

Computation of Pheromone Values in AntNet Algorithm

Anuj K. Gupta¹, Harsh Sadawarti² and Anil K. Verma³

^{1,2} Dept. of Computer Science & Engineering, RIMT Institute of Engineering & Technology, Punjab, India.
anujgupta@rime.ac.in, harshsadawarti@rimt.ac.in

³ Dept. of Computer Science & Engineering, Thapar University, Punjab, India.
akverma@thapar.edu

Abstract – In this paper, we discuss the basic routing technique of ants and study the change in pheromone values at each node. Also the optimal paths can then be computed based on the shortest cumulative pheromone count between source and destination nodes. AntNet is a distributed multi-agent system inspired by the stigmergy model of communication observed in ant colonies. The ants or control packets collect information about the network conditions and are used to update and maintain the routing tables. Ants based routing is gaining more popularity because of its adaptive and dynamic nature. A number of Swarm Intelligence based, more specially Ant Colony Optimization (ACO) based routing algorithms are proposed by researchers. A version of ant routing protocol called AntNet has been implemented to work within the network simulator ns-2. Routing tables and Pheromone tables have been computed for each node in the network. On the basis of these tables we have tried to compute the shortest and most optimal path between source node and destination node.

Index Terms – Swarm Intelligence, ants, ant colony optimization, AntNet algorithm

I. INTRODUCTION

Ad hoc networks consist of autonomous self-organized nodes. Nodes use a wireless medium for communication, thus two nodes can communicate directly if and only if they are within each other's transmission radius [13]. Two nodes normally communicate via other nodes in a multi-hop fashion. Swarm intelligence follows the behavior of cooperative ants in order to solve hard static and dynamic optimization problems. Ants leave pheromone trails at nodes or edges which increases the likelihood of other ants to follow these trails. Routing paths are then found dynamically on the fly, using this so called notion of stigmergy. Ant algorithm is a class of Swarm Intelligence algorithms. Swarm intelligence is a soft computing technique that has gained considerable attention in the research community over the last couple of years.

The basic principle of Swarm Intelligence is: the ability of ants to self-organize is based on four principles [9]. They are positive feedback, negative feedback, randomness and multiple interactions.

- **Positive feedback** – This is used to improve the good solution. When ants move from one node to another, the concentration of the pheromone along that path increases. This helps other ants to travel in this path.
- **Negative feedback** – This is mainly used to destroy bad solution. It can be done by decay of pheromone concentration with respect to time. The rate of decay is problem specific. Low decay rate encourages the bad solution not being destroyed for longer time and higher decay rate destroys good solution early [3].
- **Randomness** – Path to be taken by ant is completely random, hence there is possibility of generation of new solutions.
- **Multiple interactions** – The solution is found by interaction of many agents. In food searching process, one ant cannot find the food, as the pheromone would decay. Hence more ants can find food faster [8].

A. General Characteristics of Swarm Intelligence:

- SI provide network adaptive feature and generates multiple path for routing. SI algorithms are capable of adapting for change in network topology and traffic while giving equivalent performance.
- It relies on both passive and active information for gathering and monitoring. They collect non local information about the characteristics of solution set, like – all possible paths.
- It makes use of stochastic components. It uses stochastic component like pheromone table for user agents. User agents are autonomous and communicate each other through stigmergy.
- It sets path favoring load balancing rather than pure shortest path. The algorithm also supports for multiple paths, so that load balancing can be achieved.

B. Advantages of Swarm Intelligence:

- **Multipath routing** - Possible to generate multiple paths between pairs of nodes.
- **Fast route recovery** - If optimal path fails, then packets can easily be sent to other neighbor's by computing next hop probability, i.e., choosing second best path.
- **Distributed and fault tolerance** – SI algorithm are inherently distributed. There is no centralized control

mechanism, so if any node or link fails, there is no heavy loss.

- *Scalability and adaptation*– Population of ants may change based on the size of network. The agents may die or reproduce, with little effect on performance.
- *Speed* – Change in the network can be adapted very fast.

