mathematics

MDPI

Article

A Novel Optimized Link-State Routing Scheme with Greedy and Perimeter Forwarding Capability in Flying Ad Hoc Networks

Omar Mutab Alsalami [1], Efat Yousefpoor [2], Mehdi Hosseinzadeh [3,*] and Jan Lansky [4,*]

[1] Department of Electrical Engineering, College of Engineering, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia; o.alsalami@tu.edu.sa
[2] Center of Research and Strategic Studies, Lebanese French University, Kurdistan Region, Erbil 44001, Iraq; efat.yousefpoor@lfu.edu.krd
[3] Pattern Recognition and Machine Learning Lab, Gachon University, 1342 Seongnamdaero, Sujeonggu, Seongnam 13120, Republic of Korea
[4] Department of Computer Science and Mathematics, Faculty of Economic Studies, University of Finance and Administration, 10100 Prague, Czech Republic
* Correspondence: mehdi@gachon.ac.kr (M.H.); lansky@mail.vsfs.cz (J.L.)

Abstract: A flying ad hoc network (FANET) is formed from a swarm of drones also known as unmanned aerial vehicles (UAVs) and is currently a popular research subject because of its ability to carry out complicated missions. However, the specific features of UAVs such as mobility, restricted energy, and dynamic topology have led to vital challenges for making reliable communications between drones, especially when designing routing methods. In this paper, a novel optimized link-state routing scheme with a greedy and perimeter forwarding capability called OLSR+GPSR is proposed in flying ad hoc networks. In OLSR+GPSR, optimized link-state routing (OLSR) and greedy perimeter stateless routing (GPSR) are merged together. The proposed method employs a fuzzy system to regulate the broadcast period of hello messages based on two inputs, namely the velocity of UAVs and position prediction error so that high-speed UAVs have a shorter hello broadcast period than low-speed UAVs. In OLSR+GPSR, unlike OLSR, MPR nodes are determined based on several metrics, especially neighbor degree, node stability (based on velocity, direction, and distance), the occupied buffer capacity, and residual energy. In the last step, the proposed method deletes two phases in OLSR, i.e., the TC message dissemination and the calculation of all routing paths to reduce routing overhead. Finally, OLSR+GPSR is run on an NS3 simulator, and its performance is evaluated in terms of delay, packet delivery ratio, throughput, and overhead in comparison with Gangopadhyay et al., P-OLSR, and OLSR-ETX. This evaluation shows the superiority of OLSR+GPSR.

Keywords: flying ad hoc networks (FANETs); routing; unmanned aerial vehicles (UAVs); fuzzy logic; machine learning (ML)

MSC: 68M10

check for updates

Citation: Alsalami, O.M.; Yousefpoor, E.; Hosseinzadeh, M.; Lansky, J. A Novel Optimized Link-State Routing Scheme with Greedy and Perimeter Forwarding Capability in Flying Ad Hoc Networks. *Mathematics* **2024**, *12*, 1016. https://doi.org/10.3390/math12071016

Academic Editor: Faheim Sufi

Received: 22 February 2024
Revised: 21 March 2024
Accepted: 25 March 2024
Published: 28 March 2024

1. Introduction

A flying ad hoc network (FANET) is a type of ad hoc network made up of unmanned aerial vehicles (UAVs). This network is rapidly progressing. UAVs have become a vital part of real-world applications, for example, wireless coverage, search and rescue, geographic mapping, and monitoring of smart products and transportation systems [1–3]. FANETs are a branch of mobile ad hoc networks (MANETs) and vehicular ad hoc networks (VANETs). In recent years, an attractive research area has been the participation of FANETs with other ad hoc networks, especially VANETs, to increase their efficiency and performance in smart transportation systems. In VANET, UAVs act as relay nodes to prevent frequent interruptions in communications between vehicles [4–6].

FANETs have specific features such as highly mobile nodes, low density, frequent changes in network topology, and low scalability. These features have led to challenging issues such as routing and reliable communications in FANET because the communication links are typically created and distorted quickly [7–9]. This leads to packet loss. These challenges attract the attention of many researchers. In the recent years, researchers have studied and investigated the existing routing protocols in MANETs and VANETs [10–12]. These protocols cannot make effective communications in FANETs and must be improved to obtain good performance in these networks because they ignore the specific features of FANET such as the high dynamic topology, highly mobile nodes, and limited energy resources. Hence, it is very challenging to develop a routing protocol that provides reliable communication in FANETs [13–15].

In this paper, a novel optimized link-state routing scheme with greedy and perimeter forwarding capability called OLSR+GPSR is presented in flying ad hoc networks. In this scheme, OLSR is enhanced to reduce overhead and delay in the routing process. OLSR is a proactive routing protocol whose advantage is its sensitivity to changes in the network topology. When these changes occur in the network topology, OLSR updates routing tables stored in UAVs to maintain routes formed between network nodes. However, this method faces serious challenges such as highly unstable communication links in FANET because it is very difficult to access the new network topology. Therefore, it is difficult and time consuming to choose multi-point relays (MPRs) and calculate routing paths between all UAVs. Furthermore, discovered paths will often be invalid before use in the data transmission process. This issue imposes communication costs and a high delay to rebuild these paths in FANET. In contrast, GPSR works well in terms of delay and throughput and uses a unicast technique to send data packets to the destination. As a result, its overhead is low. These advantages are our main motivation for designing OLSR+GPSR, which merges OLSR and GPSR to improve the routing process in OLSR and fix its challenges in FANET. In this regard, OLSR+GPSR, unlike OLSR, removes two phases, namely the dissemination of topology control (TC) messages and the calculation of routing paths between all network nodes due to high communication overhead, and adds the greedy and perimeter forwarding phase to the routing process. In summary, the main innovations in OLSR+GPSR are as follows:

- In OLSR+GPSR, a fuzzy system is provided to regulate the time period for broadcasting hello messages based on the two inputs, namely velocity and position prediction error. This fuzzy system causes the adaptability of the proposed method to FANET so that high-speed UAVs employ a shorter time interval to propagate hello messages, and low-speed UAVs apply a longer time period to disseminate hello messages in the network.
- In OLSR+GPSR, unlike OLSR, MPR nodes are determined based on several metrics, including neighbor degree, node stability (based on velocity, direction, and distance), the occupied buffer capacity, and residual energy. This increases the stability of MPRs, improves energy efficiency, and decreases delay and communication overhead in FANET.
- In OLSR+GPSR, two phases, namely TC message dissemination and route calculation in OLSR, have been deleted to reduce the overhead. Then, after the broadcast of hello messages and the selection of MPRs, the greedy and perimeter forwarding phase is started to find routing paths through MPR nodes, and data packets are sent to the destination.

The structure of the present paper includes the following sections: Section 2 presents the related works performed on this research subject. Section 3 explains both optimized link-state routing (OLSR) and greedy perimeter stateless routing (GPSR) in summary. Section 4 describes the network model in OLSR+GPSR. Section 5 expresses the proposed method in detail. Section 6 presents the evaluation results of OLSR+GPSR. Finally, Section 7 concludes this paper.

2. Related Works

Ref. [16] proposes a modified OLSR based on the position information of UAVs to choose MPRs efficiently. This method is compatible with the specific features of FANET, such as the high mobility of UAVs, frequent topology changes, and energy constraints. Moreover, this routing protocol does not rely only on the node connections as a standard to select MPRs but uses the estimated transmission counts (ETX) along with their position information and a weighted cost function based on residual energy and neighbor degree in the MPR selection process. In this scheme, ETX is improved based on the expiration time of links between UAVs. Note that based on this improved ETX criterion, this scheme does not select low-energy UAVs with high neighbor degrees as MPRs. In this scheme, when the estimated ETX related to a UAV is less than one and the value of its cost function (calculated based on residual energy and neighbor degree) is minimum, this UAV can be selected as an MPR node. The purpose of this method is to increase PDR and throughput and reduce latency and routing overhead. Simulation results indicate the superior performance of this protocol compared to P-OLSR and OLSR-ETX.

Ref. [17] introduces a routing protocol called P-OLSR to maintain communications between drones in FANET. P-OLSR uses GPS information to predict the positions of UAVs. Furthermore, this method is compatible with the rapid changes in the network topology and forms reliable multi-hop communications between UAVs to prevent path failures. In addition, P-OLSR has improved the formats of both control packets, namely the hello message and TC message. Spatial coordinates (longitude, altitude, height) related to each UAV are shared with other neighboring nodes to obtain the overall topology of FANET. Then, this scheme is compared with OLSR in flying ad hoc networks. Simulation results show that P-OLSR is significantly better than OLSR when frequent changes occur in the network topology.

Ref. [18] offers a new routing technique called the enhanced OLSR-ETX inspired by OLSR in FANETs. This algorithm is compatible with fast changes in the network topology and prevents path failures in the network. In enhanced OLSR-ETX, the main novelty is to use position information obtained from the global positioning system (GPS) to evaluate the expiration time of links between UAVs. It also pays attention to the residual energy of UAVs in the routing process. In addition, in this scheme, ETX is improved based on connection time and the residual energy of the nodes. Then, this improved ETX is considered for selecting MPRs, instead of the neighbor degree of UAVs in the original OLSR protocol. Therefore, if a UAV goes out of the communication range of the relevant UAV or has less energy than its other neighboring nodes, the enhanced OLSR-ETX does not select this UAV as an MPR. This method is implemented in NS3 and the results indicate its successful performance in terms of PDR, latency, and overhead compared to OLSR. In addition, the enhanced OLSR-ETX guarantees reliable communications between source and destination in FANET.

Ref. [19] provides a mobility-based weighted cluster routing scheme called MWCRSF to improve the performance of FANET. MWCRSF employs the sparrow search algorithm (SSA) and introduces a new SSA-based clustering algorithm to form clusters in the network. It considers several criteria such as the remaining energy, correlation degree, relative velocity, distance between each UAV and BS, and centrality for picking up the most suitable cluster heads in the network. Additionally, MWCRSF presents an efficient cluster maintenance algorithm to update clusters. To transmit data packets to BS in a timely and reliable manner, MWCRSF finds an optimal next-hop node with regard to link longevity, remaining energy, and the distance between UAVs and BS. Because of the use of skyline operators, MWCRSF makes multi-objective decisions, reduces the search area, and enhances efficiency when finding cluster heads in the clustering process and the next-hop node in the routing process. The simulation results state that MWCRSF has a better performance than other routing schemes in terms of cluster stability, energy efficiency, throughput, and network lifetime.

Ref. [20] provides an aerial greedy geographic routing algorithm (AGGR) in FANET. AGGR includes an adaptive hello strategy and a greedy forwarding technique based on the mobility information of UAVs. In this approach, UAVs are divided into two categories, namely active nodes and idle nodes. Then, AGGR determines the broadcast period of hello messages, related to active nodes based on their mobility information relative to their upstream UAVs, and the broadcast time of hello messages related to idle UAVs based on the relative mobility information of their neighboring UAVs. In addition, the greedy transmission process considers the relative mobility information of UAVs to increase the adaptability of AGGR to the highly dynamic network topology in FANET. In addition, it considers the time to enter the transmission area of destination as a criterion for selecting the next-hop UAV. The simulation results indicate that AGGR significantly reduces two problems, namely temporary communication blindness (TCB) and high path failures in FANET, and improves network performance.

Ref. [21] proposes an OLSR-based routing approach based on fuzzy logic called OLSR+ in flying ad hoc networks. In this scheme, UAVs use a distributed technique to find their neighbors. Additionally, in this step, real-time link duration is calculated using a new solution, which depends on link quality, distance, relative velocity, and direction. The next step introduces a fuzzy model to specify MPRs for each UAV. This model considers three fuzzy inputs, namely neighbor degree, link duration, and remaining energy, and if a UAV has more values of these three inputs than other neighboring UAVs, it gets a higher score than other UAVs in the MPR selection process and may be selected as an MPR. Then, OLSR+ makes some changes in the structure of the TC message and adds two new metrics, namely path energy and path longevity to this framework. In the last step, it obtains routing tables stored in the memory of UAVs based on the number of hops and these two new metrics (i.e., path longevity and energy). OLSR+ is run on NS3, and the results obtained from this process show that this routing scheme has an acceptable performance compared to G-OLSR and OLSR, except that it has high routing overhead.

