Analysis of Reference Point Group Mobility Model in Mobile Ad hoc Networks with an Ant Based Colony Protocol

Mrs.M. Sivajothi* & Dr. E.R. Naganathan**

Abstract - Different routing protocols for mobile ad hoc networks have been evaluated with respect to different mobility models. Reference Point Group Mobility (RPGM) model applied for ad hoc network protocol performance evaluations have been studied here.

In this paper an ant based colony routing protocol in mobile ad hoc networks is described. The algorithm consists of both reactive and proactive components. In a reactive path set up phase, multiple paths are built between the source and destination of a data session. During the course of the session, paths are continuously monitored and improved in a proactive way. The algorithm makes use of ant-like mobile agents establishing multiple stable paths between source and destination nodes. By simulation experiments, it is shown that the performance of the proposed protocol outperforms the standard AODV routing algorithm. Specifically, how the performance results out of an ad hoc network protocol drastically resulting in the change of the mobility model simulated in terms of end-to-end delay and delivery ratio is illustrated.

1. Introduction

A mobile Ad Hoc Network (MANET) is a collection of wireless mobile nodes dynamically forming a temporary network without the use of existing network infrastructure or centralized administration. In such a network, each node operates not only as a host, but also as a router, forwarding packets for mobile nodes that may not be within the direct transmission range of each other.

The set of applications for MANETs is diverse, ranging from small, static networks that are constrained by power sources, to large-scale, mobile, highly dynamic networks. The design of network protocols for these networks is a complex issue. Regardless of the application, MANETs need efficient distributed algorithms to determine network organization, link scheduling, and routing. However, determining viable routing paths and delivering messages in a decentralized environment where the network topology fluctuates is not a well-defined problem.

* SG Lecturer & Head, Dept of Computer Science, Sri Parasakthi College for Women, Courtallam, Tamil Nadu, India. E-Mail: rksmano@yahoo.com.
** Prof & Head, Dept of Information Technology, Sikkim Manipal University, Manipal, Karnataka. E-Mail: *ern_jo@yahoo.com*

The general problem of modeling the behavior of the nodes belonging to a mobile network has not a unique and straightforward solution. Mobility and disconnection of mobile hosts pose a number of problems in designing proper routing schemes for effective communication between any source and destination. The mobility models that are commonly used to simulate MANETs can be classified into two categories: individual-based and group-based. An individual-based model describes node mobility independently of any other nodes. With regard to group-based mobility models, individual node movement is dependent on the movement of close-by-nodes.

The objective of this paper is to show the performance of the RPGM model on the behavior of a routing protocol and to present the critical factors that must be taken into consideration when optimizing the behavior of a routing protocol for MANETs based on ant colony optimization framework.

Ant Colony Optimization (ACO), called ant system was inspired by studies of the behavior of ants. As a multi-agent approach to different combinatorial optimization problems, like the traveling salesman problem and the quadratic assignment problem the ant-colony meta-heuristic framework enabled ACO to be applied to a range of combinational optimization problems. Ant colony algorithms have been founded on an observation of real ant colonies. By living in colonies, ants' social behavior is directed more to the survival of the colony as an entity rather than to that of an individual member of the colony. An interesting and significantly important behavior of ant colonies is their foraging behavior and, in particular, their ability to find the shortest route between their nest and a food source, realizing that they are almost blind [2].

Ant colony optimization algorithms have been used to produce near-optimal solutions to the traveling salesman problem. They have an advantage over simulated annealing and genetic algorithm approaches when the graph may change dynamically; the ant colony algorithm can be run continuously and adept to changes in real time. This is of interest in network routing and urban transportation systems [3].

The remainder of the paper is organized as follows. Section 2 reviews some of the existing work in this area. Section 3 presents our AntHocNetM routing algorithm. The experimental results are presented in Section 4 and conclusion part of the paper in Section 5.

2. Related Work

Alena Shmygelska et al [2] present a substantially improved version of the ACO algorithm proposed for solving an abstract variant of one of the most challenging problems in computational biology: the prediction of a protein's structure from its amino acid sequence. Genomic and proteomic sequence information is now available for an increasing number of organisms, and genetic engineering methods for producing proteins are well developed.

