electronics

Article

An Optimized Load Balancing Using Firefly Algorithm in Flying Ad-Hoc Network

Manjit Kaur [1], Deepak Prashar [1,*], Mamoon Rashid [2,*], Zeba Khanam [3], Sultan S. Alshamrani [4] and Ahmed Saeed AlGhamdi [5]

[1] School of Computer Science and Engineering, Lovely Professional University, Phagwara 144411, India; manjit.12438@lpu.co.in
[2] Department of Computer Engineering, Faculty of Science and Technology, Vishwakarma University, Pune 411048, India
[3] College of Computing and Informatics, Saudi Electronic University, P.O. Box 15515, Dammam 31454, Saudi Arabia; z.khanam@seu.edu.sa
[4] Department of Information Technology, College of Computer and Information Technology, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia; susamash@tu.edu.sa
[5] Department of Computer Engineering, College of Computer and Information Technology, Taif University, P.O. Box 11099, Taif 21994, Saudi Arabia; asjannah@tu.edu.sa
* Correspondence: deepak.prashar@lpu.co.in (D.P.); mamoon.rashid@vupune.ac.in (M.R.); Tel.: +91-7814346505 (M.R.)

Abstract: In flying ad hoc networks (FANETs), load balancing is a vital issue. Numerous conventional routing protocols that have been created are ineffective at load balancing. The different scope of its applications has given it wide applicability, as well as the necessity for location assessment accuracy. Subsequently, implementing traffic congestion control based on the current connection status is difficult. To successfully tackle the above problem, we frame the traffic congestion control algorithm as a network utility optimization problem that takes different parameters of the network into account. For the location calculation of unknown nodes, the suggested approach distributes the computational load among flying nodes. Furthermore, the technique has been optimized in a FANET utilizing the firefly algorithm along with the traffic congestion control algorithm. The unknown nodes are located using the optimized backbone. Because the computational load is divided efficiently among the flying nodes, the simulation results show that our technique considerably enhances the network longevity and balanced traffic.

Keywords: firefly algorithm; flying ad hoc networks (FANETs); geographic position mobility oriented routing (GPMOR); load balancing; unmanned aerial vehicles (UAVs)

check for updates

Citation: Kaur, M.; Prashar, D.; Rashid, M.; Khanam, Z.; Alshamrani, S.S.; AlGhamdi, A.S. An Optimized Load Balancing Using Firefly Algorithm in Flying Ad-Hoc Network. *Electronics* **2022**, *11*, 252. https://doi.org/10.3390/electronics11020252

Academic Editors: Jaime Lloret and Joel J. P. C. Rodrigues

Received: 9 December 2021
Accepted: 11 January 2022
Published: 13 January 2022

1. Introduction

Unmanned aerial vehicles (UAVs) have received much attention in the current decade because of their ability to assist humans in a variety of tasks. A Flying Ad-Hoc Network (FANET) is a critical component of an ad hoc network and UAVs are commonly used to form FANETs. UAVs are small, self-contained drones that can be controlled remotely [1,2]. UAVs have been employed in a variety of industries, including warfare, agriculture, medicine, photography, and environmental applications, among others. UAVs were initially only employed by the military for surveillance and rescue missions. Nowadays, with the advancement in technology, UAVs are widely employed in a variety of fields, including product shipping and delivery, soil analysis, agricultural monitoring, and so on [3,4]. With the introduction of FANETs, a variety of applications have emerged, including a cargo of goods, domestic package delivery, crop monitoring, agricultural surveillance, and rescue operations. The latest techniques of FANETs are mobility, modeling, and theoretical proofs of communications in FANETs; traffic models and network control for FANETs; security, privacy, and trust in FANETs; performance, scalability, reliability, and efficiency of FANETs;

UAV-assisted packages; energy efficiency in FANETs; the emerging Internet of Things (IoT) and cloud applications with FANETs; routing strategies for FANETs; application of future Internet architectures in FANETs; the energization of FANETs. Furthermore, this work explores into the modeling and performance evaluation of drones and UAVs in emerging applications, describing novel strategies, new models for energy, communication, and routes for UAVs, as well as interesting applications.

Load balancing is the act of dispersing system traffic flow over various servers, and localization is the process of estimating the location of unknown nodes placed in FANETs. As computing moves more and more to the network, load balancing plays an increasingly critical security function [5]. In such networks, the accuracy of the node's position estimation is determined by two key processes. The first is node location estimation, which involves calculating the position of unidentified nodes, and the second is node position confirmation, which involves comparing the computed location to the real site. To expand the correctness of the localization process, several expressions have been developed.

Congestion is always an undesirable circumstance in wired and wireless networks since it can degrade the communication environment. At the media access control address level or above this level, congestion can cause packet loss and retransmission. The task of planning and executing a traffic congestion control algorithm along with a firefly algorithm is difficult since numerous elements must be considered. The congestion control system is not equipped to deal with the unique characteristics of shared wireless multi-UAVs in the network. Packet loss and retransmissions can occur for a variety of reasons, including route pause due to mobility and concealed terminal problems in the wireless networks. Unpredictable packet delivery ratio and the loss of packet rates are caused by affected fluctuations in topology and a weak wireless network, posing a challenge for congestion control in FANETs [6]. Furthermore, due to the random link-state induced by UAVs' fast mobility, it is hard to control overall transmission delay within a specific threshold. Consequently, it is required to plan an accurate traffic congestion control of flying nodes algorithm that can adjust to the different status of the link, and guarantees the interruption prerequisites and different parameters.

To the best of our knowledge, only a few earlier studies have looked into the load-balancing problem in FANETs. In [3], the flying ad hoc networks are described along with mobility models, features, and routing protocols. In [7], the Geographical Position Mobility Oriented Routing protocol (GPMOR) is described where the main aim is to reduce the number of hops based on the Gauss-Markov (GM) mobility model, which further improved the performance of routing by efficient packets. In summary, we make two contributions:

- For FANETs, we propose the specific functions F_1 and F_2 with the optimization process by taking different parameters such as end-to-end delay (EED), packet delivery ratio (PDR), fuel emission, and throughput. The technique is used to resolve the constraints of the optimization problem with the firefly algorithm, which is used to estimate the exact match of the dynamic network topology. The primary problem is therefore converted into a distributed solvable problem, allowing senders to compute the attractiveness of flying nodes to execute congestion control.
- To reach the best solution, we present a distributed traffic congestion control algorithm that incorporates the delay constraints. We propose Rs, Rd, and Rp variables for all flying nodes to verify the incoming flow of the flying nodes and outgoing flow of the flying nodes probability to exploit network utilization and decrease transmission delay in a circulated manner. Finally, we examine the optimization method's performance and demonstrate its convergence using a simulator.

In this paper, we use the GPMOR protocol and the firefly algorithm to develop a load-balancing technique for position estimation in FANETs. Section 2 discusses related research in this field. The suggested network model, shown in Section 3, is completed in two phases: (i) optimizing the flying nodes using the firefly method in Section 3.1; (ii) using the traffic congestion control algorithm to predict the position of all unknown nodes in

Section 3.2. Section 4 contains the findings and analyses. Lastly, Section 5 concludes the research paper.