Ref. [22] suggests a fuzzy routing technique in FANET. This technique designs an improved route discovery phase. It evaluates UAVs based on their remaining energy, direction, link quality, and stability to remove weak UAVs and find the most suitable relay nodes for broadcasting RREQs in the network. This reduces routing overhead and manages the flood of control messages disseminated for discovering new paths in the network. Then, a fuzzy system evaluates the discovered paths to choose routes with short delay and low hops. In the last step, this scheme presents three techniques, including a prevention technique and a modification and identification technique for nearly failed paths and a replacement technique for failed paths. The simulation results indicate the successful performance of this method, but it has high routing overhead.

Ref. [23] designs an energy-aware OLSR-based routing scheme for FANET. This scheme calculates two parameters, namely the ratio of the number of the delivered hello packets to the number of the sent data packets to obtain the connection quality between UAVs. Furthermore, MPRs associated with each UAV are picked up from its neighboring nodes using the firefly algorithm (FA). According to this FA-based MPR selection algorithm, four parameters, including energy, connection quality, neighbor degree, and willingness are intended to select the optimal MPRs. In this regard, if a neighboring node maximizes these four parameters, it obtains a higher score to be an MPR than the other neighbors. Then, the routes between UAVs are made based on energy and connection quality. The simulation results confirm the successful performance of this method compared to G-OLSR and OLSR in terms of latency, throughput, PDR, and energy consumption. However, it has led to a slightly more routine overhead.

Table 1 expresses the most important strengths and weaknesses of the related works.

Table 1. Benefits and drawbacks of various methods.

Technique	Strengths	Weaknesses
[16]	Improving various quality of services (QoS) requirements such as PDR, delay, throughput and energy consumption, suitable for the FANET environment, high link stability, managing routing overhead, high reliability	Low scalability
P-OLSR [17]	Improving various QoS requirements such as PLR, throughput, and delay, increasing link stability	Low scalability, low adaptability to FANET because of not calculating an adaptive hello period
OLSR-ETX [18]	Improving various QoS requirements such as PLR, throughput, delay, and energy consumption	Low scalability, lack of adaptability to the FANET environment, low reliability, low link stability
MWCRSF [19]	Improving various QoS requirements such as throughput, delay, energy consumption, and PDR, increasing network scalability, rapid cluster construction, high cluster stability	High time complexity, not calculating an adaptive hello period
AGGR [20]	Improving various QoS requirements such as PDR, throughput, delay, and energy consumption, decreasing routing overhead, high adaptability to FANET	Low scalability, not solving local optimization issue, dealing with routing loops
OLSR+ [21]	Improving various QoS requirements such as PLR, throughput, and energy consumption, high adaptability to FANET, increasing reliability	High routing overhead, low scalability
[22]	Improving various QoS requirements such as PDR, throughput, delay, and energy consumption, increasing reliability	High routing overhead, low scalability, low adaptability to FANET
[23]	Improving various QoS requirements such as PDR, throughput, energy consumption, and delay, enhancing reliability	High routing overhead, low scalability, not calculating an adaptive hello period

3. Basic Concepts

Here, two routing protocols namely, OLSR and GPSR, are briefly explained due to their use in OLSR+GPSR.

3.1. OLSR

Optimized link-state routing protocol (OLSR) [24] is the most famous proactive routing scheme in ad hoc networks due to its flexibility against frequent changes in the network topology. OLSR consists of four main phases: obtaining neighboring information, selecting MPRs, broadcasting topology control messages, and calculating routing paths.

- **Phase (1) Obtaining neighboring information**: This phase is carried out through the periodic exchange of hello messages. According to this phase, each node obtains information about its one-hop and two-hop neighbors and records this information in a neighbor table.
- **Phase (2) Selecting multi-point relays**: This phase presents an MPR selection algorithm to choose some one-hop and symmetric neighboring nodes of each node as its MPRs. These nodes are responsible for transmitting control messages and data packets in the network. When selecting these MPRs, this algorithm must meet a condition so that the MPR nodes corresponding to each node can access all of its two-hop neighbors.
- **Phase (3) Broadcasting topology control messages**: This phase is related to the regular exchange of TC messages to get the global network topology.
- **Phase (4) Calculating routing paths**: In this phase, communication paths between all nodes are made based on hop counts. Each node maintains a routing table that

contains information about routes. This routing table is regularly updated through the periodic dissemination of TC messages in the network.

FANET is a highly dynamic network, so the use of OLSR in this network faces serious challenges that must be resolved. For example, high dynamics in this network lead to instability in communication links between UAVs and frequent and fast link failures. Under such circumstances, reaching the updated network topology, selecting valid and stable MPR nodes, and calculating the updated routing tables require a long time, and the formed paths will be valid for a very short time interval [25–27]. This challenge has led to difficulty in using OLSR in FANET because it imposes a lot of delay and communication overhead on the network. In OLSR, MPRs are determined only by one parameter, i.e., the neighbor degree. This parameter alone is not enough in the dynamic environment of FANET because these nodes may not be a suitable option in terms of link lifetime, velocity, movement direction, and remaining energy, and consequently, this issue leads to short network longevity and a lot of lost packets in the network [28,29].

3.2. GPSR

The greedy perimeter stateless routing (GPSR) protocol [30] belongs to the family of geographic routing protocols. It relies on two main mechanisms, namely the hello propagation mechanism, which is done periodically, and the greedy forwarding mechanism, which is dependent on the distance to the destination. In this method, each UAV is searching for its closest neighbor to the destination to choose it as the next-hop node. Now, if this UAV is not successful in finding such a node and cannot find a neighbor closer to the destination than itself, the greedy forwarding operation will be broken down due to the occurrence of the local optimum problem. In such conditions, GPSR solves this problem using the path recovery phase, and the next-hop node is found by the right-hand rule (perimeter forwarding mode). The GPSR can reduce the delay in the routing process and increase the throughput of the network. Moreover, in GPSR, UAVs unicast data packets to the destination. Consequently, another advantage of GPSR is that the unicast forwarding mode has less routing overhead compared to the broadcast mode.

4. Network Model

Figure 1 shows the network model in OLSR+GPSR. According to this model, n UAVs, i.e., $UAV_1, UAV_2, \ldots, UAV_n$ are available in FANET. The network environment is three-dimensional, and UAVs are mobile in this network. They are connected to a positioning system so that these nodes are aware of their position (x_i, y_i, z_i) and velocity $(v_{i,x}, v_{i,y}, v_{i,z})$ at any moment. Furthermore, IEEE 802.11b standard supports wireless communications in the MAC layer. UAVs communicate with the ground station (GS) via direct or multi-hop communications.

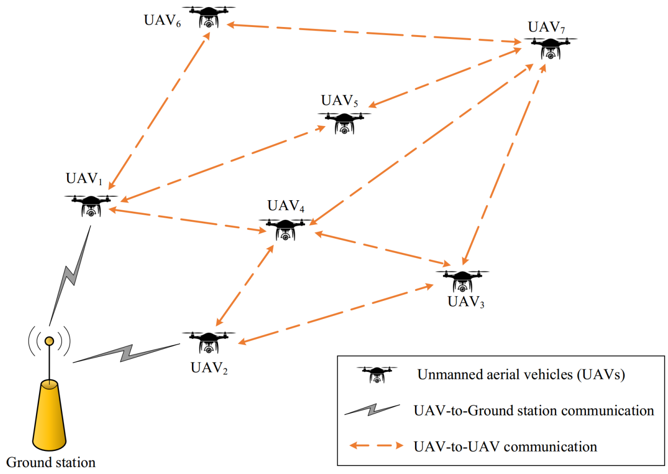

UAV₆
UAV₇
UAV₅
UAV₁
UAV₄
UAV₃
UAV₂
Ground station
Unmanned aerial vehicles (UAVs)
UAV-to-Ground station communication
UAV-to-UAV communication

Figure 1. Network model in OLSR+GPSR.

5. Proposed Method

In this section, a novel optimized link-state routing scheme with greedy and perimeter forwarding capability called OLSR+GPSR is presented in flying ad hoc networks. The proposed method includes three phases:

- Propagating hello messages based on a fuzzy system;
- Selecting MPR nodes;
- Greedy and perimeter forwarding techniques.

5.1. Propagating Hello Messages Based on a Fuzzy System

In this phase, UAVs propagate hello messages to their surrounding nodes, and at the same time, they store the information obtained from the hello messages of their surrounding nodes in a neighbor table. See Figure 2.

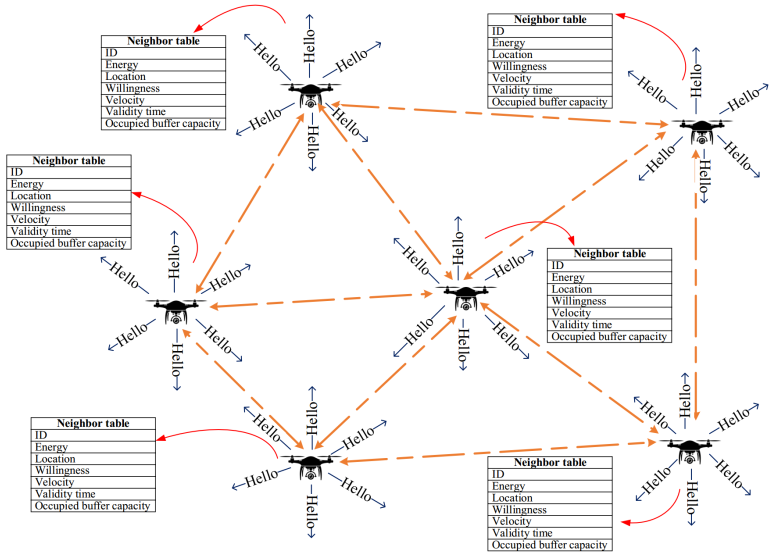
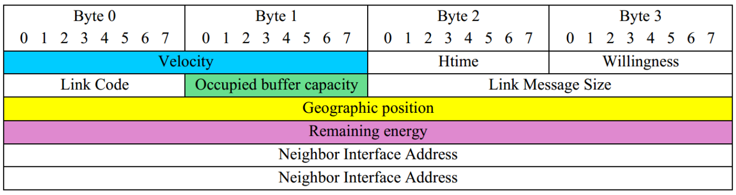


Figure 2. Broadcasting hello messages in FANET.

There are three important points in the hello propagation process:

- The propagation range of the hello message is local. Hence, this message should not be broadcast in the entire network.
- Compared with OLSR, the hello message in OLSR+GPSR includes four additional fields, i.e., geographic location (yellow field), remaining energy (pink field), occupied buffer capacity (green field), and velocity (blue field). See the format of this message in Figure 3.



Byte 0	Byte 1	Byte 2	Byte 3
0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7
Velocity		Htime	Willingness
Link Code	Occupied buffer capacity	Link Message Size	
Geographic position			
Remaining energy			
Neighbor Interface Address			
Neighbor Interface Address			

Figure 3. Hello message format in OLSR+GPSR.

- The hello propagation process is repeated periodically at a specific time interval so that the information of the neighbor table is always up to date. In OLSR+GPSR, each UAV adjusts its propagation interval according to a fuzzy system. Note that fuzzy logic is a useful tool for presenting mathematical sentences and reasoning propositions. Fuzzy logic moves from the absolute truth to partial truth and follows fuzzy set theory, which is a generalized version of classical set theory [31]. To achieve this fuzzy system, the initial propagation interval of hello messages in OLSR+GPSR is equal to one second. Then, the proposed fuzzy system decides whether to decrease, increase, or fix this interval. This fuzzy system follows the structure of the Mamdani fuzzy control system [32] and includes two input elements, one output, and a rule base.

5.1.1. Fuzzy Inputs

The fuzzy system first receives its inputs and uses a specific membership function to convert them into fuzzy sets and determine their membership degree. The fuzzy system applied in OLSR+GPSR has two inputs called velocity and position prediction error.