Swarm is considered as biological insects like ants, bees, wasps, fish etc. In this paper we have considered ant for our study [1].

C. Ants possess the following characteristics

- *Scalability*: The ants can change their group size by local and distributed agent interactions. This is an important characteristic by which the group is scaled to the desired level.
- *Fault tolerance*: Each ant follows simple rule. They do not rely on a centralized control mechanism, graceful, scalable degradation.
- *Adaptation*: Ants always search for new path by roaming around their nest. Once they find the food their nest members follow the shortest path. While nest members follow shortest path, some of the members of the colony always search for other shortest path. To accomplish this they change, die or reproduce for the colony.
- *Speed*: In order to make other ants to know the food source, they move faster to their nest. Other ants find more pheromone on the path and follow the path to the food source. Thus changes are propagated very fast to communicate to other nest mates in order to follow the food source.
- *Modularity*: Ants follow simple rule of following the path which has higher level of pheromone concentration. They do not interact directly and act independently to accomplish the task.
- *Autonomy*: No centralized control and hence so no supervisor is needed. They work for the colony and always strive to search food source around their colony.
- *Parallelism*: Ants work independently and the task of searching food source is carried out by each ant in parallelism. It is parallelism due to which they change their path, if a new food source is found near their colony.

These characteristics of biological insects such as ants resemble the characteristics of Mobile Ad Hoc Networks. This helps us to apply the food searching characteristics of ants for routing packets in Mobile Ad Hoc Networks.

Paper outline

This section has described briefly about the ad hoc networks, swarm intelligence, its characteristics and advantages. Also characteristics of biological insects i.e. ants have been presented. In section 2 a detailed review of various types of ant based algorithms has been discussed with a greater stress on AntNet algorithm. Later a comparison among all these has been carried out. Section 3 explains in detail the working of AntNet

algorithm, how pheromone values are computed and tables are generated. Section 4 describes the ACO meta-heuristic and section 5 shows the experimental setup. Section 6 computes the results and finally section 7 concludes the paper.

II. REVIEW OF ANT BASED ALGORITHMS

Ant Based Control is the first ACO [7] application to dynamic problems proposed by Schoonerwoerd [5]. ABC is another stigmergy-based ant algorithm designed for packet-switched telephone networks and uses distance vector routing. It shares many similarities with AntNet, but also incorporates certain differences [15]. ABC replaces routing tables with pheromone tables. Another difference is that ABC only uses a single class of ants (i.e. forward ants), which are initiated at regular time intervals from every source to a randomly chosen destination. Ants move from node to node selecting next node to move according to the probabilities in pheromone tables for their destination node. Arriving at a node, ants update the pheromone tables' entries for corresponding node. The ants die after reaching their destination. Here the probability increase is represented by ΔP as shown below in the probability update equation (1),

$$P_{nd} = \frac{P_{nd} + \Delta P}{1 + \Delta P} \quad (1)$$

Schoonerwoerd et al [4] introduced two new aspects – age and delay. They set all pheromone levels to an initial value and intend to primarily encourage the ants to find short paths, and to secondly prevent the agents from visiting heavily congested nodes. The first goal is accomplished by reducing the ΔP value, which reinforces the routing table, gradually over time. A relative pheromone update scheme was implemented in which the amount of pheromone updated by each ant is inversely proportional to the age of ant. The second objective is obtained by adding an artificial delay to the ants when passing congested nodes. Though ABC is a simple algorithm designed for packet-switched networks, it performs better than other ACO based algorithms.

A number of routing algorithms for routing in Mobile Ad Hoc Networks have been proposed. All the existing ant based routing algorithms are enlisted in the Table 1 as per their existence.

