An Ant Colony Based Routing Protocol to Support Multimedia Communication in Ad Hoc Wireless Networks

M. Sivajothi and Dr.E.R.Naganathan

Sri Parasakthi College for Women,TN, India Alagappa University, Karaikudi, TN, India

Summary

In mobile ad hoc networks, the protocols proposed for multimedia data communication have failed to ensure a stable QoS, because of changing link stability and link-failures. In this paper we describe AntHocNetM, a routing algorithm to support multimedia communications in mobile ad hoc networks based on Ant Colony Optimization framework. The algorithm consists of both reactive and proactive components In a reactive path setup phase, multiple paths are built between the source and destination of a data session. The multimedia data are send over the stable, failure-free paths. 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 establishes multiple stable paths between source and destination nodes. By simulation experiments, we show that the performance of AntHocNetM outperforms the standard AODV routing algorithm in terms of end-to-end delay, delivery ratio and jitter.

Key words:

Ant Colony Optimization, Multimedia, Wireless Networks, Routing Protocol, AODVs.

1. Introduction

In the next generation of wireless communication systems, there will be a need for the rapid deployment of independent mobile users. Significant examples include establishing survivable, efficient, dynamic communication for emergency/rescue operations, disaster relief efforts, and military networks. Such network scenarios cannot rely on centralized and organized connectivity, and can be conceived as applications of Mobile Ad Hoc Networks. A MANET is an autonomous collection of mobile users that communicate over relatively bandwidth constrained wireless links. Since the nodes are mobile, the network topology may change rapidly and unpredictably over time. The network is decentralized, where all network activity including discovering the topology and delivering messages must be executed by the nodes themselves, i.e., routing functionality will be incorporated into mobile nodes [1].

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 network topology fluctuates is not a well-defined problem. While the shortest path (based on a given cost function) from a source to a destination in a static network is usually the optimal route, this idea is not easily extended to MANETs. Factors such as variable wireless link quality, propagation path loss, fading, multiuser interference, power expended, and topological changes, become relevant issues. The network should be able to adaptively alter the routing paths to alleviate any of these effects. Moreover, in a military environment, preservation of security, latency, reliability, intentional jamming, and recovery from failure are significant concerns. Military networks are designed to maintain a low probability of intercept and/or a low probability of detection. Hence, nodes prefer to radiate as little power as necessary and transmit as infrequently as possible, thus decreasing the probability of detection or interception. A lapse in any of these requirements may degrade the performance and dependability of the network.

Mobility and disconnection of mobile hosts pose a number of problems in designing proper routing schemes for effective communication between any source and destination. In this paper we describe AntHocNetM, a routing algorithm to support multimedia communications in mobile ad hoc networks 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 was 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

Manuscript received July 5, 2008.
Manuscript revised July 20, 2008.

an entity rather than to that of an individual member of the colony. An interesting and significantly important behavior of ant colonies is their for aging 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 adapt to changes in real time. This is of interest in network routing and urban transportation systems [3].

It has been experimentally observed that ants in a colony can converge on moving over the shortest among different paths connecting their nest to a source of food. The main catalyst of this colony-level shortest path behavior is the use of a volatile chemical substance called *pheromone*: ants moving between the nest and a food source deposit pheromone, and preferentially move in the direction of areas of higher pheromone intensity. Shorter paths can be completed quicker and more frequently by the ants, and will therefore be marked with higher pheromone intensity. These paths will therefore attract more ants, which will in turn increase the pheromone level, until there is convergence of the majority of the ants onto the shortest path. The local intensity of the pheromone field, which is the overall result of the repeated and concurrent *path sampling* experiences of the ants, encodes a spatially distributed *measure of goodness* associated with each possible move.[4].

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 conclude the paper in Section 5.

2. Related Work

Alena Shmygelska et al [2] we 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 aminoacid 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] 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 which sample full paths between source and destination nodes in a Monte Carlo fashion.

Romit RoyChoudhuri et al [5] address the problem of maintaining a stable route in a resource limited, dynamic architecture like MANET, proposing an agentbased protocol that guarantees uninterrupted message transfer against the backdrop of minimal congestion and delay.

Siriluck Lorpunmanee et al [6] addresses 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 is also discussed.