- **Velocity ($|V_i|$)**: The time interval associated with hello messages and the velocity of UAV_i are inversely related to each other. Therefore, if UAV_i has a high velocity, then it should propagate its hello messages in a shorter time period to update its status information in a timely manner because its communication links are highly unstable, whereas if UAV_i has a low velocity, it considers a longer time period for broadcasting the hello message due to more stable communication links. Note that UAV_i is connected to the positioning system and consequently can access its velocity vector $\left(v_{i,x}, v_{i,y}, v_{i,z}\right)$ at any moment. Therefore, its velocity length ($|V_i|$) is calculated through Equation (1).

$$|V_i| = \sqrt{\left(v_{i,x}\right)^2 + \left(v_{i,y}\right)^2 + \left(v_{i,z}\right)^2} \tag{1}$$

Additionally, a normalization operation is performed on $|V_i|$ using Equation (2) to limit this value to $[0, 1]$.

$$\left|V_i^{norm}\right| = \frac{|V_i| - |V_{\min}|}{|V_{\max}| - |V_{\min}|} \tag{2}$$

so that $|V_{\max}|$ and $|V_{\min}|$ are the highest and lowest allowed velocity of UAV_i in the network, respectively. Furthermore, $\left|V_i^{norm}\right|$ represents the normalized value of $|V_i|$. Now, UAV_i must convert $\left|V_i^{norm}\right|$ into a fuzzy parameter according to the membership diagram presented in Figure 4. According to Figure 4, this fuzzy parameter contains three modes, i.e., slow, medium, and fast.

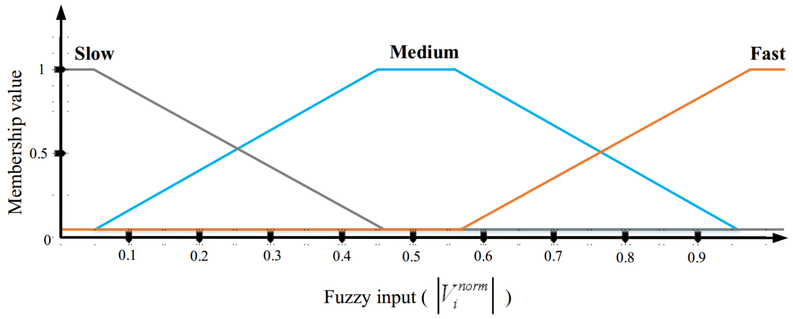


Figure 4. Fuzzy membership diagram of $\left|V_i^{norm}\right|$.

- **Position prediction error (e_i)**: The propagation interval of hello messages and e_i have an opposite relationship with each other, so if UAV_i estimates its future position with a large error, it should shorten its hello propagation interval to reduce the position prediction error. Note that UAV_i has access to its real position through the positioning device. Additionally, it uses the movement equation to estimate its new position. In OLSR+GPSR, each UAV_i assumes that its velocity vector $\left(v_{i,x}, v_{i,y}, v_{i,z}\right)$ is fixed in the time period between two consecutive hello messages. As a result, UAV_i can calculate its future position in the next hello message based on its previous position $\left(x_i^{previous}, y_i^{previous}, z_i^{previous}\right)$ and its current velocity vector $\left(v_{i,x}, v_{i,y}, v_{i,z}\right)$.

$$\tilde{x}_i^{next} = v_{i,x}\, HT_i + x_i^{previous} \tag{3}$$

$$\tilde{y}_i^{next} = v_{i,y}\, HT_i + y_i^{previous} \tag{4}$$

$$\tilde{z}_i^{next} = v_{i,z}\,HT_i + z_i^{previous} \tag{5}$$

where $\left(\tilde{x}_i^{next}, \tilde{y}_i^{next}, \tilde{z}_i^{next}\right)$ is the estimated spatial coordinates of UAV_i when receiving the next hello message, and HT_i expresses the hello propagation interval.
Now, the error value (e_i) between the actual position, i.e., $\left(x_i^{next}, y_i^{next}, z_i^{next}\right)$, and the estimated position, i.e., $\left(\tilde{x}_i^{next}, \tilde{y}_i^{next}, \tilde{z}_i^{next}\right)$, is obtained from Equation (6).

$$e_i = \sqrt{\left(x_i^{next} - \tilde{x}_i^{next}\right)^2 + \left(y_i^{next} - \tilde{y}_i^{next}\right)^2 + \left(z_i^{next} - \tilde{z}_i^{next}\right)^2} \tag{6}$$

Finally, e_i is normalized according to Equation (7) to limit this value to $[0,1]$.

$$e_i^{norm} = \frac{e_i}{e_{\max}} \tag{7}$$

where $e_{\max}$ is the maximum error of the estimated position in the network, and e_i^{norm} represents the normalized value of e_i. Now, UAV_i must convert e_i^{norm} into a fuzzy parameter according to the membership diagram presented in Figure 5. According to Figure 5, this fuzzy parameter involves three states, i.e., *low*, *medium*, and *high*.

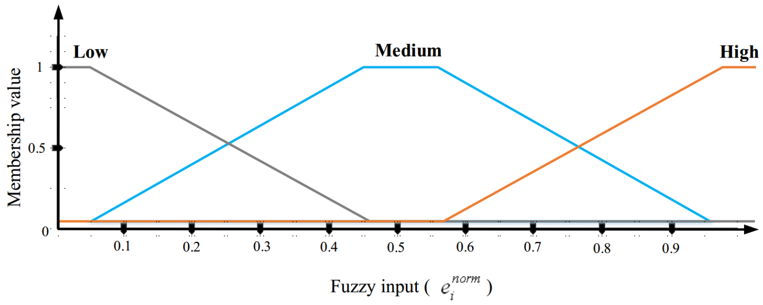


Figure 5. Fuzzy membership diagram of e_i^{norm}.

5.1.2. Fuzzy Output

The output of this fuzzy system is the change rate (Δt_i) of the hello propagation interval for each UAV_i. After determining this change rate, the hello propagation interval (HT_i) will be equal to $HT_i + \Delta t_i$. If the change rate Δt_i experiences negative growth, HT_i will be shorter, and hello messages will be sent faster. In contrast, if this change rate Δt_i grows positively, HT_i will be longer, and hello messages will be sent to neighbors at longer intervals. Δt_i is converted to a crisp value using a defuzzification process called averaging. The membership diagram of this fuzzy parameter is presented in Figure 6. According to this figure, the change rate of the hello propagation interval includes five states, i.e., *increment*, *changeless*, *low decrement*, *decrement*, and *high decrement*.

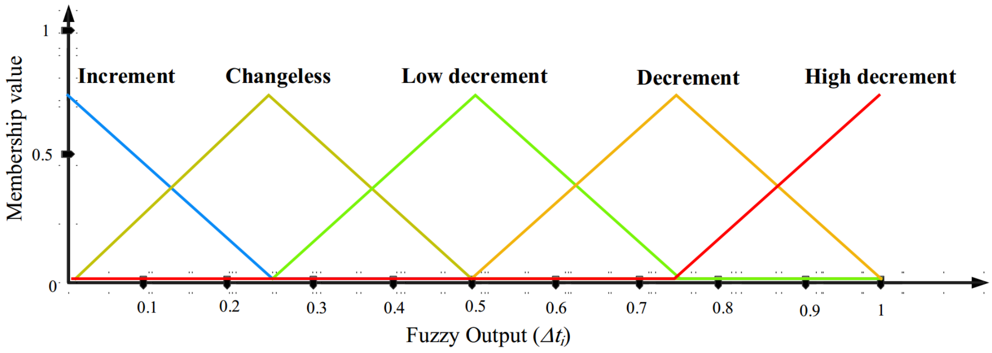


Figure 6. Fuzzy membership diagram of Δt_i.

5.1.3. Rule Base

Here, fuzzy rules (IF-THEN rules) are introduced in the proposed fuzzy system. This rule base contains nine fuzzy rules as shown in Table 2. To clarify the concept of each rule, the first rule is defined as follows.

Rule 1: IF the velocity of UAV_i is *slow* **AND** the position prediction error is *low*, **THEN** the change rate of the hello propagation interval will be *increment*.

Algorithm 1 describes the pseudo code related to this phase. Note that the time complexity of this algorithm is equal to $O(N_i)$ so that N_i indicates the number of adjacent drones of UAV_i.

Algorithm 1 Hello message broadcast

```
Input: $UAV_i$: $i = 1, 2, ..., n$
    $HT_i = 1s$: Default period of time for propagating hello messages in $UAV_i$.
    $ST$: Simulation time
    $N_i$: The number of adjacent drones of $UAV_i$.
Output: $HT_i$: The period of time for propagating hello messages in $UAV_i$.
    Begin
    repeat
        if $ST$ mod $HT_i = 0$ then
            $\mathbf{UAV_i}$: Connect to a positioning system such as GPS;
            $\mathbf{UAV_i}$: Get the next position $\left(x_i^{next}, y_i^{next}, z_i^{next}\right)$ and its velocity $\left(v_{i,x}, v_{i,y}, v_{i,z}\right)$ from GPS;
            $\mathbf{UAV_i}$: Record its position, velocity, energy, and occupied buffer capacity in the hello message;
            $\mathbf{UAV_i}$: Propagate this hello message for adjacent UAVs;
            $\mathbf{UAV_i}$: Obtain the size of its velocity, i.e., $|V_i|$ according to Equation (1);
            $\mathbf{UAV_i}$: Get the normalized $|V_i|$ (i.e., $\left|V_i^{norm}\right|$) based on Equation (2);
            $\mathbf{UAV_i}$: Transform $\left|V_i^{norm}\right|$ into a fuzzy variable in accordance with the membership graph in Figure 4;
            $\mathbf{UAV_i}$: Approximate the next position $\left(\tilde{x}_i^{next}, \tilde{y}_i^{next}, \tilde{z}_i^{next}\right)$ based on Equations (3)–(5);
            $\mathbf{UAV_i}$: Compute the error value ($e_i$) between $\left(x_i^{next}, y_i^{next}, z_i^{next}\right)$ and $\left(\tilde{x}_i^{next}, \tilde{y}_i^{next}, \tilde{z}_i^{next}\right)$ based on Equation (6);
            $\mathbf{UAV_i}$: Obtain the normalized error value ($e_i^{norm}$) in accordance with Equation (7);
            $\mathbf{UAV_i}$: Transform $e_i^{norm}$ into a fuzzy variable in accordance with the membership graph in Figure 5;
            $\mathbf{UAV_i}$: Get the broadcast period change rate ($\Delta t_i$) based on Figure 6 and the rule base in Table 1;
            $\mathbf{UAV_i}$: Obtain $HT_i = HT_i + \Delta t_i$;
        end if
        for $j = 1$ to $N_i$ do
            if $UAV_i$ gets a new hello message from $UAV_j$ then
                $\mathbf{UAV_i}$: Record the information of $UAV_j$ in its neighbor table;
            end if
        end for
        $ST = ST - 1$;
    until $ST = 0$
    End
```

Table 2. Rule base.