Frederick Ducatelle et al [4] present an algorithm for routing in mobile ad hoc networks based on ideas from the Nature-inspired Ant Colony Optimization framework. The algorithm consists of both reactive and proactive components. Data are stochastically spread over the different paths, according to their estimated quality. During the course of the session, paths are continuously monitored and improved in a proactive way. Link failures are dealt with locally. The algorithm makes extensive use of ant-like mobile agents with full paths between source and destination nodes in a Monte Carlo fashion. Romit Roy Choudhuri et al [5] address the problem of maintaining a stable route in a resource limited, dynamic architecture like MANET, proposing an agent based protocol that guarantees uninterrupted message transfer against the backdrop of minimal congestion and delay.

Siriluck Lorpunmanee et al [6] address the problem by developing a general framework of grid scheduling using dynamic information and an ant colony optimization algorithm to improve the decision of scheduling. The performance of various dispatching rules such as First Come First Served (FCFS), Earliest Due Date (EDD), Earliest Release Date (ERD), and an Ant Colony Optimization (ACO) are compared. Moreover, the benefit of using an Ant Colony Optimization for performance improvement of the grid Scheduling finds a place.

Rafael S. Parpinelli et al [7] present an overview of Ant-Miner, an ACO algorithm for discovering classification rules in data mining. In the classification task, each case of the data being mined consists of two parts: a goal attribute whose value is to be predicted and a set of predictor attributes. The aim is to predict the value of the goal attribute for a case given the values of the predictor attributes for that case.

3. AntHocNetM

AntHocNetM is a hybrid multipath algorithm, designed along the principles of ACO routing. It consists of both reactive and proactive components. It does not maintain routes to all possible destinations at all times (like the original ACO algorithms for wired networks), but only sets up paths when they are needed at the start of a data session. This is done in a reactive route set up phase, where ant agents called reactive forward ants are launched by the source in order to find multiple paths to the destination, and backward ants return to the source to set up the paths. According to the common practice in ACO algorithms, the paths are set up in the form of pheromone tables indicating their respective quality. After the route set up, data packets are routed stochastically over the different paths following these pheromone tables. While the data session is going on, the paths are monitored, maintained and improved proactively using different agents, called proactive forward ants. The algorithm reacts to link failures with either a local route repair or by warning preceding nodes on the paths [4].

3.1 Establishing Multiple paths

When a source node *s* starts a communication session with a destination node *d,* and it does not have routing information for *d* available, it broadcasts a reactive forward ant F^s_d. Due to this initial broadcasting, each neighbor of *s* receives a replica of F^s_d. We refer to the set of replicas which originated from the same original ant as an *ant generation.* The task of each ant of the generation is to find a path connecting *s* and *d.* At each node, an ant is either unicast or broadcast, according to whether or not the node has routing information for *d.* The routing information of a node *i* is represented in its pheromone table T^i. The entry $T^i_{nd} \varepsilon R$ of this table is the pheromone value indicating the estimated goodness of going from *i* over neighbor *n* to reach destination *d.* If pheromone information is available, the ant chooses its next hop *n* with probability P_{nd}:

$$P_{nd} = \frac{(T^i_{nd})^\beta}{\Sigma j \varepsilon N_d^i (T^i_{jd})^\beta}, \quad \beta \geq 1 \qquad (1)$$

where N_d^i is the set of neighbors of *i* over which a path to *d* is known, and β is a parameter value which can control the exploratory behavior of the ants (although in current experiments β is kept to 1).

If no pheromone information is available for d, the ant is broadcast. Due to this broadcasting, ants can proliferate quickly over the network, following different paths to the destination (although ants which have reached a maximum number of hops, related to the network diameter, are deleted). When a node receives several ants of the same generation, it will compare the path traveled by each ant to that of the previously received ants of this generation: only if its number of hops and travel time are both within an acceptance factor a1 of that of the best ant of the generation, it will forward the ant. Using this policy, overhead is limited by removing ants, which follow bad paths, while there is still the possibility to find multiple good paths. However, it does have as an effect that the ant that arrives first in a node is let through, while subsequent ants meet with selection criteria set by the best of the ants preceding them, which means that they have higher chances of being rejected. This can lead to "kite-shaped" paths. In order to obtain a mesh of sufficiently disjoint multiple paths which provide much better protection in case of link failures, we also consider in the selection policy the first hop taken by the ant. If this first hop is different from those taken by previously accepted ants, we apply a higher (less restrictive) acceptance factor a2 than in the case the first hop had already seen before (in the experiments a2 was set to 2 as opposed to a1 = 0:9). Each forward ant keeps a list P of the nodes [1,..., *n]* it has visited. Upon arrival at the destination d, it is converted into a backward ant, which travels back to the source retracing P (if this is not possible because the next hop is not there, for instance due to node movements, the backward ant is discarded).

