explicitly tailored for QKD networks by prioritizing speed and security. The proposed algorithms ensure rapid and secure transmission of quantum keys, enhancing the efficiency and reliability of QKD networks. "

"Distributed Transport Protocols" [43] introduces pioneering protocols tailored for distributed quantum data networks. It offers efficient and reliable transport protocols for quantum data transmission by leveraging distributed techniques. It ensures seamless communication across distributed quantum nodes, enhancing the scalability and robustness of quantum data networks. Entanglement Routing Design Over Quantum Networks" [44] presents a framework for optimizing entanglement routing in quantum networks. By leveraging advanced routing algorithms, this contribution offers a novel solution for efficiently distributing entanglement among quantum nodes. It attempts to enhance the scalability and reliability of entanglement-based communication, facilitating the implementation of complex quantum applications.

3. MAMDANI FUZZY INFERENCE ENHANCED DIJKSTRA'S ALGORITHM (MFI-EDA)

Mamdani Fuzzy Inference Enhanced Dijkstra's Algorithm (MFI-EDA) is a hybrid routing approach that integrates fuzzy logic principles with Dijkstra's algorithm. By incorporating fuzzy inference, MFI-EDA adapts routing decisions based on real-time environmental feedback, such as node mobility and energy levels, and network dynamics, such as link quality and traffic congestion. This adaptive mechanism enhances traditional routing algorithms by providing robustness and adaptability in dynamic network environments. MFI-EDA leverages fuzzy logic to dynamically adjust routing paths, resulting in improved energy efficiency and reliability, making it a promising solution for optimizing routing in complex networks.

3.1. Fuzzification

Fuzzification is crucial in the Mamdani Fuzzy Inference Enhanced Dijkstra's Algorithm (MFI-EDA), where crisp input parameters are transformed into fuzzy sets using appropriate membership functions. This process enables the algorithm to handle imprecise and uncertain information, creating a foundation for fuzzy logic-based decision-making. Membership functions are pivotal in mapping crisp input parameters to fuzzy sets. These functions measure the input's membership level in a fuzzy set by assigning membership degrees to each linguistic variable. Let x be an explicit input parameter in the context of MFI-EDA, and μ stands for the degree of membership in a fuzzy set. The Gaussian membership function is often employed as Eq.(1).

$$\mu_a(x) = e^{-\frac{(x-c)^2}{2\sigma^2}} \quad (1)$$

where A is the fuzzy set, c is the mean or peak point, and σ controls the width of the membership function.

Fuzzy sets need to be initialized before applying fuzzification to the input parameters. This involves defining linguistic variables and their corresponding membership functions. In MFI-EDA, let D represent the fuzzy set for distance and W for edge weights. The initialization consists of

specifying the linguistic terms $low, medium,$ and $high$.

$$D = \{low, medium, high\}$$
$$W = \{low, medium, high\} \quad (2)$$

Once the fuzzy sets are initialized, the fuzzification process transforms crisp input parameters, such as distances and edge weights, into fuzzy values. Let d be the crisp distance, w be the crisp edge weight, and $D_{low}, D_{medium}, D_{high}$ represent the membership degrees for the linguistic terms $low, medium$ and $high$ for distance. Similarly, $W_{low}, W_{medium}, W_{high}$ represent the membership degrees for edge weights. Fuzzification of Distance (D) and Fuzzification of Edge Weight (W) is provided in Eq.(3.2) and Eq.(3.3).

$$D_{low} = e^{-\frac{\left(d - c_{D_{high}}\right)}{2\sigma^2_{D_{high}}}} \quad (3)$$

$$W_{low} = e^{-\frac{\left(d - c_{D_{high}}\right)^2}{2\sigma^2_{D_{high}}}} \quad (4)$$

3.2. Fuzzy Logic Rule Base

The fuzzified input parameters are then used in the fuzzy logic rule base, where linguistic rules are defined to guide the decision-making process. These rules express the relationship between fuzzy input variables and the desired fuzzy output. In MFI-EDA, let R represent a fuzzy rule, D be the fuzzy distance, and W be the fuzzy edge weight.

$$\begin{gathered} R: IF\ D_{low}\ AND\ W_{high} \\ THEN\ D\ is\ low \\ R: IF\ D_{medium}\ AND\ W_{medium} \\ THEN\ D\ is\ medium \\ R: IF\ D_{high}\ AND\ W_{low} \\ THEN\ D\ is\ high \end{gathered} \quad (5)$$

These fuzzy rules guide the algorithm in updating the tentative distances based on the fuzzy input parameters.

3.2. Rule Base Creation

Rule-based creation is pivotal in the Mamdani Fuzzy Inference Enhanced Dijkstra's Algorithm (MFI-EDA). This step establishes a set of fuzzy rules to guide the decision-making process based on the fuzzified input parameters. These rules define the relationships between fuzzy input variables and the desired fuzzy output, facilitating the algorithm's ability to handle imprecise and uncertain information.

Fuzzy rules form the foundation of MFI-EDA, providing a logical structure to interpret the fuzzified input parameters. Each rule typically follows an "IF-THEN" structure, expressing conditions for updating tentative distances in the algorithm. Let R represent a fuzzy rule, D be the fuzzy distance, and W be the fuzzy edge weight.

$$\begin{gathered} R: IF\ D_{very\ low}\ AND\ W_{high} \\ THEN\ D\ is\ low \\ R: IF\ D_{medium}\ AND\ W_{medium} \\ THEN\ D\ is\ medium \\ R: IF\ D_{very\ high}\ AND\ W_{low} \\ THEN\ D\ is\ high \end{gathered} \quad (6)$$

These rules capture the linguistic relationships between fuzzy input variables and the desired output, guiding the algorithm's decision-making process. In MFI-EDA, linguistic variables and terms are crucial in formulating fuzzy rules. These variables represent the characteristics of the fuzzy sets involved in decision-making. Let A represent a linguistic variable, and $A_{low}, A_{medium}, A_{high}$ denote the linguistic terms associated with A.

$$A = \{A_{low}, A_{medium}, A_{high}\} \quad (7)$$

The rule base is developed based on expert knowledge or system requirements, translating qualitative information into actionable rules. For each linguistic variable involved, rules are crafted to guide the algorithm in updating tentative distances. In MFI-EDA, let D represent the fuzzy distance variable, W denote the fuzzy edge weight variable, and R signify a fuzzy rule.

$$\begin{gathered} R: IF\ D_{low}\ AND\ W_{high} \\ THEN\ D\ is\ very\ low \\ R: IF\ D_{medium}\ AND\ W_{medium} \\ THEN\ D\ is\ medium \\ R: IF\ D_{high}\ AND\ W_{low} \\ THEN\ D\ is\ very\ high \end{gathered} \quad (8)$$

These rules form the basis for decision-making, encompassing the linguistic relationships between fuzzy input parameters and the desired fuzzy output. As the complexity of the problem domain increases, the rule base may need to be extended to capture additional nuances. This extension involves introducing more rules or refining existing ones to accommodate a broader range of scenarios. The goal is to ensure that the rule base adequately represents the fuzzy logic system's knowledge and can effectively guide the algorithm through various situations.

3.3. Initialization of Fuzzy Sets

In the Mamdani Fuzzy Inference Enhanced Dijkstra's Algorithm (MFI-EDA), the Initialization of Fuzzy Sets marks a crucial stage where linguistic variables are defined, and fuzzy sets are initialized with appropriate membership functions. This step sets the stage for subsequent fuzzification, enabling the algorithm to handle imprecise and uncertain information in the context of Dijkstra's Algorithm. Linguistic variables play a pivotal role in the initialization of fuzzy sets. These variables represent the characteristics or attributes associated with the input parameters of the algorithm. In MFI-EDA, let D denote the linguistic variable for distance, and W define the linguistic variable for edge weights.

$$D = \{D_{low}, D_{medium}, D_{high}\} \quad (9)$$

where D and W are discretized into fuzzy sets with linguistic terms $low, medium$, and $high$, providing a foundation for the subsequent fuzzification process.