Number	**Fuzzy Inputs**		**Fuzzy Output**
	$\left\|\boldsymbol{V_i^{norm}}\right\|$	$\boldsymbol{e_i^{norm}}$	$\boldsymbol{\Delta t_i}$
1	Slow	Low	Increment
2	Slow	Medium	Changeless
3	Slow	High	Low decrement
4	Medium	Low	Changeless
5	Medium	Medium	Low decrement
6	Medium	High	Decrement
7	Fast	Low	Low decrement
8	Fast	Medium	Decrement
9	Fast	High	High decrement

5.2. Selecting MPR Nodes

In OLSR+GPSR, the MPR selection algorithm is used to obtain the information of MPRs associated with each UAV_i. The task of MPRs in OLSR+GPSR is different from their task in OLSR. In OLSR, these nodes broadcast TC messages throughout the network so that network topology information is globally available to all UAVs to calculate routing paths and update their routing table. This work is extremely costly in terms of communication overhead and energy consumption and increases delay in the routing process. Furthermore, this information is often invalid quickly due to the highly mobility of UAVs in FANET, and the routes formed between UAVs are often broken even without use in the data transmission process and must be updated again. According to the mentioned points, OLSR+GPSR seeks to reduce the tasks of MPRs to lower energy consumption and improve network lifetime. In this regard, MPRs do not need to propagate TC messages throughout the network and UAVs do not need to calculate all routes in the network. In OLSR+GPSR, MPRs are responsible for sending data packets to the destination in a greedy manner. They use the routing technique provided in GPSR to reduce routing overhead and delay. In this regard, each UAV_i must specify its own MPR set, i.e., $S_i^{MPR} = \left\{UAV_{1,i}^{MPR}, UAV_{2,i}^{MPR}, ..., UAV_{r,i}^{MPR}, ..., UAV_M^{MPR}\right\}$ so that UAV_i has at least one communication path to its two-hop neighbors through its MPRs. Note that the number of $UAV_{r,i}^{MPR}$ in S_i^{MPR} is equal to M. To achieve this MPR set, UAV_i must first make two sets, namely a one-hop neighbor set $N_i^{1-hop} = \left\{n_{1,i}^{1-hop}, n_{2,i}^{1-hop}, ..., n_{p,i}^{1-hop}, ..., n_{P,i}^{1-hop}\right\}$ and a two-hop neighbor set $N_i^{2-hop} = \left\{n_{1,i}^{2-hop}, n_{2,i}^{2-hop}, ..., n_{q,i}^{2-hop}, ..., n_{Q,i}^{2-hop}\right\}$. This process includes two steps:

- Calculating the score of one-hop neighboring nodes;
- MPR selection algorithm.

5.2.1. Calculating the Score of One-Hop Neighboring Nodes

In OLSR, each UAV_i determines its MPRs ($UAV_{r,i}^{MPR}$, so that $r = 1, 2, ..., M$) only based on the neighbor degree, whereas it is not enough because MPRs such as $UAV_{r,i}^{MPR}$ may go out of the communication range of UAV_i in a very short time due to the high mobility of UAVs in FANET. As a result, communication links are quickly cut and the information about the entire network topology must be updated. This is associated with a lot of communication overhead, high energy consumption, and a lot of delay. For this reason, this problem has been solved in OLSR+GPSR. Hence, in addition to the neighbor degree, each UAV_i should pay attention to node stability, occupied buffer capacity, and residual energy to choose each $UAV_{r,i}^{MPR}$. To achieve this goal, UAV_i calculates a score S_p for each one-hop neighbor $n_{p,i}^{1-hop}$ (where $p = 1, 2, ..., P$) based on these four components.

- **Neighbor degree ($\boldsymbol{ND_p}$)**: In OLSR+GPSR, similar to OLSR, UAV_i pays attention to the neighbor degree of $n_{p,i}^{1-hop}$ (i.e., ND_p), which is equal to the number of its one-hop neighbor to determine $S_i^{MPR} = \left\{UAV_{1,i}^{MPR}, UAV_{2,i}^{MPR}, ..., UAV_{r,i}^{MPR}, ..., UAV_R^{MPR}\right\}$. ND_p is obtained from its neighbor table. This component represents the number two-hop neighboring UAVs of UAV_i that $n_{p,i}^{1-hop}$ covers in the network. Thus, UAV_i can choose suitable one-hop neighbors such as $n_{p,i}^{1-hop}$ as $UAV_{r,i}^{MPR}$ because $n_{p,i}^{1-hop}$ with high ND_p has more communication links with other nodes and creates stable paths in the network. This decreases routing overhead. ND_p and S_p are directly related to each other. Therefore, UAV_i increases the score of $n_{p,i}^{1-hop}$, which has a higher ND_p so that $n_{p,i}^{1-hop}$ gets a higher priority to be selected as $UAV_{r,i}^{MPR}$. UAV_i uses Equation (8) to perform a normalization operation on ND_p corresponding to $n_{p,i}^{1-hop}$ (where $p = 1, 2, ..., P$) so that this element is limited to $[0, 1]$.

$$ND_p^{norm} = \frac{ND_p}{ND_{\max}} \tag{8}$$

Here, ND_p^{norm} is the normalized value of ND_p, and $ND_{\max}$ is equal to the maximum degree of the one-hop neighbors in $N_i^{1-hop} = \left\{n_{1,i}^{1-hop}, n_{2,i}^{1-hop}, ..., n_{p,i}^{1-hop}, ..., n_{P,i}^{1-hop}\right\}$ (i.e., $ND_{\max} = \max\limits_{\substack{n_{p,i}^{1-hop} \in N_i^{1-hop} \\ AND \\ p=1,2,...,P}} \{ND_p\}$).

- **Node stability (NS_p)**: In OLSR+GPSR, UAV_i measures the stability of $n_{p,i}^{1-hop}$ (i.e., NS_p) based on the effective distance, relative velocity, and movement direction of $n_{p,i}^{1-hop}$ to determine the most stable $UAV_{r,i}^{MPR}$ in S_i^{MPR}. NS_p is calculated according to Equation (9).

$$NS_p = \ell_1 ED_{ip} + \ell_2 \left(1 - \left|\vec{V}_{ip}^{norm}\right|\right) + \ell_3 \left(1 - \theta_{ip}^{norm}\right) \tag{9}$$

Here, $\sum\limits_{i=1}^{3} \ell_i = 1$ and ℓ_1, ℓ_2, and ℓ_3 are the weight coefficients to determine the effect of each component (i.e., effective distance, relative velocity, and movement direction) on NS_p. In OLSR+GPSR, the effect of all three parameters, including effective distance (ED_{ip}), relative velocity (V_{ip}), and movement direction (θ_{ip}), on NS_p is the same (i.e., $\ell_1 = \ell_2 = \ell_3$). These three parameters are explained below.

  - **Effective distance (ED_{ip})**: The distance between UAV_i and $n_{p,i}^{1-hop}$ has an important effect on node stability. If this distance is close to the effective area of UAV_i, $n_{p,i}^{1-hop}$ is a stable neighbor. Otherwise, it is not suitable to choose this node as the MPR. In Figure 7, the white area shows the effective area of UAV_i.

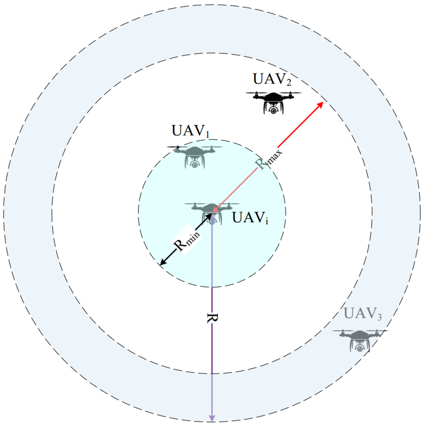


Figure 7. Effective distance between UAV_i and $n_{p,i}^{1-hop}$.

In general, the stability of $n_{p,i}^{1-hop}$ and its effective distance have a direct relationship with each other. Hence, if the distance between UAV_i and $n_{p,i}^{1-hop}$ is very short (i.e., $d_{ip} < R_{\min}$), the selection of $n_{p,i}^{1-hop}$ as $UAV_{r,i}^{MPR}$ is not a useful option because $UAV_{r,i}^{MPR}$ plays the role of the next-hop node in the routing process, and its proximity to UAV_i will cause an excessive increase in the number of hops in the route. In Figure 7, the distance between UAV_i and UAV_1 is less than $R_{\min}$. Thus, UAV_1 should not be selected as an MPR node. On the other hand, if the distance between UAV_i and $n_{p,i}^{1-hop}$ is very long (i.e., $d_{ip} > R_{\max}$), the selection of $n_{p,i}^{1-hop}$ as $UAV_{r,i}^{MPR}$ is not appropriate because it may go out of the communication range of UAV_i very quickly. In Figure 7, the distance between UAV_i and UAV_3 is more than $R_{\max}$. Thus, UAV_3 should not be selected as an MPR node, whereas if the distance between UAV_i and $n_{p,i}^{1-hop}$ is in the effective range (i.e., $R_{\min} \leq d_{ip} \leq R_{\max}$), the selection of $n_{p,i}^{1-hop}$ as $UAV_{r,i}^{MPR}$ is a very good option. In Figure 7, the distance between UAV_i and UAV_2 is between $R_{\min}$ and $R_{\max}$. Thus, UAV_2 is a stable neighbor and can be selected as an MPR node. This issue is stated in Equation (10).

$$ED_{ip} = \begin{cases} \frac{d_{ip}}{R_{\min}}, & d_{ip} < R_{\min} \\ 1, & R_{\min} \leq d_{ip} \leq R_{\max} \\ \frac{d_{ip} - R_{\max}}{R - R_{\max}}, & R_{\max} < d_{ip} \leq R \end{cases} \tag{10}$$

where R is the communication radius of UAVs, and d_{ip} is the Euclidean distance between $n_{p,i}^{1-hop}$ and UAV_i obtained from Equation (11).

$$d_{ip} = \sqrt{\left(x_i^{current} - x_p^{current}\right)^2 + \left(y_i^{current} - y_p^{current}\right)^2 + \left(z_i^{current} - z_p^{current}\right)^2} \tag{11}$$

– **Relative velocity ($\left|\vec{V}_{ip}\right|$)**: The relative velocity of $n_{p,i}^{1-hop}$ with regard to UAV_i is an important factor in determining node stability. This factor determines whether these two UAVs (i.e., UAV_i and $n_{p,i}^{1-hop}$) fly at a relatively similar velocity. If this factor approaches zero, then the velocities of UAV_i and $n_{p,i}^{1-hop}$ are similar. Therefore, they do not leave the neighboring area of each other quickly. In this case, $n_{p,i}^{1-hop}$ is a stable one-hop neighbor, which can create stable paths and decrease the number of path failures. Now, if the relative velocity is large, $n_{p,i}^{1-hop}$ has weak stability. The length of the relative velocity of $n_{p,i}^{1-hop}$ with regard to UAV_i (i.e., $\left|\vec{V}_{ip}\right|$) is calculated according to Equation (12).

$$\left|\vec{V}_{ip}\right| = \sqrt{\left(v_{i,x} - v_{p,x}\right)^2 + \left(v_{i,y} - v_{p,y}\right)^2 + \left(v_{i,z} - v_{p,z}\right)^2} \tag{12}$$

where $\left(v_{i,x}, v_{i,y}, v_{i,z}\right)$ and $\left(v_{p,x}, v_{p,y}, v_{p,z}\right)$ are the velocity vectors of UAV_i and $n_{p,i}^{1-hop}$, respectively.
$\left|\vec{V}_{ip}\right|$ is normalized through Equation (13).

$$\left|\vec{V}_{ip}^{norm}\right| = \frac{\left|\vec{V}_{ip}\right|}{\left|\vec{V}_{\max}\right|} \tag{13}$$

Here, $\left|\vec{V}_{ip}^{norm}\right|$ is the normalized value of $\left|\vec{V}_{ip}\right|$. $\left|\vec{V}_{\max}\right|$ is the maximum relative velocity of one-hop neighbors in $N_i^{1-hop} = \left\{n_{1,i}^{1-hop}, n_{2,i}^{1-hop}, \ldots, n_{p,i}^{1-hop}, \ldots, n_{P,i}^{1-hop}\right\}$, i.e., $\left|\vec{V}_{\max}\right| = \max\limits_{\substack{n_{p,i}^{1-hop} \in N_i^{1-hop} \\ AND \\ p=1,2,\ldots,P}} \left\{\left|\vec{V}_{ip}\right|\right\}$.