The backward ant incrementally computes an estimate $\widehat{T}p$ of the time it would take of a data packet to travel over *P* towards the destination, which is used to update routing tables. $\widehat{T}p$ is the sum of local estimates $\widehat{T}_{i+1}^{i}$ in each node $i \in P$ of the time to reach the next hop *i* + 1:

3.2. Routing Multimedia Data

The path set up phase described above creates a number of good paths between source and destination, indicated in the routing tables of the nodes. Data can then be forwarded between nodes according to the values of the pheromone entries. Nodes in AntHocNetM forward data stochastically. When a node has multiple next hops for the destination d of the data, it will randomly select one of them, with probability P_{nd}. P_{nd} is calculated in the same way as for the reactive forward ants, but with a higher exponent (in the experiments set to 2), in order to be more greedy with respect to the better paths:

$$P_{nd} = \frac{(\mathcal{T}_{nd}^{i})^{\beta_2}}{\sum_{j \in \mathcal{N}_d^i} (\mathcal{T}_{jd}^{i})^{\beta_2}}, \quad \beta_2 \geq \beta_1 .$$

According to this strategy, we do not have to choose as a priori how many paths to use: their number will be automatically selected in function of their quality.

The probabilistic routing strategy leads to data load spreading according to the estimated quality of the paths. If the estimates are kept up-to-date (which is done using the proactive ants described in Subsection 3.3), this leads to automatic load balancing. When a path is clearly worse than others, it will be avoided, and its congestion will be relieved. Other paths will get more traffic, leading to higher congestion, which will make their end-to-end delay increase. By continuously adapting the data traffic, the nodes try to spread the data load evenly over the network.

Due to the ant agents, the nodes can now determine the best route locally and initiate the sending of data packets through it. After a point of time, if the source node finds that the chosen route has attained a low stability [4], the node computes a new better stable route from the local information cache and redirects data packets through the later. Due to this adaptive route selection, multiple paths can communicate data continuously. The route discovery delay is minimal. So we can predict that because of the connectivity of two nodes, they can able to communicate by selecting at least one route.

3.3. Link failures

In AntHocNetM, each node tries to maintain an updated view of its immediate neighbors at each moment, in order to detect link failures as quickly as possible, before they can lead to transmission errors and packet loss. The presence of a neighbor node can be confirmed when a hello message is received, or after any other successful interception or exchange of signals. The disappearance of a neighbor is assumed when such an event has not taken place for a certain amount of time, defined by hello X allowed - hello-loss, or when a unicast transmission to this neighbor fails.

When a neighbor is assumed to have disappeared, the node takes a number of actions. In the first place, it removes the neighbor from its neighbor list and all the associated entries from its routing table. Further actions depend on the event, which was associated with the discovered disappearance. If the event was a failed transmission of a control packet, the node broadcasts a *link failure notification* message. Such a message contains a list of the destinations to which the node lost its best path and the new best estimated end-to-end delay and number of hops to this destination (if it still has entries for the destination). All its neighbors receive the notification and update their pheromone table using the new estimates. If they in turn lost their best or their only path to a destination due to the failure, they will broadcast the notification further, until all concerned nodes are notified of the new situation.

If the event was the failed transmission of a data packet, the node sends the link failure notification only about the destinations for which it lost its best next hop and if this was not the only next hop. For the destinations for which it lost its only next hop, the node starts a *local route repair.* The node broadcasts a *route repair ant* that travels to the involved destination like a reactive forward ant: it follows available routing information when it can, and is broadcast otherwise. One important difference is that it has a maximum number of broadcasts, so that its proliferation is limited. The node waits for a certain time and if no backward repair ant is received by then, it concludes that it was not possible to find an alternative path to the destination. Packets that were in the meantime buffered for this destination are discarded, and the node sends a link failure notification about the lost destinations.

4. Experimental results.

This section describes the simulation model and presents the results obtained using the NS2 simulator [6]. In all simulation the data at PHY layer is 11 Mbps and the routing protocol is AODV [7].

Our simulation is implemented on the NS2 simulator. Simulation environment consists of 50 wireless nodes forming an ad hoc network, moving in a 1000 x 800 m rectangular region for 50 seconds simulation time. The data packets used for simulations are 512 bytes and the data sending rate is 0.1Mb. The MAC layer uses the distributed coordination function (DCF) of IEEE 802.11b protocol for the wireless standard. The radios use two-ray ground propagation model and have a receiving range of 250 meters. Movements of the nodes are obtained using the Reference Point Group Mobility (RPGM) model of NS2. Each node moves independently with the same average speed.

