Fusion Routing Algorithm for Aerial Wireless Network

Gayathiri E[1], Mukundhan T.E[2], Vishnu Ch[3], E. Elamaran[4]
[1,2,3]*Department of electronics and communication engineering SRM Institute of Science and Technology, SRM Nagar, Kattankulathur 603203, Kanchipuram, Chennai, T.N., India*
[4]*Assistant Professor (OG), Electronics and Communication Dept., SRM Institute of Science and Technology, SRM Nagar, Kattankulathur 603203 Kanchipuram, Chennai, T.N., India*

Abstract **- Flying Ad-Hoc Networks (FANETs) are designed for providing seam- less connectivity between mobile devices in both air and ground level communications. The problem of neighbor discovery and data exchange relies on the availability and distribution of the infrastructure units. This limits the available routes in communicating the neighbors in mobile environments. This project introduces a novel quality of service (QoS) routing for improving the efficiency of FANETs. In this routing, the conventional ant colony optimization (ACO) and greedy methods are integrated for providing improved routing. ACO agents are responsible for discovering optimal routes to the destination whereas the greedy method filters the QoS defacing factors for improving the routing solution. The ant agents identify local solutions through periodic pheromone update and generate maximum possible solutions. In the greedy filtering process, the QoS factors are identified, and the best-afford nodes are used for routing and transmission. The delay and congestion factors are used for determining best-afford route using the greedy process. The number of ACO solutions is refined based on the QoS output in the destination that makes up the best solution. The above processes are recurrent until the transmission between the source and destination is successful with better QoS.**

Index Terms **- ACO, FANET, Greedy Algorithm, QoS Routing.**

I.INTRODUCTION

There has been a significant increase in technological advances in electronic sectors, communication sectors, transmission, and sensor technologies. Flying Ad-hoc networks such as UAV systems fly autonomously and are operable remotely without the help of any humans onboarding them. They have many good qualities such as easy installation and comparatively fewer expenses for operating them. They are also highly flexible and highly versatile. They can be used in a variety of fields, including military applications such as wildlife management, border surveillance, search and destroy operations, and so on. They also have civilian applications, such as weather forecasting, earthquakes, floods, tsunamis and other disaster control measures, and traffic surveillance.

Single UAV systems are being used for many years until now so rather than creating and working with a large UAV that is singular it is better to employ a cluster of little UAVs since they give out many plus points rather than the former idea. But creating a cluster of multiple UAVs working together brings out new problems such as communicating with each other nodes of UAVs.

FANETS also known as Flying Ad-hoc Networks is a network of hoc between UAVs and they are monitored as a replacement network. Usually, FANETS are flying without a specific plan and they are a network where a small cluster of UAVs are attached in an ad-hoc way, where they are integrated into another group to form a high-level aim.

FANET UAVs has grown in popularity and importance in recent operations such as tracking locations, intranet onsite and internet onsite, search and rescue missions, as well as shipping of goods and services, patrolling, weather monitoring, traffic sensing, and so on, due to their flexibility and comparatively lower operating costs and ease of installation. They are designed specifically to provide easy connection between devices at air and ground level with mobile communications. Though they have multiple advantages, they also have new challenges such as communication, and also as the number of drones is increased in UA Systems the design of efficient network architectures is a difficult problem to handle.

The problem of discovering their next neighbours and exchanging data between them depends on the supply of their units. Hence, FANETs are a very important aspect while achieving application-dependent goals. To collaborate, give signals and try to coordinate with each other Routing is very essential.

To improvise the pathfinding between the nodes this project implements a novel and completely new Quality of Service (QoS). This routing process improves routing by including the help of Ant Colony Optimization (ACO) and greedy algorithms.

A.Ant Colony Optimization:

ACO is a probability-based method for solving computational problems to seek out reduced paths or shortest paths through graphs. The working is directly modelled to the behaviour of ant colonies. When the agents are exploring the solutions, they constantly search the space of the site and parameters for all possible solutions that are optimal, which is very close to how real ants drop pheromones that help other ants find food and resources. During this process by multiple iterations, the optimal solution is obtained. ACO Agents are liable for discovering optimal routes to the destination. The ant agents identify local solutions through periodic pheromone updates and generate the utmost possible solutions.