TABLE 1. YEAR-WISE LIST OF PROPOSED ANT BASED ROUTING ALGORITHMS FOR MANET

Algorithm	Year	Proposed by	Type	Path
ABC	1997	Ruud Schoonderwood	Proactive	Single
AntNet	1997	Gianni Di Caro	Proactive	Single
ARA	2002	Mesut Gunes, Udo Sorges and I.Bouazizi	Reactive	Multipath
MABR	2003	Heissen and Braun	Proactive	Single
PERA	2003	John S Baras, Harsh Mehta	Proactive	Single
TERMITE	2003	Martin Roth and Stephen Wicker	Proactive	Multipath

AntHocNet	2004	Gianni Di Caro, Frederick Ducatellet LM Gambardella	Hybrid	Single
POSANT	2008	Shabab Kamali and Jaroslav Optarny	Reactive	Multipath
HOPNET	2008	Jianping Wanga, Eseosa Osagiea, Parimala Thulasiramam and Ruppa K Thulasiramam	Hybrid	Multipath
PACONET	2008	Eseosa Osagie, Parimala Thulasiramam and Rupple Thulasiramam	Reactive	Single
PAR	2009	S P Prasad, Y P Singh, C S Rai	Hybrid	Multipath

A. ARA

Ant-colony based Routing Algorithm was proposed by Gunes et al [4]. Route discovery is done by flooding a forward ant to the destination, similar to Route Request in AODV. Duplicate packets are identified by the use of a sequence number and are deleted by system. When a route is found to the destination, a backward ant is created similar to Route Reply in AODV. The backward ant follows the path with the shortest trip time detected by the forward ant. The amount of pheromone deposited by both ants is a function of route length associated with pheromone. The route maintenance phase is responsible for updating the routing information during the communication. The algorithm updates both the forward and the backward path with an equal amount of pheromone (presumed as 0.1). ARA also allows for the evaporation of pheromones. Every second the amount of pheromone $\Delta\tau$ is decreased by a multiplicative factor f as shown below in equation (2):

$$\Delta\tau = (1 - f) \cdot \Delta\tau, f \in [0,1] \quad (2)$$

ARA achieves loop free paths using sequence numbers. If a node receives a duplicate packet, it sets the DUPLICATE_ERROR flag and returns the packet to the previous node, and removes the link. According to Gunes et al [5] ARA performs comparatively better in terms of overhead ratio and delivery rate.

B. AntHocNet

DiCaro and Gambardella et al [2] proposed a hybrid multipath routing algorithm called AntHocNet that combines both proactive and reactive components with ACO. Routes are discovered reactively, where reactive forward ants (FA) are sent by the source node to find multiple paths towards the destination node. Unlike AODV, AntHocNet compares both travel time and hop count of subsequent forward ant and rebroadcasts only if both the criteria are within a certain factor of the best forward ant. The FA gathers route information while moving towards destination node. After reaching there it becomes backward ant (BA) which traces its path back to

source node and creates a route with the help of intermediate visited nodes. Each node m stores a routing table T^m which contains a measure T_{nd}^m of the desirability of using node n as the next hop to destination d . The amount of pheromone $\Delta\tau$ deposited by BA is measured as the average of the inverse of cost (in terms of delay and number of hops) of travelling to the node d through node n . For each destination-neighbor pair T^m stores a probability value P_{nd} calculated using equation (3):

$$P_{nd} = \frac{\tau_{nd}^2}{\sum_{m \in N_d} \tau_{md}^2} \quad (3)$$

Once the route is created, the source node sends one proactive forward ant to the destination node on every n^{th} data packet. One important thing to notice is that the proactive forward ant can discover new routes in the vicinity (within the boundary) of existing route only. The main purpose of using proactive forward ants is to search for any route improvements or variations and not to search for entirely new route. Like AODV [11], AntHocNet also uses HELLO messages to improve local connectivity of nodes. In case of link failure the node updates its routing table and send a notification to its neighbors. But Di Caro et al did not mention about the status of data packet in case of link failure, whether the packet is to be dropped or backtracked to any upstream node in the network. However, AntHocNet gives a superior delivery ratio and better packet delay than AODV.