2. Related Work

There are several assumptions such as device hardware, network configuration, operating setting, signal propagation models, time management, communication costs, error requirements, timing, energy requirements, and node mobility used to address the localization problem in FANET.

Some studies formalized congestion control as an effectiveness maximization issue [8], which was then explained using optimization approaches to diminish network congestion and to propose an innovative cross-layer architecture for optimizing congestion control, direction-finding, and control in ad hoc networks, to address the routine limits imposed by the nonexistence of collaboration between layers. The research paper [9] proposed a framework for wireless networks based on the concept of utility maximization of the network that considers routing and congestion as well as power controls all at the same time while taking into account the continuing or temporary random attenuation of ad hoc networks. Khodaian et al. [10] observed the delay in multiple access networks, addressed the delay-based effectiveness maximization issue, and tried to find the best balance of delay, PDR, and energy parameters. Li et al. [11] explained congestion control in two modes, complex mode and non-complex mode of delay traffic, and proposed an innovative decentralized system for the same problem. Under predicted contact limits, [12] investigated the combined enhancement of power control as well as congestion control in other ad hoc networks. A disseminated cross-layer enhancement framework was planned at the same time to meet the simultaneous design aim of power control and blockage control from one source to another node at the physical layer and transport layer, respectively.

In [13], authors discussed a new paradigm, software-defined networking, for the load-balancing technique to deal with the issue of load irregularity in the organization, and they planned unified programming characterized by LTE remote access network structure. To overcome the problem of throughput expansion, [14] suggested an algorithm for congestion control-based scheduling. In heterogeneous QoS-aware applications, [15] solves the rate assignment problem to optimize the system value of closure in a certain time frame and enable every data flow to satisfy the criteria of lengthy average end-to-end delay constraints. Alaei et al. [16] suggested a distributed congestion control strategy that holds the congestion issue in wireless multimedia sensor networks. When it is discovered, a local binary tree is created at the congestion link to resolve the issue. The established tree is used to transfer packets to the next possible congestion node, whereas the mobile sink node is used to facilitate communication when the established tree is unable to reduce congestion.

Silva et al. [17] investigated congestion management in time lag and disruption-tolerant networks (DTNs) and a novel disruption-tolerant network congestion control system that modifies its operation dynamically based on the underlying network aspects. The authors of [18] proposed a novel, impartial and dispersed congestion control method for neighborhood area networks to assure network access equality. Lubna et al. [19] proposed improving throughput by applying a unique strategy for dynamically controlling the reduction factor of the congestion control algorithm in multipath TCP based on the interval between packet losses. Verma et al. [20] investigated a novel congestion control policy for the Internet of Things, which aims to reduce packet drops and improve throughput by promptly adapting the data transfer rate whensoever the existing bandwidth and delay changes. The description of traffic congestion control with parameters of FANET is shown in Table 1 below:

Table 1. Traffic congestion control parameters of FANET.

Work	EED	PDR	Fuel Emission	Throughput
D'Aronco et al. [8]	✓	×	×	×
Mehta et al. [9]	✓	×	×	✓
Khodaian et al. [10]	✓	✓	×	×
Li et al. [11]	✓	✓	×	✓
Zhang et al. [12]	✓	×	×	✓
Rangisetti et al. [13]	✓	×	×	✓
Kafi et al. [14]	✓	✓	×	✓
Hajiesmaili et al. [15]	✓	×	×	×
Alaei et al. [16]	✓	×	×	✓
Silva et al. [17]	×	✓	×	×
Leon et al. [18]	×	✓	×	✓
Lubna et al. [19]	✓	×	×	✓
Verma et al. [20]	✓	✓	×	✓
Proposed work	✓	✓	✓	✓

3. Proposed Network Model

3.1. Problem Formulation

There are different categories of load balancing as follows:

- Load balancing using software-defined networking (SDN),
- User Datagram Protocol (UDP),
- Transmission Control Protocol (TCP),
- Server load balancing (SLB),
- Virtual load balancing, multi-site load balancing, and elastic load balancing, also known as global server load balancing (GSLB), and
- Geographic load balancing.

Geographic load balancing reallocates user traffic among data centers in multiple locations for maximum efficiency and security [21]. GPMOR is proposed as a method for determining the best available next hop to effectively reduce the impact of network latency caused by highly dynamic mobility. To begin, we used the Gauss-Markov mobility model to predict node position to reduce routing failure. Second, we used the deployment relationship to more precisely select the next hop for routing. We looked at the firefly algorithm, which has three major aspects that are relevant to the optimization technique. The firefly algorithm is simple and efficient. The benefits of such computation are that they are usually efficient for specific problems, require a small number of rounds, and are capable of dealing with highly nonlinear, multimodal optimization problems naturally and efficiently. The firefly algorithm has a very fast convergence rate in terms of the probability of finding the globally-optimized answer. It does not employ velocities and does not necessitate a good initial solution to begin its iterative process. It can be combined with other optimization techniques to create hybrid tools. Internal load balancing takes place inside a centralized environment, whereas geographic load balancing takes place across numerous sites. The mathematical notations are described in Table 2.

Table 2. Mathematical Notations.

Symbol/Notation	Description
$\sum_F$	The notation depicts a limited set that contains all of the UAVs that are free to fly in the specified area.
L_i	Indicates a link linking a pair of UAVs.
L	Denotes the set $\{\forall l \in L\}$.
U_j and U_i	If the distance between U_j and U_i is below the communication radius, $j \in Ne_i$, where Ne_i is a set of U_i's neighbors.
S	A session initiated by a source UAV.
E	A collection of all consecutive sessions.
L(s)	A collection of links followed by session Us.
$S(l) = \{Us \in \sum S \mid L_i \in L(s)\}$	A collection of all sources that use link $L_{i.}$
$\sum D_i < \theta$	The entire delay along the path L(s) < threshold (θ).
Cn	Capacity of the node-link
$D_n = P/(Cn - \sum S * r)$	This is expressed as a single-hop delay where P is the length of the packet and r is the rate of the source node.

Assume that each flying node can achieve function $fn(r)$ by creating a packet flow rate of r, where $fn(r) = \omega \log(r)$ and ω is a constant. This effort purposes at exploiting the overall function of all the flying nodes in the network under the node-link capacity of the network and total delay along with the multiple flying nodes. Hence, the problem can be formulated as a function F_1, which is defined as below:

$$F_1 = max\sum_{s}^{E} fn(r) \tag{1}$$

Here, the optimization problem F_1 is the main problem of optimization of flying nodes in the network. Here, the Equation (2) is defined as below:

$$\sum_{s}^{S} r \leq \mathrm{Cn} \tag{2}$$

$$\sum_{\mathrm{Li}}^{\mathrm{Ls}} Dn \leq \theta \tag{3}$$

3.2. Solution of the Problem

It should be noticed that the function defined in Equation (4) is more complex. As a result, the following equations may be used to breakdown the relationship in Equation (4)

$$\sum_{s}^{S} r \leq \mathrm{Cn} - \mathrm{p} \tag{4}$$

where $p = K/\overline{D}i$, K is the constant value, and $\overline{D}i$ is the network's single-hop delay limit. The value of p should be greater than zero, i.e., $p > 0$.