B. Greedy process

The greedy algorithm is a simple but effective algorithm that can solve a range of problems. When we have to discover an optimal solution in multiple node routing problems. The algorithm has certain steps and after each step, it checks for the optimal solution. It employs Dijkstra's algorithm to obtain graphs, resulting in the most cost-effective solution. The use of QoS factors aids in the further refinement of the algorithm and the selection of the best possible solution for our needs. This project incorporates a brand-new Quality of Service model (QoS).

II.OVERVIEW OF THE PROJECT

The major drawback of neighbour node discovery and transfer of data packets is its availability. There are many routing protocols used, but the result is the same, with no improvements. The current protocol does a crude job of multicasting and optimizing objectives. It doesn't take end-to-end delay and congestion into account. Packet drop delay and power are unique to each node. Packet drop cannot be completely wiped out but can only be minimized. There are two types of protocol during packet transfer, TCP and UDP. TCP will receive acknowledgments but UDP will not. It will limit the available routes in communicating. QoS routing makes FANETs more efficient. Thus, 2 algorithms are used, one is ACO (Ant Colony Optimisation), and the other is the Greedy algorithm.

Ant colony optimization, also known as ACO, has a probabilistic technique for its solutions, which are very promising in finding appropriate routes (which have very little packet drop and more efficiency). It finds all the appropriate paths from the start node to the end node. ACO is dependent on communication between organic ants, which secrete pheromones. Similarly, simulation agents that act as biological ants have a value of pheromones for each pathway based on the chosen pathway. This is the most popular standard. Real ants secrete pheromones that lead towards resources while exploring the environment, simulated ants similarly record their positions and the quality of their solutions. In subsequent simulated iterations, more ants locate better solutions. Therefore, after discovering several local solutions, Greedy is used to find specific solutions because not every local solution found can be used simultaneously. Greedy is used to choose the shortest path according to the change of pheromones reported by ACO. A greedy algorithm is a simple and intuitive algorithm used for optimization issues. Greedy is an algorithmic paradigm that constructs a piece-by-piece solution, always selecting the next piece that provides the most obvious and immediate advantage. Thus, problems where the optimal local choice also leads to a comprehensive solution are better suited to Greedy. Greedy algorithms work by recursively constructing an array of objects from the smallest possible pieces. Greedy finds the cumulative and average fitness for a neighbour and then decides the shortest path. The solution found in the existing system would remain, whether the solution exists or not, but that is not the case in the proposed system. Additionally, in the existing, the greedy process is a probabilistic method, but the type that exists in the proposal is a combination and partly genetic. The existing one gives only one solution whereas the proposed one gives many solutions and the main advantage is it has premature converging. The current process ends after communicating with fewer nodes (e.g., 75), preventing premature convergence and allowing more nodes to communicate. This proposal focuses on QoS routing by addressing congestion and delay constraints in identifying neighbours in the network. The adaptability of the devices to the network topology and infrastructures helps to address the convergence issues. The metrics are valuable for identifying optimal connecting neighbours without disturbing the performance. The considering metrics are either independent or dependent based on the transmission requirement. This serves as the novel part in providing seamless communication and congestion-free transmissions.

III.SYSTEM ANALYSIS

FANET nodes have been created virtually Using NS2. Initially, ACO acts and finds the neighbour node, utilizing its level of pheromones. The delay and congestion conditions determine the pheromone value. Delay and congestion are significant factors in locating a path to neighbours. Less delay and less congestion imply a higher pheromone value. Less delay and moderate congestion mean a lower pheromone value. The pheromone value is ranked based on these two conditions. Under Pheromones, there are two subcategories one is congestion, and the other is availability. Congestion happens as the number of packets sent through the network exceeds the network's packet processing capability. Nonetheless, the time delay must be minimal. ACO is revised after the whole process is completed. After each transmission pheromone update takes place. The pheromone which has a high value is chosen as the first neighbour node. The nodes which have low pheromone value are discarded and new nodes are rediscovered by the ant initialization.