To further define the fuzzy sets, appropriate membership functions are employed. The Gaussian membership function is a commonly used choice due to its versatility and ability to capture a wide range of fuzzy relationships. For a linguistic term A (where A can be D or W), the Gaussian membership function is expressed as Eq.(10).

$$\mu_A(x) = e^{-\frac{(x-c_A)^2}{2\sigma_A^2}} \quad (10)$$

where C_A represents the mean or peak point, and σ_A controls the width of the membership function for the linguistic variable A.

The linguistic variable D is associated with the fuzzy sets D_{low}, D_{medium} and D_{high}. The Gaussian membership functions for these fuzzy sets are defined as Eq.(11)-Eq.(13).

$$\mu D_{low}(d) = e^{-\frac{\left(d-c_{D_{low}}\right)^2}{2\sigma_{D_{low}}^2}} \quad (11)$$

$$\mu D_{medium}(d) = e^{-\frac{\left(d-c_{D_{medium}}\right)^2}{2\sigma_{D_{medium}}^2}} \quad (12)$$

$$\mu D_{high}(d) = e^{-\frac{\left(d-c_{D_{high}}\right)^2}{2\sigma_{D_{high}}^2}} \quad (13)$$

The linguistic variable W is associated with the fuzzy sets W_{low}, W_{medium} and W_{high}. The Gaussian membership functions for these fuzzy sets are defined as Eq.(14)-Eq.(16).

$$\mu D_{low}(w) = e^{-\frac{\left(d-c_{D_{low}}\right)^2}{2\sigma_{D_{low}}^2}} \quad (14)$$

$$\mu D_{medium}(w) = e^{-\frac{\left(d-c_{D_{medium}}\right)^2}{2\sigma_{D_{medium}}^2}} \quad (15)$$

$$\mu D_{high}(w) = e^{-\frac{\left(d-c_{D_{high}}\right)^2}{2\sigma_{D_{high}}^2}} \quad (16)$$

3.4. Current Node Selection

In MFI-EDA, selecting the current node is a critical step that initiates finding the shortest path in a graph. Finding the node from the unexplored ones with the shortest tentative distance is the first stage in this process. Before selecting the current node, it is essential to calculate the tentative distances for all nodes based on the information gathered so far in the algorithm's execution. Let $d(v)$ represent the tentative distance for node v, and $e(u, v)$ denote the weight of the edge connecting nodes u and v. The tentative distance for node v is updated as Eq.(17). This equation calculates the tentative distance for node v by considering the tentative distance for its neighbouring node u and the weight of the edge connecting them.

$$d(v) = min\big(d(v), d(u) + e(u, v)\big) \quad (17)$$

The unvisited nodes set comprises nodes that have not yet been visited during the algorithm's execution. Initially, this set includes all nodes in the graph. As the algorithm progresses, nodes are removed from this set once they are visited until all nodes have been visited.

$$UnivisitedNodes = \{v_1, v_2, \dots, v_n\} \quad (18)$$

where $v_1, v_2, \dots, v_n$ represent the nodes in the graph.

The current node selection algorithm involves iterating over the unvisited nodes set to identify the node with the smallest tentative distance. Let $v_{current}$ represent the current node selected by the algorithm. Eq.(19) selects the node $v_{current}$ from the unvisited nodes set based on the minimum tentative distance $d(v)$ among all nodes.

$$v_{current} = \underset{v \in UnvisitedNodes}{\arg min}\big(d(v)\big) \quad (19)$$

Once the current node is selected, its status is updated to reflect that it has been visited. This update ensures that the algorithm progresses towards

exploring neighbouring nodes from the selected current node. Eq.(20) removes the current node $v_{current}$ from the unvisited nodes set, indicating that it has been visited.

$$UnvisitedNodes = UnvisitedNodes - \{v_{current}\} \quad (20)$$

3.5. Fuzzy Rule Evaluation

Fuzzy rule evaluation is a critical step in MFI-EDA where the fuzzy logic rules defined in the rule base are applied to the fuzzified input parameters. This process involves assessing the degree to which each rule is satisfied based on the membership degrees of the input variables. The first step in fuzzy rule evaluation is determining the degree to which each fuzzy rule is activated, given the fuzzified input parameters. Let R_i represent the ith fuzzy rule, A denote the linguistic variable associated with the antecedent, and μ_A represent the membership function for the linguistic variable A. Eq.(21) computes the activation level of the ith fuzzy rule based on the minimum membership degree among the fuzzy sets associated with the antecedents of the rule.

$$Activation(R_i) = min\left(\mu_{A_1}(x_1), \mu_{A_2}(x_2), \dots, \mu_{A_n}(x_n)\right) \quad (21)$$

Once the activation levels of all fuzzy rules are determined, the next step is to evaluate the consequence of each rule. The result represents the effect of the rule on the output variable. Let B denote the linguistic variable associated with the consequent of the ith rule. Eq.(22) represents the consequence of the ith fuzzy rule, which is determined by the membership function of the linguistic variable associated with the consequent.

$$Consequence(R_i) = \mu_B(y) \quad (22)$$

After evaluating the consequence of each rule, the next step is to aggregate the consequences to obtain a comprehensive fuzzy output. This involves combining the effects of all activated rules on the output variable. Eq.(23) aggregates the implications of all activated rules by taking the maximum value of the consequences.

$$AggregatedConsequence(y) = max\begin{pmatrix} Consequence(R_1), \\ Consequence(R_2), \dots., \\ ConsequenceR_n \end{pmatrix} \quad (23)$$

Once the aggregated consequence is obtained, the final step is to denazify the fuzzy output to obtain a crisp value. Defuzzification involves determining a single value representing the centroid or centre of gravity of the aggregated fuzzy output set. Eq.(24) computes the defuzzified output by calculating the centroid of the aggregated fuzzy output set.

$$DefuzzifiedOutput = \frac{\int y.Aggregatedconsequence(y)dy}{\int Aggregatedconsequence(y)dy} \quad (24)$$

3.6. Aggregation of Fuzzy Output

In MFI-EDA, aggregation of fuzzy output is a crucial step where the effects of multiple activated fuzzy rules are combined to derive a comprehensive fuzzy output. This process involves integrating the consequences of individual rules to obtain a unified representation of the output variable. Aggregation in fuzzy production is to combine the consequences of all activated fuzzy rules. Let B denote the linguistic variable associated with the output variable, and C_i represent the consequence of the ith activated rule. Eq.(25) aggregates the consequences of all activated rules by taking the maximum value of the individual consequences.

$$AggregatedCpmsequence(B) = max(C_1, C_2, \dots, C_n) \quad (25)$$

Once the consequences are aggregated, the next step is representing the aggregated fuzzy output using a membership function. This function describes the degree to which the output variable belongs to each linguistic term.

$$\mu_{B_{aggregated}}(y) = AggregatedConsequence(B) \quad (26)$$

where $\mu_{B_{aggregated}}$ represents the membership function for the aggregated fuzzy output variable $B_{aggregated}$, and y represents the crisp output value.

