

# Fuzzy Logic Ant Colony based Routing for Mobile Ad-Hoc Networks towards Improved Quality of Services

J. Jayaperiya<sup>1</sup>, S. Saravana Kumar<sup>2</sup>

<sup>1</sup>MPhil Research Scholar, Sri Jayendra Saraswathi Maha Vidhalaya College of Arts and Science, India

<sup>2</sup>Assistant Professor, Sri Jayendra Saraswathi Maha Vidhalaya College of Arts and Science

**Abstract:** Mobile Ad Hoc Network (MANET) is a dynamic multihop wireless network which is established by a set of mobile nodes on a shared wireless channel. One of the major issues in MANET is routing due to the mobility of the nodes. QoS routing plays an important role for providing QoS in wireless ad hoc networks. The biggest challenge in this kind of networks is to find a path between the communication end points satisfying user's QoS requirement. In our proposed work we have used the fuzzy based ant colony optimization to determine the vagueness in routing protocol to provide high quality of service based on the negative consequences that usually occur in manet. Swarm intelligence, as demonstrated by a natural biological swarm, such as an ant colony, has many powerful properties that are desirable for effective routing in communications networks. In this paper, we propose an intelligent routing algorithm that we are calling Fuzzy Logic Ant-based Routing (FLAR), which is inspired by ant colonies and enhanced by fuzzy logic techniques. Using a fuzzy system as an intelligent and expert mechanism allows multiple constraints to be considered in a simple and intuitive way. Simulation results and a comparison of the proposed method with two state-of-the-art routing algorithms show better performance and a higher fault tolerance for our approach, particularly in regard to link failures

**Keywords:** dynamic, routing, fuzzy logic, Ant-based routing

## 1. Introduction

Mobility management has long been recognized as a major challenge in mobile ad hoc networks (MANETs). A MANET generally has the following characteristics:

- 1) New members can join and leave the network any time
- 2) No base station is available to provide connectivity to backbone hosts or to other mobile hosts.
- 3) It is difficult to implement sophisticated scheme for handover and location management
- 4) Each node acts as a router, forwarding packets from others nodes
- 5) Communication connectivity is usually "weak" in the sense that it is easily broken due to node movement.

To ensure better tracking quality for a moving target, it is beneficial to dynamically move nodes to advantageous locations. In this project, we present an efficient mobility management scheme that can be implemented in a fully distributed manner. The proposed mobility management scheme is a general framework that incorporates both the positive and negative consequences of node movement; it allows a node to autonomously decide whether it should move and where it should move to. It is based on concepts taken from Bayesian estimation theory.

## 2. Related Work

MAODV [1] is a shared-tree-based protocol that is an extension of AODV [2] to support multicast routing. With the unicast route information of AODV, MAODV constructs the shared tree more efficiently and has low control overhead. Each node will broadcast a Hello message to its neighbors if it does not send any packet within a period of

time. The lack of a Hello message indicates that the link between a node and its neighbor is broken. Then the node locally floods a join message towards the group leader. Only those tree nodes which are closer to the group leader and have fresher paths to the group leader respond to this join message. The shared tree may be partitioned due to node mobility and hence two or more group leaders may co-exist.

AMRIS [3] is an on-demand shared-tree-based protocol which dynamically assigns every node in a multicast session an id-number. The multicast tree is rooted at a special node called Sid and the id-numbers of surrounding nodes increase in numerical value as they radiate from the Sid. These id-numbers help nodes know which neighbors are closer to the Sid and this reduces the cost to repair link failures. BEMRP [4] is a sender-tree-based protocol which emphasizes on high multicast efficiency. BEMRP requires each new member to set up a branch with the fewest new forwarding nodes being added to the multicast tree. A route optimization process is also introduced to detect and remove unnecessary forwarding nodes. When a new node P wishes to join a group, it globally floods a join message into the network. Only tree nodes respond to the join message and will reply the one along the shortest path to P. P then selects the shortest path to connect to the tree node

ADMR [5] is an on-demand sender-tree-based protocol which adapts its behavior based on the application data sending pattern. It does not require periodic floods of control packets, periodic neighbor sensing, or periodic routing table exchanges. The application layer behavior allows efficient detection of link breaks and expiration of routing state.