The greedy method shows any possible output and finds the best optimal solution. The Quality of Service (QOS) factor consists of Transmission rate, Delay, and Packet drop. After the improvisation of Quality Of Service, the transmission path is updated after each node. Based on the preceding process, the greedy algorithm finds a possible route combination. The four combinations for the output are Low Pheromones and available, Low Pheromones and unavailable, High Pheromones and unavailable, High Pheromones and unavailable. Under these conditions, the greedy chooses the best route that has high pheromone value, availability, and a high rate of transmission. Then the pheromone level is updated, where the delay factor is taken into consideration. New neighbors are found, according to the level of congestion. When a new neighbour is uncovered, a new path is discovered and updated.

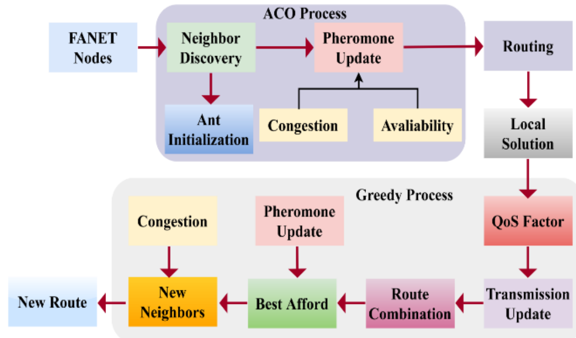

Fig.1. System Block Diagram

The greedy method shows any possible output and finds the best optimal solution. The Quality of Service (QOS) factor consists of Transmission rate, Delay, and Packet drop. After the improvisation of Quality Of Service, the transmission path is updated after each node. Based on the preceding process, the greedy algorithm finds a possible route combination. The four combinations for the output are Low Pheromones and available, Low Pheromones and unavailable, High Pheromones and unavailable, High Pheromones and unavailable. Under these conditions, the greedy chooses the best route that has high pheromone value, availability, and a high rate of transmission. Then the pheromone level is updated, where the delay factor is taken into consideration. New neighbours are found, according to the level of congestion. When a new neighbour is uncovered, a new path is discovered and updated.

IV.SYSTEM FLOW

a) Virtual ants (agents) are deployed during Ant initialization to look for neighboring nodes.
b) Following the identification of neighboring nodes, the nodes that are immediately adjacent to each other are chosen to be the neighboring nodes. Pheromone values are identified and modified.
c) If the pheromone value of i is higher than j, where i and j denote the two neighboring nodes, then the local solution is the one with a high pheromone value (i).
d) The lesser pheromone value of the node (j) is updated, and another virtual agent is introduced to locate other neighboring nodes.

e) After locating a local solution, delay and congestion values are determined and modified. The conditional values of the whole path and the node next to it are compared.
f) The node delay is determined if the path congestion exceeds the node congestion.
g) In comparison to the previous congestion value, path delay is computed and availability is validated.
h) Node delay is computed when it is available. When this is not the case, there is a reduction in the number of neighbors due to unavailability.
i) After the node reduction, an additional ant is dispatched to find a node to compensate for the reduction.
j) The delay between the path and the node is compared.
k) If the path delay exceeds a node, it implies the presence of an undiscovered hidden node.
l) This adds one to the number of neighboring nodes.
m) If the path delay is less than the node delay, a local solution is updated.