- **Movement direction (θ_{ip})**: The direction of $n_{p,i}^{1-hop}$ relative to UAV_i (i.e., θ_{ip}) is an effective factor to evaluate the stability of $n_{p,i}^{1-hop}$. If $n_{p,i}^{1-hop}$ moves in the opposite direction to UAV_i, $n_{p,i}^{1-hop}$ is known as an unstable node, and vice versa. θ_{ip} is calculated based on Equation (14).

$$\theta_{ip} = \cos^{-1}\left(\frac{v_{i,x}v_{p,x}+v_{i,y}v_{p,y}+v_{i,z}v_{p,z}}{\sqrt{(v_{i,x})^2+(v_{i,y})^2+(v_{i,z})^2}\times\sqrt{(v_{p,x})^2+(v_{p,y})^2+(v_{p,z})^2}}\right), \quad 0 \le \theta_{ip} \le \pi \tag{14}$$

where $(v_{i,x}, v_{i,y}, v_{i,z})$ and $(v_{p,x}, v_{p,y}, v_{p,z})$ are the velocity vectors of UAV_i and $n_{p,i}^{1-hop}$, respectively.

Now, θ_{ip} is normalized based on Equation (15).

$$\theta_{ip}^{norm} = \frac{\theta_{ip}}{\pi} \tag{15}$$

So θ_{ip}^{norm} is the normalized value of θ_{ip} in $[0,1]$.

- **Occupied buffer capacity (NB_p)**: In OLSR+GPSR, the occupied buffer capacity of $n_{p,i}^{1-hop}$ (i.e., NB_p) is used to determine $UAV_{r,i}^{MPR}$ in S_i^{MPR}. NB_p is calculated based on the ratio of the number of data packets in the buffer queue to the size of the buffer. This parameter is exchanged through hello messages. There is an inverse relationship between NB_p and S_p because if NB_p is a large value, it shows the presence of congestion and high traffic in $n_{p,i}^{1-hop}$. This congestion decreases energy efficiency and network lifetime. As a result, UAV_i gives a lower score to $n_{p,i}^{1-hop}$ with a higher NB_p so that $n_{p,i}^{1-hop}$ has a lower priority to be selected as $UAV_{r,i}^{MPR}$. UAV_i uses Equation (16) to perform a normalization operation on NB_p corresponding to $n_{p,i}^{1-hop}$ $(p = 1, 2, \ldots, P)$ to limit this element to $[0,1]$.

$$NB_p^{norm} = \frac{NB_p}{NB_{\max}} \tag{16}$$

Here, NB_p^{norm} is the normalized value of NB_p, and $NB_{\max}$ is equal to the maximum occupied buffer capacity of one-hop neighbors in N_i^{1-hop} (i.e., $NB_{\max} = \max\limits_{\substack{n_{p,i}^{1-hop} \in N_i^{1-hop} \\ AND \\ p=1,2,\ldots,P}} \{NB_p\}$).

- **Residual energy (NE_p)**: In OLSR+GPSR, UAV_i applies the remaining energy of $n_{p,i}^{1-hop}$ (i.e., NE_p) to determine $UAV_{r,i}^{MPR}$ in S_i^{MPR}. When deciding on MPRs, this component is really vital for calculating the score of $n_{p,i}^{1-hop}$ because each MPR node such as $UAV_{r,i}^{MPR}$ experiences higher routing overhead than other nodes. Hence, its energy consumption is high. NE_p is obtained from the neighbor table, and there is a direct relationship between NE_p and S_p. If NE_p is high, $n_{p,i}^{1-hop}$ is more stable and has priority to be selected as $UAV_{r,i}^{MPR}$. UAV_i normalizes NE_p corresponding to $n_{p,i}^{1-hop}$ $(p = 1, 2, \ldots, P)$ through Equation (17) so that this parameter is limited to $[0,1]$.

$$NE_p^{norm} = \frac{NE_p}{NE_{\max}} \tag{17}$$

Here, NE_p^{norm} is the normalized value of NE_p, and $NE_{\max}$ is equal to the maximum remaining energy in $N_i^{1-hop} = \left\{n_{1,i}^{1-hop}, n_{2,i}^{1-hop}, \ldots, n_{p,i}^{1-hop}, \ldots, n_{P,i}^{1-hop}\right\}$ (i.e., $NE_{\max} = \max\limits_{\substack{n_{p,i}^{1-hop} \in N_i^{1-hop} \\ AND \\ p=1,2,\ldots,P}} \{NE_p\}$).

Therefore, the score of each $n_{p,i}^{1-hop}$ (S_p) is obtained from Equation (18).

$$S_p = \lambda_1 ND_p^{norm} + \lambda_2 NS_p^{norm} + \lambda_3\left(1 - NB_p^{norm}\right) + \lambda_4 NE_p^{norm} \tag{18}$$

Here, λ_1, λ_2, λ_3, and λ_4 are the weight coefficients to express the effect of each parameter (i.e., neighbor degree, node stability, occupied buffer capacity, and remaining energy) on S_p and $\sum\limits_{i=1}^{4} \lambda_i = 1$. In OLSR+GPSR, the effect of all four parameters is considered to be the same (i.e., $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4$).

5.2.2. MPR Selection Algorithm

In OLSR+GPSR, the MPR selection algorithm includes the following steps:

1. UAV_i makes three sets, $S_i^{MPR} = \{\}$, $N_i^{1-hop} = \left\{n_{1,i}^{1-hop}, n_{2,i}^{1-hop}, \ldots, n_{p,i}^{1-hop}, \ldots, n_{P,i}^{1-hop}\right\}$, and $N_i^{2-hop} = \left\{n_{1,i}^{2-hop}, n_{2,i}^{2-hop}, \ldots, n_{q,i}^{2-hop}, \ldots, n_{Q,i}^{2-hop}\right\}$.

2. UAV_i calculates S_p for each $n_{p,i}^{1-hop}$ in N_i^{1-hop} according to Section 5.2.1.

3. If UAV_i finds a $n_{p,i}^{1-hop}$ with $Willingness = WILL - ALWAYS$, it inserts $n_{p,i}^{1-hop}$ into S_i^{MPR}. Willingness is an index recorded in hello messages to determine the willingness of UAVs to transmit traffic, and its value is limited to $[1,7]$. Nodes with $Willingness = WILL - NEVER$ are never selected as an MPR, whereas nodes with $Willingness = WILL - ALWAYS$ always become an MPR.

4. If UAV_i and $n_{q,i}^{2-hop}$ have only one communication path through $n_{p,i}^{1-hop}$, UAV_i should select $n_{p,i}^{1-hop}$ as a $UAV_{r,i}^{MPR}$.

5. UAV_i is responsible for removing all one-hop neighboring nodes $n_{p,i}^{1-hop}$ determined as $UAV_{r,i}^{MPR}$ from N_i^{1-hop}.

6. UAV_i is responsible for removing all two-hop neighbors $n_{q,i}^{2-hop}$, which have at least one communication path to UAV_i through MPRs, from N_i^{2-hop}.

7. This step is repeated until N_i^{2-hop} is empty (i.e., $N_i^{2-hop} = \{\}$):

- UAV_i sorts the members of N_i^{1-hop} (i.e., $n_{p,i}^{1-hop}$ so that $p = 1, 2, \ldots, P$) based on S_p.
- If some of $n_{p,i}^{1-hop}$ have the same score, those with a higher neighbor degree have a higher priority.
- UAV_i chooses a $n_{p,i}^{1-hop}$ with the highest score (the first member of N_i^{1-hop}, i.e., $n_{1,i}^{1-hop}$) and inserts it as $UAV_{r,i}^{MPR}$ into S_i^{MPR}.
- UAV_i removes $n_{1,i}^{1-hop}$ determined as $UAV_{r,i}^{MPR}$ from N_i^{1-hop}.
- All IDs of two-hop neighbors (such as $n_{q,i}^{2-hop}$) that have at least one communication path to UAV_i through MPRs in S_i^{MPR} are removed from N_i^{2-hop}.

The pseudo-code related to this phase is described in Algorithm 2. The time complexity of this algorithm is $O(QP\log P)$ so that P and Q indicate the sizes of two sets N_i^{1-hop} and N_i^{2-hop}, respectively.

Algorithm 2 The selection process of multi-point relays (MPRs)

```
Input: $UAV_i$: $i = 1, 2, ..., n$
    $HT_i = 1s$: Default period of time for propagating hello messages in $UAV_i$.
    $UAV_{r,i}^{MPR}$: $r = 1, 2, ..., M$
    $n_{p,i}^{1-hop}$: A one-hop neighbor of $UAV_i$.
    $S_p$: The score of $n_{p,i}^{1-hop}$.
    $N_i^{1-hop} = \left\{ n_{1,i}^{1-hop}, n_{2,i}^{1-hop}, ..., n_{p,i}^{1-hop}, ..., n_{P,i}^{1-hop} \right\}$
    $N_i^{2-hop} = \left\{ n_{1,i}^{2-hop}, n_{2,i}^{2-hop}, ..., n_{q,i}^{2-hop}, ..., n_{Q,i}^{2-hop} \right\}$
Output: $S_i^{MPR} = \left\{ UAV_{1,i}^{MPR}, UAV_{2,i}^{MPR}, ..., UAV_{r,i}^{MPR}, ..., UAV_M^{MPR} \right\}$
    Begin
    for $p = 1$ to $P$ do
        UAV_i: Obtain $ND_p$ corresponding to $n_{p,i}^{1-hop}$ from its neighbor table;
        UAV_i: Normalized $ND_p$ according to Equation (8);
        UAV_i: Gain $NS_p$ corresponding to $n_{p,i}^{1-hop}$ based on Equation (9);
        UAV_i: Extract $NB_p$ corresponding to $n_{p,i}^{1-hop}$ from its neighbor table;
        UAV_i: Calculate the normalized $NB_p$ (i.e., $NB_p^{norm}$) based on Equation (16);
        UAV_i: Obtain $NE_p$ corresponding to $n_{p,i}^{1-hop}$ from its neighbor table;
        UAV_i: Get the normalized $NE_p$ (i.e., $NE_p^{norm}$) in accordance with Equation (17);
        UAV_i: Compute the score ($S_p$) related to $n_{p,i}^{1-hop}$ according to Equation (18);
    end for
    UAV_i: Consider an empty set $S_i^{MPR} = \{\}$ to record the selected MPRs;
    for $q = 1$ to $Q$ do
        for $p = 1$ to $P$ do
            if $n_{p,i}^{1-hop}$ in $N_i^{1-hop}$ has $Willingness = WILL - ALWAYS$ then
                UAV_i: Determine this $n_{p,i}^{1-hop}$ as a member of $S_i^{MPR}$;
                UAV_i: Delete this $n_{p,i}^{1-hop}$ from $N_i^{1-hop}$;
                UAV_i: Remove the 2-hop neighbors ($n_{q,i}^{2-hop}$) covered by $n_{p,i}^{1-hop}$ from $N_i^{2-hop}$;
            else if there is a $n_{q,i}^{2-hop}$ in $N_i^{2-hop}$ covered only by $n_{p,i}^{1-hop}$ then
                UAV_i: Determine this $n_{p,i}^{1-hop}$ as a member of $S_i^{MPR}$;
                UAV_i: Eliminate this $n_{p,i}^{1-hop}$ from $N_i^{1-hop}$;
                UAV_i: Delete the 2-hop neighbors ($n_{q,i}^{2-hop}$) covered by $n_{p,i}^{1-hop}$ from $N_i^{2-hop}$;
            end if
        end for
    end for
    while $N_i^{2-hop} = \{\}$ do
        UAV_i: Sort the members of $N_i^{1-hop}$ based on their $S_p$;
        if two one-hop neighbor have the same $S_p$ then
            UAV_i: Prioritize these one-hop neighbors based on their $ND_p$;
        end if
        UAV_i: Select $n_{p,i}^{1-hop}$ the first member of $N_i^{1-hop}$ as $UAV_{r,i}^{MPR}$;
        UAV_i: Eliminate this $n_{p,i}^{1-hop}$ from $N_i^{1-hop}$;
        UAV_i: Delete the 2-hop neighbors ($n_{q,i}^{2-hop}$) covered by $n_{p,i}^{1-hop}$ from $N_i^{2-hop}$;
    end while
    End
```

5.3. Greedy and Perimeter Forwarding Techniques

In OLSR+GPSR, the routing process follows the GPSR protocol, except that the next-hop nodes are only selected from $S_i^{MPR} = \left\{UAV_{1,i}^{MPR}, UAV_{2,i}^{MPR}, \ldots, UAV_{r,i}^{MPR}, \ldots, UAV_R^{MPR}\right\}$. In this process, UAV_i uses the greedy technique to choose the $UAV_{r,i}^{MPR}$ closest to the destination. Now, if UAV_i cannot find $UAV_{r,i}^{MPR}$ in S_i^{MPR} that is closer to the destination than itself, then UAV_i faces the local optimum problem and leaves the greedy forwarding technique. In this case, it enters the path recovery phase and decides on the next hop node based on the right-hand rule (perimeter forwarding mode). The path recovery phase in OLSR+GPSR is completely similar to that in GPSR. See the pseudo code of this process in Algorithm 3. In the worst case, the time complexity of this algorithm is $O(nM)$ so that n and M are the total number of UAVs and the number of MPRs related to UAV_i, respectively.