Ahmed Al-Ani [7] presents a novel method that utilizes the ACO algorithm to implement a feature subset search procedure. Initial results obtained using the classification of speech segments are very promising.

Rafael S. Parpinelli et [8] 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 consisits 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 alues of the predictor attributes for that case.

Philip Hingston [9] shows how an ant colony optimisation algorithm may be used to enumerate knight’s tours for variously sized chessboards. We have used the algorithm to enumerate all tours on 5x5 and 6x6 boards, and, while the number of tours on an 8x8 board is too large for a full enumeration, our experiments suggest that the algorithm is able to uniformly sample tours at a constant, fast rate for as long as is desired.

Osvaldo Gomez et al [10] propose the Omicron ACO (OA), a novel population-based ACO alternative originally designed as an analytical tool. To experimentally prove OA advantages, this work compares the behavior between the OA and theMMAS as a function of time in two well-known TSP problems. A simple study of the behavior of OA as a function of its parameters shows its robustness.

Matthijs den Besten [11] present an algorithm based on the Ant Colony Optimization (ACO) metaheuristic for the single machine total weighted tardiness problem, a well known NP–hard scheduling problem. Our ACO algorithm is currently among the best algorithms known for this problem type. It discusses three elements that enable it to find very good solutions quickly.

Panagiotis Papadimitratos et al [12] present a route discovery protocol that mitigates the detrimental effects of such malicious behavior, as to provide correct connectivity information. This protocol guarantees that fabricated, compromised, or replayed route replies would either be rejected or never reach back the querying node. Furthermore, the protocol responsiveness is safeguarded under different types of attacks that exploit the routing protocol itself. The sole requirement of the proposed scheme is the existence of a security association between the node initiating the query and the sought destination.

Emmanuel A. Gonzalez et al. [13] algorithm for PID controllers that is pre-tuned using the quarter-wave Ziegler-Nichols PID tuning method. The retuning algorithm uses a Sequential Ant Colony Optimization (SeqACO) metaheuristic in order to determine the optimal values of the PID constants, represented by a discrete combinatorial optimization problem in the form of a directed graph.

A. Shmygelska et al. [14] present a substantially improved version of the ACO algorithm for solving an abstract variant of one of the most challenging problems in computational biology: the prediction of a protein's structure from its aminoacid 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.

Ines Alaya [15] a generic algorithm based on Ant Colony Optimization to solve multi-objective optimization problems. The proposed algorithm is parameterized by the number of ant colonies and the number of pheromone trails. We compare different variants of this algorithm on the multi-objective knapsack problem. We compare also the obtained results with other evolutionary algorithms from the literature.

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 setup 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 setup, 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. Reactive path setup

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_d^s Due to this initial broadcasting, each neighbor of s receives receives a replica $F_d^s(k)$ of F_d^s. In what follows, we will also refer to the set of replicas which originated from the same original ant as an ant generation. The task of each ant $F_d^s(k)$ 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 current node has routing information for d. The routing information of a node i is represented in its pheromone table $\mathcal{T}^i$. The entry $\mathcal{T}_{nd}^i \in \mathbb{R}$ 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 will choose its next hop n with the probability P_{nd}:

$$P_{nd} = \frac{(\mathcal{T}_{nd}^i)^{\beta_1}}{\sum_{j \in \mathcal{N}_d^i} (\mathcal{T}_{jd}^i)^{\beta_1}}, \quad \beta_1 \geq 1,$$

where $\mathcal{N}_d^i$ is the set of neighbors of i over which a path to d is known, and β_1 is a parameter value which can lower the exploratory behavior of the ants (although in current experiments β_1 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 which 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, as shown in Figure 1. In order to obtain a mesh of sufficiently disjoint multiple paths as shown in Figure 2, which provides 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 was 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 ^ TP of the time it would take a data packet to travel over *P* towards the destination, which is used to update routing tables. $\hat{T}_{\mathcal{P}}$ is the sum of local estimates $\hat{T}^i_{i+1}$ in each node i 2 P of the time to reach the next hop $i+1$.