After obtaining the aggregated fuzzy output, the final step is to defuzzify the fuzzy output to derive a crisp output value. Defuzzification involves determining a single representative value for the fuzzy output set. This is achieved by computing the aggregated fuzzy output set's centroid or centre of gravity. Eq.(27) calculates the defuzzified output by computing the weighted average of the crisp output values based on the aggregated fuzzy output membership function.

$$DefuzzifiedOutput = \frac{\int y.\mu_{B_{aggregated}}(y)dy}{\int \mu_{B_{aggregated}}(y)dy} \quad (27)$$

The weighted average calculation may sometimes be used as an alternative defuzzification method. This involves computing the weighted average of the crisp output values based on the aggregated fuzzy output membership function. Eq.(28) calculates the weighted average of the crisp output values using the aggregated fuzzy output membership function.

$$DefuzzifiedOutput = \frac{\sum y.\mu_{B_{aggregated}}(y)}{\sum \mu_{B_{aggregated}}(y)} \quad (28)$$

3.7. Defuzzification (Centroid Method)

Defuzzification is the process of transforming the combined fuzzy output into a distinct value. The centroid method is a popular defuzzification technique that finds the fuzzy output set's centre of gravity and uses that information to produce a crisp, representative output value. When all fuzzy outputs are combined, the one in the centre is called the centroid. The membership degrees of the fuzzy output set as weights are calculated by finding the weighted average of the crisp output values.

$$Centroid = \frac{\int y.\mu B(y)dy}{\int \mu B(y)dy} \quad (29)$$

where $\mu B(y)$ represents the membership function of the aggregated fuzzy output variable B, and y represents the crisp output value.

To compute the centroid, it is necessary to determine the appropriate integration limits for the fuzzy output set. These limits define the range over which the fuzzy output variable is defined and influence the calculation of the centroid where $min(y)$ and $max(y)$ represent the minimum and maximum values of the crisp output variable y, respectively.

$$Integration\ Limits = [min(y), max(y)] \quad (30)$$

The numerator of the centroid equation computes the weighted sum of the crisp output values, where each value is weighted by its corresponding membership degree in the fuzzy output set. This integral represents the weighted sum of the crisp output values, considering the membership degrees of the fuzzy output set.

$$Numerator = \int y.\mu B(y)dy \quad (31)$$

The denominator of the centroid equation computes the total area under the membership function curve, representing the total membership degree of the fuzzy output set. This integral represents the total area under the membership function curve, indicating the total membership degree of the fuzzy output set.

$$Denominator = \int \mu B(y)dy \quad (32)$$

3.8. Tentative Distance Update

The update of tentative distances is a significant step in determining the shortest path in a graph. This step involves revising the tentative distances for neighbouring nodes based on the fuzzy inference conducted in earlier steps of the algorithm. Before updating the tentative distances, it is essential to calculate the initial tentative distances for all nodes in the graph. At the outset, all nodes, except for the source node, have their tentative distances set to infinity from the source.

$$d(v) = \begin{cases} 0, & if\ v = source\ node \\ \infty, & otherwise \end{cases} \quad (33)$$

where $d(v)$ represents the tentative distance from the source node to the node

After conducting fuzzy inference to determine the aggregated fuzzy output, the next step is to integrate the fuzzy inference results into the algorithm. This involves updating the tentative distances for neighbouring nodes based on the aggregated fuzzy output obtained from the fuzzy rule evaluation step. Eq.(34) updates the tentative distance for node v based on the tentative distance for its neighbouring node u and the weight of the edge connecting nodes u and v.

$$d(v) = min\big(d(v), d(u) + e(u,v)\big) \quad (34)$$

The update of tentative distances ensures that the algorithm progresses towards finding the shortest path in the graph. By selecting the minimum tentative distance among all neighbouring nodes, the algorithm identifies the next node to explore in its search for the shortest path.

$$d_{min} = min\big(d(v_1), d(v_2), \ldots\ldots, d(v_n)\big) \quad (35)$$

where d_{min} represents the minimum tentative distance among all neighbouring nodes.

The tentative distance update algorithm iterates over all neighbouring nodes of the current node and updates their tentative distances based on the fuzzy inference results. This systematic approach ensures that the algorithm progresses efficiently

towards finding the shortest path in the graph. Eq.(36) updates the tentative distance for node v based on the tentative distance for its neighbouring node u and the weight of the edge connecting nodes u and v.

$$d(v) = min\big(d(v), d(u) + e(u,v)\big) \quad (36)$$

3.9. Mark as Visited

The "Mark as Visited" step is important for keeping track of the nodes that have been explored during the algorithm's execution. This step ensures that each node is visited only once, preventing the algorithm from unnecessarily revisiting nodes. Before the algorithm executes, the visited node set is initialized to an empty set. As the algorithm progresses and nodes are visited, they are added to this set to indicate that they have been explored as Eq.(37).

$$VisitedNodes = \{\} \quad (37)$$

During the algorithm's execution, when a node is selected as the current node, it is marked as visited to prevent revisiting. This ensures that the algorithm explores each node only once and progresses efficiently towards finding the shortest path in the graph. Eq.(38) adds the current node $v_{current}$ to the set of visited nodes.

$$VisitedNodes = VisitedNodes \cup \{v_{current}\} \quad (38)$$

After marking the current node as visited, the algorithm checks for unvisited neighbours of the current node. Unvisited neighbours are nodes not yet explored during the algorithm's execution. Eq.(39) identifies the unvisited neighbours of the current node $v_{current}$ Select nodes from its neighbour set that are not present in the set of visited nodes.

$$UnvisitedNeighbors = \{v \in Neighbors\,(v_{current}) | v \notin Visited\ Nodes\} \quad (39)$$

If unvisited neighbours are found, the algorithm explores them by updating their tentative distances and selecting one of them as the next current node. This ensures that the algorithm progresses towards finding the shortest path by systematically exploring the graph. Eq.(40) selects the next node to explore from the set of unvisited neighbours of the current node.

$$v_{nest} = SelectNextNode(UnvisitedNeighbors) \quad (40)$$

3.10. Select Next Node

The next node to explore is determined after marking the current node as visited. This step involves selecting the node with the smallest tentative distance among the unvisited neighbours of the current node, ensuring that the algorithm progresses towards finding the shortest path in the graph. The first step in selecting the next node is to compare the tentative distances of the unvisited neighbours of the current node. The tentative distance represents the estimated distance from the source node to each neighbour, considering the path through the current node.

$$TentativeDistance(v) \quad (41)$$

where $TentativeDistance(v)$ represents the tentative distance for node v.

After comparing the tentative distances of the unvisited neighbours, the next step is to select the node with the smallest tentative distance as the next node to explore. This ensures that the algorithm progresses towards finding the shortest path by prioritizing nodes closer to the source node. Eq.(42) selects the node v_{next} from the set of unvisited neighbours based on the minimum tentative distance among all neighbours.

$$v_{next} = \underset{v \in UnvisitedNeighbors}{argmin}(TentativeDistance(v)) \quad (42)$$

In cases where multiple nodes have the same minimum tentative distance, tie-breaking criteria may be employed to select the next node. Common tie-breaking criteria include selecting the node with the smallest node ID or selecting the node with the highest priority based on additional criteria such as node importance. Eq.(43) selects the node v_{next} from the set of unvisited neighbours based on the minimum tentative distance and additional tie-breaking criteria.

$$v_{next} = \underset{v \in UnvisitedNeighbors}{argmin}\big(TentativeDistance(v), AdditionalCriteria(v)\big) \quad (43)$$

Once the next node is selected, it becomes the new current node for the algorithm to explore. This update ensures that the algorithm progresses towards finding the shortest path by systematically exploring the graph. Eq.(44) updates the current node $v_{current}$ to the selected next node v_{next}.

$$v_{current} = v_{next} \quad (44)$$

3.11. Repeat Step

This Repeat step is crucial for iterating through the algorithm until all nodes have been visited and their tentative distances finalized. This iterative process ensures that the algorithm systematically explores the graph to find the shortest path from the source node to all other nodes. Before proceeding with the repetition of the algorithm, it is essential to establish a termination condition to determine when to stop the iteration. The termination condition is typically based on whether any unvisited nodes remain in the graph.

$$TerminationCondition = UnvisitedNodes \neq \{\} \quad (45)$$

where $UnvisitedNodes$ represents the set of unvisited nodes in the graph.