Algorithm 3 Routing process

```
Input: $UAV_i$: $i = 1, 2, \ldots, n$
    $S_i^{MPR} = \left\{UAV_{1,i}^{MPR}, UAV_{2,i}^{MPR}, \ldots, UAV_{r,i}^{MPR}, \ldots, UAV_M^{MPR}\right\}$
Output: Routing path between source and destination.
    Begin
    repeat
        if $UAV_i$ find the nearest $UAV_{r,i}^{MPR}$ to the destination then
            $\mathbf{UAV_i}$: Choose this $UAV_{r,i}^{MPR}$ as the next-hop node;
            $\mathbf{UAV_i}$: Forward the data packet to this $UAV_{r,i}^{MPR}$ using the greedy forwarding
            mode in GPSR;
        else if there is no $UAV_{r,i}^{MPR}$ close to the destination compared the current node then
            $\mathbf{UAV_i}$: Select the next-hop node using the perimeter forwarding mode in GPSR;
            $\mathbf{UAV_i}$: Forward the data packet to the next-hop node;
        end if
    until the data packet reaches the destination
    End
```

6. Simulation and Evaluation of Results

In this section, OLSR+GPSR is run on the network simulator version 3 (NS3) [33–35]. Then, a comparison is made between its performance and the three routing methods, namely Gangopadhyay et al. [16], P-OLSR [17], and improved OLSR-ETX [18]. The simulation process considers two modes to evaluate the performance of different schemes: (1) the change in network density and (2) the change in the velocity of UAVs. To perform this operation, it is assumed that the network environment includes a three-dimensional space with the size of 3000 × 3000 × 3000 m^3. In the network environment, 10 to 50 UAVs are randomly deployed, and their velocity varies between 10 and 50 m/s. The transmission radius of these UAVs is 500 m, and their initial energy is 100 Joules. Furthermore, the movement model of UAVs follows the three-dimensional Gauss Markov model (3D GM) [36,37]. In the simulation operation, the traffic model has a constant bit rate (CBR), and the size of data packets is 512 bytes. In addition, the MAC layer protocol supports IEEE 802.11b standard. The execution time of this simulation process is 500 s. For increasing the accuracy of the simulation results, the simulation process is repeated 10 times. The simulation parameters are stated in Table 3 in summary. The evaluation process examines four criteria, namely end-to-end delay, packet delivery rate, routing overhead, and throughput.

Table 3. Simulation parameters.

Parameter	Value
Simulation tool	NS3
Network environment	$3000 \times 3000 \times 3000$ m^3
Number of UAVs	10–50
Velocity of UAVs	10–50 m/s
Communication range of UAVs	500 m
Initial energy of UAVs	100 J
Execution time	500 s
Packet size	512 bytes
Traffic type	CBR
Mobility model	3D GM
Mac protocol	IEEE 802. 11b

6.1. End-to-End Delay (EED)

Delay is an evaluation metric obtained through Equation (19). This means the time period from when a data packet is sent from the source UAV until it arrives at the destination UAV.

$$EED = \frac{\sum\limits_{PK_i \in P = \{PK_1, \ldots, PK_n\}} (T_R(PK_i) - T_S(PK_i))}{\sum\limits_{r=1}^{n_r} PK_r} \tag{19}$$

where PK_r and n_r are the packets delivered to the destination UAV and their number, respectively. PK_i means the i-th packet from the set of all packets (P). In addition, $T_R(PK_i)$ and $T_S(PK_i)$ represent the reception moment and the sending moment, respectively.

Figure 8 compares the end-to-end delay in different schemes so that the number of UAVs is between 10 and 50, and their speed is set to 20 m/s. As shown in this figure, OLSR+GPSR has reduced EED by 22.43%, 38.01%, and 47.44% compared to Gangopadhyay et al., P-OLSR, and enhanced OLSR-ETX. Its main reason is that OLSR+GPSR has removed two steps, namely broadcasting TC messages and calculating transmission routes between all UAVs, and instead, it uses a greedy forwarding technique, whereas Gangopadhyay et al., P-OLSR, and improved OLSR-ETX use the TC message broadcast process to find all possible routes between UAVs but it is very time-consuming. Another reason for this issue is that OLSR+GPSR determines MPRs based on several parameters, including the neighbor degree, node stability (which is obtained from the position and movement information of UAVs), occupied buffer capacity, and remaining energy. This issue leads to the selection of stable MPRs, which reduces the need to update these nodes and, consequently, lowers the delay in FANET. Another point inferred from Figure 8 is that EED and the density of UAVs have a direct relationship with each other, and if the density of nodes is high, EED in the routing process will increase in all schemes. Its reason can be the increase in traffic and congestion in the network. It causes a number of lost data packets, which must be retransmitted. In addition, Figure 9 shows EED in different methods, so that the number of nodes is set to 30, and their velocity is between 10 and 50 m/s. According to this figure, OLSR+GPSR has reduced this evaluation metric by 33.94%, 43.19%, and 53.32% in comparison with Gangopadhyay et al., P-OLSR, and enhanced OLSR-ETX. We have stated the reasons above. Moreover, in Figure 9, it can be seen that EED in the routing process is increasing when the velocity of UAVs is rising because increasing the velocity of UAVs will lead to faster topology changes in the network. As a result, MPRs must be updated in a shorter period of time. This issue will increase EED in the routing process.

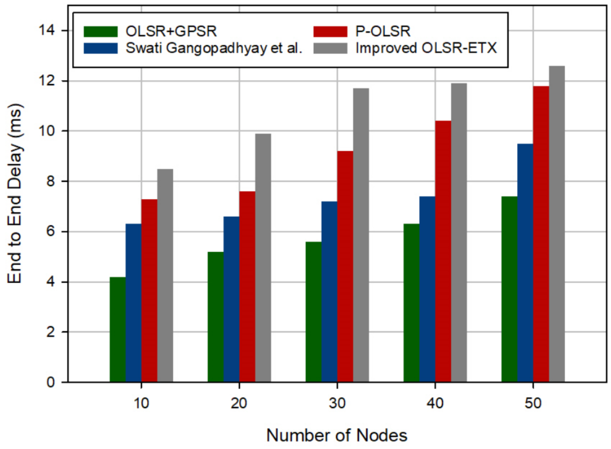


Figure 8. Evaluation of delay based on the number of UAVs.

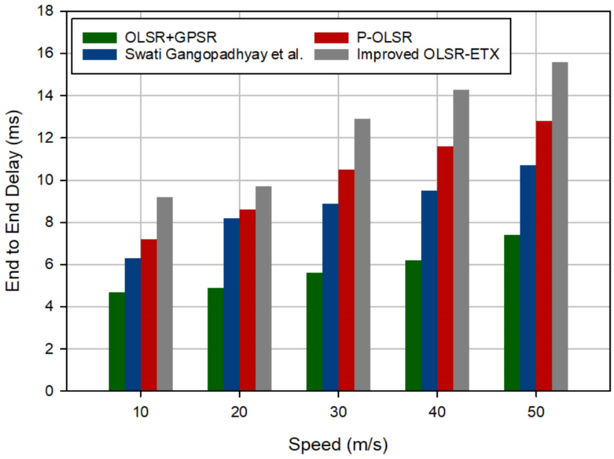


Figure 9. Evaluation of delay based on the speed of UAVs.

6.2. Packet Delivery Rate (PDR)

PDR is an evaluation metric calculated based on Equation (20). It measures the ratio of packets received by the destination UAV to packets sent from the source UAV.

$$PDR = \frac{\sum\limits_{r=1}^{n_r} PK_r}{\sum\limits_{s=1}^{n_s} PK_s} \times 100 \tag{20}$$

where PK_r and n_r are the packets entered to the destination UAV and their number, respectively. Furthermore, PK_s and n_r represent the packets sent from the source UAV and their number, respectively.

Figure 10 shows PDR in different schemes. It is assumed that the number of UAVs is between 10 and 50, and their velocity is a fixed value (20 m/s). According to this figure, OLSR+GPSR has increased this evaluation metric by 4.13%, 16.86%, and 21.51% in comparison with Gangopadhyay et al., P-OLSR, and improved OLSR-ETX, respectively. This is because the proposed method pays attention to node stability along with neighbor degree when choosing MPRs. The stability scale is calculated based on the position and mobility (i.e., velocity and movement angle) information of UAVs. As a result, MPRs in OLSR+GPSR are more stable than other methods. This reduces the number of lost packets and, consequently, increases PDR. On the other hand, in the proposed method, the buffer information of UAVs is considered in the MPR selection process to prevent congestion in UAVs. This also has a positive effect on increasing PDR. As shown in Figure 10, PDR in

OLSR+GPSR does not change based on the number of UAVs, while PDR in other methods is decreasing when the number of UAVs is increasing. This issue is rooted in the fact that the increase in the number of UAVs causes an increase in congestion in FANET. As a result, some packets are lost due to collisions or high traffic in the network, but OLSR+GPSR takes into account the buffer capacity of the flying nodes in the MPR selection process and prevents congestion in the network, and consequently, PDR in our scheme experiences an almost constant rate. Furthermore, Figure 11 makes a comparison between different methods in terms of PDR when the number of UAVs is equal to 30, and their velocity is between 10 and 50 m/s. Based on this figure, OLSR+GPSR has increased PDR by 14%, 23.27%, and 16.11% in comparison with Gangopadhyay et al., P-OLSR, and improved OLSR-ETX. The reasons related to this issue were stated above. On the other hand, different methods experience a lower PDR when increasing the velocity of UAVs. This is caused by higher changes in the network topology. The better performance of the proposed method in comparison with other methods is due to considering the mobility information of UAVs in the routing process. OLSR+GPSR adjusts the propagation period of hello messages according to the velocity of UAVs. This mechanism makes more adaptability between OLSR+GPSR and the dynamic environment of FANET.

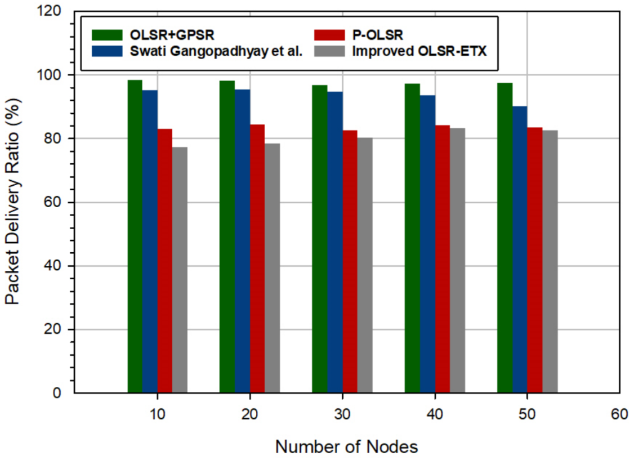


Figure 10. Evaluation of PDR based on the number of UAVs.