DDM [6] is a sender-tree-based protocol that is designed for

small group. DDM has no any multicast routing structure. It encodes the addresses of group members in each packet header and transmits the packets using the underlying unicast routing protocol. When a node P is interested in a multicast session, it unicasts a join message toward the sender of the session. The sender adds P into its member list (ML) and unicasts an ACK message back to P. DDM has two operation modes: stateless mode and soft-state mode.

MCEDAR [7] is a multicast extension to the CEDAR [8] architecture which provides the robustness of mesh structures and the efficiency of tree structures. MCEDAR uses a mesh as the underlying infrastructure, but the data forwarding occurs only on a sender-rooted tree. MCEDAR is particularly suitable for situations where multiple groups coexist in a MANET. At first, MCEDAR partitions the network into disjoint clusters. Each node exchanges a special beacon with its one hop neighbors to decide that it becomes a dominator or chooses a neighbor as its dominator. A dominator and those neighbors that have chosen it as a dominator form a cluster. A dominator then becomes a core node and issues a message to nearby core nodes for building virtual links between them. All the core nodes form a core graph.

ODMRP [9] provides richer connectivity among group members and builds a mesh for providing a high data delivery ratio even at high mobility. It introduces a "forwarding group" concept to construct the mesh and a mobility prediction scheme to refresh the mesh only necessarily. The first sender floods a join message with data payload piggybacked. The join message is periodically flooded to the entire network to refresh the membership information and update the multicast paths. An interested node will respond to the join message.

DCMP [10] aims at mitigating the high control overhead problem in ODMRP. DCMP dynamically classifies the senders into different categories and only a portion of senders need issue control messages. In DCMP, senders are classified into three categories: active senders, core senders, and passive senders. Active senders flood join messages at regular intervals. Core senders are those active senders which also act as the core node for one or more passive senders. A passive sender does not flood join messages, but depends on a nearby core sender to forward its data packets. ACMRP [11] presents an adaptive core mechanism in which the core node adapts to the network and group status. In general mesh-based protocols, the mesh provides too rich connectivity and results in high delivery cost. Hence, ACMRP forces only one core node to take responsibility of the mesh creation and maintenance in a group. MANSI [12] relies on only one core node to build and maintain the mesh and applies swarm intelligence to tackle metrics like load balancing and energy conservation. Swarm intelligence refers to complex behaviors that arise from very simple individual behaviors and interactions. NSMP [13] utilizes the node locality concept to lower the overhead of mesh maintenance. For initial path establishment or network partition repair, NSMP occasionally floods control messages through the network. CAMP [14] is a receiver-initiated protocol. It assumes that an underlying unicast routing protocol provides correct distances to known destinations. CAMP establishes a

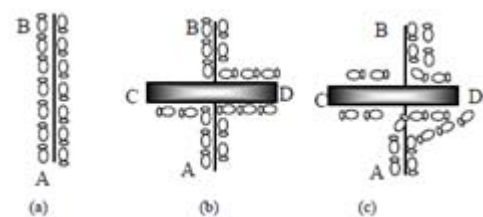
mesh composed of shortest paths from senders to receivers. One or multiple core nodes can be defined for each mesh, and core nodes need not be part of the mesh, and nodes can join a group even if all associated core nodes are unreachable.