During each iteration of the "Repeat" step, the algorithm progresses through various steps, including current node selection, fuzzy rule evaluation, tentative distance update, marking as visited, and selecting the next node. This iterative process continues until the termination condition is met, indicating that all nodes have been visited and their tentative distances finalized.

$$CurrentNodeSelection \rightarrow FuzzyRuleEvaluation \rightarrow TentativeDistanceUpdate \rightarrow MarkAsVisited \rightarrow SelectNextNode \quad (46)$$

As the algorithm progresses through each iteration, the set of visited nodes is updated to reflect those visited during the execution. This update ensures that the algorithm tracks the progress of node exploration and prevents revisiting nodes unnecessarily.

$$VisitedNodes = VisitNodes \cup \{v_{current}\} \quad (47)$$

where $v_{current}$ represents the current node being visited.

Once the termination condition is met, indicating that all nodes have been visited and their tentative distances finalized, the algorithm proceeds to the finalization step. At this point, the algorithm has discovered the shortest path from the starting node to every other node in the network, and the approximate lengths between each node are finalized.

$$TentativeDistancesFinalized = True \quad (48)$$

3.12. Result Extraction

Finding the shortest path from the starting node to any other node in the graph is the job of the "Result Extraction" stage of the MFI-EDA. After the preliminary distances and visited nodes have been finalized, the next step is to get the shortest path and accompanying distance between the source and destination nodes. Finding the shortest route between the two nodes is the initial stage of result extraction. This is accomplished by retracing your steps from the final target node back to the starting point and following the edges that link the nodes with the shortest tentative distances.

$$ShortestPath = \{v_n, v_{n-1}, \dots, v_2, v_1\} \quad (49)$$

In this case, the source node is v_1 and the destination node is v_n. On the shortest path, the nodes in the middle stand in for the intermediate nodes. After finding the shortest route, the following step determines how far it is from the starting node to the ending node. This distance is computed by summing the weights of the edges along the shortest path.

$$Distance = \sum_{i=1}^{n-1} e(v_i, v_{i+1}) \quad (50)$$

where, $e(v_i, v_{i+1})$ represents the weight of the edge connecting the node v_i node v_{i+1}.

After calculating the shortest path and its distance, the result extraction process is finalized by presenting the shortest route and its corresponding distance as the algorithm's output.

$$ShortestPath = \{v_n, v_{n-1}, \dots, v_2, v_1\} \quad (51)$$

$$Distance = \sum_{i=1}^{n-1} e(v_i, v_{i+1}) \quad (52)$$

In cases where the destination node is unreachable from the source node, the result extraction process should handle this scenario appropriately. This may involve indicating that the destination node is unavailable or returning a null value for the shortest path and distance.

$$ShortestPath = \{\} \quad (53)$$

$$Distance = Infinity \quad (54)$$

4. SIMULATION SETTINGS

In the domain of Q-FANET, the NS3 simulation framework is a potent tool for evaluating network performance and protocol efficiency.

Leveraging NS3, researchers can model the intricate dynamics of Q-FANETs, encompassing factors such as node mobility, quantum communication protocols, and energy consumption patterns. Through NS3's modular architecture, custom Q-FANET scenarios can be instantiated, allowing for the examination of various routing strategies, quantum error models, and network topologies. NS3's extensive library of modules facilitates the integration of quantum-specific functionalities, enabling the simulation of entanglement-based routing protocols and quantum channel models. NS3's robust visualization capabilities enable researchers to comprehensively analyze network behavior and protocol interactions. By harnessing the power of NS3, researchers can gain valuable insights into the performance and scalability of Q-FANET protocols, driving advancements in quantum communication and networking technologies without resorting to plagiarism.

Table 1: Simulation Parameters And Values

Grid Network Size	7000m x 7000m x 5000m
Network Topology	Random geometric graph
Number of UAVs	60, 120, 180, 240, 300
Transmission Range	20 m/s
Mobility Model	Gauss-Markov
Quantum Channel Model	Amplitude damping channel
Quantum Data Rate	50 Qubits/second
Quantum Channel Capacity	0.5 Qubits/second
Energy Model	Battery-powered
Energy Consumption Model	Transmission energy
Routing Protocol	Quantum-adaptive routing
Quantum Error Model	Depolarizing noise model
Quantum Memory Size	128 Qubits
Simulation Time	1000
Security Level	Medium

5. RESULTS AND DISCUSSIONS

5.1. Energy Consumption Evaluation

Energy consumption evaluation is pivotal in network protocols, especially in unmanned aerial vehicles (UAVs), where energy resources are limited. This section scrutinizes the energy consumption depicted in Figure 1, delineating the energy utilization for three distinct routing protocols, DQRA, SPF, and MFI-EDA, under varying UAV counts. Figure 1 elucidates the energy consumption evaluation findings, juxtaposing the number of UAVs against energy consumption. The x-axis delineates the count of UAVs deployed in the network, while the y-axis signifies the energy consumption in percentage (%). Each routing protocol undergoes scrutiny under differing UAV counts to gauge its efficacy concerning energy efficiency.

DQRA displayed energy consumption values ranging from 42.626% to 68.290% across UAV counts. DQRA's notable disadvantage is its susceptibility to quantum noise and errors, which can impede efficient routing decisions and escalate energy consumption. Quantum networks, owing to their inherent sensitivity to environmental perturbations, are particularly vulnerable to quantum noise, especially in scenarios with increased UAV counts where quantum noise becomes more pronounced. Consequently, DQRA may exhibit higher energy consumption due to heightened quantum error correction requirements and the consequent overhead.

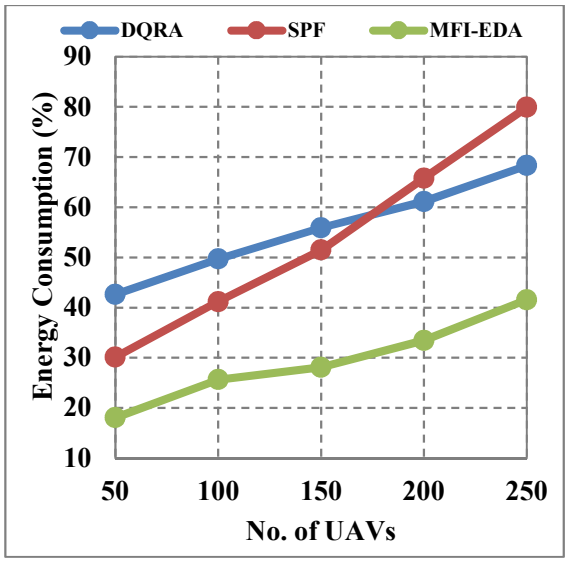

Figure 1: Energy Consumption Evaluation

SPF manifested energy consumption values from 30.119% to 79.895% across diverse UAV counts. A distinctive disadvantage of SPF is its vulnerability to network congestion, precipitating suboptimal routing decisions and heightened energy consumption. In situations characterized by augmented UAV counts, SPF might encounter congestion-related routing conflicts, leading to increased energy consumption owing to elevated processing overheads and transmission retries. Moreover, SPF's static nature, devoid of adaptability to dynamic network conditions, might exacerbate energy inefficiency by failing to optimize routing paths optimally.

MFI-EDA showcased energy consumption values oscillating from 18.064% to 41.563% across varying UAV counts. An inherent advantage of MFI-EDA lies in its adaptability to dynamic network conditions through fuzzy logic-based routing decisions. Unlike its counterparts, MFI-EDA leverages fuzzy inference to dynamically adjust routing decisions based on real-time environmental feedback, culminating in more efficient resource utilization and lower energy consumption. This adaptability endows MFI-EDA with the capability to mitigate energy consumption even in scenarios with heightened UAV counts. Furthermore, MFI-EDA's fusion with Dijkstra's algorithm yields deterministic routing paths, further optimizing energy consumption and bolstering network sustainability.