In this mobility model, each node belongs to a group in which every node follows a logical centre reference point. The nodes in a group are usually randomly distributed around the reference point. The different nodes use their own mobility model and are then added to the reference point, which drives them in the direction of the group.

We compare our AntHocNetM protocol with AODV. We evaluate mainly the performance according to the following metrics.

We vary the following parameters:

- Number of Nodes
- Node speed
- Pause time
- Simulation time

and measure the packet delivery ratio and end-to-end delay.

First we vary the number of nodes as 10,20,30,40 and 50 and evaluated the delivery ratio and delay. From Figures 1 and 2, we can see that AntHocNetM gives better results when compared with AODV for all settings. Figures 3 and 4 show results of delivery ratio and average delay for increasing the node's pause time (from 10 to 50s) in a RPGM model. It can be seen that AntHocNetM clearly outperforms AODV. Figures 5 and 6 show that evaluation of the delivery ratio and delay for increasing node speed (from 10 to 40s) in a RPGM model. Once again AntHocNetM outperforms AODV. Figures 7 and 8 show the evaluation of the delivery ratio and delay for increasing simulation time from (10 to 100s). Once again AntHocNetM outperforms AODV clearly for delivery ratio and delay.

5. Conclusion

This paper gives an analysis of the behavior of a routing protocol with an ant colony based algorithm when group mobility is involved. The main aim was to prove that the mobility model can deeply affect the performance results of a routing protocol. The behavior of a classical reactive protocol, the Ad hoc on Demand distance Vector (AODV) protocol and hybrid AntHocNetM protocol have been compared and arrived at a thorough evaluation of its behavior obtained with the RPGM model. We observe the high variability of the results and the need to know exactly the behavior of the system. It is evident that RPGM model has a strong impact on the AntHocNetM routing protocol performance and should therefore be taken into consideration when designing a routing protocol.