### 3. Proposed Framework

#### 3.1 Overview of Proposed Framework

The ACO meta-heuristic is based on generic problem representation and the definition of the ant's behavior. ACO adopts the foraging behavior of real ants. When multiple paths are available from nest to food, ants do random walk initially. During their trip to food as well as their return trip to nest, they lay a chemical substance called pheromone, which serves as a route mark that the ants have taken [4]. Subsequently, the newer ants will take a path which has higher pheromone concentration and also will reinforce the path they have taken. As a result of this autocatalytic effect, the solution emerges rapidly. To illustrate this behavior, let us consider the experiment shown in Figure 1. A set of ants moves along a straight line from their nest A to a food source B (Figure 1a). At a given moment, an obstacle is put across this way so that side (C) is longer than side (D) (Figure 1b). The ants will thus have to decide which direction they will take: either C or D. The first ones will choose a random direction and will deposit pheromone along their way. Those taking the way ADB (or BDA), will arrive at the end of the obstacle (depositing more pheromone on their way) before those that take the way ACB (or BCA). The following ants' choice is then influenced by the pheromone intensity which stimulates them to choose the path ADB rather than the way ACB (Figure 1c). The ants will then find the shortest way between their nest and the food source. In most cases, an artificial ant will deposit a quantity of pheromone represented by  $H\tau_{i,j}$  only after completing their route and not in an incremental way during their advancement. This quantity of pheromone is a function of the found route quality. Pheromone is a volatile substance. An ant changes the amount of pheromone on the path  $(i, j)$  when moving from node  $i$  to node  $j$  as follows:  $\tau_{i,j} = \sigma \cdot \tau_{i,j} + \Delta\tau_{i,j}$  where  $\sigma$  is the pheromone evaporation factor. It must be lower than 1 to avoid pheromone accumulation and premature convergence. At one point  $i$ , an ant chooses the point  $j$  (i.e. to follow the path  $(i, j)$ ) according to the following probability

$$P_{i,j} = \frac{(\tau_{i,j})^\alpha \cdot (\eta_{i,j})^\beta}{\sum_{(i,k) \in C} (\tau_{i,k})^\alpha \cdot (\eta_{i,k})^\beta}$$



**Figure 1:** Behaviour of the Ants for searching the food. where,  
 •  $\tau_{i,j}$ : is the pheromone intensity on path  $(i, j)$ .  
 •  $\eta_{i,j}$ : is the ant's visibility field on path  $(i, j)$  (an ant assumes that there is food at the end of this path).

- $\alpha$  and  $\beta$  : are the parameters which control the relative importance of the pheromone intensity compared to ant's visibility field.
- C: represents the set of possible paths starting from point  $i$  ( $(i, k)$  is a path of C).

### 3.2 Network Routing Using ACO

Mobile ad hoc network routing is a difficult problem because network characteristics such as traffic load and network topology may vary stochastically and in a time varying nature. The distributed nature of network routing is well matched by the multi agent nature of ACO algorithms. The given network can be represented as a construction graph where the vertices correspond to set of routers and the links correspond to the connectivity among routers in that network. Now network route finding problem is just finding a set of minimum cost path between nodes present in the corresponding graph representation which can be done easily by the ant algorithms.

### 3.3 General Characteristics of ACO algorithms for routing

The following set of core properties characterizes ACO instances for routing problems:

- 1) Providing traffic-adaptive and multipath routing.
- 2) Relying on both passive and active information monitoring and gathering.
- 3) Making use of stochastic components.
- 4) Not allowing local estimates to have global impact.
- 5) Setting up paths in a less selfish way than in pure shortest path schemes favoring load balancing.
- 6) Showing limited sensitivity to parameter settings [4].

### 3.4 Proposed Algorithm

This thesis proposes a QoS routing algorithm. The proposed approach has two phases namely path discovery phase and path maintenance phase. When a source node has to pass data to a destination node with QoS requirements it starts with the path discovery phase. Once the path is found, the data transfer will take place. While data transmission is going on, it is also required to maintain the path to the destination. This is very much desirable and required in mobile ad hoc networks and hence is done in the path maintenance phase.

#### 3.4.1 Path Discovery Phase

STEP 1: Let the source node  $S$  has data to send to a destination  $D$  with QoS requirements higher transmission rate, less delay and more bandwidth. A list of nodes that are progressively visited by the ant is called visited nodes list. This list forms the route  $R$  from the source node to destination node.