C. Probabilistic Emergent Routing Algorithm

PERA works reactively with ants being broadcasted towards any random destination [12]. As multiple paths are set up, only the one with highest pheromone value is used by data and the other paths are available for backup. The route discovery and maintenance is done by flooding the network with ants. Both forward and backward ants are used to fill the routing tables with probabilities. These probabilities reflect the likelihood that a neighbor will forward a packet to the given destination. Multiple paths between source and destination are created. The neighbors are discovered using HELLO messages, but entries are only inserted in the routing table after receiving a backward ant from the destination node. Each neighbor receives a probability value for destination. This value is increased as a backward ant comes from that node, establishing a path towards destination. The algorithm uses sequence numbers to avoid loops and duplicate packets. Forward ant with a lower sequence number is dropped, similar to AODV Route Request packets, but discovers a set of routes instead of one. Data packets can be routed according to the highest probability in the routing table for the next node.

D. AntNet

AntNet is an ACO algorithm proposed by Gianni Di Caro and Marco Dorigo [14] for data communication networks. AntNet is one of the approaches to adaptive

learning of routing tables in wide area best-effort datagram networks. In this algorithm two types of ants are generated viz. Forward ant and Backward ants. While traveling from a source to a destination the forward ants store, in their memory, the paths and of the traffic conditions they encounter. After reaching the destination the forward ant transfers its memory to the backward ant and dies. The backward ant retraces the path traversed by the forward ant and updates the routing tables in the path. AntNet is designed in such way that the forward ants carry the information about the status of the links it traverses. This status information can be captured and can be used to find the best path. AntNet is one of the dynamic routing algorithms for learning new routes. Each node in the network consists of mainly two data structures viz., routing table and neighbor list.

Routing table: Routing table at each node stores the list of reachable nodes and their pheromone value. It is represented as structure consisting of following fields:

- *destination_id* – This represents the address of the destination node
- *next_id* – This represents the address of the adjacent node used to reach destination node.
- *pheromone* – This represents the value used by the node to calculate the probability of each adjacent node to be the next hop in order to reach the destination.

Neighbor list: Neighbor list is used to store the information of all the neighboring nodes.

AntNet algorithm was basically proposed for fixed wired networks, which derives features from parallel replicated Monte Carlo systems, previous work on artificial ant colonies techniques and telephone network routing. Dorigo et al [3] used Ant Net to describe the application of ACO to dynamic routing in packet-switched networks. It is based on source routing mechanism. The basic idea in AntNet is to use two network exploration agents – forward ants (FA) and backward ants (BA), which collect information about the delay, congestion status and path in the network. Each node n generates a FA at regular time intervals to a randomly selected destination d . FA uses the current routing tables to find a path to node d and records the route taken. On reaching at the destination, FA creates another ant called BA moves back to the source node n reverse the path of FA. The BAs get their information from the FAs and use it to achieve routing updates at the nodes. After creating a BA, the FA dies. Each node stores a routing table T_k organized as in distance-vector algorithms storing probability values. For each destination-neighbor pair (d, n) , T_k stores a probability value P_{nd} which helps in choosing n as next-hop node to destination d , such that

$$\sum_{n \in N_k} P_{nd} = 1, d \in [1, N], N_k = \{neighbors(k)\} \quad (4)$$

According to Dorigo the AntNet algorithm provides better performance in terms of average delay. The last

step is to update the routing table when a BA arrives. The action involves increase in corresponding routing probability of FA and decrease in probabilities of all other neighboring nodes. The values are increased and decreased such that the sum of all probabilities will remain 1. Table 2 gives the comparison of different ant based routing algorithms based on various parameters [15].