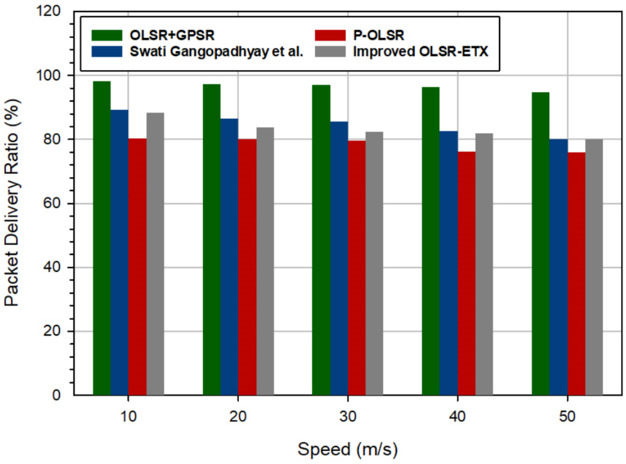


Figure 11. Evaluation of PDR based on the speed of UAVs.

6.3. Routing Overhead

Routing overhead is an evaluation metric that is calculated using Equation (21). It means the ratio of number of sent control packets to all data packets. Note that OLSR-based routing methods have two types of control packets: hello packet and TC packet.

$$RO = \frac{\sum_{h=1}^{n_{Hello}} Hello_h + \sum_{t=1}^{n_{TC}} TC_t}{\sum_{s=1}^{n_s} PK_s} \tag{21}$$

where PK_s and n_s represent the packets sent from the source UAV and their total number, respectively. $Hello_h$ and n_{Hello} are the h-th hello packet and the total number of hello packets in the routing process, respectively. Furthermore, TC_t and n_{TC} indicate the t-th TC packet and the total number of TC packets in the routing process.

Figure 12 shows the routing overhead in different methods when the number of UAVs is changed between 10 and 50, and their speed is set to 20 m/s. According to this figure, OLSR+GPSR has reduced the routing overhead by approximately 6.33%, 11.95%, and 17.12% in comparison with Gangopadhyay et al., P-OLSR, and improved OLSR-ETX, respectively. The good performance of the proposed method in comparison with other methods has two reasons. First, OLSR+GPSR uses a fuzzy system to dynamically adjust the hello message propagation interval based on velocity and position prediction error so that high-speed UAVs have a shorter hello propagation period and low-speed UAVs experience a longer hello propagation period. As a result, the proposed method is more compatible with the dynamic environment of FANET than other routing methods. In addition, OLSR+GPSR eliminates the two phases, namely broadcasting TC messages and calculating transmission routes, and applies greedy and perimeter forwarding techniques. Hence, OLSR+GPSR reduces routing overhead significantly. In addition, according to Figure 12, it can be seen that there is a direct relationship between the routing overhead and the number of UAVs. Therefore, when the number of UAVs is increasing, the routing overhead is also increasing. This is because when the number of UAVs is high, more control messages will be exchanged between them in the network, meaning that, the routing overhead is high. Figure 13 shows a comparison between routing overhead in different methods. In this experiment, it is assumed that the number of UAVs is 30, and their velocity is between 10 and 50 m/s. This figure shows that OLSR+GPSR has reduced the routing overhead by 8.68%, 14.77%, and 18.52% in comparison with Gangopadhyay et al., P-OLSR, and improved OLSR-ETX, respectively. We have mentioned the reasons above. Additionally, this figure confirms the existence of a direct relationship between the routing overhead and the velocity of UAVs because the higher velocity of UAVs will increase the instability of UAVs, which causes higher topology changes in the network. This issue increases the number of control packets due to the need for faster updating MPRs.

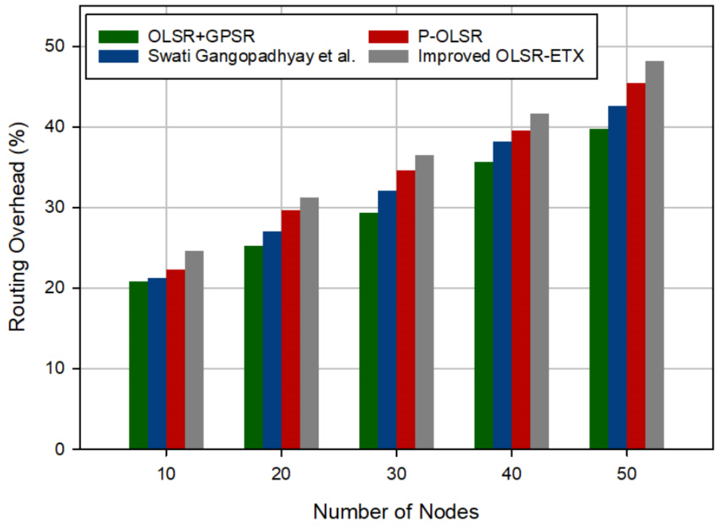


Figure 12. Evaluation of routing overhead based on the number of UAVs.

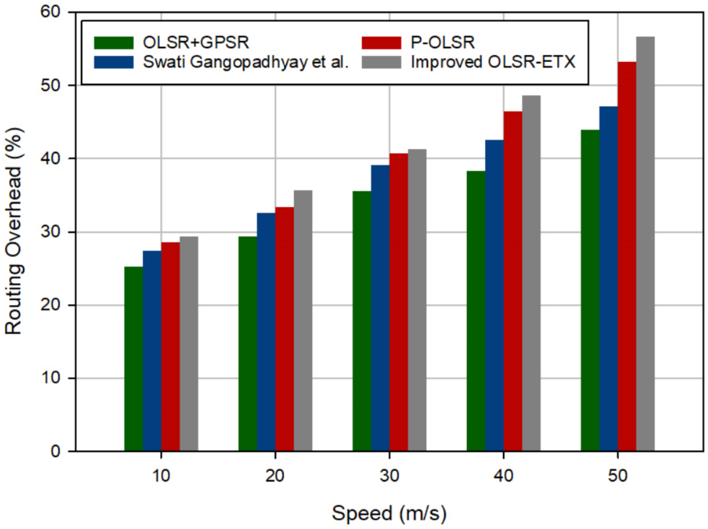


Figure 13. Evaluation of routing overhead based on the speed of UAVs.

6.4. Throughput

Throughput is an evaluation metric obtained through Equation (22). It means the ratio of all packets entered at the destination UAV to the time needed to transfer them.

$$Throughput = \frac{\sum\limits_{r=1}^{n_r} PK_r}{T_{required}} \tag{22}$$

where PK_r and n_r indicate the packets entered to the destination UAV and their total number, respectively. Furthermore, $T_{required}$ is the time needed to send these packets.

Figure 14 shows a comparison between the throughput of different methods when the number of UAVs is between 10 and 50, and their velocity is set to 20 m/s. According to this figure, OLSR+GPSR increases throughput by 27.78%, 30.01%, and 32.33% in comparison with Gangopadhyay et al., P-OLSR, and improved OLSR-ETX, respectively. As stated in Sections 6.1 and 6.2, the proposed method has improved delay and PDR because it designs the fuzzy system to dynamically adjust the hello broadcast interval according to the velocity of UAVs, selects stable MPRs, and removes the TC message propagation process. This issue has increased throughput in OLSR+GPSR. According to Figure 14, throughput and the density of UAVs are oppositely related to each other. When the number of UAVs is high, OLSR+GPSR increases network traffic. Hence, more data packets are lost in the network, and on the other hand, more time is needed to send other packets to the destination. These events reduce throughput in all methods. Figure 15 shows a comparison between throughput in different methods when the number of UAVs is equal to 30, and their velocity changes from 10 to 50. According to this figure, OLSR+GPSR has increased throughput by 60.86%, 94.43%, and about two times compared to Gangopadhyay et al., P-OLSR, and enhanced OLSR-ETX, respectively. The reasons for this were stated above. This figure shows an inverse relationship between the velocity of UAVs and throughput. In all methods, when the velocity of UAVs is increasing, the topology changes in the network is more. Eventually, the number of lost packets is greatly increased. It will result in a lower throughput.

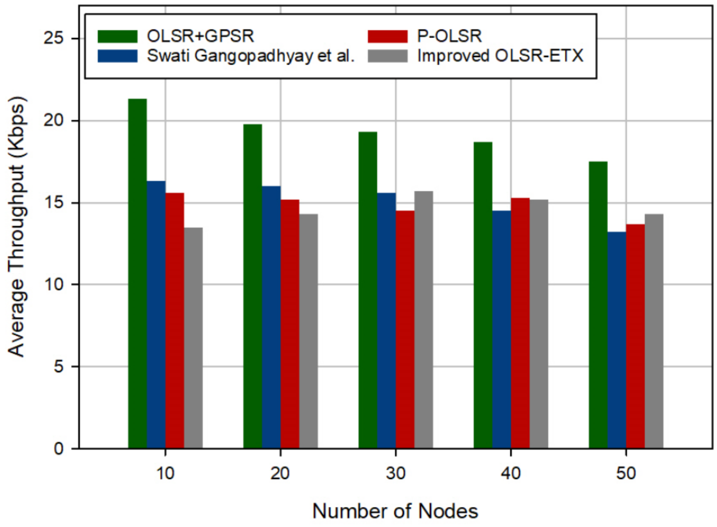


Figure 14. Efficiency evaluation based on the number of nodes.

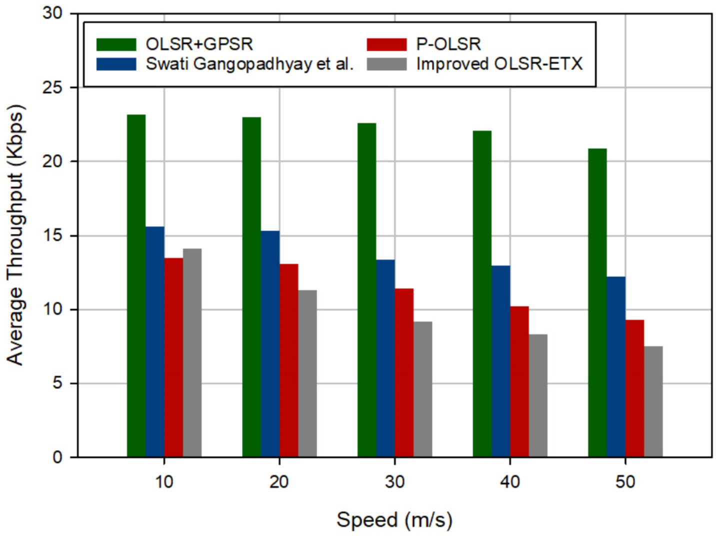


Figure 15. Evaluation of efficiency based on the speed of nodes.

7. Conclusions

In this paper, a novel optimized link-state routing scheme with greedy and perimeter forwarding capability called OLSR+GPSR was proposed in flying ad hoc networks. This method includes three main steps: propagating hello messages based on a fuzzy system, selecting MPR nodes, and greedy and perimeter forwarding techniques. In the first phase, a fuzzy system was designed to regulate the hello propagation interval. It includes two inputs, the velocity of UAVs and position prediction error. In the second phase, OLSR+GPSR selects MPRs based on four parameters, including neighbor degree, node stability, occupied buffer capacity, and remaining energy. Finally, in the third phase, OLSR+GPSR eliminates the two phases, namely broadcasting TC messages and calculating transmission routes in OLSR, and uses greedy and perimeter forwarding techniques to send data packets to the destination. In order to evaluate the performance of OLSR+GPSR, this method was executed in NS3, and its performance in two modes, i.e., changing the network density and changing the velocity of UAVs, was compared with Gangopadhyay et al., P-OLSR, and improved OLSR-ETX. Based on these tests, when the density of UAVs is changing, OLSR+GPSR reduces the delay by 22.43%, 38.01%, and 47.44%, increases the packet delivery rate by 4.13%, 16.86%, and 21.51%, lowers the routing overhead by 6.33%, 11.95%, and 17.12%, and improves the throughput by 27.78%, 30.01%, and 32.33% in comparison with Gangopadhyay et al., P-OLSR, and enhanced OLSR-ETX, respectively. Furthermore, when the velocity of UAVs is changing in the network, OLSR+GPSR reduces the delay by 33.94%, 43.19%, and 53.32%, improves the packet delivery rate by 14%, 23.27%, and 16.11%, decreases the routing overhead by 8.68%, 14.77%, and 18.52%, and enhances the throughput by 60.86%, 94.43%, and about two times in comparison with Gangopadhyay et al., P-OLSR, and improved OLSR-ETX, respectively. In future research, we will attempt to validate the proposed fuzzy model for adjusting the hello propagation intervals to ensure its effectiveness across diverse operational scenarios. Furthermore, we seek to estimate the hello broadcast interval based on metaheuristic methods such as the dragonfly algorithm (DA) or gray wolf optimizer (GWO) or machine learning (ML) techniques to increase its compatibility with the dynamic environment of FANET. In addition, a comparative analysis can be performed between OLSR+GPSR and the integration of OLSR and other alternative routing paradigms to highlight the proposed method's strengths and limitations. In the future, a detailed energy consumption analysis is performed to evaluate OLSR+GPSR's efficiency in conserving energy and highlight potential avenues for enhancing energy-aware routing strategies.