STEP 2: Initially choose the source node  $S$ . The visited nodes list will be initialized to source node ( $S$ ).

STEP3:  $S$  initiates a Path\_Request\_Ant to destination  $D$  through all its neighbors which are in 1-hop distance from  $S$ . The Path\_Request\_Ant contains source address, destination address, hop count and bandwidth.

STEP 4: After that the pheromone evaporation of all the 1-hop distance nodes will be calculated. Each node ( $i$ ) maintains a table called "PMtab" a table of Pheromones specifying the quantity of available pheromone on each link ( $V_i, V_j$ ). This quantity is initialized to constant  $C$ .

STEP 5: Then we calculate the pheromone evaporation of all the 2-hop distance nodes.

STEP 6: At last we calculate the path preference probability value of each path from source  $S$  with the help of pheromone evaporation of every node. A node  $j$  from a set of adjacent nodes ( $j, k, \dots, n$ ) of  $i$  is selected as MPR node such that it covers all the 2-hop distance nodes and its path preference probability is better than others.

STEP 7: If calculated path preference probability value is better than the requirements, the path is accepted and stored in memory.

STEP 8: When the Path\_Request\_Ant reaches the destination, it will be converted as Path\_Reply\_Ant and forwarded towards the original source. The Path\_Reply\_Ant will take the same path of the corresponding Path\_Request\_Ant but in reverse direction.

STEP 9: The path with higher path preference probability will be considered as the best path and data transmission can be started along that path.

#### 3.4.1 Path Maintenance Phase

When the data transmission is going on, the paths are reinforced positively making it more desirable for further selection. Also when session is going on, the load on the selected path may increase causing more delay and less available bandwidth; Nodes might have moved causing link failures. In such case, the path preference probability will automatically decrease and hence alternate routes can be used which are found during route discovery phase. The alternate routes are also periodically checked for their validity even though they are not currently used.

### 3.5 Quantitative Evaluation

Let  $G = (V, E)$  represents a mobile ad hoc network, where  $V$  denotes the set of network nodes and  $E$  denotes the set of bi-directional links. To find a route, our routing algorithm uses the pheromone accumulation ants laid on crossed links. This deposited quantity is the same on any link ( $V_i, V_j$ ) along the route  $R$ . It is noted  $H\tau(V_i, V_j)$  and it is a function of the global quality of the route  $R$ . It is expressed by the following equation:

$$\Delta\tau(V_i, V_j) = \frac{B(R)^{\beta_B} + T(R)^{\beta_T}}{D(R)^{\beta_D}}$$

where,

- $B(R) = \min(B(V_i, V_{i+1}), B(V_{i+1}, V_{i+2}), \dots, B(V_{k-1}, V_k))$ ; The smallest link bandwidth in route  $R$  while  $V_i$  is the source node and  $V_k$  the destination node.



- $T(R) = \min((T(V_i, V_{i+1}), T(V_{i+1}, V_{i+2}), \dots, T(V_{k-1}, V_k))$ ; The smallest link expiration time in route  $R$ .
- $D(R) = D(V_i, V_{i+1}) + D(V_{i+1}, V_{i+2}), \dots + D(V_{k-1}, V_k)$ ; The sum of delays on all links in route  $R$ .
- $\beta_B, \beta_T$  and  $\beta_D$  denote the link weight factors that show the relative significance of each QoS parameter during pheromone update on a route  $R$  [2]. Although the pheromone is deposited on the link, its quantity (equation 3) is only defined after finding the route. It permits to appreciate in the same way all links forming the route. Nevertheless, the local quality of a link is taken into account in its choice. This local quality represents the heuristic factor or the visibility of the ant. It is given by the following equation:

$$\eta_{ij} = \frac{B(V_i, V_j)^\alpha + T(V_i, V_j)^\alpha}{D(V_i, V_j)^\alpha}$$