TABLE 2. COMPARISON OF DIFFERENT ANT BASED ALGORITHMS

Parameters	ABC	AntNet	AntHocNet	ARA
Ants sending	Periodic	Periodic	Periodic	Periodic
Types of ants	FA	FA, BA	FA, BA (Reactive & Proactive)	FA, BA
Information collected by Forward Ants	Launching time	Time elapsed between ant launch and arrival at each node	While travelling to destination, reactively	Link costs
Ant structure	SIP, DIP, Ageof Ant	SIP, DIP Seq num, memory	SIP, DIP, Next Hop IP add, Stack, Hop count, Seq num	SIP, DIP Seq num, Hop count
Parameters considered in choosing next hop	Ph	Ph	Ph	Ph
Amount of deposited pheromone	Depends on ant age	Constant amount	Function of cost of travelling from src to destination	Non-decreasing function of links' costs
Routing table structure	Dest add, Next hop, Ph value	Dest add, Each neighbour, Phvalue	Goodness of next hop, Destination address, Next hop	Dest add, Next hop, Ph value
Traffic Statistics Structure	Not used	Used	Used	Not used

Ph – Pheromone, SIP – source IP address, DIP – destination IP address, FA – forward ant, BA – backward ant

We have represented the wired network as a graph $G(N, L)$ consisting of N nodes and L links. All the links in the network are considered bidirectional. Each node is considered both as a host and router. Every node in the network maintains a buffer composed of a single queue in which pheromone values of neighboring nodes are maintained. This buffer is then consulted to form a routing table. All the packets within the network can be divided into two different classes: Data packets and Mobile agents. In ant-routing, data packets do not maintain any routing information but use the information stored at routing tables for travelling from the source to the destination node. Forward ants and backward ants act as Mobile agents that are used to update the routing tables and distribute information about the traffic load in the network. When a node receives a packet from any of its neighbors, the packet is first stored in its buffer. A routing table T_r , organized as a matrix with probabilistic pheromone entries is formed at each node. Each row in the routing table corresponds to one source, next node

and pheromone value at that particular node while traversing the shortest path to the destination from current node. T_r stores a probability value P_{nd} expressing the probability of choosing n as the next node while moving towards the destination node [17].

III. THE ANTNET ALGORITHM

The AntNet algorithm [1] can be described as follows:

At regular intervals, from every network node, a forward ant is launched with a randomly selected destination node. While travelling towards their destination nodes, the forward ants store their paths and the traffic conditions. At each node, each forward ant chooses the next node as follows:

- If all the neighboring nodes have not been visited, then the next neighbor is chosen among the nodes that have not been visited.
- If all the neighboring nodes have been visited previously, then the next node is chosen uniformly among all the neighbors. The forward ant is forced to return to a previously visited node.
- With a small probability, the next node may be chosen uniformly among all the neighboring nodes.
- If a cycle is detected, that is, if the ant is forced to return to an already visited node.
- When the destination node is reached, the forward ant generates a backward ant. The backward ant takes the same path as the corresponding forward ant, but in the opposite direction. Arriving at a node coming from a neighbor node, the backward ant updates the routing table T_r , for all the entries corresponding to the destination node.

The AntNet router contains a special routing table i.e. pheromone table, where each destination node is associated to all the neighbor nodes and each node has a certain probability. Every router sends a forward ant with random destination over the network, which collects information about the state of network [18]. The next hop is determined on the basis of probabilities in the routing table. When the forward ant reaches its final destination, the path based on Dijkstra algorithm is computed and the forward ant is transformed into a backward ant. The backward ants use the information collected by forward ants to update the routing table probabilities in each router along their path. When the backward ant reaches the first router, it dies [19].

A. Pheromone Computation

The pheromone values at any edge (i, j) are updated by all the ants that have completed the path length as follows:

$$\tau_{ij} = (1 - \lambda) \cdot \tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k \quad (5)$$

where m is the number of ants that have completed the path.

$\lambda \in (0,1)$ is the evaporation constant that determines the evaporation rate of the pheromone [15].

$\Delta\tau_{ij}^k$ is the quantity of pheromone deposited by ant k on edge (i, j).

B. AntNet Limitations

Although AntNet possess some benefits in controlling congestion and load balancing, but still there exists certain limitations such as high probability values of Backward Ants than Forward Ants.