$$\hat{T}_{\mathcal{P}} = \sum_{i=1}^{n-1} \hat{T}^i_{i+1} .$$

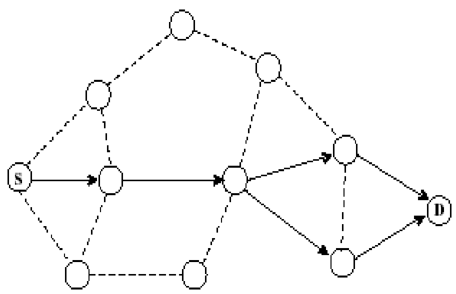

Fig. 1 “Kite-shaped” multiple paths

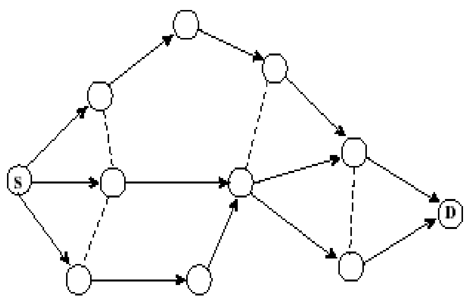

Fig. 2 A mesh of multiple paths

The value of $\hat{T}^i_{i+1}$ is defined as the product of the estimate of the average time to send one packet, $\hat{T}^i_{mac}$, times the current number of packets in queue (plus one) to be sent at the MAC layer, Q^i_{mac}:

$$\hat{T}^i_{i+1} = (Q^i_{mac} + 1)\hat{T}^i_{mac} .$$

$\hat{T}^i_{mac}$, is calculated as a running average of the time elapsed between the arrival of a packet at the MAC layer and the end of a successful transmission. So if t^i_{mac} is the time it took to send a packet from node i, then node i updates its estimate as:

$$\hat{T}^i_{mac} = \alpha \hat{T}^i_{mac} + (1-\alpha) t^i_{mac},$$

with $\alpha \in [0,1]$. Since $\hat{T}^i_{mac}$, is calculated at the MAC layer it includes channel access activities, so it takes into account local congestion of the shared medium. Forward ants calculate a similar time estimate $\hat{T}_{\mathcal{P}}$, which is used for the filtering of the ants, as mentioned above.

At each intermediate node $i \in \mathcal{P}$, the backward ant virtually sets up a path towards the destination d, creating or updating the pheromone table entry $\mathcal{T}^i_{nd}$ in $\mathcal{T}^i$. The pheromone value in $\mathcal{T}^i_{nd}$ represents a running average of the inverse of the cost, in terms of both estimated time and number of hops, to travel to d through n. If $\hat{T}^i_d$ is the travelling time estimated by the ant, and h is the number of hops, the value $\hat{T}^i_d$ used to update the running average is defined as:

$$\tau^i_d = \left(\frac{\hat{T}^i_d + hT_{hop}}{2} \right)^{-1},$$

where T_{hop} is a fixed value representing the time to take one hop in unloaded conditions. Defining τ^i_d like this is a way to avoid possibly large oscillations in the time estimates gathered by the ants (e.g., due to local bursts of traffic) and to take into account both end-to-end delay and number of hops. The value of $\mathcal{T}^i_{nd}$ is updated as follows:

$$\mathcal{T}^i_{nd} = \gamma \mathcal{T}^i_{nd} + (1-\gamma)\tau^i_d, \ \gamma \in [0,1],$$

Where γ and α were both set to 0:7 in the experiments.

3.2. Routing Multimedia Data

The path setup 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:

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

According to this strategy, we do not have to choose 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.

For real time/multimedia communication, uninterrupted communication and delay minimization are the most important constraints. In our protocol, the local topology information is regularly updated through the ant agents. Whenever a best route is selected by the protocol, the source node start sending the data packets through that path. It checks periodically, whether a stable better route exists for the destination. When such a route is found, the data packets are redirected through this new route. To define the quality of route, we have used the link stability and number of hops as parameters.

Due 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, data can be communicated continuously by multiple paths. 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. The availability of multiple routes always leads to a best route so that the uninterrupted multimedia communication session can be established, through this 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 canlead 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 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 (which we set to 2 in our experiments), so that its proliferation is limited. The node waits for a certain time (empirically set to 5 times the estimated end-to-end delay of the lost path), 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 which were in the meantime buffered for this destination are discarded, and the node sends a link failure notification about the lost destinations.

