Proposal on Multi agent Ants based Routing Algorithm for Mobile Ad-Hoc Networks

Siva Kumar.D* and Bhuvaneswaran.R.S**

***Department of Electrical and Electronics, Research Scholar, Anna University, Chennai 600 025, India**
***** Ramanujan Computing Center, Asst. Professor, Anna University, Chennai 600 025, India***

Summary:
Single path routing protocol, known as Ad Hoc On-demand Distance Vector, has been widely studied for mobile ad hoc networks. AODV needs a new route discovery whenever a path breaks. Such frequent route discoveries cause route discovery latency. To avoid such inefficiency, in this paper we present Multi agent Ants based Routing Algorithm (MARA), a new algorithm for routing in mobile ad hoc networks. The proposed hybrid protocol reduces route discovery latency and the end-to-end delay by providing high connectivity without requiring much of the scarce network capacity. Multi agent Ants based Routing Algorithm (MARA), is based on ideas from Ant Colony Optimization with Multi agent systems technique. In simulation tests we show that Multi agent Ants based Routing Algorithm (MARA), can outperform AODV, one of the most important current state-of-the-art algorithms, both in terms of end-to-end delay and packet delivery ratio.

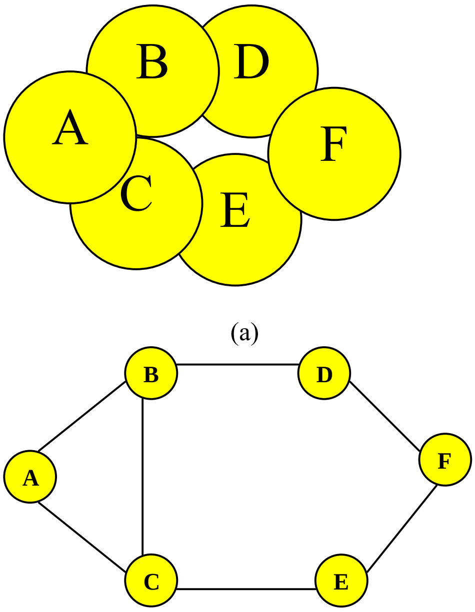
Keywords:
Routing, Mobile Ad hoc Network, Multi agent Ants based Routing Algorithm

1. Introduction

As mobile hosts and wireless networking equipments have become widely available, an entirely new class of applications has been created that wired network infrastructure cannot achieve. These applications include battlefield communications, disaster recovery, and rescue. These applications all rely on a quickly deployable wireless network infrastructure. One type of infrastructure is the ad hoc network, which can be rapidly deployed in a given area. Mobile ad hoc networks are collections of mobile nodes connected by wireless links. If two nodes are not within radio range, all communication between them must pass through one or more intermediate nodes that act as routers. The nodes are free to move, thus the network topology may change dynamically. Therefore, routing protocols must be able to find paths (sequences of intermediate nodes to a destination) quickly in such dynamic conditions. On-demand protocols that initiate routing activities on an on-demand basis have been widely studied because of their low routing overhead. Well-known on-demand protocols are AODV (Ad Hoc On-demand Distance Vector protocol) [1, 6, 16] and DSR [2, 4] (Dynamic Source routing protocol). Although AODV outperforms DSR in many cases [2, 4, 13,], AODV is a single path routing that needs a new route discovery whenever path breaks. Such frequent route discoveries cause route discovery latency. To avoid such inefficiency, several studies such as [3, 12] and [17] have been proposed that extend AODV to compute multiple paths. They send data packets by alternative paths (sequences of intermediate nodes to a destination) without executing a new route discovery, if the primary path breaks. This reduces route discovery latency that contributes to the end-to-end delay of data packets. However, [1] does not perform well by increasing the number of communication sessions and [3, 5] can not find paths even they actually exist when there are two or more common intermediate nodes on the paths. An ad hoc network is a collection of wireless mobile hosts, located in a given region. A transceiver with a certain transmission range is associated with each mobile host. Figure l (a) shows an example of an ad hoc network. Two mobile hosts can communicate directly in this network if both are within the transmission range of each other. When mobile hosts are not within range the assistance of other hosts is used to forward packets. This is called multihopping. An ad hoc network can be modeled as an undirected graph G = (V, E) in which the nodes set V represents the collection of wireless mobile hosts. The link set E represents the collection of any node pairs that can communicate directly. Figure l(b) shows the graph corresponding to Figure l (a).