C. AntNet Vs Dijkstra's Algorithm

There is a major difference between AntNet and Dijkstra's shortest path algorithm. In Dijkstra's shortest path algorithm, the cost of shortest path is computed as the sum of individual link weights on the path. However in AntNet, the cost of the shortest path used for updating the routing tables is assumed to be the sum of link weights along the path and the transmission and queuing delays. The delay in the queues is included in the AntNet algorithm to account for the traffic conditions in the network. Thus, there is no static shortest path in the AntNet algorithm.

IV. ANT COLONY OPTIMIZATION META-HEURISTIC

Ant colony optimization (ACO) has been formalized into a meta-heuristic for combinatorial optimization problems by Dorigo et al. The model proposed is used to define the pheromone model of ACO in which each component is associated with a pheromone value. All the ants build a solution by traversing the fully connected construction graph $G = \{V, E\}$, where V is set of all nodes in the network and E is set of all edges or links. The ants move from node to node along the edges of the graph and deposit a certain amount of pheromone on the components i.e. node or edge.

Step1: Set parameters; initialize pheromone values

Step2: Do while (destination node not reached):

Construct Ant Solutions,

Traverse next node (if not destination)

Step 3: Update Pheromones

Step 4: End

A number of solutions are generated each step by the ants; which are then improved and finally the pheromone is updated. Each node maintains a routing table T for choosing the optimal path to transfer data. Table 3 shows the probability of choosing next node to destination, where ant lays in the node i (1 to n), d is number of node that node i can chose as destination and n is the total number of neighbors.

TABLE 3. ROUTING TABLE

$P_{1,1}$	$P_{1,2}$...	$P_{1,d}$
$P_{2,1}$	$P_{2,2}$...	$P_{2,d}$
...
$P_{n,1}$	$P_{n,2}$...	$P_{n,d}$

V. EXPERIMENTAL SETUP

Our simulation is based on the AntNet implementation [16] on ns2 version 2.33 [6, 10]. We have implemented new agent called AntNet and new packet type Ant on a network of 12 nodes having arbitrary topology as shown figure 1. The output of the simulation gives a Routing table T_r , on the basis of which shortest paths can be calculated between any two nodes using pheromone values at each of their neighbor nodes. The network model used in our simulation is composed by mobile nodes and links that are considered bidirectional and wireless. Each node is considered a communication endpoint i.e. host and a forwarding unit i.e. router. The routing tables are generated at each node. At each node the pheromone value is computed using the equation mentioned in section 3.1 above. The pheromone value lies between the ranges 0 to 1. The more the value the more is the possibility of shortest path from source node to destination node. 1 represents direct link and 0 represents longest or no link. In case of leaf node i.e. the single link node, the pheromone value is 1 because there is no other link existing for that node to connect to the other nodes, no intermediate node exist between them.

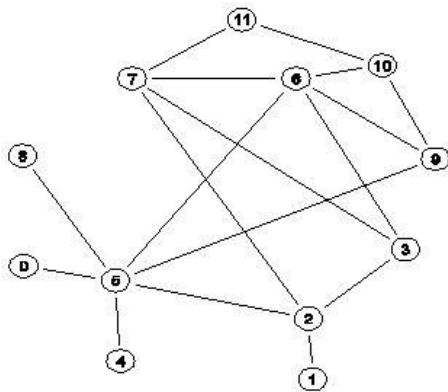


Figure 1: An arbitrary Network with 12 nodes

As an example the routing table for node 3 is shown in table 4 below. It is having 3 columns, first one represents the destination node, second represents the immediate next node and last column represents the pheromone value for that link.