From Equations (2) and (4), we can formulate

$$F_2 = max\sum_{s}^{E} fn(r) \tag{5}$$

Here, Equation (6) is subject to

$$\sum_{s}^{S} r \leq \mathrm{Cn} - \mathrm{p} \tag{6}$$

$$\sum_{\mathrm{Li}}^{\mathrm{Ls}} \overline{\mathrm{D}}\mathrm{i} \leq \theta \tag{7}$$

It is rather challenging to tackle the traffic congestion problem of flying nodes in the network with various settings in a centralized manner. To ease the optimization process, Equation (5) can be denoted as χ. As a result, the new form of F_2 can be expressed as

$$S = max\left(\sum_{\mathrm{s}}^{\mathrm{S}} \mathrm{r} + \sum_{\mathrm{Li}}^{\mathrm{Ls}} \chi - \left(\mathrm{Cn} - \left(\mathrm{p} + \sum_{\mathrm{s}}^{\mathrm{S}} \mathrm{r}\right)\right)\right) \tag{8}$$

After reordering Equation (9), a new form can be obtained as

$$S = max\left(\sum_{\mathrm{s}}^{\mathrm{S}}\left(\mathrm{r} - \mathrm{r}\sum_{\mathrm{Li}}^{\mathrm{Ls}} \chi\right) - \sum_{\mathrm{Li}}^{\mathrm{Ls}} (\chi * \mathrm{p})\right) \tag{9}$$

Furthermore, we have two different equations from Equation (10), as follows

$$S1 = max \sum_{\mathrm{s}}^{\mathrm{S}}\left(\mathrm{r} - \mathrm{r}\sum_{\mathrm{Li}}^{\mathrm{Ls}} \chi\right)$$

and

$$S2 = min \sum_{\mathrm{Li}}^{\mathrm{Ls}} (\chi * \mathrm{p}) \tag{10}$$

The Proposed Firefly Algorithm is defined as follows:

Algorithm 1: (Proposed Firefly Algorithm) With the help of two properties, this approach is used to determine the direct path (shortest) in a network. The first is the firefly's brightness, which is proportionate to its mate selection and prey attractiveness. The other is that the difference between the couples (two) of fireflies is inversely proportional to the difference between them [20].

Algorithm 1. Proposed Firefly Algorithm

Step 1: Begin by initializing the objective function.
Step 2: Create a small population of fireflies (nodes).
Step 3: Calculate the light intensity and the state absorption coefficient.
Step 4: Repeat Steps 5–8 until the maximum generation value is reached (maximum iteration).
Step 5: Repeat for I = 1 to N, where N represents all of the 'N' fireflies.
Step 6: Repeat for J = 1 to I.
Step 7: If J's light intensity is larger than I's light intensity, then set: change mate selection and prey attractiveness with their distance.
Step 8: Reposition the firefly based on I's attraction to J and test different solutions.
Compute attractiveness value of the fireflies using Equation (11)
(The end of the If structure)
(At the end of the Inner for structure.)
(At the end of the Outer for structure.)
Step 9: If the result cannot be discovered, proceed to step 4.
Step 10: Show the best-desired outcome.

$$\beta = \underset{0}{\beta} x e^{-y r^2} - 1! = 0 \tag{11}$$

where β0 ! = 0

Here, the proposed firefly algorithm is compared with the existing firefly algorithm [20]. After comparing the light intensity in [20], then move the fireflies based on the attraction I towards J and evaluate new solutions. In the proposed firefly algorithm, further to this, reposition the firefly based on I's attraction to J and test different solutions, then the algorithm computes the attractiveness value of the firefly using Equation (11). The main difference between the algorithms is the computation of the latest attractiveness value of the fireflies, which shows the best-desired and most efficient result for the flying nodes. In conclusion, we have two different solutions of flying nodes, S1 and S2, which can be considered as the final method to solve the traffic congestion problem of flying nodes in the network. In the decentralized environment, the topology changes of flying nodes can be the issue of flying nodes in the network, so the solution of the particular problem is to calculate the speed (Rs), distance (Rd), and path (Rp) constraints of the flying nodes. We can calculate the actual values of Rs, Rd, and Rp as defined, as follows

$$Rs = \max\,(r - s * \sum_{\mathrm{Li}}^{\mathrm{Ls}} \chi) \tag{12}$$

$$Rd = \max\,(r - Cn * \sum_{\mathrm{Li}}^{\mathrm{Ls}} \chi * \mathrm{p}) \tag{13}$$

and

$$Rp = \exp\,\{-(Cn - \sum_{\mathrm{s}}^{\mathrm{S}} \mathrm{Rs}) * \mathrm{Rd}\} \tag{14}$$

Similarly, the traffic congestion control algorithm can be implemented as:

Algorithm 2: (Traffic Congestion Control Algorithm) This approach is based on the following scenario: when a large number of flying nodes are present in the network yet their performance diminishes, the network is said to be congested. The following algorithm is used to resolve or balance the traffic of flying nodes:

Algorithm 2. Traffic Congestion Control Algorithm

Step 1: First of all, we need to initialize different parameters such as Rs, Rd, and Rp.
Step 2: If the flying node arrives at link Li:
Step 3: Then we have to calculate the value of ∑Di, which is based on Equation (7)

$$\sum \overline{\mathrm{Di}} = -\mathrm{n}\chi / (\mathrm{Cn} - \sum_{\mathrm{s}}^{\mathrm{S}} \mathrm{Rs}) + \chi 1)$$

Here, n is the error of flying nodes due to environmental issues, and χ1 is the delay errors of flying nodes at node-link Cn.
Step 4: Further, calculate the value of χ1.
Step 5: χ = χ + χ1, where χ = 0.
Step 6: As per the firefly algorithm, update the attractiveness as described in Equation (12).
Step 7: Calculate Rs based on Equation (12).
Step 8: Update the value of attractiveness until β remains unchanged.
Step 9: Stop.

The value of χ can help to alleviate traffic congestion. If more packets are dropped because the threshold θ is set too high, a higher value χ is necessary. Algorithms 1 and 2 are used in the implementation.

4. Results and Discussion

Simulations have been run to evaluate the performance of the proposed scheme. Because standard FANET simulator tools have limitations, realistic mobility models are very useful in designing and implementing routing protocol performance. To reflect the effects of a routing protocol based on environmental variables, a flying mobility model should be proactively easily adjustable. The results are validated using the NS2 simulator as a network simulator and the cbrgen utility to generate traffic files. Setdest is a program in the NS2 simulator that has an average pause of 2 s between node movements. The simulation is terminated after 200 s, and the topology perimeter is defined as 500 × 500.

In this paper, we use the NS2 simulator to pretend the network scenario. NS-2 packages are made up of the following components:

- geo utility.h contains geometrical utility functions such as the projection of a 3D graph to a 2D graph and the communications network between two flying nodes;
- geo pkt.h contains the new geo packet header definition;
- geo node.h and geo node.cc files define and implement the geographic node;
- geo.h and geo.cc files contain the definition and implementation of the geographic agent prototype;
- the proposed algorithms are defined and implemented in geo next node.h and geo next node.cc.