where  $\alpha_B, \alpha_T$  and  $\alpha_D$  are parameters showing the relative importance of each QoS parameter during the link selection. The pheromone quantity on the link  $(V_i, V_j)$  is:

$$\tau(V_i, V_j) = \rho \cdot \tau(V_i, V_j) + \Delta\tau(V_i, V_j)$$

where  $\rho$  is the evaporation factor ( $0 < \rho < 1$ ). It is used to avoid the infinite increment of pheromone on the link that may lead to stagnation route. When an ant searches for a route, it chooses probabilistically one node among its neighbour's nodes that are not visited yet. The routing probability value between  $V_i$  and  $V_j$  is computed by the composition of the strength of pheromone values (equation 5) and the visibility values (equation 4).

$$P_{ij} = \frac{[\tau(V_i, V_j)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{k \in M} [\tau(V_i, V_k)]^\alpha \cdot [\eta_{ik}]^\beta}$$

where:

- $\tau(V_i, V_j)$  is the amount of pheromone on the link  $(V_i, V_j)$ .
- $\eta_{ij}$  is the visibility of the link  $(V_i, V_j)$ .
- $\alpha$  and  $\beta$  are two parameters that show the relative significance of the pheromone and the visibility during the process of QoS route discovery.
- $M$ : is the set of all possible neighbor nodes  $V_k$ , not visited yet by the ant.

### 3.6 The Proposed Algorithm (FLAR)

In this paper, our novel FLAR approach is presented. FLAR is constructed with the communication model observed in ant colonies and in fuzzy logic technique. In this section we will describe the fuzzy inference system (FIS) designed for FLAR, and then explain the FLAR Algorithm in detail.

In this section we describe our novel FLAR algorithm in detail. FLAR is constructed with the communication model observed in ant colonies, which is then enhanced by fuzzy logic technique. The sequence of FLAR algorithm is outlined as follows:

- 1) Each source node launches forward ants to destinations at regular time intervals.
- 2) The forward ants find a path to the destination randomly based on the current routing tables, but the data packets choose the path to the destination with the highest probability.
- 3) Each forward ant creates a stack, pushing in delay time and amount of buffer utilization for every traversed route (or link) to a node. The delay can be the sum of the time spent

waiting in queue and the transmission time for each visited node  $n$ .

- 4) When the destination is reached the backward ants inherit the stack.
- 5) The backward ant pops the stack entries, including delay time and utilization amount, and takes in the path in reverse to update to the routing tables of visited nodes. The total delay of a path is defined as the sum of all delays of the intermediate routes from the current node  $n$  to the destination node  $d$  via a neighboring node  $j$  Eq. (2).

$$D_{j,d}^n = \sum_{i=1}^s \text{delay}_i$$

where  $s$  is the total number of routes (or links) in the path traversed by forward ant. The utilization of each buffer on the path is calculated as in Eq. (3). Each node has an incoming packet buffer with a maximum capacity of  $Q$ . The sum of these utilization measures is taken and used to generate a weighting measure  $k_{ui}$  for each buffer  $i$  as in Eqs. (4) and (5). Finally, the estimated path utilization  $U_{n,j;d}$  from the current node  $n$  to the destination node  $d$  via the neighboring node  $j$ , is calculated by multiplying the number of packets in each buffer by its corresponding weight factor  $k_{ui}$ , which is shown in Eq. (6).

$$u_i = \frac{q_i}{Q}, \quad i = 1, \dots, s$$

$$B = \sum_{i=1}^s u_i, \quad \lambda_{u_i} = \frac{u_i}{B}, \quad i = 1, \dots, s$$

$$U_{j,d}^n = \sum_{i=1}^s \lambda_{u_i} \cdot u_i$$

where  $q_i$  is used as the queue buffer,  $Q$  is the maximum capacity of an incoming packet buffer for each node  $i$ , and  $s$  is the total number of nodes traversed by the forward ant. Using the calculated values pair of  $D_{n,j;d}$  and  $U_{n,j;d}$  as crisp inputs, we determine the congestion route for each eligible path via fuzzification (based on the membership functions shown in Figures 2-4), fuzzy inference (based on the rule shown in Table 1 and the mamdani implication), and defuzzification (based on the centers mean method). The Path congestion  $j;d$ , the amount of the congestion rate to go from a current node  $n$  to a destination node  $d$  via a neighboring node  $j$ , is expressed in Eq. (7).