TABLE 4. ROUTING TABLE FOR NODE 3

Destination node	Next node	Pheromone value
0	7	0.437617
0	6	0.416205
0	2	0.146178
1	7	0.280460
1	6	0.015005
1	2	0.704535
2	7	0.291052
2	6	0.000703
2	2	0.708245
4	7	0.312407
4	6	0.602697
4	2	0.084896
5	7	0.390440
5	6	0.485925
5	2	0.123636

6	7	0.351311
6	6	0.627780
6	2	0.020908
7	7	0.950641
7	6	0.011946
7	2	0.037412
8	7	0.402235
8	6	0.206698
8	2	0.391067
9	7	0.578002
9	6	0.363279
9	2	0.058720
10	7	0.700919
10	6	0.271622
10	2	0.027459
11	7	0.922854
11	6	0.044369
11	2	0.032778

VI. RESULTS

In table 5 the pheromone values for each next node (from every node acting source) to the destination node 11 are shown. Initially route discovery routine is called by the node 0. This results in generation of following routing table. Let us suppose we want to compute the shortest distance from node 0 to node 11. Once the data discovery is done, the data packets from source node 0 will be sent to destination node 11 along the path 0-5-9-10-11 or 0-5-6-10-11, as the probability of these two paths is same. If the node 6 fails, then source 0 can redirect the traffic through node 9 will no degradation in throughput. But if the node 5 fails then no data can be sent to any other node from node 0. Similarly paths are discovered for all other nodes for destination node 11 and best optimal path then can be chosen taking the pheromone value into consideration. More the pheromone value more optimal is the path.

TABLE 5. PHEROMONE VALUES FOR DESTINATION 11 FROM EACH NODE

Source	Next node	Pheromone value
0	5	1.000000
1	2	1.000000
2	7	0.579234
	5	0.192586
	3	0.228054
	1	0.000126
3	7	0.661520
	6	0.102287
	2	0.236193
4	5	1.000000
5	9	0.880370
	8	0.000045
	6	0.119165
	4	0.000045
	2	0.000329
6	0	0.000045
	10	0.418304
	9	0.084475
	7	0.101621
	5	0.000282
7	3	0.395319
	11	0.922854

7	11	1.000000
	6	0.000000
	3	0.000000
8	2	0.000000
	5	1.000000
9	10	0.808539
	6	0.191442
	5	0.000019
10	11	1.000000
	9	0.000000
	6	0.000000

VII. CONCLUSIONS

We observed that the experimental data provided by this paper will help researchers in simulating Ant based routing protocols. This paper presents an enhanced algorithm for generating all possible paths between source node and destination node in mobile ad hoc network using swarm intelligence. Route maintenance is done periodically to retain optimal path. This is done through data packets. This paper describes the method for handling loss of ants and avoids retransmission of lost packets by reserving resources at nodes, which in turn enhances the performance. We have studied that we can use AntNet algorithm to build optimal routes with the help of the forward ants. After generating the routing table we can compute the best route with the help of the Dijkstra's algorithm. The algorithm discussed in the paper uses hop count as metric for optimization. Further, it can be extended for other matrices for optimization like load and multimedia data. Number of ants in the network is to be controlled as it may not cause congestion. It can be concluded that Swarm Intelligence based approach offers to be a powerful means to solve routing problems in Mobile Ad Hoc Networks. As a part of future work, AntNet shall be performed and evaluated against other existing ant based routing protocols for mobile ad hoc networks. And their performance shall be enhanced.

ACKNOWLEDGEMENTS

The authors would like to thank all the anonymous reviewers and experts for their valuable and critical comments in bringing out more useful information and enhancements to the present study.