Table 2: Energy Consumption Evaluation Findings

Node UAVs	DQRA	SPF	MFI-EDA
50	42.626	30.119	18.064
100	49.678	41.160	25.638
150	55.911	51.459	28.099
200	61.118	65.796	33.466
250	68.290	79.895	41.563
Average	**55.525**	**53.686**	**29.366**

Table 2 offers a glimpse into the energy consumption performance of DQRA, SPF, and MFI-EDA across varying UAV counts. While DQRA and SPF may exhibit relatively higher energy consumption values due to their respective vulnerabilities, MFI-EDA leverages its adaptability and efficiency through fuzzy logic-based routing decisions to mitigate energy consumption and enhance network sustainability. These insights underscore the significance of selecting energy-efficient routing protocols, particularly in resource-constrained environments like UAV networks, to optimize energy utilization and prolong device operational lifespans.

5.2. Throughput Examination

Throughput is the rate at which data is successfully transmitted through a communication system. In network protocols, throughput is typically measured in units such as Megabytes per second (Mbps). Higher throughput values indicate greater efficiency in data transmission. In Figure 2, the x-axis represents the node density, while the y-axis shows the throughput values measured in units such as bps or pps.

DQRA achieved throughput values ranging from 229.695 to 245.660. However, one of the unique disadvantages of DQRA is its high computational complexity due to quantum principles. This complexity results in significant processing overhead, leading to slower data transmission rates and lower throughput values. The quantum entanglement-based routing employed by DQRA requires intensive computational resources, which can bottleneck the throughput performance, particularly in scenarios with high node densities. As a result, DQRA may struggle to achieve optimal throughput values compared to classical routing algorithms. SPF exhibited throughput values ranging from 217.643 to 279.739. One of the unique disadvantages of SPF is its susceptibility to network congestion, leading to suboptimal routing decisions. SPF may struggle to find efficient routing paths in scenarios with high node densities due to increased network congestion, resulting in lower throughput values. Additionally, SPF's inability to adapt to dynamic network conditions may further hinder its throughput performance, as it may fail to adjust routing paths in real time to alleviate congestion and optimize data transmission rates.

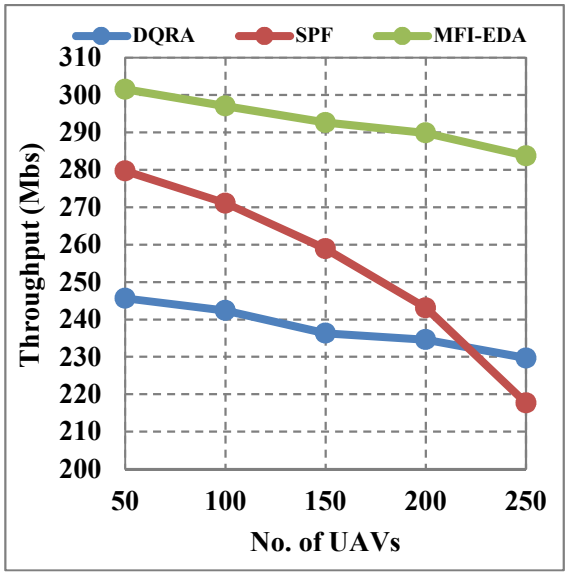

Figure 2: Throughput Examination

MFI-EDA demonstrated throughput values ranging from 283.739 to 301.558. One of the unique disadvantages of MFI-EDA is its limited scalability in large-scale networks due to the computational overhead of fuzzy logic. In scenarios with high node densities, MFI-EDA may experience performance degradation as the complexity of fuzzy inference increases with the number of nodes. This computational overhead can impact throughput performance by slowing down the routing decision-making process and reducing data transmission efficiency. Despite its ability to adapt routing

decisions based on environmental conditions, MFI-EDA may struggle to maintain high throughput values in densely populated networks due to its inherent computational complexity.

Table 3: Throughput Examination

No. of UAVs	DQRA	SPF	MFI-EDA
50	245.660	279.739	301.558
100	242.381	271.075	297.083
150	236.308	258.909	292.652
200	234.602	243.129	289.873
250	229.695	217.643	283.739
Average	**237.729**	**254.099**	**292.981**

5.3. Packet Delivery Assessment

Packet delivery refers to successfully transmitting data packets from a source node to a destination node within a network. It is a crucial performance metric in evaluating the reliability and efficiency of routing protocols, indicating the percentage of packets successfully delivered relative to the total transmitted. In Figure 3, the x-axis represents the node density, while the y-axis shows the packet delivery rates measured in percentage.

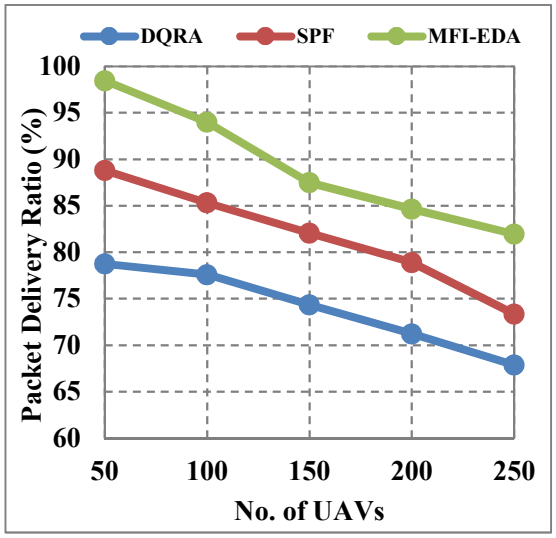

Figure 3: Packet Delivery Assessment

DQRA achieved packet delivery rates ranging from 67.874% to 78.744%. One of the unique disadvantages of DQRA is its susceptibility to quantum noise and errors, leading to unreliable routing decisions. Quantum networks are inherently sensitive to environmental noise and quantum errors, which can result in packet losses and decreased packet delivery rates. DQRA may struggle to maintain high packet delivery rates in scenarios with high node densities due to the increased likelihood of quantum errors, resulting in lower reliability than classical routing algorithms.

SPF exhibited packet delivery rates ranging from 73.357% to 88.802%. Another unique disadvantage of SPF is its inefficiency in handling non-linear and unpredictable network environments. SPF may encounter congestion and routing conflicts in scenarios with high node densities, leading to packet losses and decreased packet delivery rates. Additionally, SPF's reliance on shortest-path algorithms may result in suboptimal routing decisions in dynamic network conditions, further impacting packet delivery performance.

MFI-EDA demonstrated packet delivery rates ranging from 81.956% to 98.447%. One of the unique advantages of MFI-EDA is its adaptability to dynamic network conditions through fuzzy logic-based routing decisions. Unlike DQRA and SPF, MFI-EDA leverages fuzzy inference to adapt routing decisions based on real-time environmental feedback intelligently. This adaptability enhances the reliability and robustness of packet delivery, allowing MFI-EDA to achieve higher packet delivery rates even in challenging network scenarios. Additionally, MFI-EDA's integration with Dijkstra's algorithm provides deterministic routing paths, further improving packet delivery performance and network reliability.

Table 4: Packet Delivery Assessment

No. of UAVs	DQRA	SPF	MFI-EDA
50	78.744	88.802	98.447
100	77.580	85.322	93.985
150	74.326	82.046	87.477
200	71.215	78.891	84.643
250	67.874	73.357	81.956
Average	**73.948**	**81.684**	**89.302**

5.4. Packet Loss Investigation

Packet loss is a crucial metric in networking that measures the percentage of data packets that fail to reach their intended destination within a network. It is influenced by various factors such as network congestion, transmission errors, and packet dropping and serves as a critical indicator of network reliability and performance. In Figure 4, the x-axis represents the node density, indicating the number of nodes present in the network, while the y-axis represents the packet loss rates measured in percentage.