In the beginning phase, flying nodes are dispersed arbitrarily, the altitude of UAVs is 40 m, and the directional gain is 10 dBi with a frequency range of 2.4 GHz. The value of the transmission power is 0.005 W for each session is set to the speed of the UAVs, which varies up to 60 m/s. In addition, we use the queue type as the priority queue to simulate a wireless physical medium channel and further to estimate the link quality of the nodes. The detailed definitions of simulation parameters can be found in Table 3. The goal of this study is to regulate the congestion level of the entire network by satisfying various characteristics of flying nodes.

Table 3. Dimensions of UAVs in the network.

Parameter Type	**Value**
Number of UAVs	100
Queue Type	Priority queue
Altitude of UAVs	70 m
Traffic Type	CBR
Directional Gain	10 dBi
Frequency	2.4 GHz
Wireless Medium	Wireless physical medium
Data Rates	54 Mbps
Packet Interval (s)	Exponential (1)
Routing Protocol	GPMOR
Packet Size (byte)	1024
Fuel (kg)	80
Simulation Time	200 s
Pause Time	Variable
Antenna Type	Omni-Directional
Transmission Power	0.005 W
Speed of UAVs	Can vary up to 60 m/s

The network is initialized with the help of multiple flying nodes with the altitude of UAVs at 70 m. The transmission power is constrained by the connectivity between terrestrial base stations and UAVs. To prevent these, UAVs can communicate with one another using purely ad hoc architecture. In Figure 1, the initialization of the flying nodes is defined. The syntax for defining a node is set as n0 ($ns node). We created a node that is represented as node 0 by the variable n0. When we refer to that node in the code, we will use $n0. After we have defined a few nodes, we can define the links that connect them, and so on. In ns2, a node's output queue is implemented as a component of each link whose input is that node. The definition of the link then includes a method for dealing with queue overflow. Figure 2 depicts the specific source and destination nodes as defined by RoadSide Units (RSU). Furthermore, we need to connect the traffic source to the traffic sink. This maintains the RSU's proper infrastructure for sending route packets to flying nodes. It manages traffic signals; the RSU is intended for use at road intersections and serves as a resource for information.

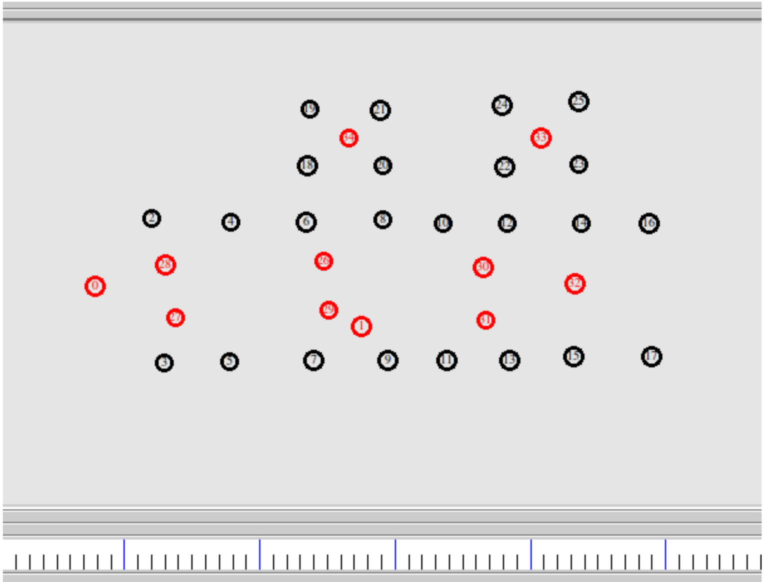

Figure 1. Initialization of flying nodes.

Figure 2. Source node and destination node.

Furthermore, such wireless communication may be used to enable multi-node communications and other applications if a data packet needs to be delivered to another node that is outside of the range. In the network, the node with name 26 and node 30 are described as the root nodes mentioned in Figure 3.

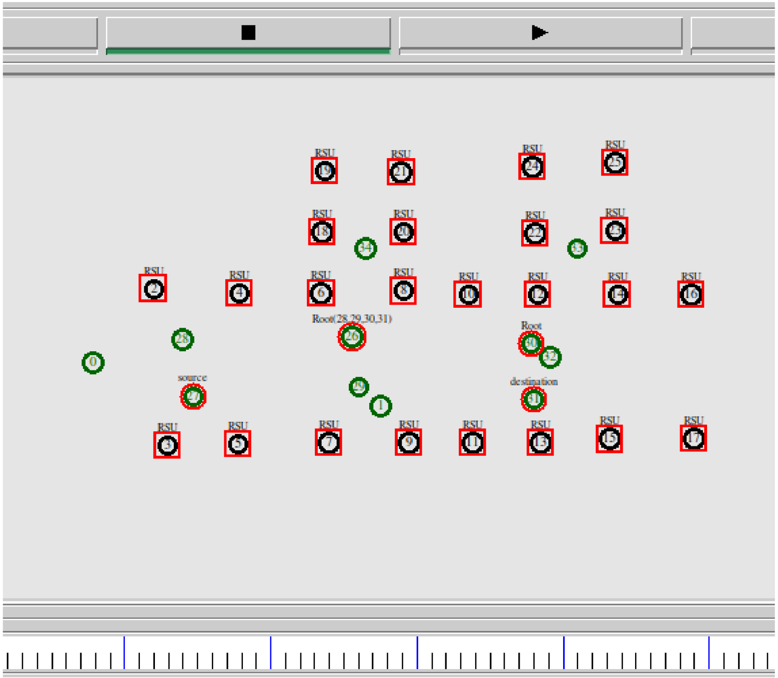

Figure 3. Root nodes in the network.

In such a case, each node selects a random destination, then travels with a random velocity and pauses at the destination. When the stop time expires, the node chooses a random destination with a random velocity and a similar pause duration based on set probability. Furthermore, the root node sends a Route Reply Packet (RRP) to the next neighbor node as shown in Figure 4. A few entries of statistical data from the source node to the destination with appropriate time are shown in Figure 5 below.

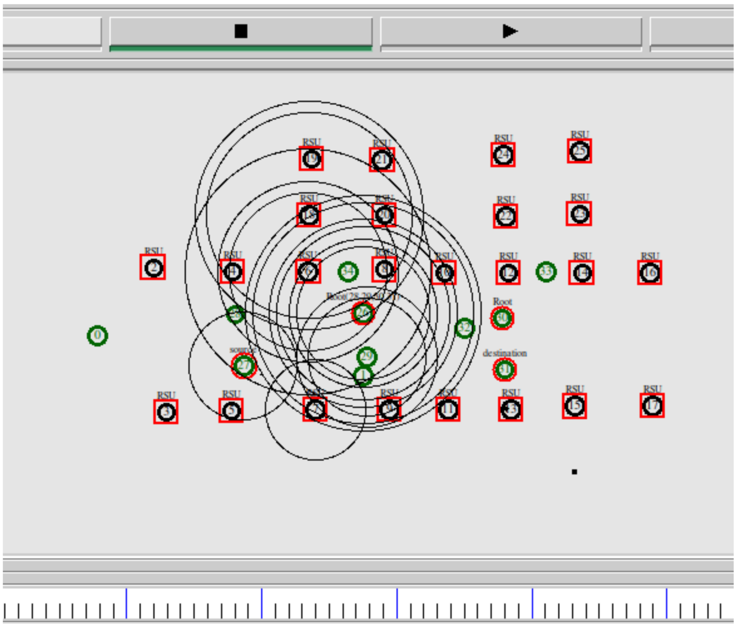

Figure 4. Root node sends a Route Reply Packet.