Link failure notifications keep routing tables on paths up-to-date about upstream link failures. However, they can sometimes get lost and leave dangling links. A data packet following such a link arrives in a node where no further pheromone is available. The node will then discard the data packet and unicast a warning back to the packet's previous hop, which can remove the wrong routing information

4 Experimental Results

4.1 Simulation Model and Parameters

We use NS2 to simulate our proposed algorithm. In our simulation, the channel capacity of mobile hosts is set to the same value: 2 Mbps. We 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 our simulation, mobile nodes move in a 1000 meter x 800 meter rectangular region for 50 seconds simulation time. Initial locations and movements of the nodes are obtained using the random waypoint (RWP) model of NS2. We assume each node moves independently with the same average speed. All nodes have the same transmission range of 250 meters.

In this mobility model, a node randomly selects a destination from the physical terrain. It moves in the direction of the destination in a speed uniformly chosen between the minimal speed and maximal speed. After it reaches its destination, the node stays there for a pause time and then moves again.

The simulated traffic is Constant Bit Rate (CBR). For each scenario, ten runs with different random seeds were conducted and the results were averaged.

We compare our AntHocNetM protocol with AODV We evaluate mainly the performance according to the metrics given in the next section.

We vary the following parameters

- No.of Nodes
- Node Speed
- Pause time
- No.of Multimedia flows

and measure the Packet Delivery ratio, End-to-end delay , Jitter , Packets Dropped.

4.2 Performance Results

First we vary the no. of multimedia data flows from 1 to 5 and studied the performance of average delay , jitter and drop. Figures 3 and 4 give the results for delay and drop respectively while figure 5 gives the results of jitter. We can observe that AntHocNetM clearly outperforms AODV in all the 3 cases.

Next we vary the no. of nodes as 10,20,30,40 and 50 and evaluated the delivery ratio and average delay. From Figures .6 and 7, we can see that AntHocNetM gives better results when compared with AODV for all settings, while the performance difference between the algorithms remains more or less constant.

.Figures 8 and 9 shows results of delivery ratio and average delay for increasing the node's pause time (from 10 to 40s) in a random waypoint model. We can see that AntHocNetM clearly outperforms AODV

Figures 10 and 11 show the evolution of the delivery ratio and average delay for increasing node speed (from 10 to 40 m/s) in a random waypoint model. Once again AntHocNetM outperforms AODV.

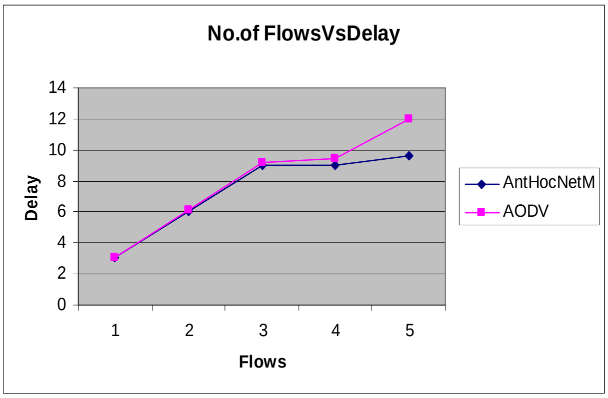

Fig.3 Flow Vs Delay (AODV.AntHocNetM)

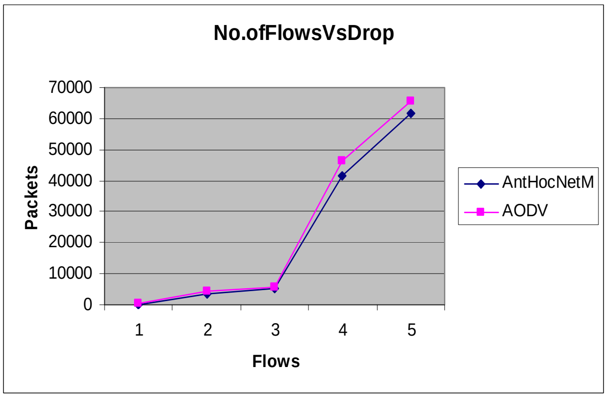

Fig. 4 Flow Vs Drop (AODV/AntHocNetM)

