Adhoc Network Routing Optimization and Performance Analysis of ACO Based Routing Protocol

Anubhuti Verma

***Abstract*— Ant Colony Optimization is based on the capability of real ant colonies of finding the shortest path from nest to food source by depositing and following the trail of pheromone on the path. Pheromone is a chemical deposited on ground by the ants while walking and it affect their moving decisions based on the intensity. The main components for ACO optimization are: set of software agent (artificial ant), use of memory and strategies of collective and distributed learning. Many ACO algorithms were designed and implemented for TSP which was very successful in finding optimal tour.**
In MANET routing is a challenging issue because of dynamically changing network topology which needs to be addressed. In this paper we analyze an ACO based routing algorithm inspired by the foraging behavior of ants, to route packets through shorter and feasible routes. Each ant while moving towards destination collects information about the time length, congestion status and address of each visited node of the followed path. During backward travel, local network traffic model and routing table is modified by ant based on the goodness of the followed path. The protocol is capable of discovering path for dynamic changing topology and varying traffic over the network and gives high redundancy and fault tolerance. Analysis shows that the proposed ACO based routing protocol gives performance improvement over other routing protocol by continuously checking for better paths in the network with less overhead.

***Index Terms*— ACO, MANET, Routing Protocol**

1. INTRODUCTION

A mobile ad hoc network (MANET) is an infrastructure less networks consisting of mobile nodes, with constantly changing topologies, the nodes are mobile and they communicates through a wireless medium. Due to infrastructure less and dynamic nature of such networks, there is requirement of new set of networking strategies which is to be implemented for efficient end-to-end communication. This, along with the diverse application of these networks in many different scenarios such as battlefield and disaster recovery, has seen Mobile Ad Hoc Networks (MANETs) being researched by many different organizations and academia. MANETs employ the traditional TCP/IP structure to provide end-to-end communication between the nodes. However, due to their mobility and the limited resource in wireless networks, each layer in the TCP/IP model requires redefinition or modifications to function efficiently in

MANETs. One interesting research area in MANET is routing. In MANETs the transmission range of nodes is limited so the nodes in MANETs act as both hosts and routers. The routing protocol for MANETs needs to be flexible enough to adapt to the continuously changing network topologies, and to support bandwidth management and energy management as the nodes are power constrained.
Recently, nature's self-organizing systems (Swarm Intelligence) [3] such as ant colony optimization algorithms [1] are proved to be efficient in developing routing algorithm for MANET. Ant Colony Optimization (ACO) is based on the capability of real ant colonies of finding the shortest path from nest to food source [1]. The ants coordinate their activities via stigmergy (a form of indirect communication mediated by modifications of the environment). Ant uses pheromone to communicate with other ants. Pheromone is a chemical deposited on ground by the ants while walking and it affect their moving decisions based on the intensity. In this paper ACO based network routing algorithm is analyzed inspired by the foraging behavior of the ants, to route packets through shorter and feasible routes.

The remainder of this paper is organized as follows. In section 2 we describe related work. The algorithm is described in section 3. In section 4, performance analysis of the algorithm is done. Finally a conclusion is given in section 5.

2. RELATED WORK

In this section we describe related literature. In 2.1 a short introduction to MANET routing algorithms is given, and
in 2.2 we give an overview of Ant Colony Optimization. Then in 2.3 we describe the basic elements of ANT based routing.

2.1 ROUTING IN MANET

In recent years several protocols have been proposed to address the routing problem in MANETs. These protocols can be characterized as proactive, reactive or hybrid based on their design approach. Proactive routing protocol maintains global topology information in the form of tables at every node E.g. DSDV [6]. In reactive protocol node only gather routing information E.g. AODV [5]. Reactive algorithms are in general more scalable since they are never prepared for disrupted events. Hybrid protocols make use of both proactive and reactive protocol to achieve better performance E.g. ZRP [7].