Manuscript received June 5, 2007
Manuscript revised June 20, 2007

Figure 1 (a) An example for the Adhoc Network
and
(b) The corresponding graph model



(a)

(b)

Current routing protocols for mobile ad hoc networks MANETs suffer from certain inherent shortcomings. On the one side the proactive routing schemes like Destination Sequenced Distance Vector (DSDV) [1] continuously update the routing tables of mobile nodes consuming large portion of the scarce network capacity for exchanging huge chunks of routing table data. This reduces the available capacity of the network for actual data communication. The on-demand routing protocols like Ad Hoc On-Demand Distance Vector and Dynamic Source routing [2] on the other hand launch route discovery, and require the actual communication to be delayed until the route is determined. This may not be suitable for real-time data and multimedia communication applications. Mobile agents similar to ants [7, 8, 9, and 10] can be used for efficient routing in a network and discover the topology, to provide high connectivity at the nodes. However the ant-based algorithms in wireless ad hoc networks have certain drawbacks. In that the nodes depend solely on the ant agents to provide them routes to various destinations in the network. This may not perform well when the network topology is very dynamic and the route lifetime is small. In pure ant-based routing, mobile nodes have to wait to start a communication, till the ants provide them with routes. In some situations it may also happen that the nodes carrying ants suddenly get disconnected with the rest of the network. This may be due to their movement away from all other nodes in the network or they might go into sleep mode or simply turned off. In such situations the amount of ants left for routing are reduced in the network, which leads to ineffective routing. In this paper, we focus on AODV and propose Multi agent Ants based Routing Algorithm (MARA), a new algorithm for routing in mobile ad hoc networks. The MARA hybrid routing protocol is able to reduce the end-to-end delay and route discovery latency by providing high connectivity compared to AODV and ant-based routing schemes. The hybrid scheme also does not overload the available network capacity with control messages like the proactive protocols.

2. Background description of AODV and Ant-based routing protocols

2.1. AODV Routing Protocol

The specific challenges and possible applications of MANETs have made this a very popular research area, and a lot of routing algorithms have been proposed. People traditionally classify these algorithms as either proactive or reactive. In purely proactive protocols (e.g., DSDV [14]) nodes try to maintain at all times routes to all other nodes. This means that they need to keep track of all topology changes, which can become difficult if there are a lot of nodes or if they are very mobile. Therefore, reactive protocols (e.g., AODV or DSR) are in general more scalable (see [1, 2, 5, and 12]). In these protocols, nodes only gather routing information on demand only when they have data for a certain destination they construct a path, and only when the path becomes infeasible they search a new path. In this way they greatly reduce the routing overhead, but they can super from oscillations in performance since they are never prepared for disruptive events. Hybrid algorithms like ZRP have both a proactive and a reactive component, in order to try to combine the best of both worlds. Most of the algorithms are single path: at any time, they use only one path .The rest ant-based routing algorithms were ABC and Ant Net [7, 8, and 10]. Both algorithms follow a similar general strategy. Nodes send ant agents out at regular intervals to randomly chosen destinations. The main aim of the ants is to sample the paths, assign a quality to them, and use this information to update the routing tables in the nodes they pass. These routing tables contain an entry for each destination and each neighbor, indicating the goodness of going over this neighbor on the way to the destination. This goodness value is called pheromone. This pheromone information is used for the routing of both ants and data packets. All packets are routed stochastically, choosing with a higher probability those links with higher pheromone values. If enough ants are sent to the different destinations, nodes keep up-to-date information about the best paths, and automatically adapt their data load spreading to this. Ant-based routing algorithms have a number of properties which are desirable in MANETs: they are highly adaptive to network changes, use active path sampling, are robust to agent failures, provide multipath routing, and take care of data load spreading. However, the fact that they crucially rely on repeated path sampling can cause significant overhead if not dealt with carefully. There have been a number of attempts to design ant-based routing algorithms for MANETs. Examples are ARA [10] and PERA [11]. However, these algorithms loose much of the proactive sampling and exploratory behavior of the original ant-based algorithms in their attempt to limit the overhead caused by the ants.