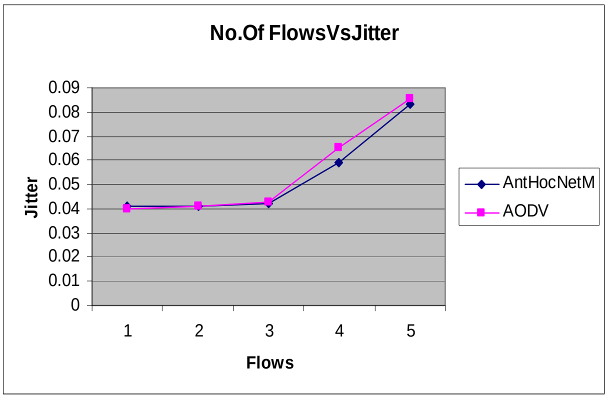

Fig.5 Flow Vs Jitter (AODV/AntHocNetM)

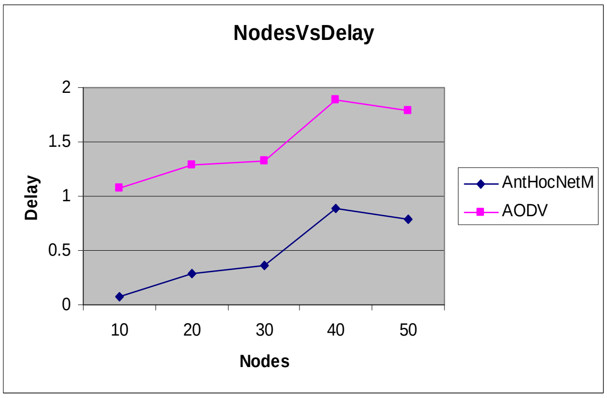

Fig.6 Number Of NodesVsMean Delay

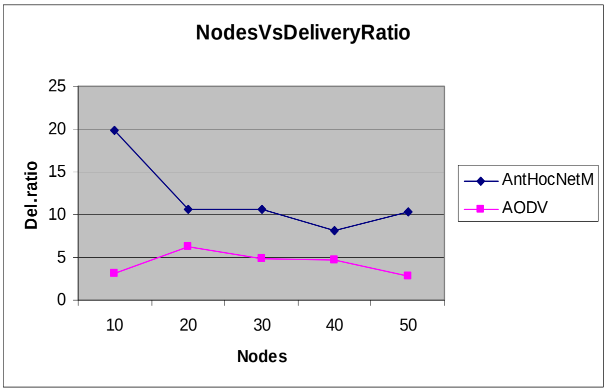

Fig.7 NumberOfNodesVsDelRatio

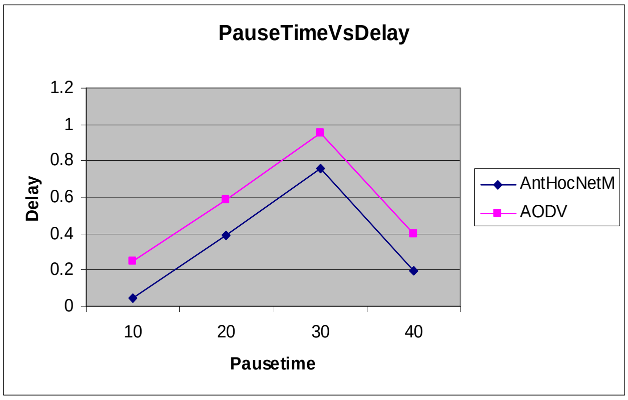

Fig.8 PauseTimeVsDelay

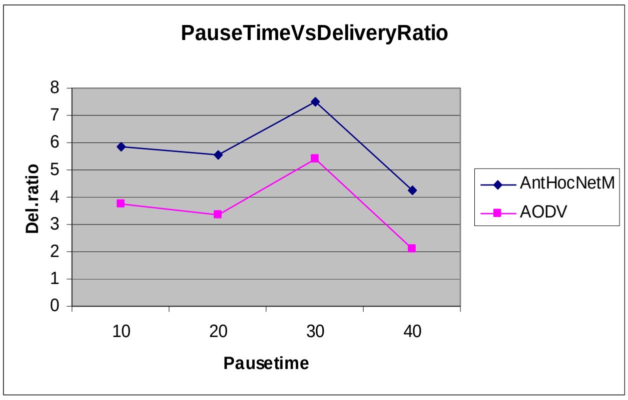

Fig.9 PauseTimeVsDelRatio

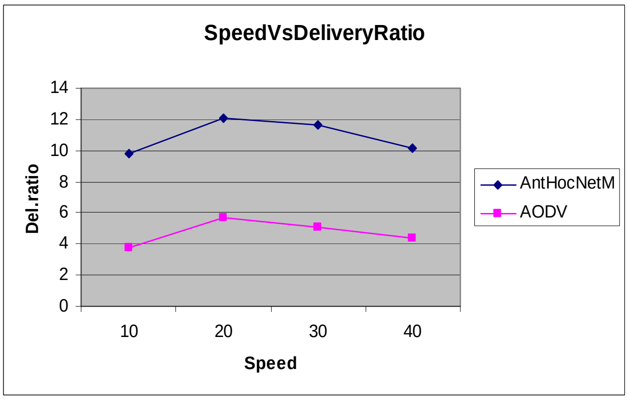

Fig.10 SpeedVsDelRatio

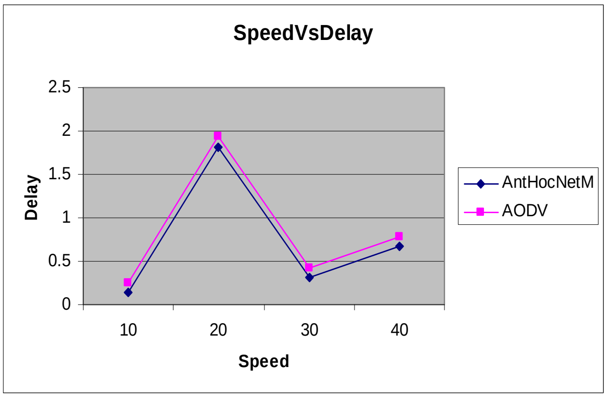

Fig.11 SpeedVsDelay

5. Conclusion

In this paper we have proposed a routing algorithm AntHocNetM, to support multimedia communications in mobile ad hoc networks based on Ant Colony Optimization framework. The algorithm consists of both reactive and proactive components In a reactive path setup phase, multiple paths are built between the source and destination of a data session. The multimedia data are send over the stable, failure-free paths. 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 establishes multiple stable paths between source and destination nodes. Our simulation results show that the performance of AntHocNetM outperforms the standard AODV routing algorithm in terms of end-to-end delay, delivery ratio and jitter.

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] Ahmed Al-Ani "Ant Colony Optimization for Feature Subset Selection" World Academy Of Science, Engineering And Technology Volume 4 February 2005.

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

[9] Philip Hingston "Enumerating Knight's Tours using an Ant Colony Algorithm" IEEE Congress on Evolutionary Computation Sept. 2005.