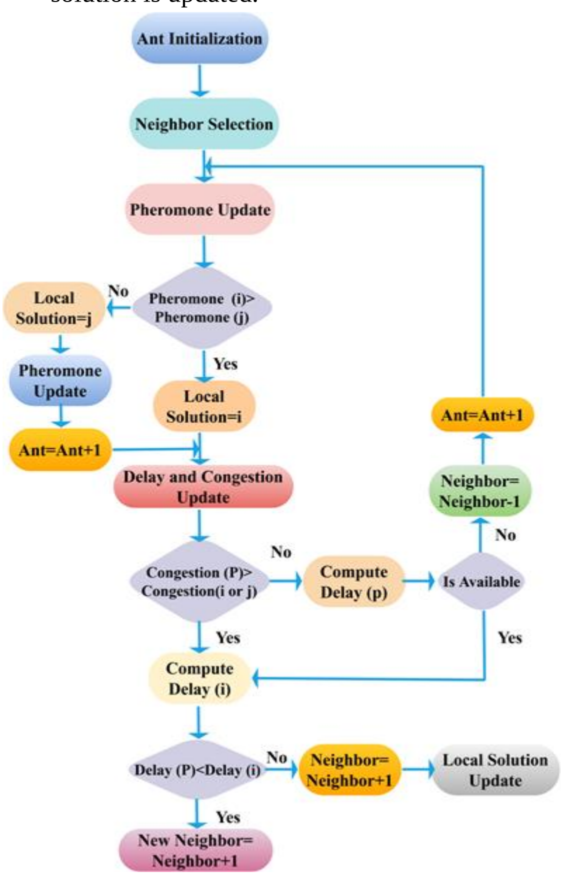

Fig.2. Flowchart

Graphical Representation of the Nodes:

A stimulating setting was developed, consisting of 50 nodes that are each 730.5 meters apart. There is 'n' number of nodes that are adjacent to each other. The frame, the link id are generated, and the frame transmit time is calculated. Broadcasts (larger circles) occur between nodes within and beyond the group. It is used to locate the nearest neighboring node and to transfer data utilizing routing protocols. The distance between neighboring nodes is calculated using the Euclidian formula.

$$d(x,\ y)\ =\ \sqrt{\sum_{i=1}^{n}(y_i - x_i)^2}$$

where x and y are the node coordinates. A single node would be in control of data accumulation. In that case, the zero node serves as a transmitter, moving in a random wave pattern toward other nodes when interacting. It can communicate with any node and travel instantly in any direction. The Random wave point movement of this node can be configured per the requirements. The falling square represents a packet drop. TCP packets provide acknowledgments, while UDP packets do not. Thus, TCP is used for communication. When a packet is dropped, retransmission occurs, using the newly created connection id. There are three types of color change for power analysis:

1.Communicating - light blue

2.Next communication phase - brown

3.No communication-green

The conversation pattern changes after the algorithm are enabled. Ant agents generate a large number of link ids. The ACO will initiate a pheromone update and continue to iterate until several new solutions are discovered. Finally, greedy selects the ultimate optimal solution.

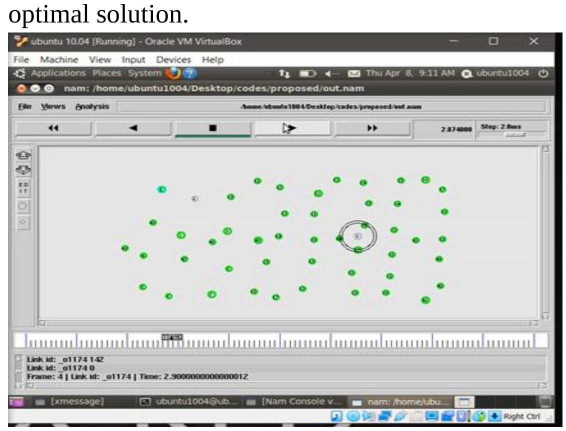

Fig.3. NAM Window

The environment set for the process:

- Number of Nodes: 50
- Distance between Nodes:730.5metres
- Channel: Wireless Channel
- Frequency: 2.4GHz
- Standard: IEEE 802.1 (MAC type)
- Propagation: 2-way ground propagation
- Antenna model: Omni antenna

Routing protocol: DSR (Dynamic Source Routing.