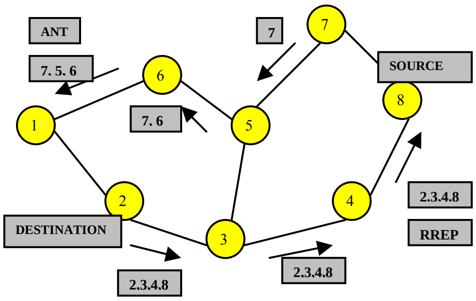
2.2. Ants Based Routing Algorithm

Ants Based Routing Algorithm is a hybrid multipath algorithm. When a data session is started at node s with destination d, s checks whether it has up-to-date routing information for d. If not, it reactively sends out ant-like agents, called reactive forward ants, to look for paths to d. These ants gather information about the quality of the path they followed, and at their arrival in d they become backward ants, which trace back the path and update routing tables. The routing table T^{i} in node i contains for each destination d and each possible next hop n a value T^{i}_{nd} € IR.T^{i}_{nd} is an estimate of the goodness of the path over n to d, which we call pheromone. In this way, pheromone tables in different nodes indicate multiple paths between s and d, and data packets can be routed from node to node as data grams. They are stochastically spread over the paths: in each node they select the next hop with a probability proportional to its pheromone value. Once paths are set up and the data session is running, s starts to send proactive forward ants to d. These ants follow the pheromone values similarly to data packets. In this way they can monitor the quality of the paths in use. Moreover, they have a small probability of being broadcasted, so that they can also explore to their neighbors such that these can update.

2.3. MARA: Multi agent Ant based routing Algorithm

Multi agent Ant based routing Algorithm forms a hybrid of both ant based routing and Multi agent systems technique to overcome some of their inherent drawbacks. The hybrid technique enhances the node connectivity and decreases the end-to-end delay and route discovery latency. Route establishment in conventional ant based routing techniques is dependant on the ants visiting the node and providing it with routes. If a node wishes to send data packets to a destination for which it does not have a fresh enough route, it will have to keep the data packets in its send buffer till an ant arrives and provides it with a route to that destination. Also, in ant routing algorithms implemented so far there is no local connectivity maintenance as in AODV. Hence when a number of data packets being dropped. AODV on the other hand takes too much time for connection establishment due to the delay in the route discovery process whereas in ant based routing if a node has a route to a destination it just starts sending the data packets without any delay. This long delay in AODV before the actual connection is established may not be applicable in real-time communication applications.

Figure 2 Propagation of route relay and traversal of ant packet in MARA routing protocol



In Ant-AODV ant agents work independently and provide routes to the nodes as shown in fig. 2. The nodes also have capability of launching on-demand route discovery (fig. 2) to find routes

to destinations for which they do not have a fresh enough route entry. The use of ants with AODV increases the node connectivity (the number of destinations for which a node has un-expired routes), which in turn reduces the amount of route discoveries. Even if a node launches a RREQ (for a destination it does not have a fresh enough route), the probability of its receiving replies quickly (as compared to AODV) from nearby nodes is high due to the increased connectivity of all the nodes resulting in reduced route discovery latency. Lastly, as ant agents update the routes continuously, a source node can switch from a longer (and stale) route to a newer and shorter route provided by the ants. This leads to a considerable decrease in the average end-to end delay as compared to both AODV and ant-based routing. Ant-AODV uses route error messages (RERR) to inform upstream nodes of a local link failure similar to AODV. Routing table in Ant-AODV is common to both ants and AODV. Frequent HELLO broadcasts are used to maintain a neighbor table. This table is used to select a randomly chosen next hop (avoiding the previously visited node) from the list of neighbors by the ant agents.