[10] Osvaldo Gomez "Omicron ACO. A New Ant Colony Optimization Algorithm" CLEI ELECTRONIC JOURNAL AUGUST 2005

[11] Matthijs den Besten "An Ant Colony Optimization Application to the Single Machine Total Weighted Tardiness Problem" Proceedings of ANTS'2000 - From Ant Colonies to Artificial Ants: 2nd International Workshop on Ant Algorithms, 2000.

[12] Panagiotis Papadimitratos et al. " Secure Routing for Mobile Ad hoc Networks" SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002), January 27-31, 2002.

[13] E. A. Gonzalez et al. "Optimal retuning of quarter-wave Ziegler-Nichols PID controllers using sequential ant colony optimization", 7th National Electrical, Electronics, Computer, and Communications Engineering Conference (EECCECON November 30-December 2, 2006.

[14] A. Shmygelska et al. "An improved ant colony optimization algorithm for the 2D HP protein folding problem", in: Proceedings of the Canadian Conference on Artificial Intelligence, LNCS2671, 2003, pp. 400-417.

[15] Ines Alaya "Ant Colony Optimization for Multi-objective Optimization Problems" 19th IEEE International Conference on Tools with Artificial Intelligence, 2007.

M.Sivajothi Born in Tiruppathur, Tamil Nadu state in India in 1965, received the M.Sc in Computer Applications from Alagappa University, Karaikudi in 1998 and M.Phil in Computer Science from Mother Teresa Women's University, Kodaikkanal in 2000. Currently she is doing Ph.D in Computer Science at Mother Teresa Women's University, Kodaikkanal. Now she 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 Computer Science and Engineering, Alagappa University, Karaikudi, 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.