NODE: 6	Time: 0.726284	Dest: 26	Next: 26	Seq: 4
NODE: 6	Time: 0.726284	Dest: 8	Next: 26	Seq: 4
NODE: 6	Time: 0.726284	Dest: 4	Next: 4	Seq: 6
NODE: 6	Time: 0.802832	Dest: 26	Next: 26	Seq: 4
NODE: 6	Time: 0.802832	Dest: 8	Next: 8	Seq: 6
NODE: 6	Time: 0.802832	Dest: 4	Next: 4	Seq: 6
NODE: 8	Time: 0.806447	Dest: 4	Next: 26	Seq: 4
NODE: 8	Time: 0.806447	Dest: 26	Next: 26	Seq: 4
NODE: 8	Time: 0.806447	Dest: 6	Next: 6	Seq: 6
NODE: 8	Time: 0.806447	Dest: 10	Next: 10	Seq: 4
NODE: 8	Time: 0.893216	Dest: 4	Next: 26	Seq: 4
NODE: 8	Time: 0.893216	Dest: 26	Next: 26	Seq: 4
NODE: 8	Time: 0.893216	Dest: 6	Next: 6	Seq: 6
NODE: 8	Time: 0.893216	Dest: 10	Next: 10	Seq: 6
NODE: 26	Time: 1.516071	Dest: 27	Next: 27	Seq: 4
NODE: 26	Time: 1.516071	Dest: 8	Next: 8	Seq: 12
NODE: 26	Time: 1.516071	Dest: 4	Next: 4	Seq: 4
NODE: 26	Time: 1.516071	Dest: 6	Next: 6	Seq: 6
NODE: 26	Time: 1.827767	Dest: 31	Next: 31	Seq: 6
NODE: 26	Time: 1.827767	Dest: 27	Next: 27	Seq: 4
NODE: 26	Time: 1.827767	Dest: 8	Next: 8	Seq: 12
NODE: 26	Time: 1.827767	Dest: 4	Next: 4	Seq: 4
NODE: 26	Time: 1.827767	Dest: 6	Next: 6	Seq: 6
NODE: 4	Time: 2.522810	Dest: 8	Next: 18	Seq: 12
NODE: 4	Time: 2.522810	Dest: 26	Next: 27	Seq: 8
NODE: 4	Time: 2.522810	Dest: 6	Next: 6	Seq: 6
NODE: 4	Time: 2.522810	Dest: 31	Next: 34	Seq: 6

Figure 5. Statistical data from the source node to the destination at the appropriate time.

In the simulation, a large number of flying nodes with defined direction or speed changes are used. We investigated various degrees of flying node density, velocity, and network activity from source to destination during the simulation.

Furthermore, in Figure 6, the source node starts sending data to the furthest node, accordingly, to identify the traffic in the entire network. Then, the source node multicast route request packets (Figure 7) and the root node again sends a Route Reply Packet to the next neighbor node (Figure 8). The neighbor node's data with the exact distance value are shown in Table 4 below. Finally, the source node starts sending data to the next proceeding nodes to achieve the target for traffic balance in-network as shown in Figure 9.

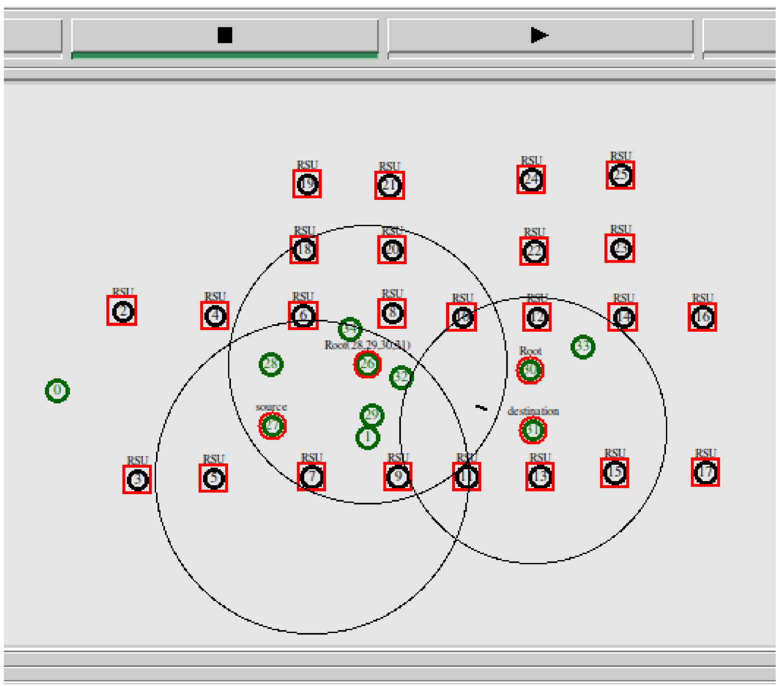

Figure 6. Source node starts sending data.

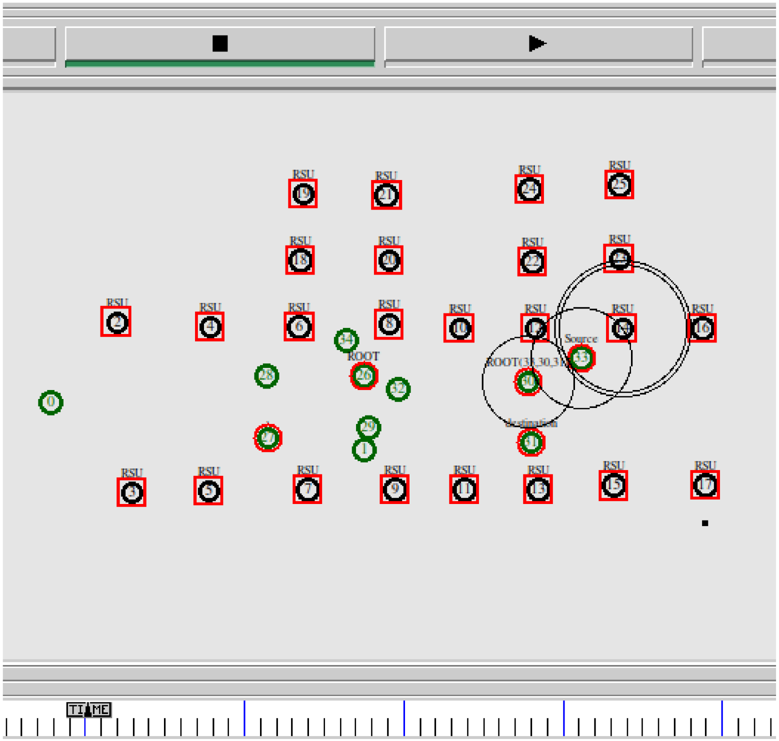

Figure 7. Source node multicast route request packets.