Author Contributions: Conceptualization, O.M.A. and E.Y.; methodology, M.H., E.Y. and J.L.; validation, J.L., E.Y. and O.M.A.; investigation, O.M.A., M.H. and J.L.; resources, M.H. and E.Y.; writing—original draft preparation, O.M.A., E.Y. and J.L.; supervision, M.H.; project administration, E.Y. and M.H. All authors have read and agreed to the published version of the manuscript.

Funding: The result was created through solving the student project "Security analysis and developing lightweight ciphers and protocols" using objective oriented support for specific university research from the University of Finance and Administration, Prague, Czech Republic. The research was funded by TAIF University, TAIF, Saudi Arabia (TU-DSPP-2024-258).

Data Availability Statement: Data are contained within the article.

Acknowledgments: The authors extend their appreciation to TAIF University, Saudi Arabia, for supporting this work through project number (TU-DSPP-2024-258). Authors thank Michal Merta and Zdeněk Truhlář for their help with the research connected with the topic of the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Swain, S.; Khilar, P.M.; Senapati, B.R. A reinforcement learning-based cluster routing scheme with dynamic path planning for mutli-uav network. *Veh. Commun.* **2023**, *3*, 100605. [CrossRef]
2. Gharib, M.; Afghah, F.; Bentley, E.S. LB-OPAR: Load balanced optimized predictive and adaptive routing for cooperative UAV networks. *Ad Hoc Netw.* **2022**, *3*, 102878. [CrossRef]
3. Yang, J.; Sun, K.; He, H.; Jiang, X.; Chen, S. Dynamic virtual topology aided networking and routing for aeronautical ad hoc networks. *IEEE Trans. Commun.* **2022**, *3*, 4702–4716. [CrossRef]
4. Zhang, M.; Dong, C.; Yang, P.; Tao, T.; Wu, Q.; Quek, T.Q. Adaptive routing design for flying ad hoc networks. *IEEE Commun. Lett.* **2022**, *3*, 1438–1442. [CrossRef]
5. Kumar, S.; Raw, R.S.; Bansal, A. LoCaL: Link-oriented cone-assisted location routing in flying ad hoc networks. *Int. J. Commun. Syst.* **2023**, *36*, e5375. [CrossRef]
6. Oubbati, O.S.; Lakas, A.; Zhou, F.; Güneş, M.; Yagoubi, M.B. A survey on position-based routing protocols for flying ad hoc networks (FANETs). *Veh. Commun.* **2017**, *10*, 29–56. [CrossRef]
7. Messaoudi, K.; Oubbati, O.S.; Rachedi, A.; Lakas, A.; Bendouma, T.; Chaib, N. A survey of UAV-based data collection: Challenges, solutions and future perspectives. *J. Netw. Comput. Appl.* **2023**, *216*, 103670. [CrossRef]
8. Kumar, S.; Raw, R.S.; Bansal, A.; Singh, P. UF-GPSR: Modified geographical routing protocol for flying ad hoc networks. *Trans. Emerg. Telecommun. Technol.* **2023**, *34*, e4813. [CrossRef]
9. Hosseinzadeh, M.; Tanveer, J.; Rahmani, A.M.; Aurangzeb, K.; Yousefpoor, E.; Yousefpoor, M.S.; Darwesh, A.; Lee, S.W.; Fazlali, M. A Q-learning-based smart clustering routing method in flying Ad Hoc networks. *J. King Saud-Univ.-Comput. Inf. Sci.* **2024**, *3*, 101894. [CrossRef]
10. Hosseinzadeh, M.; Mohammed, A.H.; Alenizi, F.A.; Malik, M.H.; Yousefpoor, E.; Yousefpoor, M.S.; Ahmed, O.H.; Rahmani, A.M.; Tightiz, L. A Novel Fuzzy Trust-based Secure Routing Scheme in Flying Ad Hoc Networks. *Veh. Commun.* **2023**, *44*, 100665. [CrossRef]

11. Hosseinzadeh, M.; Ali, S.; Mohammed, A.H.; Lansky, J.; Mildeova, S.; Yousefpoor, M.S.; Yousefpoor, E.; Ahmed, O.H.; Rahmani, A.M.; Mehmood, A. An energy-aware routing scheme based on a virtual relay tunnel in flying ad hoc networks. *Alex. Eng. J.* **2024**, *91*, 249–260. [CrossRef]
12. Singh, V.; Sharma, K.P.; Verma, H.K. ABNT: Adaptive beaconing and neighbor timeout for geographical routing in UAV networks. *Peer-Peer Netw. Appl.* **2022**, *3*, 2079–2100. [CrossRef]
13. Ren, Z.; Hussain, K.; Faheem, M. K-means online-learning routing protocol (K-MORP) for unmanned aerial vehicles (UAV) adhoc networks. *Ad Hoc Netw.* **2024**, *3*, 103354. [CrossRef]
14. Zhang, H.; Lv, X.; Liu, Y.; Zou, X. Hedge transfer learning routing for dynamic searching and reconnoitering applications in 3D multimedia FANETs. *Multimed. Tools Appl.* **2024**, *3*, 7505–7539. [CrossRef] [PubMed]
15. Shokrollahi, S.; Dehghan, M. TGRV: A trust-based geographic routing protocol for VANETs. *Ad Hoc Netw.***2023**, *3*, 103062. [CrossRef]
16. Gangopadhyay, S.; Jain, V.K. A Position-based modified OLSR Routing Protocol for Flying Ad Hoc Networks. *IEEE Trans. Veh. Technol.* **2023**, *72*, 12087–12098. [CrossRef]
17. Rosati, S.; Krużelecki, K.; Heitz, G.; Floreano, D.; Rimoldi, B. Dynamic routing for flying ad hoc networks. *IEEE Trans. Veh. Technol.* **2015**, *3*, 1690–1700. [CrossRef]
18. Xie, P. An enhanced OLSR routing protocol based on node link expiration time and residual energy in ocean FANETS. In Proceedings of the 2018 24th Asia-Pacific Conference on Communications (APCC), Ningbo, China, 12–14 November 2018; pp. 598–603. [CrossRef]
19. Khedr, A.M.; Salim, A.; PV, P.R.; Osamy, W. MWCRSF: Mobility-based weighted cluster routing scheme for FANETs. *Veh. Commun.* **2023**, *3*, 100603. [CrossRef]
20. Zheng, B.; Zhuo, K.; Zhang, H.; Wu, H.X. A novel airborne greedy geographic routing protocol for flying Ad hoc networks. *Wirel. Netw.* **2022**, 1–15. [CrossRef]
21. Rahmani, A.M.; Ali, S.; Yousefpoor, E.; Yousefpoor, M.S.; Javaheri, D.; Lalbakhsh, P.; Ahmed, O.H.; Hosseinzadeh, M.; Lee, S.W. OLSR+: A new routing method based on fuzzy logic in flying ad hoc networks (FANETs). *Veh. Commun.* **2022**, *3*, 100489. [CrossRef]
22. Lee, S.W.; Ali, S.; Yousefpoor, M.S.; Yousefpoor, E.; Lalbakhsh, P.; Javaheri, D.; Rahmani, A.M.; Hosseinzadeh, M. An energy-aware and predictive fuzzy logic-based routing scheme in flying ad hoc networks (fanets). *IEEE Access* **2021**, *3*, 129977–130005. [CrossRef]
23. Lansky, J.; Rahmani, A.M.; Malik, M.H.; Yousefpoor, E.; Yousefpoor, M.S.; Khan, M.U.; Hosseinzadeh, M. An energy-aware routing method using firefly algorithm for flying ad hoc networks. *Sci. Rep.* **2023**, *3*, 1323. [CrossRef] [PubMed]
24. Clausen, T.; Jacquet, P. (Eds.) *RFC3626: Optimized Link State Routing Protocol (OLSR)*; The Internet Society: Reston, VA, USA, 2003. [CrossRef]
25. Parihar, A.S.; Chakraborty, S.K.; Sharma, A.; Saini, M. A comparative study and proposal of a novel distributed mutual exclusion in UAV assisted flying ad hoc network using density-based clustering scheme. *Wirel. Netw.* **2023**, *3*, 2635–2648. [CrossRef]
26. Xie, H.; Zheng, J.; He, T.; Wei, S.; Hu, C. A blockchain-based ubiquitous entity authentication and management scheme with homomorphic encryption for FANET. *Peer-Peer Netw. Appl.* **2024**, 1–16. [CrossRef]
27. Wang, X.; Shi, S.; Xue, J.; Wu, C. Controller placement in software defined FANET. *Wirel. Netw.* **2023**, 1–14. s11276-023-03252-5. [CrossRef]
28. Parihar, A.S.; Chakraborty, S.K. Flying ad hoc network (fanet): Opportunities, trending applications and simulators. In Proceedings of the 2022 IEEE Pune Section International Conference (PuneCon), Pune, India, 15–17 December 2022; pp. 1–5. [CrossRef]
29. Gupta, A.; Barthwal, A.; Vardhan, H.; Kakria, S.; Kumar, S.; Parihar, A.S. Evolutionary study of distributed authentication protocols and its integration to UAV-assisted FANET. *Multimed. Tools Appl.* **2023**, *3*, 42311–42330. [CrossRef]
30. Karp, B.; Kung, H.T. GPSR: Greedy perimeter stateless routing for wireless networks. In Proceedings of the 6th Annual International Conference on Mobile Computing and Networking, Boston, MA, USA, 6–11 August 2000; pp. 243–254. [CrossRef]
31. Yang, J.; Zhang, T.; Hong, J.; Zhang, H.; Zhao, Q.; Meng, Z. Research on driving control strategy and Fuzzy logic optimization of a novel mechatronics-electro-hydraulic power coupling electric vehicle. *Energy* **2021**, *3*, 121221. [CrossRef]
32. Nguyen, A.T.; Taniguchi, T.; Eciolaza, L.; Campos, V.; Palhares, R.; Sugeno, M. Fuzzy control systems: Past, present and future. *IEEE Comput. Intell. Mag.* **2019**, *3*, 56–68. [CrossRef]
33. Manzoor, S.; Manzoor, M.; Manzoor, H.; Adan, D.E.; Kayani, M.A. Which Simulator to Choose for Next Generation Wireless Network Simulations? NS-3 or OMNeT++. *Eng. Proc.* **2023**, *3*, 36. [CrossRef]
34. Gomez, J.; Kfoury, E.F.; Crichigno, J.; Srivastava, G. A survey on network simulators, emulators, and testbeds used for research and education. *Comput. Netw.* **2023**, *3*, 110054. [CrossRef]
35. Ergun, S.; Sammour, I.; Chalhoub, G. A survey on how network simulators serve reinforcement learning in wireless networks. *Comput. Netw.* **2023**, *234*, 109934. [CrossRef]

36. Alqahtani, M.; Hu, M. Dynamic energy scheduling and routing of multiple electric vehicles using deep reinforcement learning. *Energy* **2022**, *3*, 122626. [CrossRef]
37. Xie, J.; Wan, Y.; Kim, J.H.; Fu, S.; Namuduri, K. A survey and analysis of mobility models for airborne networks. *IEEE Commun. Surv. Tutor.* **2013**, *3*, 1221–1238. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.