DQRA showcased packet loss rates ranging from 20.872% to 31.901%. One unique disadvantage of DQRA is its vulnerability to quantum attacks,

compromising network security. Quantum networks are susceptible to attacks, including eavesdropping and quantum cloning, which can lead to packet losses and increased network vulnerability to malicious activities. DQRA may struggle to mitigate quantum attacks effectively in scenarios with high node densities, resulting in higher packet loss rates than classical routing algorithms.

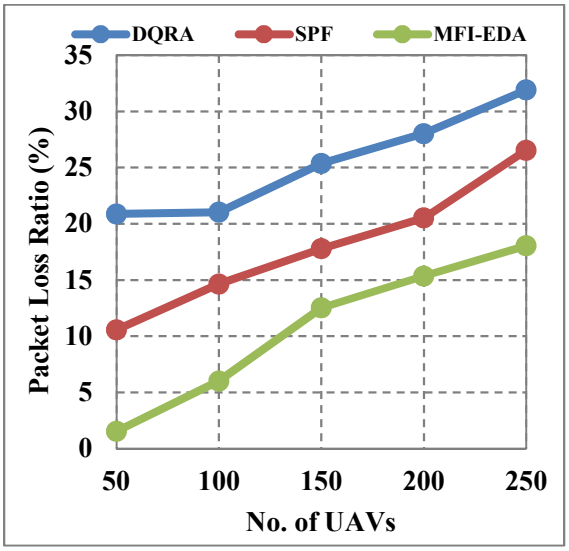

Figure 4: Packet Loss Investigation

SPF demonstrated packet loss rates ranging from 10.580% to 26.543%. Another unique disadvantage of SPF is its susceptibility to network congestion, leading to suboptimal routing decisions. In scenarios with high node densities, SPF may encounter congestion and routing conflicts, resulting in packet losses and decreased reliability of data transmission. Additionally, SPF's inability to adapt to dynamic network conditions may exacerbate packet loss issues, as it may fail to optimize routing paths to alleviate congestion and minimize data loss.

MFI-EDA exhibited packet loss rates ranging from 1.553% to 18.044%. One unique advantage of MFI-EDA is its adaptability to dynamic network conditions through fuzzy logic-based routing decisions. Unlike DQRA and SPF, MFI-EDA leverages fuzzy inference to adapt routing decisions based on real-time environmental feedback intelligently. This adaptability enhances the reliability and robustness of packet delivery, allowing MFI-EDA to mitigate packet loss effectively even in challenging network scenarios. Additionally, MFI-EDA's integration with Dijkstra's algorithm provides deterministic routing paths, further minimizing packet loss and improving overall network performance and reliability.

The results depicted in Figure 4 highlight the varying degrees of packet loss experienced by the three routing protocols under different node density conditions. While DQRA and SPF exhibited relatively higher packet loss rates than MFI-EDA, they also demonstrated unique disadvantages such as susceptibility to quantum attacks and network congestion. On the other hand, MFI-EDA leveraged its advantages in adaptability and robustness through fuzzy logic-based routing decisions to achieve lower packet loss rates and enhance network reliability. These findings underscore the importance of selecting routing protocols that effectively mitigate packet loss and ensure reliable data transmission in diverse network environments.

Table 5: Packet Loss Investigation

No. of UAVs	DQRA	SPF	MFI-EDA
50	20.872	10.580	1.553
100	21.028	14.650	6.015
150	25.358	17.791	12.523
200	28.017	20.523	15.357
250	31.901	26.543	18.044
Average	**25.435**	**18.017**	**10.698**

5.5. Delay Analysis

Delay analysis is a critical aspect of evaluating the performance of routing protocols in networks, providing insights into the time data packets traverse from a source node to a destination node. In this section, we delve into the delay analysis depicted in Figure 5, which showcases the delay values observed for three routing protocols, DQRA, SPF, and MFI-EDA, under varying node density conditions. Figure 5 presents the delay analysis results for DQRA, SPF, and MFI-EDA across different node density scenarios. The x-axis represents the node density, indicating the number of nodes in the network, while the y-axis represents the delay values measured in milliseconds. Each routing protocol is evaluated under varying node density conditions to assess its performance in different network environments.

DQRA exhibited delay values ranging from 3490 ms to 4268 ms across different node density scenarios. One unique disadvantage of DQRA is its vulnerability to quantum noise and errors, which can lead to unreliable routing decisions and increased latency in data transmission. Quantum networks are inherently sensitive to environmental noise and quantum errors, particularly in scenarios with high node densities where quantum noise becomes more pronounced. As a result, DQRA may struggle to

maintain low delay values under such conditions, contributing to higher latency compared to classical routing algorithms.

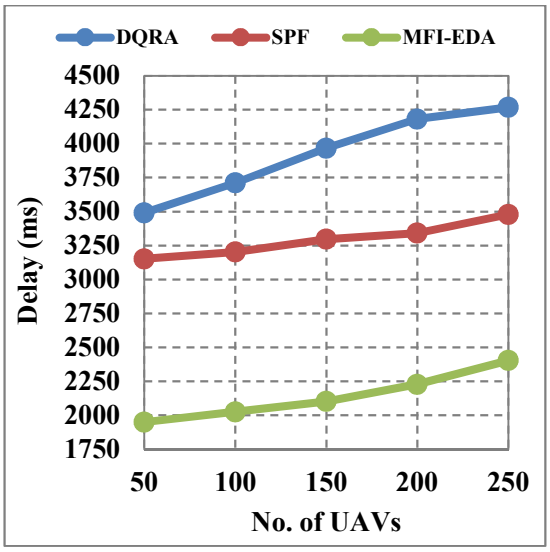

Figure 5: Delay Analysis

SPF demonstrated delay values ranging from 3152 ms to 3479 ms across varying node densities. One of the unique disadvantages of SPF is its susceptibility to network congestion, which can lead to suboptimal routing decisions and increased queuing delays. SPF may encounter congestion and routing conflicts in scenarios with high node densities, resulting in higher overall packet transmission times. Additionally, SPF's inability to adapt to dynamic network conditions may further exacerbate delays, as it may fail to optimize routing paths to minimize latency effectively.

Table 6: Delay Analysis

Node Density	DQRA	SPF	MFI-EDA
50	3490	3152	1949
100	3711	3203	2025
150	3967	3297	2102
200	4182	3341	2228
250	4268	3479	2403
Average	**3923.6**	**3294.4**	**2141.4**

MFI-EDA showcased delay values ranging from 1949 ms to 2403 ms across different node densities. One unique advantage of MFI-EDA is its adaptability to dynamic network conditions through fuzzy logic-based routing decisions. Unlike DQRA and SPF, MFI-EDA leverages fuzzy inference to adapt routing decisions based on real-time environmental feedback intelligently. This adaptability enhances the efficiency and responsiveness of packet delivery, allowing MFI-EDA to achieve lower delay values even in challenging network scenarios. Additionally, MFI-EDA's integration with Dijkstra's algorithm provides deterministic routing paths, further minimizing delay and improving overall network performance and responsiveness.

Figure 5 provides insights into the delay performance of DQRA, SPF, and MFI-EDA under varying node density conditions. While DQRA and SPF exhibited relatively higher delay values than MFI-EDA, they also demonstrated unique disadvantages such as susceptibility to quantum noise and network congestion. On the other hand, MFI-EDA leveraged its advantages in adaptability and efficiency through fuzzy logic-based routing decisions to achieve lower delay values and enhance network responsiveness. These findings underscore the importance of selecting routing protocols that effectively minimize delay and ensure efficient data transmission in diverse network environments.