V.SYSTEM DESIGN

A. Study of Software:

NS stands for Network Simulator, by name defines as simulates network. NS2 channel log contains new updated contents and availability in it. NS was first invented in 1989 and has advanced in recent years. DARPA started funding Network Simulator for years after its invention. It also supported and gave an up-hand in its advancement. NS was designed to simulate and study networks. It can operate both through wired and wireless mediums and also supports TCP ad multicast protocols. NS was built on the basic building blocks of C++ and OTcl, to provide an appropriate interface for networking studies. Writing codes in NS requires knowledge of C++ and object-oriented TCL. Tool Command language is the abbreviation for TCL

It contains six parts: Global declaration is where the input command is uploaded to set the environment like topology, standards used, number of nodes, type of propagation, the distance between nodes, and many more. Object definition is a program that was built on building blocks of javascript. It defines objects like packet drop, delay, and many others. Node deployment /placement where nodes, defined earlier will now be deployed and placed according to the user. It may be a particular placement or random wave motion. Agent definition comes after the placement of nodes according to the user. UDP and TCP, both the protocols are defined and implemented in this process. Application definition where there are two types of messages, one is constant and variable messages. Finish the procedure, at the end, if a graph is required the user can call out the function and xgraph and a nam window opens following the input command codes given by the user. It's mandatory what details are been given as input, this tool works perfectly for non-sophisticated systems. In the case of a complex system, many details may be skipped by the tool and end up in non-manageable simulation and give wrong or no output. So it is necessary to keep into account not to include irrelevant information.

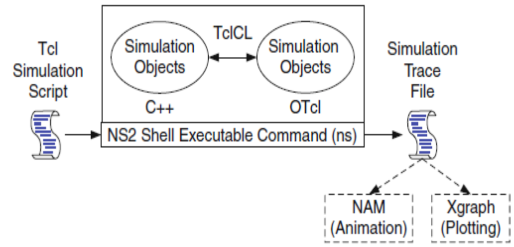

Fig.4. NS2 Architecture

NS accepts command (input) only if it's OTcl script (which is a simulation program). After the input is given to NS, all the C++ libraries and OTcl libraries act and read the input data given by the user. NS simulator library consists of Event Scheduler Objects, Network Component Objects, Network Setup Helping, Modules (plumbing Modules). C++ defines the backend of the system and is responsible for major changes in the code.NS2 shell command consists of C++ and OTcl libraries. After the complete run of the software using NS libraries, simulation results are shown which exists in two windows accordingly, one in Xgraph and the other in NAM (Network Animator one in Xgraph and the other in NAM

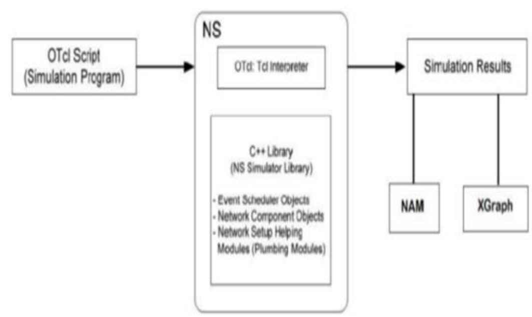

Fig.5. Basic Structure of NS

To start the simulation the user writes the OTcl code also known as a script file, this file is used as simulation code filed to start the simulation, which we had to line the topology victimization of the network object and pipeline functions from the library and tell the availability once to start and stop-transfer packets, by event schedule, once associate OTcl code translator receives associate OTcl file, it'll set a physical surrounding parameter which is already mentioned above, consistent with the necessity for our surroundings. If a user desires to make a replacement network object instead of writing a replacement file, produce a specific object from the library, then bend the data path thereto object.

B. NAM window (Network Animator Window)

Network Animator may be a TCL tool supported action or an interpreter-based graphical visualizing network simulator that traces the real-world packet paths. It uses various data inspection, topology layout, and packet-level action.

C. Xgraph

X graph is a utility package tool for displaying the outcome of a simulation in a graphical format. It plots the graph on the X-Y axis using the coordinates of every data file, or the standard point if no files are defined.