$$\text{Path\_congestion}_{j,d}^n = \frac{\sum_{l=1}^M \bar{Y}^l (\mu_{A^l(d)} \cdot \mu_{A^l(j)})}{\sum_{l=1}^M (\mu_{A^l(d)} \cdot \mu_{A^l(j)})}$$

where the parameters are:

- $i$ : the node an ant is coming from
- $j$ : the node where an ant wants to move
- $M$ : the number of fuzzy rule bases used ( $M \approx 25$ )
- $\bar{Y}^l$ : the mean value of each membership function in fuzzy set
- $|\Lambda| \delta D \beta$ : the amount of membership functions for delay

$|\Delta|$  : the amount of membership functions for path utilization.

Then, the output of the fuzzy system (Path congestion<sub>j;d</sub>) is applied to the FLAR algorithm, which can be used as a criterion for updating routing tables for each visited node.

7. The new estimation of Path congestion<sub>j;d</sub> is computed as expressed in Eq. (8).

$$Path\_congestion_{j,d}^n(t) = (1 - \rho)Path\_congestion_{j,d}^n(t-1) + \rho Path\_congestion_{j,d}^n$$

where  $\rho$  is the learning factor that is set to 0.15 in this experiment. Finally, the routing table probabilities of each traversed node are updated by Eq. (9) on the basis of the Path congestion<sub>j,d</sub><sup>n</sup>

$$P_{j,d}^n(t) = \frac{\left[ \frac{1}{Path\_congestion_{j,d}^n(t)} \right]}{\sum_{l \in Neighborsof(n)} \left[ \frac{1}{Path\_congestion_{l,d}^n(t)} \right]}$$

## 4. Experimental Results

### 4.1 Simulation Settings

In this thesis to implement the proposed work Vb.net is used to simulate proposed technique. In the simulation, the channel capacity of mobile hosts is set to the same value: 2 Mbps. It use the distributed coordination function (DCF) of IEEE 802.11 for wireless LANs as the MAC layer protocol. It has the functionality to notify the network layer about link breakage.

In the simulation, the number of nodes is varied as 25, 30, 35, 40, 45 and 50. The mobile nodes move in a 1250 meter x 1250 meter square region for 50 seconds simulation time. It assumes each node moves independently with the same average speed. All nodes have the same transmission range of 250 meters. The speed is varied from 5 m/s to 25m/s. Random Way Point mobility model is used. The simulated traffic is Constant Bit Rate (CBR) and the number of connections in each scenario is 4.

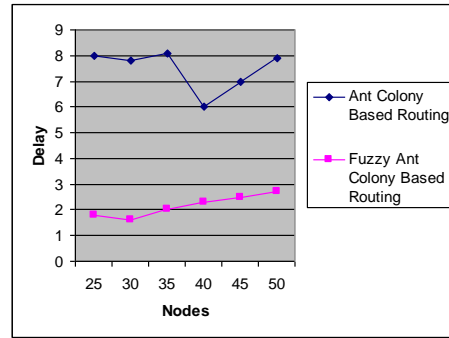
### 4.2 Performance Analysis

#### 4.2.1 Effect of varying number of nodes

In the experiment, the simulation was varied the number of nodes from 25 to 50.