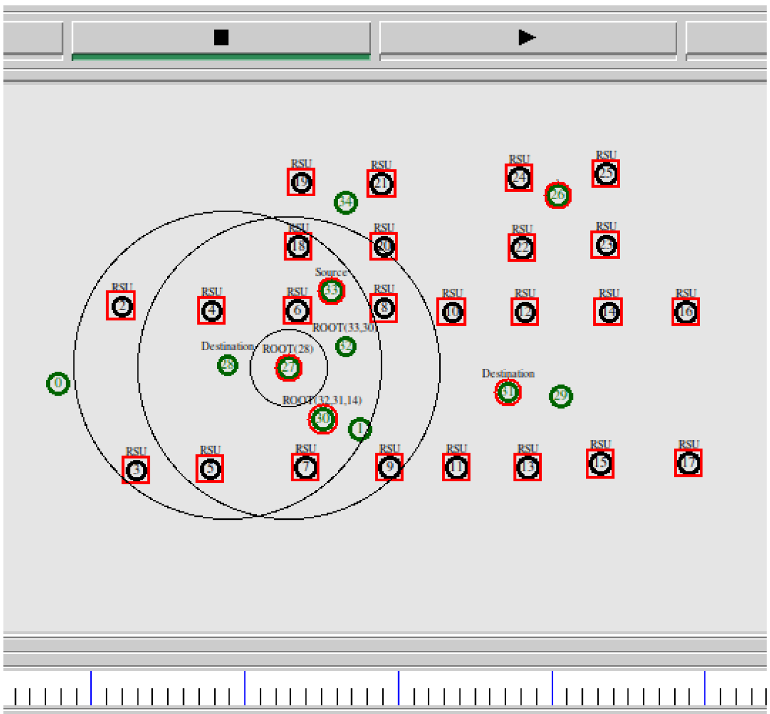

Figure 8. After multicasting, root node again sends a Route Reply Packet.

Different performance metrics (parameters) are to be taken further in this research work such as delay analysis, fuel emission analysis, packet delivery ratio analysis, and throughput analysis of flying nodes. There is a comparison between two routing protocols such as GPMOR and Greedy Perimeter Stateless Routing (GPSR).

Table 4. Neighbor data with a distance value.

Source	Neighbor	SX-Pos	SY-Pos	Distance (d)
0	2	−247	358	161
1	6	239	284	216
1	7	239	284	106
1	8	239	284	198
1	9	239	284	78
1	11	239	284	168
1	26	239	284	115
1	27	239	284	159
1	28	239	284	209
1	29	239	284	36
1	30	239	284	229
1	31	239	284	204
2	0	−145	483	161
2	4	−145	483	145
2	28	−145	483	225
3	0	−122	218	187
3	5	−122	218	120
3	27	−122	218	220
4	2	0	475	145
4	6	0	475	139
4	18	0	475	173

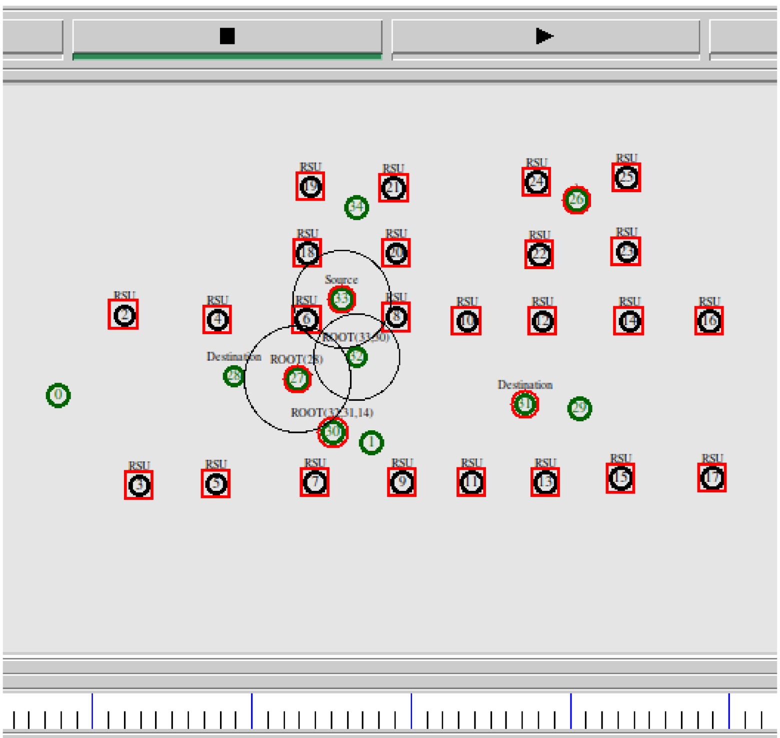

Figure 9. Appropriate load balancing between source to destination.

Delay: The average time it takes for data packets to move across the network from the source flying node to the destination flying node is referred to as delay. The delay of a communications network is a significant design and performance aspect. The processing and transmission delays of a network link are all included in end-to-end delays.

Figure 10 depicts the working of a network in terms of end-to-end delay when the number of UAVs, speed, and area magnitudes are varied. The X-axis depicts the time in m/s, while the Y-axis denotes the delay in seconds. The end-to-end delay decreased with the number of UAV nodes. This is because packets are more likely to be routed rather than captured in the suspension buffer. Once the delay for each of these measures was compared, GPSR (purple bar) had the longest delay, even when the region was smaller. This was because when a route request was given, the destination answered to every RREQs that it received, making determining the least populated route take longer. When compared to the GPSR protocol, GPMOR had the shortest delay.

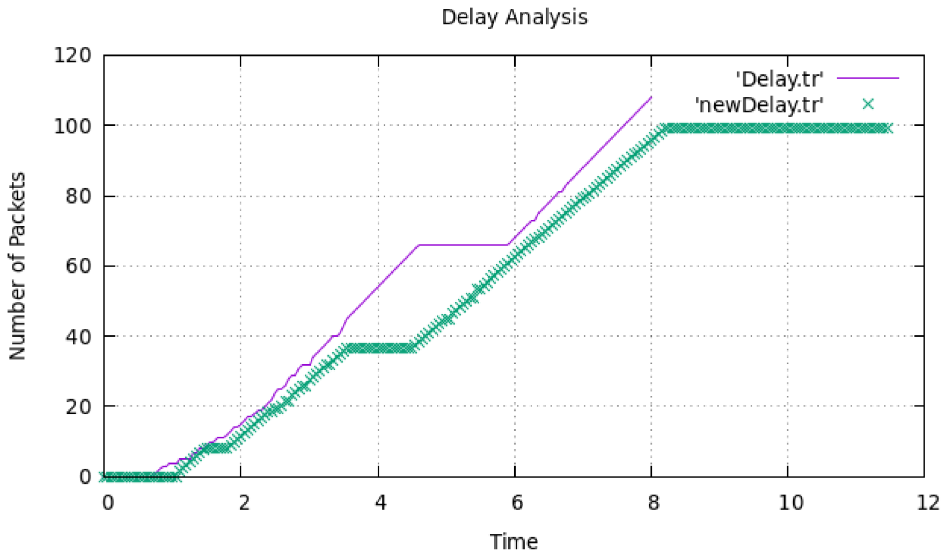

Figure 10. Delay analysis of flying nodes.