2.2 OVERVIEW OF ANT COLONY OPTIMIZATION

Ant Colony Optimization (ACO) takes inspiration from the behavior of the real ant colony. These ants deposit pheromone on the ground in order to mark some favorable path to reach the food source that should be followed by other member of the colony. Ant colony optimization exploits similar mechanism for solving optimization problems. The algorithm can find the optimum solution by generating artificial ants. As the real ants search their environment for food, the artificial ants search the solution space. The pheromone trail-laying and trail-following behavior of some ant species is investigated by several experiments. One brilliant experiment was designed and run by Goss [4].

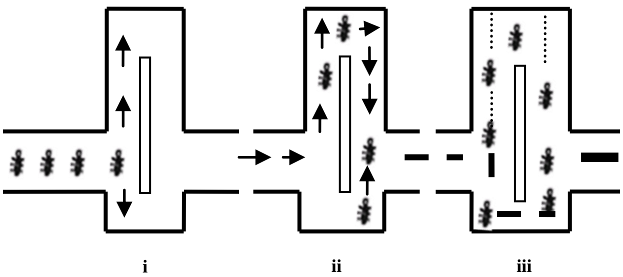

Figure 1: (i) Initially ants choose path randomly (ii) Ant choosing shorter path will reach first (iii) Pheromone deposited more quickly on shorter path, number of ants keep on increasing on sorter path.

These studies have given insight into the foraging behavior of ants showing that they are always able to determine, over time, the shortest path from their nest to a food source and also adapt easily adapt to path disruptions that may occur. This phenomenon can be explained using Figures (i) to (iii). While moving ants deposit pheromone on their path which evaporates over time. The ants are attracted towards a higher concentration of pheromone which enables them to choose shortest paths since these would retain a higher concentration each time than longer Ants choose whether to turn left or right with equal probability. The ant that choose the shorter path by chance will acquire a path between the nest and food that is faster and will reach the destination first than those taking the alternative path as in Figure (ii). This path will have higher pheromone concentration and number of ant will keep increasing as shown in Figure (iii).

2.3 ANT BASED ROUTING

The basic idea behind ACO algorithms for routing [8] is the acquisition of routing information through sampling of paths using agents, which are called ants. An ant going from source node s to destination node d collects information about the quality of the path (e.g. end-to-end delay, number of hops, etc.), and uses this on its way back from d to s to update the routing information at the intermediate nodes.

The routing tables contain for each destination a vector of real-valued entries, one for each known neighbor node. These entries are a measure of the goodness of going over that neighbor on the way to the destination. They are termed pheromone variables, and are continually updated according to path quality values calculated by the ants. The ant searches for a number of paths at each node each with an estimated measure of quality. In turn, the ants use the routing tables at each node to stochastically choose a next hop, giving higher probability to those links which are associated with higher pheromone values.
This process of ant agent in routing is very similar to the pheromone lying and following behavior of real ant colonies. The pheromone information is used for routing data packets, choosing with higher probability those links associated with higher pheromone values. In this way data for same destination are spread over multiple paths (but with more packets going over the best paths).

3. ACO ROUTING ALGORITHM FOR MANET

The ACO algorithm is designed to help solve the problem in mobile ad hoc network. Many different ACO based routing algorithms have been proposed and designed so far to tackle the problem of routing in MANET. The ACO technique for routing in MANETs uses stigmergy (indirect communication by modifying the environment) process to determine the best possible routes from a source node to a destination node. Artificial ants are placed at each node and they mark their trails with pheromone as they move within the network. The level of concentration of pheromone on a trail depends upon the quality of the path.

The proposed algorithm is a modification of AntNet algorithm [9] because AntNet has been designed for fixed network and we are going to use it in ad-hoc network, where the topology is highly dynamic. Thus, a neighbor discovery protocol is added in route maintenance phase to deal with mobility. Other phases follow the procedure given in AntNet.