Table 1 Performance comparison of proposed method with ant colony based routing based on delay

Nodes	Delay in (Seconds)	
	Ant Colony Based Routing	Fuzzy Ant Colony Based Routing
25	8	1.8
30	7.8	1.6
35	8.1	2
40	6	2.3
45	7	2.5
50	7.9	2.7



#### 4.2.2 Nodes Vs Delay

The table and chart 1 shows the results of average end-to-end delay for the increasing number of nodes. From the results, it can see that Ant-Fuzzy scheme has significantly lower delay than the ANTHOCNET protocol, since it has delay as one of the routing metrics. The path selected by the ant agents is delay aware and hence the end-to-end delay is less in Ant-Fuzzy

Table 2: Performance comparison of proposed routing algorithm with ant colony based routing Nodes Vs Packet Drop

Nodes	Packets	
	Ant Colony Based Routing	Fuzzy Ant Colony Based Routing
25	6000	250
30	8000	300
35	8500	400
40	9000	5000
45	9500	6000
50	15000	8500

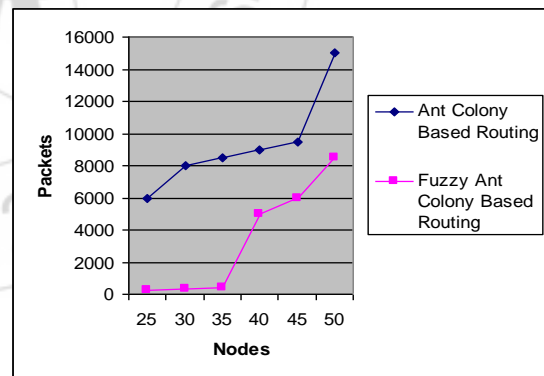
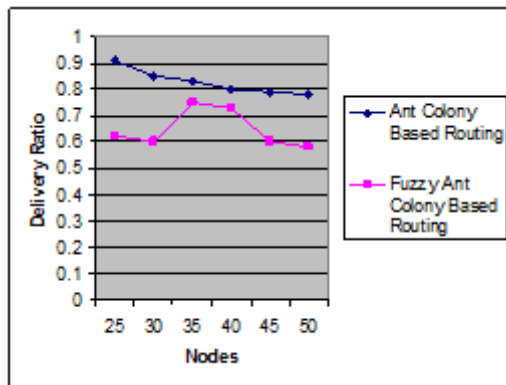


Chart 2: Nodes Vs Packet Drop

The table and chart 2 shows that the Ant-Fuzzy scheme has less packet drops than the ANTHOCNET protocol. This is because Ant-Fuzzy has the strong route maintenance scheme to mitigate the route failures. Moreover the selected path satisfies the QoS constraints delay and bandwidth and there by decreasing the number of packets dropped Since the packet drop is less, the delivery ratio is more in Ant-Fuzzy.

**Table 3:** Performance comparison of proposed routing algorithm with ant colony based routing average packet delivery ratio

Nodes	Delivery Ratio	
	Ant Colony Based Routing	Fuzzy Ant Colony Based Routing
25	0.91	0.62
30	0.85	0.6
35	0.83	0.75
40	0.8	0.73
45	0.79	0.6
50	0.78	0.58



**Chart 3:** Nodes Vs Delivery Ratio

Table and chart 3 show the results of average packet delivery ratio for the varying nodes scenario. Clearly our Ant-Fuzzy scheme achieves better delivery ratio than the ANTHOCNET scheme.

## 5. Conclusion

MANET is a collection of mobile nodes using as transmission medium radio waves, able to communicate with each other using multihops links without the need of any infrastructure. Any mobile node communicates directly using the radio waves with nodes in its transmission range, and uses intermediate nodes to communicate with other nodes which are out of its transmission range. This paper, described an ant based routing along with the fuzzy based route failure management technique. This technique helps in finding a path between the communication end points satisfying users QoS requirement without any occurrence of path failure. By simulation results, we have shown that Ant-Fuzzy has the strong route maintenance scheme and it attains good packet delivery ratio with less packet drop and delay than the ANTHOCNET scheme. There is tremendous potential for extending this technique and adapting other ACO principles to wireless networks

## References

[1] E. M. Royer and C. E. Perkins, "Multicast Operation of the Ad-hoc On-demand Distance Vector Routing Protocol", in Proc. ACM MOBICOM, pp. 207-218, Aug. 1999.  
 [2] C. E. Perkins and E. M. Royer, "Ad-hoc On-demand Distance Vector Routing", in Proc. IEEE WMCSA, pp. 90-100, Feb. 1999.