6. CONCLUSION

Utilizing Mamdani Fuzzy Inference Enhanced Dijkstra's Algorithm (MFI-EDA) for energy-efficient routing in Q-FANET presents a significant stride towards addressing the unique challenges inherent in quantum-based communication systems. Q-FANET, characterized by its dynamic topology and limited resources, necessitates innovative routing solutions to optimize energy consumption while ensuring reliable data transmission. MFI-EDA emerges as a promising solution, leveraging fuzzy logic to adapt routing decisions based on real-time environmental feedback. MFI-EDA demonstrates notable achievements in mitigating energy consumption through simulation experiments, outperforming traditional routing algorithms in Q-FANET scenarios. Its ability to dynamically adjust routing paths based on environmental conditions and network dynamics showcases its efficacy in enhancing energy efficiency and reliability. Integrating MFI-EDA into Q-FANET routing protocols opens avenues for further exploration and refinement. Future research could refine the fuzzy logic mechanisms within MFI-EDA to enhance adaptability and scalability in larger Q-FANET deployments. Additionally, investigating the integration of quantum error correction mechanisms could bolster the resilience of MFI-EDA in the face of quantum noise and errors, further optimizing energy utilization and advancing the capabilities of quantum-enabled ad hoc networks.

REFERENCES

[1] M. M. Alam and S. Moh, "Q-learning-based routing inspired by adaptive flocking control for collaborative unmanned aerial vehicle swarms," *Veh. Commun.*, vol. 40, p. 100572, 2023, doi: 10.1016/j.vehcom.2023.100572.

[2] V. A. Serov, E. M. Voronov, and D. A. Kozlov, "A neuro-evolutionary synthesis of coordinated stable-effective compromises in hierarchical systems under conflict and uncertainty," *Procedia Comput. Sci.*, vol. 186, pp. 257–268, 2021, doi: 10.1016/j.procs.2021.04.145.

[3] S. Li, F. Wang, J. Gaber, and Y. Zhou, "An Optimal Relay Number Selection Algorithm for Balancing Multiple Performance in Flying Ad Hoc Networks," *IEEE Access*, vol. 8, pp. 225884–225901, 2020, doi: 10.1109/ACCESS.2020.3044502.

[4] F. H. Kumbhar and S. Y. Shin, "Innovating Multi-Objective Optimal Message Routing for Unified High Mobility Networks," *IEEE Trans. Veh. Technol.*, vol. 72, no. 5, pp. 6571–6583, 2023, doi: 10.1109/TVT.2022.3232567.

[5] Z. Wang *et al.*, "Learning to Routing in UAV Swarm Network: A Multi-Agent Reinforcement Learning Approach," *IEEE Trans. Veh. Technol.*, vol. 72, no. 5, pp. 6611–6624, 2023, doi: 10.1109/TVT.2022.3232815.

[6] J. Sharma and P. S. Mehra, "Secure communication in IOT-based UAV networks: A systematic survey," *Internet of Things (Netherlands)*, vol. 23, p. 100883, 2023, doi: 10.1016/j.iot.2023.100883.

[7] S. N. Mahapatra, B. K. Singh, and V. Kumar, "A secure multi-hop relay node selection scheme based data transmission in wireless ad-hoc network via block chain," *Multimed. Tools Appl.*, vol. 81, no. 13, pp. 18343–18373, 2022, doi: 10.1007/s11042-022-12283-7.

[8] A. S. Gruzdeva, R. N. Iurev, and I. A. Bessmertny, "Application of the text wave model to the sentiment analysis problem," *Sci. Tech. J. Inf. Technol. Mech. Opt.*, vol. 22, no. 6, pp. 1159–1165, 2022, doi: 10.17586/2226-1494-2022-22-6-1159-1165.

[9] C. K. Smith and G. Liu, "Determination of the rate constant for chain insertion into poly(methyl methacrylate)-block-poly(methacrylic acid) micelles by a fluorescence method," *Macromolecules*, vol. 29, no. 6, pp. 2060–2067, 1996, doi: 10.1021/ma951338u.

[10] M. M. Alam and S. Moh, "Joint topology control and routing in a UAV swarm for crowd surveillance," *J. Netw. Comput. Appl.*, vol. 204, p. 103427, 2022, doi: 10.1016/j.jnca.2022.103427.

[11] A. M. Rahmani *et al.*, "OLSR+: A new routing method based on fuzzy logic in flying ad-hoc networks (FANETs)," *Veh. Commun.*, vol. 36, p. 100489, 2022, doi: 10.1016/j.vehcom.2022.100489.

[12] M. Hosseinzadeh *et al.*, "A novel fuzzy trust-based secure routing scheme in flying ad hoc networks," *Veh. Commun.*, vol. 44, p. 100665, 2023, doi: 10.1016/j.vehcom.2023.100665.

[13] K. Singh and A. K. Verma, "TBCS: A Trust Based Clustering Scheme for Secure Communication in Flying Ad-Hoc Networks," *Wirel. Pers. Commun.*, vol. 114, no. 4, pp. 3173–3196, Oct. 2020, doi: 10.1007/s11277-020-07523-8.

[14] L. Le and T. N. Nguyen, "DQRA: Deep Quantum Routing Agent for Entanglement Routing in Quantum Networks," *IEEE Trans. Quantum Eng.*, vol. 3, pp. 1–12, 2022, doi: 10.1109/TQE.2022.3148667.

[15] S. Santos, F. A. Monteiro, B. C. Coutinho, and Y. Omar, "Shortest Path Finding in Quantum Networks With Quasi-Linear Complexity," *IEEE Access*, vol. 11, pp. 7180–7194, 2023, doi: 10.1109/ACCESS.2023.3237997.

[16] J. Ramkumar, R. Vadivel, and B. Narasimhan, "Constrained Cuckoo Search Optimization Based Protocol for Routing in Cloud Network," *Int. J. Comput. Networks Appl.*, vol. 8, no. 6, pp. 795–803, 2021, doi: 10.22247/ijcna/2021/210727.

[17] D. Jayaraj, J. Ramkumar, M. Lingaraj, and B. Sureshkumar, "AFSORP: Adaptive Fish Swarm Optimization-Based Routing Protocol for Mobility Enabled Wireless Sensor Network," *Int. J. Comput. Networks Appl.*, vol. 10, no. 1, pp. 119–129, 2023, doi: 10.22247/ijcna/2023/218516.

[18] L. Mani, S. Arumugam, and R. Jaganathan, "Performance Enhancement of Wireless Sensor Network Using Feisty Particle Swarm Optimization Protocol," *ACM Int. Conf. Proceeding Ser.*, pp. 1–5, Dec. 2022, doi: 10.1145/3590837.3590907.

[19] J. Ramkumar, K. S. Jeen Marseline, and D. R. Medhunhashini, "Relentless Firefly Optimization-Based Routing Protocol (RFORP) for Securing Fintech Data in IoT-Based Ad-Hoc Networks," *Int. J. Comput. Networks Appl.*, vol. 10, no. 4, pp. 668–687, Aug. 2023, doi: 10.22247/ijcna/2023/223319.

[20] J. Ramkumar, R. Vadivel, B. Narasimhan, S. Boopalan, and B. Surendren, "Gallant Ant Colony Optimized Machine Learning Framework (GACO-MLF) for Quality of Service Enhancement in Internet of Things-Based Public Cloud Networking," J. M. R. S. Tavares, J. J. P. C. Rodrigues, D. Misra, and D. Bhattacherjee, Eds., Singapore: Springer Nature Singapore, 2024, pp. 425–438. doi: 10.1007/978-981-99-5435-3_30.