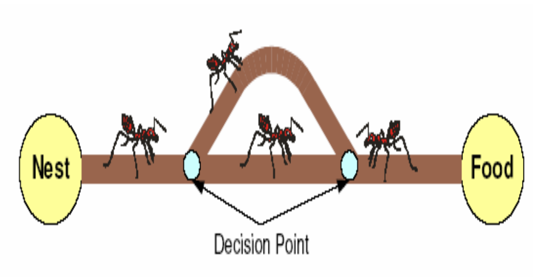
3. Ant Algorithms

Ant algorithms are a subset of swarm intelligence which models the behavior of insect swarms to solve complex tasks by cooperation. They are multi-agent systems where agents show the behavior of individual ants. See [7, 8, and 10] for more information.

3.1 Basic ant algorithm

The basic idea of the ant algorithm is taken from the food searching behavior of real ants. When ants search for food, they start from their nest and walk toward the food. When an ant reaches an intersection, it has to decide which branch to take next. While walking ants deposit pheromone1 which marks the selected route. The concentration of pheromone on a certain path is an indication of its usage. Over time the concentration of pheromone decreases due to diffusion effects. Figure 3 shows a scenario with two routes from the nest to the food. At the intersection the first ants randomly select a branch. Since the lower route is shorter than the upper one, the ants, which take this path, will reach the food place first. On their way back to the nest, the ants again have to select a path. After a while the pheromone concentration on the shorter path will be higher than on the longer path, because the ants using the shorter path will increase the pheromone concentration faster. Thus, eventually all ants will only use this path.

Figure 3.All ants take the shortest path after an initial searching time



.

This behavior of the ants can be used to find the shortest path in networks. Especially the dynamic component of this method provides for a high degree of adaptation to changes in mobile ad-hoc network topology, since in these networks the existence of links is not guaranteed and link changes occur frequently.

3.2 A simple Ant algorithm

Let G = (V, E) be a connected graph with n = │V│nodes. The simple ant colony optimization meta-heuristic can be used to find the shortest path between a source node VS and a destination node VD on the graph G. The number of nodes on the path gives the path length. A variable $\varphi_{i,j}$ (artificial pheromone), which is modified by the ants when they visit the node is associated with an edge $e(i,j) \in E$ of the graph connecting the nodes V_i andV_j. The pheromone concentration i;j is an indication of the usage of this edge. Initially $\varphi_{i,j}$ j is constant for each edge e (i; j). An ant located in node V_i uses pheromone $\alpha_{i;j}$ of node $v_j \in N_i$ to compute the probability of node vj being the next hop. Ni is the set of one-step neighbors of node V_i. The transition probabilities $P_{i;j}$ of a node Vi, i.e. the probability that the ant selects node vj after it has visited $V_{i,}$ are defined as follows

$$p_{i,j} = \begin{cases} \frac{\varphi_{i,j}}{\sum_{j \in N_i} \varphi_{i,j}} & \text{if } j \in N_i \\ 0 & \text{if } j \notin N_i \end{cases}, \qquad \sum_{j \in N_i} p_{i,j} = 1, \qquad i \in [1, N].$$

---------- (1)

During the route finding process, ants deposit pheromone on the edges. In the simplest version of the algorithm, the ants deposit a constant amount $\Delta\varphi$ of pheromone, i.e. the amount of pheromone of the edge e ($v_{i;}$ $v_{j)}$ when the ant is moving from node v_i to node v_j
is changed as follows:

$$\varphi_{i,j} := \varphi_{i,j} + \Delta\varphi$$

----------- (2)

Like real pheromone the artificial pheromone concentration decreases with time. In the simple ant algorithm this is described by:

$$\varphi_{i,j}(t+\tau) := (1-q) \cdot \varphi_{i,j}(t), \qquad q \in (0,1]$$

----------- (3)

3.3. Why ant algorithms are suitable for ad-hoc networks

The simple ant algorithm shown in the previous section illustrates different reasons why this kind of algorithms could perform well in mobile multi-hop ad-hoc networks. We discuss some by relating them to important properties of mobile ad-hoc networks.