[3] C. W. Wu and Y. C. Tay, "AMRIS: A Multicast Protocol for Ad Hoc Networks", in Proc. IEEE MILCOM, Vol. 1, pp. 25-29, Nov. 1999.  
 [4] T. Ozaki, Jaime Bae Kim, and T. Suda, "Bandwidth-efficient Multicast Routing for Multihop, Ad-hoc Wireless Networks", in Proc. IEEE INFOCOM, Vol. 2, pp. 1182-1191, Apr. 2001.  
 [5] J. G. Jetcheva and D. B. Johnson, "Adaptive Demand-driven Multicast Routing in Multi-hop Wireless Ad Hoc Networks", in Proc. ACM MOBIHOC, pp. 33-44, Oct. 2001.  
 [6] L.-S. Ji and M. S. Corson, "Explicit Multicasting for Ad Hoc Networks", Mobile Networks and Applications, Vol. 8, No. 5, pp. 535-549, Oct. 2003.  
 [7] P. Sinha, R. Sivakumar, and V. Bharghavan, "MCDAR: Multicast Core-extraction Distributed Ad Hoc Routing", in Proc. IEEE WCNC, Vol. 3, pp. 1313-1317, Sep. 1999.  
 [8] P. Sinha, R. Sivakumar, and V. Bharghavan, "CEDAR: A Core Extraction Distributed Ad Hoc Routing Algorithm", IEEE Journal on Selected Areas in Communications, Vol. 17, No. 8, pp. 1454-1466, Aug. 1999.  
 [9] S. J. Lee, W. Su, and M. Gerla, "On-demand Multicast Routing Protocol in Multihop Wireless Mobile Networks", Mobile Networks and Applications, Vol. 7, No. 6, pp. 441-453, Dec. 2002.  
 [10] S. K. Das, B. S. Manoj, and C. S. R. Murthy, "A Dynamic Core Based Multicast Routing Protocol for Ad Hoc Wireless Networks", in Proc. ACM MOBIHOC, pp. 24-35, Jun. 2002.  
 [11] S.-H. Park and D.-Y. Park, "Adaptive Core Multicast Routing Protocol", Wireless Networks, Vol. 10, No. 1, pp. 53-60, Jan. 2004.  
 [12] C.-C. Shen and C.-P. Jaikao, "Ad Hoc Multicast Routing Algorithm with Swarm Intelligence", Mobile Networks and Applications, Vol. 10, No. 1-2, pp. 47-59, Feb. 2005.  
 [13] S.-J. Lee and C.-K. Kim, "Neighbor Supporting Ad Hoc Multicast Routing Protocol", in Proc. IEEE MOBIHOC, pp. 37-44, Aug. 2000.  
 [14] J. J. Garcia-Luna-Aceves and E. L. Madruga, "The Core-assisted Mesh Protocol", IEEE Journal on Selected Areas in Communications

## Author Profile

**Miss J. Jayaperiya** received the M.Sc computer Science degree in Sri Jayendra Saraswathi Maha Vidhalaya College of Arts and Science in 2013 and presently she is doing her M.Phil in Computer Science. She is presently working as teacher Sri Jayendra Saraswathi Vidhalaya CBSE School. Her Area of interest is networking, Mobile adhoc Networks.

**Mr. S. Saravana Kumar** received his M.Phil Degree in Computer Science from Periyar university and has 10 years of experiences in teaching industry and pursuing his Phd in Computer Science.. He is working as Assistant Professor in Sri Jayendra Saraswathi Maha Vidhalaya College of Arts and Science. He has published 4 papers in International Journals. His area of interest is Data mining.