D. Software Requirements:

The simulator used in this case is the Ns2 Software platform. The operating system is Linux/Ubuntu 10.04. VM Ware is the cross platform. Ns allinone-2.35 is the package name. TCL is the scripting language. Perl is used in the analysis script.

VI.SIMULATION RESULTS

6.1. Delay Output

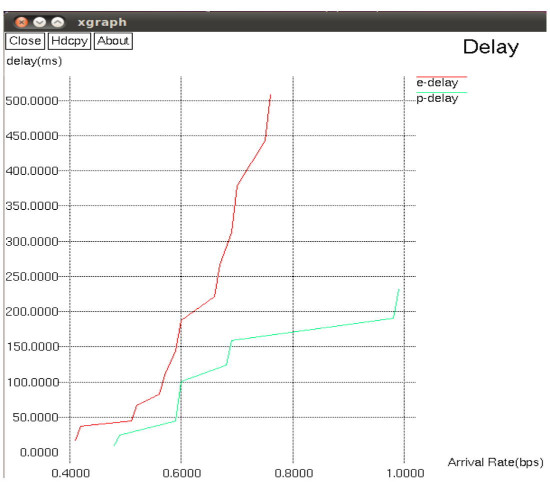

Fig.6. Delay of Data

The delay specifies the time taken between the transfer of the packet and the receipt. It should be less so the system can be more efficient. Comparison of two systems illustrated on the x graph. The RED curve shows the current system and the GREEN curve shows the proposed system. With reference to the figure above, the RED curve shows that at times the delay is very high. On the other hand, the GREEN curve is much better compared to the RED curve and the delay encountered is almost constant over a period of time.

6.2. Throughput Output

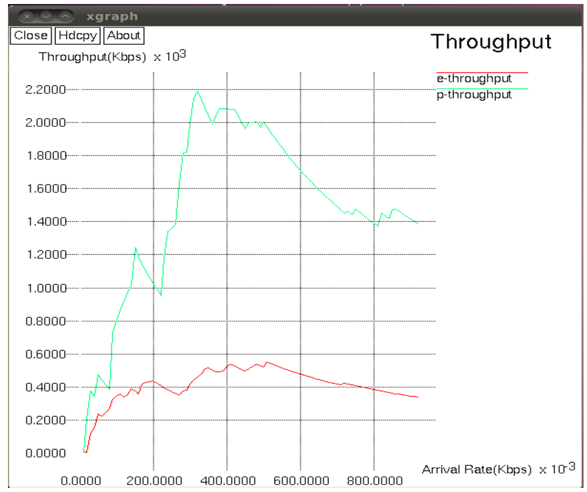

Fig.7. Throughput

Throughput is the data transfer per unit time. Comparison of two systems shown on the x-graph. RED curve indicates the existing system and GREEN curve indicates the proposed system. With reference to the figure above, the RED curve shows LOW throughput rate.. The maximal value is 0.5000. However, the GREEN curve has a much throughput value than the RED curve. The maximum is greater than 2.0000.

6.3. Mobility Overhead Output

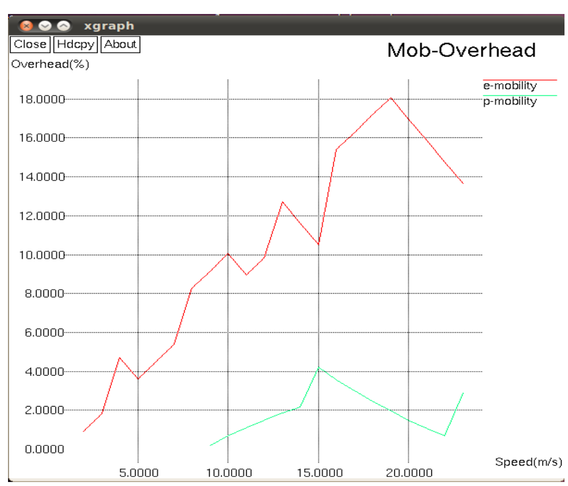

Fig.8. Mobility Overhead

Mobility overheads refer to the total number of control messages by the sum of control messages and data messages. Comparison of two systems shown on the x-graph. The RED curve shows the existing system, while the GREEN curve shows the proposed system. Based on the figure above, the RED curve shows that overhead fluctuates and is very high. On the other hand, the GREEN curve is a lot better than the RED curve and overheads values are low.