3.1 NODE DATA STRUCTURE

AntNet is an ACO algorithm for data network routing proposed by Gianni Di Caro and Marco Dorigo [2]. Mobile agents (artificial ants) act concurrently and independently, and communicate in an indirect way (stigmergically), through the pheromones they read and write locally on the nodes. Each network node k stores two data structures:

1) Routing table T_k

For each possible destination d and for each neighbor node n, T_k stores a probability value P_{nd} expressing the goodness of choosing n as next node when the destination node is d:

$$\sum_{n \in N_k} P_{knd} = 1 \quad \text{where } d \in [1, N]$$

N_k = neighbor (k) Probability value P_{nd} represents the pheromone concentration along the link from node k to neighbor node *n* for destination node *d*.

2) Traffic model M_k

M_k (μ_d, $\sigma_{d}2$, W_d) is a statistical model of the traffic situation over the network as seen by node k. It is described by the sample mean μ_d and the variance $\sigma_{d}2$ computed over the trip times experienced by the artificial ants, and by a moving observation window W_d used to store the best value W_{best_d} of the artificial ants' trip time.

3.2 ACO ALGORITHM

1) At regular intervals Δt from every network node s, a forward ant $F_{s \to d}$ is launched toward a destination d to discover a feasible, low-cost path to that node and to investigate the load status of the network along the path. If fsd is a measure (in bits or in number of packets) of the data flow s→d, then the probability of creating at node s a forward ant with node d as destination is

$$P_{sd} = \frac{f_{sd}}{\sum_{i=1}^{N} f_{si}}$$

While travelling toward their destination nodes, the forward ants keep memory of their paths and of the traffic conditions found. The identifier of every visited node i and the time elapsed since the launching time to arrive at this i-th node are stored in a memory stack.

2) At each node i, each forward ant headed toward a destination d selects the node j to move to, with a probability P_{ijd} computed as normalized sum of the pheromone τ_{ijd} with a heuristic value η_{ij} taking into account the length of the *j*-th link queue of the current node *i*:

$$P_{ijd} = \frac{\tau_{ijd} + \alpha \eta_{ij}}{1 + \alpha(|N_i| - 1)}$$

The value of α weighs the importance of the heuristic value with respect to the pheromone values.

3) If a cycle is detected, the cycle's nodes are removed and all the memory about them is deleted. When an ant reaches a node that is already in its memory, a cycle is detected and all the nodes until this recurrent node are deleted from the ants memory.

4) When the destination node d is reached, the agent $F_{s \to d}$ generates backward ant $B_{d \to s}$, transfers to it all of its memory, and is deleted.

5) The backward ant takes the same path as that of its corresponding forward ant, but in the opposite direction.

6) Arriving at a node i coming from a neighbor node, the backward ant updates the local model of the traffic M_i and the pheromone matrix T_i, for all the entries corresponding to the destination node d.

- *Update traffic model M_i*

The estimated mean and variance are updated as follows:

$$\mu_{id} \leftarrow \mu_{id} + \varsigma(o_{i \to d} - \mu_{id})$$

$$\sigma_{id}^2 \leftarrow \sigma_{id}^2 + \varsigma((o_{i \to d} - \mu_{id})^2 - \sigma_{id}^2)$$

Where $o_{i \to d}$ is the observed ant's trip time from node i to destination d. The factor ς weighs the number of most recent samples that will really affect the average.

- Update pheromone matrix Ti

The backward ant $B_{d \to s}$ moving from node f to node i increases the pheromone values $\tau_{ifd'}$

$$p_{ifd'} \leftarrow p_{ifd'} + r(1 - \tau_{ifd'})$$

3.3 NEIGHBOR DISCOVERY PROTOCOL

This protocol works in parallel with the ACO Algorithm and its aim is to maintain a list with available neighbors to forward packets. It works as follows:

1. Every node broadcasts, with periodicity HELLO_INTERVAL, a message to all neighbors indicating that is available to forward packets.
2. Also with periodicity HELLO_INTERVAL, every node checks if its neighbors are still available. If the node k has not received any HELLO message from neighbor n and current time is greater than the expire time, neighbor n is deleted from the list. Furthermore, all entries in the routing table related to this neighbor are also erased.
3. Nodes are constantly listening. When a node k receives a HELLO message from node n, it first checks if that neighbor n is already in the list. If not, the new neighbor n is added to the list, and the routing table T_k is updated, adding an entry for every destination d with an associated probability P_{nd} that is initialized to a minimum value.