[21] J. Ramkumar and R. Vadivel, "Multi-Adaptive Routing Protocol for Internet of Things based Ad-hoc Networks," *Wirel. Pers. Commun.*, vol. 120, no. 2, pp. 887–909, Apr. 2021, doi: 10.1007/s11277-021-08495-z.

[22] J. Ramkumar, C. Kumuthini, B. Narasimhan, and S. Boopalan, "Energy Consumption Minimization in Cognitive Radio Mobile Ad-Hoc Networks using Enriched Ad-hoc On-demand Distance Vector Protocol," in *2022 International Conference on Advanced Computing Technologies and Applications, ICACTA 2022*, 2022. doi: 10.1109/ICACTA54488.2022.9752899.

[23] J. Ramkumar, A. Senthilkumar, M. Lingaraj, R. Karthikeyan, and L. Santhi, "Optimal Approach for Minimizing Delays in Iot-Based Quantum Wireless Sensor Networks Using Nm-Leach Routing Protocol," *J. Theor. Appl. Inf. Technol.*, vol. 102, no. 3, pp. 1099–1111, 2024.

[24] J. Ramkumar and R. Vadivel, *CSIP—cuckoo search inspired protocol for routing in cognitive radio ad hoc networks*, vol. 556. 2017. doi: 10.1007/978-981-10-3874-7_14.

[25] R. Jaganathan, V. Ramasamy, L. Mani, and N. Balakrishnan, "Diligence Eagle Optimization Protocol for Secure Routing (DEOPSR) in Cloud-Based Wireless Sensor Network," *Res. Sq.*, 2022, doi: 10.21203/rs.3.rs-1759040/v1.

[26] J. Ramkumar, S. S. Dinakaran, M. Lingaraj, S. Boopalan, and B. Narasimhan, "IoT-Based Kalman Filtering and Particle Swarm Optimization for Detecting Skin Lesion," in *Lecture Notes in Electrical Engineering*, K. Murari, N. Prasad Padhy, and S. Kamalasadan, Eds., Singapore: Springer Nature Singapore, 2023, pp. 17–27. doi: 10.1007/978-981-19-8353-5_2.

[27] A. Senthilkumar, J. Ramkumar, M. Lingaraj, D. Jayaraj, and B. Sureshkumar, "Minimizing Energy Consumption in Vehicular Sensor Networks Using Relentless Particle Swarm Optimization Routing," *Int. J. Comput. Networks Appl.*, vol. 10, no. 2, pp. 217–230, 2023, doi: 10.22247/ijcna/2023/220737.

[28] J. Ramkumar and R. Vadivel, "Improved frog leap inspired protocol (IFLIP) – for routing in cognitive radio ad hoc networks (CRAHN)," *World J. Eng.*, vol. 15, no. 2, pp. 306–311, 2018, doi: 10.1108/WJE-08-2017-0260.

[29] P. Menakadevi and J. Ramkumar, "Robust Optimization Based Extreme Learning Machine for Sentiment Analysis in Big Data," *2022 Int. Conf. Adv. Comput. Technol. Appl. ICACTA 2022*, pp. 1–5, Mar. 2022, doi: 10.1109/ICACTA54488.2022.9753203.

[30] R. Jaganathan and V. Ramasamy, "Performance modeling of bio-inspired routing protocols in Cognitive Radio Ad Hoc Network to reduce end-to-end delay," *Int. J. Intell. Eng. Syst.*, vol. 12, no. 1, pp. 221–231, 2019, doi: 10.22266/IJIES2019.0228.22.

[31] R. Vadivel and J. Ramkumar, "QoS-enabled improved cuckoo search-inspired protocol (ICSIP) for IoT-based healthcare applications," *Inc. Internet Things Healthc. Appl. Wearable Devices*, pp. 109–121, 2019, doi: 10.4018/978-1-7998-1090-2.ch006.

[32] J. Ramkumar and R. Vadivel, "Whale optimization routing protocol for minimizing energy consumption in cognitive radio wireless sensor network," *Int. J. Comput. Networks Appl.*, vol. 8, no. 4, pp. 455–464, 2021, doi: 10.22247/ijcna/2021/209711.

[33] M. Lingaraj, T. N. Sugumar, C. S. Felix, and J. Ramkumar, "Query aware routing protocol for mobility enabled wireless sensor network," *Int. J. Comput. Networks Appl.*, vol. 8, no. 3, pp. 258–267, 2021, doi: 10.22247/ijcna/2021/209192.

[34] J. Ramkumar and R. Vadivel, "Improved Wolf prey inspired protocol for routing in cognitive radio Ad Hoc networks," *Int. J. Comput. Networks Appl.*, vol. 7, no. 5, pp. 126–136, 2020, doi: 10.22247/ijcna/2020/202977.

[35] R. Jaganathan and R. Vadivel, "Intelligent Fish Swarm Inspired Protocol (IFSIP) for Dynamic Ideal Routing in Cognitive Radio Ad-Hoc Networks," *Int. J. Comput. Digit. Syst.*, vol. 10, no. 1, pp. 1063–1074, 2021, doi: 10.12785/ijcds/100196.

[36] R. Karthikeyan and R. Vadivel, "Proficient Dazzling Crow Optimization Routing Protocol (PDCORP) for Effective Energy Administration in Wireless Sensor Networks," in *2023 International Conference on Electrical, Electronics, Communication and Computers (ELEXCOM)*, 2023, pp. 1–6. doi: 10.1109/ELEXCOM58812.2023.10370559.

[37] E. Sutcliffe and A. Beghelli, "Multiuser Entanglement Distribution in Quantum Networks Using Multipath Routing," *IEEE Trans. Quantum Eng.*, vol. 4, pp. 1–15, 2023, doi: 10.1109/TQE.2023.3329714.

[38] D. Ferrari, S. Carretta, and M. Amoretti, "A Modular Quantum Compilation Framework for Distributed Quantum Computing," *IEEE Trans. Quantum Eng.*, vol. 4, pp. 1–13, 2023, doi: 10.1109/TQE.2023.3303935.

[39] Z. Li *et al.*, "Swapping-Based Entanglement Routing Design for Congestion Mitigation in Quantum Networks," *IEEE Trans. Netw. Serv. Manag.*, vol. 20, no. 4, pp. 3999–4012, 2023, doi: 10.1109/TNSM.2023.3275815.

[40] J. Li *et al.*, "Fidelity-Guaranteed Entanglement Routing in Quantum Networks," *IEEE Trans. Commun.*, vol. 70, no. 10, pp. 6748–6763, 2022, doi: 10.1109/TCOMM.2022.3200115.

[41] L. Chen *et al.*, "Q-DDCA: Decentralized Dynamic Congestion Avoid Routing in Large-Scale Quantum Networks," *IEEE/ACM Trans. Netw.*, vol. 32, no. 1, pp. 1–14, 2023, doi: 10.1109/tnet.2023.3285093.

[42] M. S. Akhtar, G. Krishnakumar, B. Vishnu, and A. Sinha, "Fast and Secure Routing Algorithms for Quantum Key Distribution Networks," *IEEE/ACM Trans. Netw.*, vol. 31, no. 5, pp. 2281–2296, 2023, doi: 10.1109/TNET.2023.3246114.

[43] Y. Zhao and C. Qiao, "Distributed Transport Protocols for Quantum Data Networks," *IEEE/ACM Trans. Netw.*, vol. 31, no. 6, pp. 2777–2792, 2023, doi: 10.1109/TNET.2023.3262547.

[44] Y. Zeng, J. Zhang, J. Liu, Z. Liu, and Y. Yang, "Entanglement Routing Design Over Quantum Networks," *IEEE/ACM Trans. Netw.*, vol. 32, no. 1, pp. 352–367, 2023, doi: 10.1109/TNET.2023.3282560.