6.4. Power Consumption Output

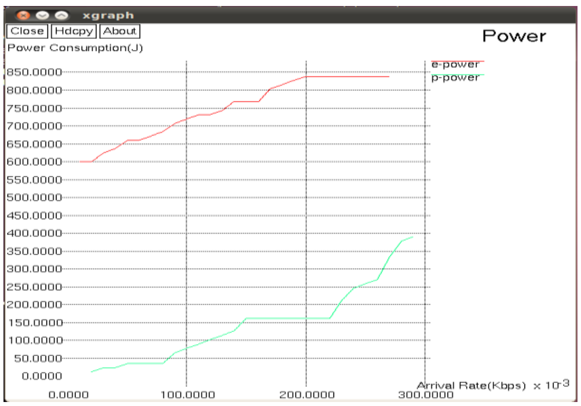

Fig.9. Power Consumption

Power consumption is the cumulative energy used by the node within a complete communication interval. Comparison of two systems shown on the x-graph. The RED curve shows the existing system, while the GREEN curve shows the proposed system. With reference to the figure above, the RED curve shows that the power requirement is high. In contrast, the extreme value of the GREEN curve is significantly lower than that of the RED curve, indicating that the proposed system requires less power for transmission.

6.5. Packet Drop Output

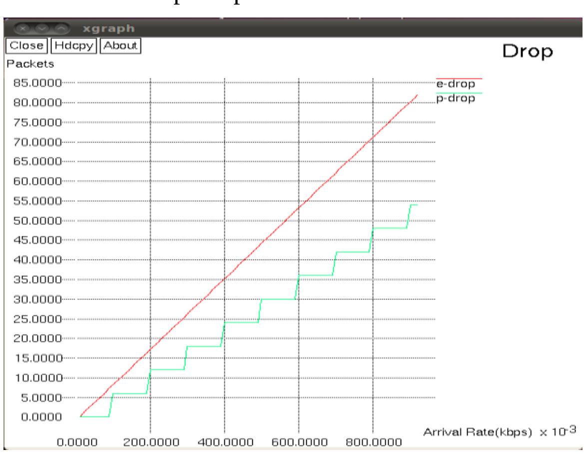

Fig.10. Packet Drop

The packet drop is the total number of packets sent less the total number of packets received. Comparison of two systems shown on the x-graph. The RED curve shows the existing system, while the GREEN curve shows the proposed system. Referring to the figure above, the RED curve increases sharply, giving a value of approximately 85.0000 indicates a LOW Package Delivery Ratio. Furthermore, the GREEN curve has maximum values below 55.0000, which indicates a HIGH PACKET DELIVERY Ratio.

VII.CONCLUSION

In this project, A routing protocol for Flying Ad-hoc Networks is being proposed and implemented. The main objectives taken into consideration were to produce minimal delay along with improved congestion transmission. The approach to achieve the objective is the hybridization of two traditional algorithms, which are, Ant colony and greedy optimization. The novelty of this project is the addition of Quality-of-Service factors into the routing algorithm which has considerably help the project to outperform the existing model and produce significantly improved results. Along with the implementation of these algorithms, a method that helps to adjust the routing parameters when the FANETs are in highly mobile and ever-changing topographical situations is also proposed.

VIII.FUTURE ENCHANCEMENT

In the future, less complex multi-flow transfer and rerouting systems are likely to be evaluated. In this process, delayed routes and less used neighbors and their availability are identified for multi-flow processing at the same time. The proposed QoS-based greedy routing is planned to be included for its scalability in multiple flow processing.