4. EXPERIMENTAL RESULTS

The protocol is simulated in NS-2. The metric for goodness of a selected path is assumed to be hop count. The protocol constructs the probabilistic routing tables, where better paths among all available paths have higher pheromone concentration. Thus, a path consisting of a series of links with highest pheromone values is always the favorable path between any source destination pair. The protocol also calculates next best path having lower pheromone value which would be helpful in selecting next best alternative path in case of link failures, without additional overheads.

The simulation modeled a network of 12 mobile nodes places randomly. Considering a single session in which they will be found in this arrangement. The channel capacity is 2Mb/s. simulation time is 11.2s, the time which the algorithm takes to converge. The time interval at which forward ants are launched is 0.03s.

Next hop

Dest.	5
0	1.000000
1	1.000000
2	1.000000
3	1.000000
4	1.000000
5	1.000000
6	1.000000
7	1.000000
9	1.000000
10	1.000000
11	1.000000

Table5: Routing table at Node 4

The table 1 to 9 represent the resulting routing tables at nodes 0 to 8 for an arbitrary topology. The first column represents the destination node. The other columns are the next hop neighbor for corresponding destination. The pheromone value, which indicates the probability of chosing corresponding next node by the packet for tha destination.

Next hop

Dest.	5
0	1.000000
1	1.000000
2	1.000000
3	1.000000
5	1.000000
6	1.000000
7	1.000000
8	1.000000
9	1.000000
10	1.000000
11	1.000000

Table1:Routing table at Node 0

Next hop

Dest.	2
0	1.000000
2	1.000000
3	1.000000
4	1.000000
5	1.000000
6	1.000000
7	1.000000
8	1.000000
9	1.000000
10	1.000000
11	1.000000

Table2: Routing table at Node 1

Next hop

Dest.	7	5	3	1
0	0.148759	0.773045	0.070153	0.008043
1	0.000000	0.000000	0.000000	1.000000
3	0.051738	0.000002	0.948258	0.000001
4	0.096252	0.867587	0.031585	0.004575
5	0.118359	0.775141	0.106500	0.000000
6	0.422056	0.015825	0.562082	0.000037
7	0.814397	0.000337	0.185266	0.000000
8	0.028491	0.905406	0.062376	0.003726
9	0.397596	0.275681	0.323983	0.002739
10	0.661876	0.051746	0.286060	0.000318
11	0.768465	0.023789	0.207734	0.000012

Table 3: Routing table at Node 2

Next hop

Dest.	11	6	3	2
0	0.303749	0.345789	0.234074	0.116388
1	0.004259	0.142377	0.409604	0.443761
2	0.000012	0.091400	0.464665	0.443924
3	0.000014	0.116202	0.831522	0.052262
4	0.256849	0.372338	0.065490	0.305323
5	0.276445	0.270299	0.284796	0.168460
6	0.219529	0.593945	0.130406	0.056120
8	0.130434	0.204408	0.474848	0.190310
9	0.750064	0.205366	0.043878	0.000692
10	0.978475	0.001516	0.019996	0.000012
11	1.000000	0.000000	0.000000	0.000000

Table 4: Routing table at Node 3

Next hop

Dest	7	6	2
0	0.181896	[illegible]	[illegible]31
1	0.163355	0.003541	0.833103
2	0.090559	0.000004	0.909436
4	0.234556	0.483971	0.281473
5	0.103101	0.505072	0.391827
6	0.043832	0.915430	0.040738
7	0.815368	0.093947	0.090685
8	0.130459	0.458348	0.411193
9	0.259107	0.650321	0.090572
10	0.483465	0.412383	0.104152
11	0.627195	0.219325	0.153480