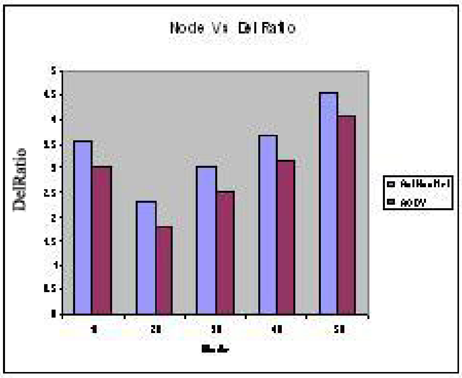

Figure 1. Node Vs Del Ratio

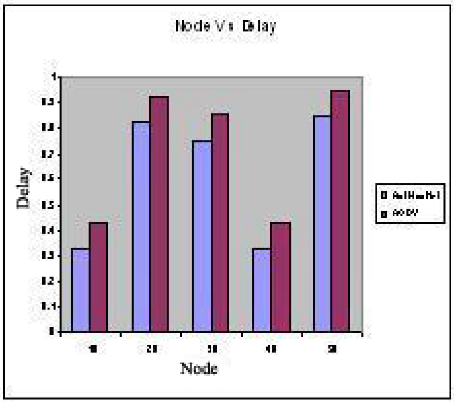

Figure 2. Node Vs Delay

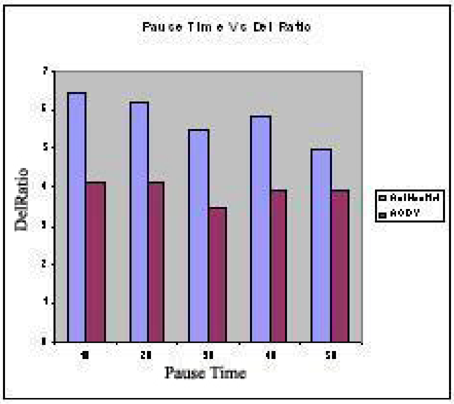

Figure 3. Pause Time Vs Del Ratio

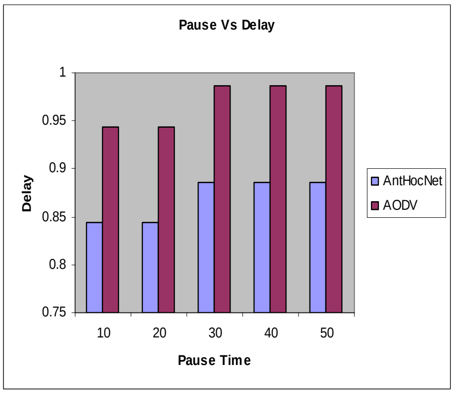

Figure 4. Pause Time Vs Delay

ISBN: 978-988-17012-2-0

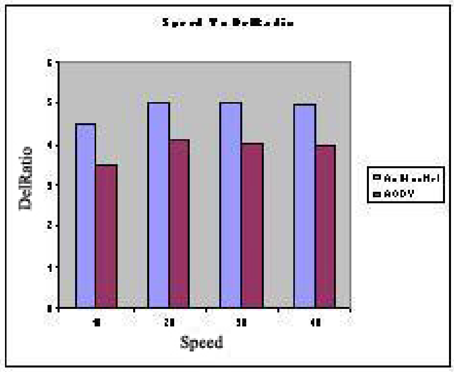

Figure 5. Speed Vs Del Ratio

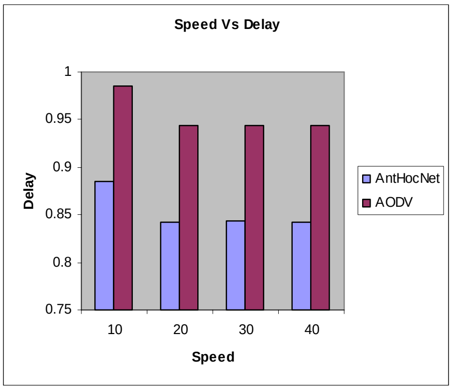

Figure 6. Speed Vs Delay

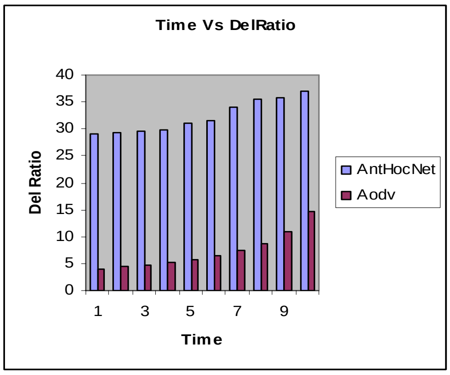

Figure 7. Time Vs Del Ratio

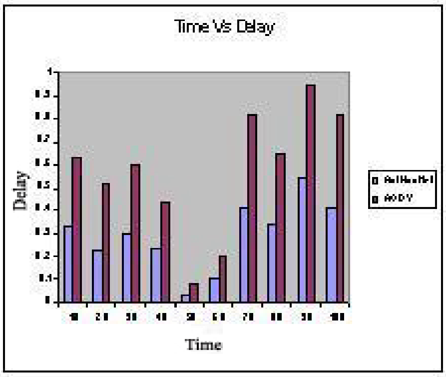

Figure 8.Time Vs Delay

References

[1] “Mobile Adhoc Networks” from http://w3.antd.nist.gov/ wahn_mahn.shtml

[2] Jalali, M. R. et al. “Improved ant colony optimization algorithm for reservoir operation” Scientia Iranica, 2006, 13(3): 295-302.

[3] “Ant colony optimization” from http://en.wikipedia.org/wiki /Ant_colony_ optimization

[4] Frederick Ducatelle et al. “Ant Agents for Hybrid Multipath Routing in Mobile Ad Hoc Networks” Second Annual Conference on Wireless On demand Network Systems and Services (WONS), St. Moritz, Switzerland, January 19-21, 2005.

[5] Romit RoyChoudhuri et al “An Agent-based Protocol to Support Multimedia Communication in Ad Hoc Wireless Networks”, 1st International Workshop on Parallel and Distributed Computing :Issues in Wireless Networks and Mobile Computing, San Francisco, USA, April 23-27, 2001.

[6] Siriluck Lorpunmanee et al "An Ant Colony Optimization for Dynamic Job Scheduling in Grid Environment" International Journal of Computer and Information Science and Engineering 2007.

[7] Rafael S. Parpinelli et al “Classification-Rule Discovery with an Ant Colony Algorithm” IEEE/WIC International Conference on Intelligent Agent Technology, 2003.

M.Sivajothi is working as the Head of the Department of Computer Science in Sri Parasakthi College for women, Courtallam. Her research interests lie in the area of Ad hoc Networks.

Dr.E.R.Naganathan is the Chairperson, School of Computer Science and Head of the Department of Information Technology, Skkim Manipal University, Manipal, Karnadaka, India. He received his Ph.D. in Computer Applications from Alagappa University, India. His main research interest is in Network Security, Cryptography, and Optimization Techniques. He is a member of ISTE, India. He has 25 publications in his credit.