ACKNOWLEDGMENT

I would like to express my deepest gratitude to my guide, Mr.E.Elamaran his valuable guidance, consistent encouragement, personal caring, timely help and providing me with an excellent atmosphere for doing research. All through the work, in spite of his busy schedule, he has extended cheerful and cordial support to me for completing this research work.

REFERENCES

[1] Qiannan, Z., Ziyang, Z., Chen, G., & Ruyi, D. (2014). Path planning of UAVs formation based on improved ant colony optimization algorithm. Proceedings of 2014 IEEE Chinese Guidance, Navigation and Control Conference.

[2] Khaledi, M., Rovira-Sugranes, A., Afghah, F., & Razi, A. (2018). On Greedy Routing in Dynamic UAV Networks. 2018 IEEE International Conference on Sensing, Communication and Networking (SECON Workshops).

[3] QIANQIAN SANG, HONGHAI WU , LING XING, (Member, IEEE), HUAHONG MA , AND PING (2020). An Energy-Efficient Opportunistic Routing Protocol Based on Trajectory Prediction for FANETs. IEEE Access Volume 8.

[4] Y Eun, H Bang, "Cooperative task assignment/path planning of multiple unmanned aerial vehicles using genetic algorithms," Journal of Aircraft, vol. 46, no. 1, pp. 338-343, 2009

[5] Berger Jean, Boukhtouta Abdeslem, Benmoussa Abdelhamid, Kettani Ossama, "A new mixed-integer linear programming model for rescue path planning in uncertain adversarial environment," Computers and Operations Research, vol. 39, no. 12, pp. 3420-3430, Dec. 2012.

[6] Yang X, Ding M Y, Zhou C P, "Fast marine route planning for UAV using improved sparse A* algorithm," International Conference on Genetic and Evolutionary Computing, China, 2010, pp. 190-193.

[7] "Gartner says almost 3 million personal and commercial drones will be shipped in 2017," Feb 2017. [Online]. Available: http://www.gartner. com/newsroom/id/3602317

[8] J. d. Zhang, Y. j. Feng, F. f. Shi, G. Wang, B. Ma, R. s. Li, and X. y. Jia, "Vehicle routing in urban areas based on the oil consumption weight - dijkstra algorithm," IET Intelligent Transport Systems, vol. 10, no. 7, pp. 495–502, 2016

[9] Z. Sun, P. Wang, M. C. Vuran, M. A. Al-Rodhaan, A. M. Al-Dhelaan, and I. F. Akyildiz, ''BorderSense: Border patrol through advanced wireless sensor networks,'' Ad Hoc Netw., vol. 9, no. 3, pp. 468–477, May 2011.

[10] M. Erdelj, E. Natalizio, K. R. Chowdhury, and I. F. Akyildiz, ''Help from the sky: Leveraging UAVs for disaster management,'' IEEE Pervas. Comput., vol. 16, no. 1, pp. 24–32, Jan. 2017.

[11] H. Ma and Y. Liu, ''On coverage problems of directional sensor networks,'' in Proc. 1st Int. Conf. Mobile Ad-Hoc Sensor Netw. (MSN). Berlin, Germany: Springer, 2005, pp. 721–731.

[12] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, ''Wireless communication using unmanned aerial vehicles (UAVs): Optimal transport theory for hover time optimization,'' IEEE Trans. Wireless Commun., vol. 16, no. 12, pp. 8052–8066, Dec. 2017

[13] F. Jiang and A. L. Swindlehurst, ''Optimization of UAV heading for the ground-to-air uplink,'' IEEE J. Sel. Areas Commun., vol. 30, no. 5, pp. 993–1005, Jun. 2012.

[14] R. Martin, I. Rojas, K. Franke, and J. Hedengren, ''Evolutionary view planning for optimized UAV terrain modeling in a simulated environment,'' Remote Sens., vol. 8, no. 1, p. 26, Dec. 2015