Table 8: Routing table at Node 7

Dist.	9	8	6	4	2	0
0	0.000000	0.000000	0.000000	0.000000	0.000000	0.999999
1	0.043916	0.000127	0.136472	0.000127	0.819231	0.000127
2	0.062009	0.000000	0.140939	0.000000	0.797052	0.000000
3	0.310379	0.000008	0.465300	0.000008	0.224296	0.000008
4	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000
6	0.373372	0.000000	0.596386	0.000000	0.030242	0.000000
7	0.324202	0.000001	0.331045	0.000001	0.344750	0.000001
8	0.000000	0.999999	0.000000	0.000000	0.000000	0.000000
9	0.908237	0.000000	0.073511	0.000000	0.018252	0.000000
10	0.715963	0.000001	0.189954	0.000001	0.094082	0.000001
11	0.485468	0.000056	0.173503	0.000056	0.340861	0.000056

Table 6: Routing table at Node 5

Next hop

Next hop

Dest.	10	9	7	5	3
0	0.076955	0.172125	0.008356	0.729907	0.012657
1	0.016061	0.145939	0.161019	0.085261	0.591721
2	0.002451	0.108759	0.212591	0.043728	0.632471
3	0.000139	0.000000	0.111298	0.000000	0.888562
4	0.001381	0.307327	0.004176	0.666984	0.020131
5	0.050645	0.302448	0.000068	0.646315	0.000523
7	0.002219	0.000005	0.701497	0.000000	0.296278
8	0.051266	0.056839	0.021544	0.862462	0.007889
9	0.082513	0.906571	0.010490	0.010490	0.000001
10	0.742717	0.187890	0.069197	0.000001	0.000194
11	0.373550	0.013431	0.594944	0.000052	0.018023

Table 7: Routing table at Node 6

Next hop

Dest.	5
0	1.000000
1	1.000000
2	1.000000
3	1.000000
4	1.000000
5	1.000000
6	1.000000
7	1.000000
9	1.000000
10	1.000000
11	1.000000

Table 9: Routing table at Node 8

5. PERFORMANCE ANALYSIS

The performance of the algorithm is evaluated in different scenarios. The performance of the algorithm is also compared to an existing algorithm AODV.
A simulation setting was created with 12 mobile nodes randomly placed to form an ad hoc network. The nodes move with a maximum velocity of 10m/s. The metrics chosen to evaluate the efficiency and effectiveness of protocol were

1) *Packet Delivery Ratio*- The ratio of the number of packets delivered to the destination to the number of packets generated by sources.
2) *Average End-to-End Delay*- This is the average time taken by a data packet to travel from the source to the destination.
3) *Control Overhead*- The control overhead is the total number of routing packets transmitted for the entire simulation time.

The performance comparison of ACO Algorithm with AODV is carried out in an increasingly dynamic environment. We vary the node mobility by decreasing the node pause time. The lower the pause time the higher the mobility.
In Figure 2, the packet delivery ratio is measured at varying pause time, both algorithms show improved performance. Their performance actually alternates, AODV starts out better but ACO Algorithm finishes with higher delivery ratio for longer pause times.

Figure 3, shows that AODV experiences a higher end-to-end delay than ACO Algorithm. The delay in ACO Algorithm is usually significant at the start of the simulation because of the initial search for routes. Intermediate nodes in AODV are able to respond to route requests thus saving the time for path discovery. In ACO Algorithm, the source node has to wait till it gets a BANT sent by the destination.

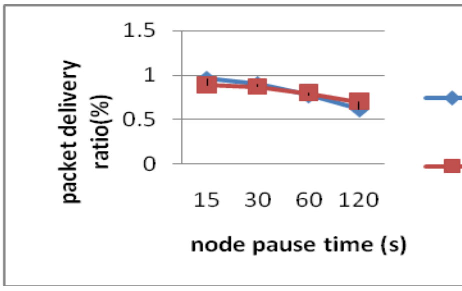

Figure 2: Packet Delivery Ratio measured against varying Pause times of mobile node

In Figure 4, ACO Algorithm experiences quite a bit of overhead due to different control packets that have to be sent around the network for route maintenance and discovery.