Fuel Emission: In fuel emission, to accomplish connection dependability and the quantity of stored fuel in terms of energy and input buffer, the system chooses a route based on the present processing status of a node. Figure 11 shows that GPSR released more fuel than the GPMOR technique. Several requirements must be satisfied, such as the minimum fuel necessary to process packets in kilobytes. The current processing state of nodes calculated in terms of fuel and input buffer. Node priority is based on a threshold value route selection which is in terms of fuel and if a node meets the threshold criteria to participate in routing. To avoid a node becoming a bottleneck, the optimal information capacity of a metric node concerning traffic and remaining fuel is employed.

Packet Delivery Ratio (PDR): A network generates certain data packets, which are then delivered through a routing mechanism. A data packet is considered delivered when it is received in full and without loss by the destination node.

$$\text{Packet delivery ratio} = (\text{all packets received by the receiver successfully} \times 100)/\text{all packets produced by the senders}$$

In the simulation results analysis, we discovered that the network connectivity giving a packet delivery ratio of more than 95% is dependent on the network characteristics of the GPMOR protocol as opposed to the GPSR protocol. Figure 12 shows the simulation results of a network of varied nodes with the transmission power of a flying node configured.

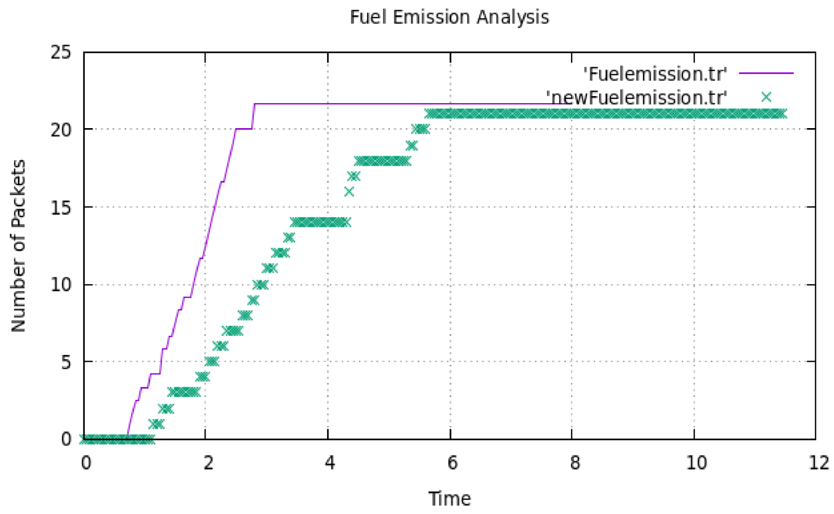

Figure 11. Fuel emission analysis of flying nodes.

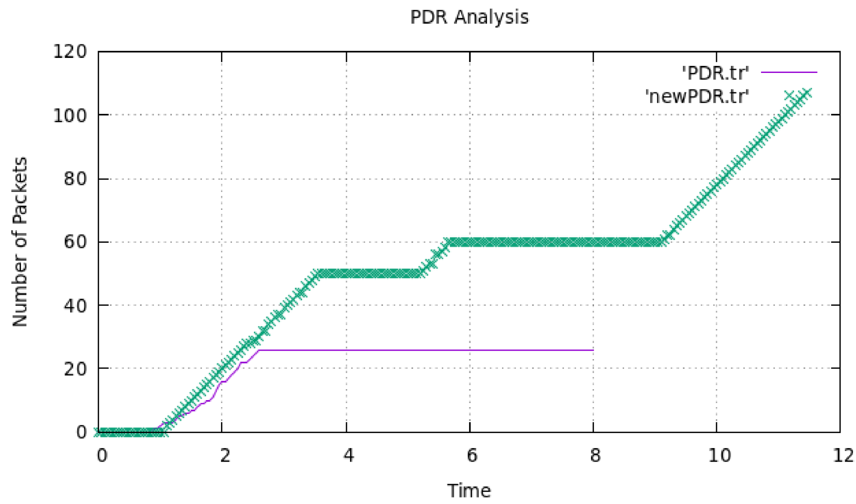

Figure 12. PDR analysis of flying nodes.

As the simulation results show, the number of packets should range from 0 to 120, and the duration should range from 0 to 12 m/s, to ensure the requisite connection between flying nodes. This is represented by two lines in the outcome, the purple and green. The green line denotes the maximum amount of data packets that must be sent to the destination. In this case, increasing the number of nodes does not enhance data quality according to the GPSR protocol; however, it does improve the data quality of flying nodes according to the suggested protocol (GPMOR). As a result, the appropriate packet delivery ratio values were obtained using the simulation settings and network configuration utilized. When the number of nodes is increased to 120, simulation results demonstrate that FANET connection with a packet delivery ratio greater than 95 percent is obtained.

Throughput: Throughput is an important measure for measuring network performance. Throughput can be affected by the distance between the transmitter and the receiver. Throughput is defined as the average data probability of a successful data packet or message passing across a communication connection from the source flying node to the destination flying node in a given time unit. Because the flying nodes' positions may be changed, the distance between two nodes can be modified, and the capacity of the related link can be tuned to increase network throughput. Here, each flying node provides its position and user location information to the ground station, which utilizes all of the flying nodes' current positions.

Figure 13 depicts the analysis of a network in connection with throughput when the number of UAVs, area sizes, and speed is varied. The X-axis denotes the time of simulation in m/s, whereas the Y-axis denotes throughput in bits per second (bps). The network's throughput grew as the number of UAVs increased, as did its performance.

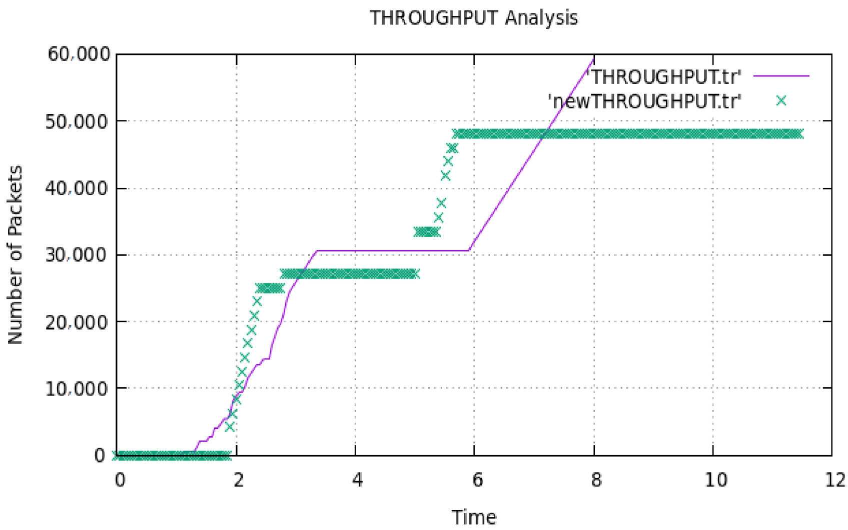

Figure 13. Throughput analysis of flying nodes.