REFERENCES

- [1] Di Caro, G., Dorigo, M, "Antnet: Distributed stigmergetic control communications networks. Journal of Artificial Intelligence Research pp 317-365, 1998.
- [2] Dorigo M & Gambardella L, "Ant colony system: A Cooperative learning approach to the traveling salesman problem", IEEE Transaction on Evolutionary Computation, Vol. 1, N1, pp53-66
- [3] M. Heissenbittel, T. Braun, "Ants-Based Routing in Large Scale Mobile Ad-Hoc Networks".
- [4] Mesut Gunes, Udo Sorges, Imed Bouazizi, "ARA-The Ant-Colony Based Routing Algorithm for MANETs" International workshop on Ad Hoc Networking (WAHN 2002) vancouver, Canada, August 18-21, 2002.
- [5] Schoonderwoerd R , Holland O Bruten J, "Ant like agents for load balancing in Telecommunication Networks", Hewlett-Packard Laboratories, Bristol-England, 1997
- [6] The NS Manual , VINT Project (Editors: Kevin Fall and Kannan Vardhan)
- [7] M. Dorigo and T. Stützle. The Ant Colony Optimization.
- [8] Metaheuristic: Algorithms, Applications and Advances. In F. Glover and G. Kochenberger, (Eds.), Handbook of Metaheuristics, Kluwer Academic Publishers, pages 251-285, 2003.
- [9] Singh Rajeshwar, D K Singh, Lalan Kumar (2010). "Ants Pheromone for Quality of Service Provisioning In Mobile Adhoc Networks". Int. J.Elect. Eng. Res., 2(1): 101-109.
- [10] ns-2, Network simulator, <http://www.isi.edu/nsnam/ns>.
- [11] C.E Perkins, E.M. Royer, and S. Das, —Ad hoc On-demand Distance Vector (AODV), RFC 3561, July 2003.
- [12] J. Baras and H. Mehta 2003. A Probabilistic Emergent Routing Algorithm for Mobile Ad hoc Networks (PERA).
- [13] S. Murthy, C. Siva Ram and B.S. Manoj, —Ad Hoc Wireless Networks: Architectures and Protocols, Prentice Hall.
- [14] G. di Caro and M. Dorigo, AntNet: distributed stigmergetic control for communications network, Journal of Artificial Intelligence Research (JAIR), Vol. 9, Pag. 317-365, 1998.
- [15] Anuj K. Gupta, Anil K. Verma, Harsh Sadawarti, "Analysis of various Swarm-based & Ant-based Algorithms", Proceedings of International Conference on Advances in Computing and Artificial Intelligence (ACAI 2011) July 2011, pp – 39-43. Available online in ACM DL: <http://dl.acm.org/citation.cfm?id=2007052>.
- [16] V. Laxmi, Lavina Jain, M.S. Gaur, "International Conference on Wireless Communication and Sensor Networks, WSCN 2006".
- [17] S.S. Dhillon, P. Van Mieghem, "Performance analysis of the AntNet algorithm", Elsevier Computer Networks 51 (2007) 2104-2125.
- [18] Firat Tekiner, Z. Ghassemlooy , S. Alkhayatt, "Investigation of AntNet routing algorithm by employing multiple ant colonies for packet switched networks to overcome the stagnation problem", Optical Communications Research Group, 2004.
- [19] Anuj K. Gupta, Harsh Sadawarti, Anil K. Verma, "MANET routing protocols based on Ant Colony Optimization", *International Journal of Modeling & Optimization (IJMO)*, ISSN: 2010-3697, vol. 2, no. 1, pp. 42-49, February 2012.



Anuj K. Gupta is a research fellow in Punjab Technical University, Punjab, India. He is pursuing PhD degree in Computer Science & Engg. His research area is Mobile ad hoc networks. He has a teaching experience of more than 10 years. He is currently working as faculty at RIMT, Punjab, India.

Email: anujgupta@rimt.ac.in.

obile ad hoc networks. He has a teaching e



Dr. Harsh Sadawarti is Professor in Dept. of Computer Science & Engg. at RIMT, Punjab, India. He has a vast teaching and research experience of more than 20 years in the field of computer science. His areas of research are ad hoc networks, parallel computing & Distributed systems.

Email: harshsadawarti@rimt.ac.in.



Dr. Anil K. Verma is faculty in Computer Sci. & Engg. Dept. at Thapar University, Patiala, Punjab, India. He has a vast teaching & research experience of more than 20 years. His areas of research are mobile ad hoc networks, wireless sensor networks & Network Security.

Email: akverma@thapar.edu.