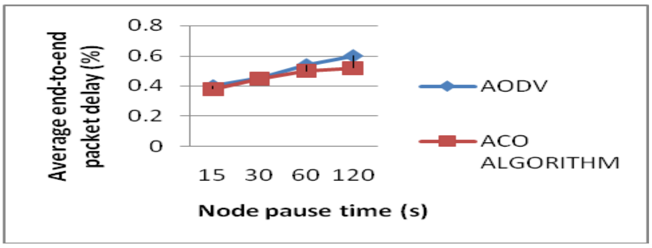

Figure 3: Average End-to-End Delay measured against varying Pause times of mobile nodes

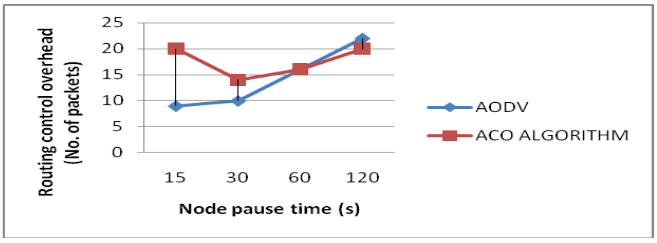

Figure 4: Routing Control Overhead measured against varying Pause times of mobile nodes

6. CONCLUSION

The protocol implementation has been successfully carried out in ns-2. The protocol implementation shows superior performance. The adaptive and probabilistic behavior is depicted in the resulting routing tables. The experiments generated correct routing tables for all topologies simulated. In this paper an improvement on previously proposed AntNet Algorithm is done for routing in MANET. It uses neighbor discovery protocol for route maintenance and handle link failures in the network. The efficiency of the protocol is compared to an existing protocol. It gives a better approach to find best route based to the current traffic load and network condition.

There are many of directions in which the current work can be extended. As the protocol is resource intensive algorithm. An obvious extension of the current work would be a variant of the algorithm that has lesser routing overhead and memory requirements.

REFERENCES

[1] "Ant Colony Optimization", Marco DArigo and Thomas Stutzle, MIT Press Landon, England

[2] "Artificial Ants as a Computation Intelligence Technique" Macro Dorigo, Mauro Birattari, and Thomas STUTZLE, Technical report No. TR/IRIDIA/2006-023 September 2006

[3] M. D. E. Bonabeau and G. Theraulaz. "*Swarm Intelligence:From natural to artifical systems*". Oxford University Press, New York, NY, 1999.

[4] J.-L. Deneubourg, S. Aron, S. Goss, and J.-M. Pasteels, "*The selforganizing exploratory pattern of the Argentine ant*," Journal of Insect Behavior, vol. 3, p. 159, 1990.

[5] C. Perkins and E. M. Royer. "Ad–hoc on-demand distance vector routing". In Proceedings of 2nd IEEE Workshop on Mobile Computing Systems and Applications,

[6] C. Perkins and P. Bhagwat. "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers". In ACM SIGCOMM' 94 Conference on Communications Architectures, Protocols and Applications.

[7] Z. J. Haas. A new Routing Protocol for Reconfigurable Wireless Networks. In Proceedings of IEEE International Conference on Universal Personal Communication, Atlanta, GA, October 1997.

[8] G. Di Caro. Ant Colony Optimization and its application to adaptive routing in telecommunication networks. PhD thesis, Facult_e des Sciences Appliqu_ees, Universit_e Libre de Bruxelles, Brussels, Belgium, 2004.

[9] G. Di Caro and M. Dorigo. AntNet: Distributed stigmergetic control for communications networks. Journal of Artificial Intelligence Research (JAIR),1998.

Anubhuti Verma received her M.Tech degree in Information Technology from Univesity School of Information Technology, Guru Govind Singh Indraprastha University, Delhi. She is Assistant Professor in Deptt. Of Information Technology, NIET, Greater Noida. Her major area of interest include mobile computing and Network routing. She has published four research papers in national and international conferences.