Thus, when UAVs' speeds were reduced to 0 m/s and 12 m/s, the GPMOR protocol beat the GPSR protocol, with the number of UAVs growing to almost 100, as shown in Figure 13. This is because GPMOR allocates time slots for packet transfers to prevent network congestion.

5. Conclusions

This work provides a traffic congestion management method for FANETs based on the firefly algorithm, which may enhance throughput and restrict EED to a specific approach while controlling other parameters, including end-to-end delay, packet delivery ratio, fuel emission, and throughput. The FANET framework is specifically utilized as the mathematical version of the primal problem. In addition, to improve the end-to-end delay limitation outcome, the single-hop delay is predicted with a new form of delay value that is merged with a dual strategy to fix the optimal solution in a distributed system using the firefly algorithm and the traffic congestion control algorithm. The simulation results show that the suggested algorithms boost network throughput and reduce packet delay rates significantly. Because of the best attributes of the firefly algorithm, the proposed approach is inspired by it. Experiments are carried out on the NS2 simulator, and the performance of the proposed approach is evaluated. It is used to compute the most recent attractiveness value of the fireflies, which displays the most desirable and efficient result for the flying nodes. The performance analysis yielded expected results, demonstrating that the proposed approach is effective at optimizing schedules by balancing loads of flying nodes. More effort will be made in the future to incorporate these issues and applicable solutions into the optimization framework and transmission-related strategy.

Author Contributions: Conceptualization, D.P.; methodology, M.K.; validation, D.P. and M.R.; formal analysis, M.K.; writing—original draft preparation, M.K.; writing—review and editing, M.R. and Z.K.; supervision, S.S.A.; funding acquisition, A.S.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Taif University Research Supporting Project number (TURSP-2020/311), Taif University, Taif, Saudi Arabia.

Data Availability Statement: The data used in this article will be shared by requesting to corresponding authors.

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

References

1. Gupta, L.; Jain, R.; Vaszkun, G. Survey of Important Issues in UAV Communication Networks. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 1123–1152. [CrossRef]
2. Kang, M.-S.; Kum, D.-W.; Bae, J.-S.; Cho, Y.-Z.; Le, A.-N. Mobility Aware Hybrid Routing protocol for mobile ad hoc network. In Proceedings of the International Conference on Information Network 2012, Bali, Indonesia, 1–3 February 2012; pp. 410–414.
3. Kaur, M.; Verma, S. Kavita Flying Ad-Hoc Network (FANET): Challenges and routing protocols. *J. Comput. Theor. Nanosci.* **2020**, *17*, 2575–2581. [CrossRef]
4. Johnson, D.B.; Maltz, D.A. Dynamic source routing in ad hoc wireless networks. In *The Kluwer International Series in Engineering and Computer Science Mobile Computing*; Springer: Berlin/Heidelberg, Germany, 1996; pp. 153–181.
5. Zheng, Z.; Sangaiah, A.K.; Wang, T. Adaptive Communication Protocols in Flying Ad Hoc Network. *IEEE Commun. Mag.* **2018**, *56*, 136–142. [CrossRef]
6. Kaur, M.; Prashar, D. Analysis of geographic position mobility oriented routing protocol for FANETs. In Proceedings of the 2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 3–4 September 2021; pp. 1–4.
7. Kaur, M.; Singh, A.; Verma, S.; Jhanjhi, N.Z.; Talib, M.N. FANET: Efficient routing in flying ad hoc networks (FANETs) using firefly algorithm. In *Intelligent Computing and Innovation on Data Science Lecture Notes in Networks and Systems*; Springer: Singapore, 2021; Volume 248, pp. 483–490. [CrossRef]
8. D'Aronco, S.; Toni, L.; Mena, S.; Zhu, X.; Frossard, P. Improved utility-based congestion control for delay-constrained communication. *IEEE/ACM Trans. Netw.* **2016**, *25*, 349–362. [CrossRef]
9. Mehta, R.; Lobiyal, D.K. Cross-layer optimization using two-level dual decomposition in multi-flow ad-hoc networks. *Telecommun. Syst.* **2017**, *66*, 639–655. [CrossRef]
10. Khodaian, A.; Khalaj, B. Delay-constrained utility maximization in multi-hop random-access networks. *IET Commun.* **2010**, *4*, 1908–1918. [CrossRef]
11. Zhang, L.; Zhuo, F.; Xu, H. A cross-layer optimization framework for congestion and power control in cognitive radio ad hoc networks under predictable contact. *EURASIP J. Wirel. Commun. Netw.* **2018**, *2018*, 57. [CrossRef]
12. Li, Y.; Papachristodoulou, A.; Chiang, M.; Calderbank, A.R. Congestion control and its stability in networks with delay sensitive traffic. *Comput. Netw.* **2010**, *55*, 20–32. [CrossRef]
13. Rangisetti, A.K.; Tamma, B.R. QoS Aware load balance in software defined LTE networks. *Comput. Commun.* **2017**, *97*, 52–71. [CrossRef]
14. Kafi, M.A.; Ben-Othman, J.; Ouadjaout, A.; Bagaa, M.; Badache, N. REFIACC: Reliable, efficient, fair and interference-aware congestion control protocol for wireless sensor networks. *Comput. Commun.* **2017**, *101*, 1–11. [CrossRef]
15. Hajiesmaili, M.H.; Talebi, M.S.; Khonsari, A. Multiperiod network rate allocation with end-to-end delay constraints. *IEEE Trans. Control. Netw. Syst.* **2018**, *5*, 1087–1097. [CrossRef]
16. Alaei, M.; Sabbagh, P.; Yazdanpanah, F. A QoS-aware congestion control mechanism for wireless multimedia sensor networks. *Wirel. Netw.* **2018**, *25*, 4173–4192. [CrossRef]
17. Silva, A.P.; Obraczka, K.; Burleigh, S.; Nogueira, J.M.; Hirata, C.M. A congestion control framework for delay- and disruption tolerant networks. *Ad. Hoc. Netw.* **2019**, *91*, 101880. [CrossRef]
18. León, J.P.A.; Begin, T.; Busson, A.; Llopis, L.J.D.L.C. A fair and distributed congestion control mechanism for smart grid neighborhood area networks. *Ad. Hoc. Netw.* **2020**, *104*, 102169. [CrossRef]
19. Lubna, T.; Mahmud, I.; Cho, Y.-Z. D-LIA: Dynamic congestion control algorithm for MPTCP. *ICT Express* **2020**, *6*, 263–268. [CrossRef]
20. Verma, L.P.; Kumar, M. An IoT based Congestion Control Algorithm. *Internet Things* **2020**, *9*, 100157. [CrossRef]
21. Kaur, M.; Prashar, D.; Rashid, M.; Alshamrani, S.S.; AlGhamdi, A.S. A novel approach for securing nodes using two-ray model and shadow effects in flying ad-hoc network. *Electronics* **2021**, *10*, 3164. [CrossRef]