Dynamic topology: This property is responsible for the poor performance of many 'classical' routing algorithms in mobile multi-hop ad-hoc networks. The ant algorithm is based on autonomous agent systems imitating individual ants. This allows a high adaptation to the current topology of the network.

Local work: In contrast to other routing approaches, the ant algorithm is based only on local information, i.e. no routing tables or other information blocks have to be transmitted to other nodes of the network.

Link quality: It is possible to integrate the connection/link quality into the computation of the pheromone concentration, especially into the evaporation process. This will improve the decision process with respect to the link quality. It is important to note that the approach can be modified so that nodes can also manipulate the pheromone concentration independent of the ants, e.g. if a node detects a change of the link quality.

Support for multi-path: Each node has a routing table with entries for all its neighbors, which also contain the pheromone concentration. The decision rule for selection of the next node is based on the pheromone concentration at the current node, which is provided for each possible link. Thus, the approach supports multi-path routing.

4. The Ant Routing Algorithm for MANETs

In this section we discuss the adaptation of the simple ant algorithm for mobile multi-hop ad-hoc networks and describe the Multi agent Abased Routing Algorithm (MARA). The routing algorithm is similar to [22] many other routing approaches and consists of three phases.

4.1. Route Discovery Phase

New routes are created in the route discovery phase. The creation of new routes requires the use of a forward ant (FANT) and a backward ant (BANT). A FANT is an agent, which establishes the pheromone track back to the source node. In analogous, a BANT establishes the pheromone track back to its origin, namely the destination node. The FANT is a small packet with a unique sequence number. Nodes are able to distinguish duplicate packets on the basis of the sequence number and the source address. A node, which receives a FANT for the first time, creates a record in its routing table. An entry in the routing table is a triple (destination address, next hop, pheromone value). The node interprets the source address of the FANT as destination address; the address of the previous node as the next hop, and computes the pheromone value depending on the number of hops it took the FANT to reach the node. The node then relays the FANT to its neighbors. Duplicate FANTs are identified through the unique sequence number, and are removed. The destination node extracts the information of the FANT, creates a BANT and returns it to the source node. The BANT's task is similar to that of the FANT, i.e. to establish a track to this node. When the sender receives the BANT from the destination node, the path is established and data packets can be sent. The forward ant only creates one pheromone track to the source node in node 6, but two tracks in node 5, via node 3 and node 4. It only creates one pheromone track to the destination node V_D in node 5 and two tracks in node 6. Thus, MARA also supports multi-path routing.

Figure 4 Demonstrates the route discovery phase of MARA

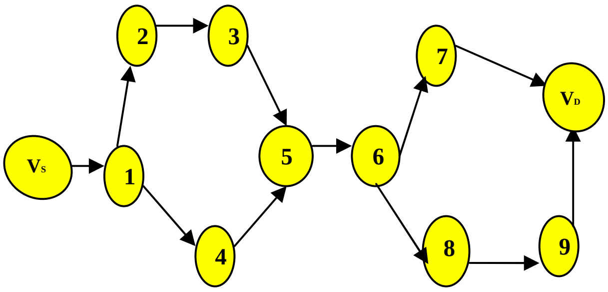


Figure 4 a) shows the establishment of the pheromone track back to the source node vs.

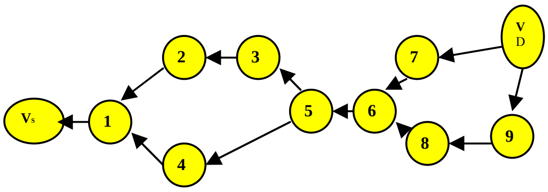


Figure 4 b) depicts analogous situation for the backward ant.

4.2 Route Maintenance

The second phase of the routing algorithm is called route maintenance. This phase is responsible for the maintenance of the routes during the communication. ARA does not need any special packets for that purpose. Once the FANT and BANT have established the pheromone tracks for the source and destination nodes regular data packets are used to maintain the path. As in biological systems, established paths do not keep their initial pheromone values forever. When a node vi relays a data packet to destination v_D to a neighbor node v_j, it increases the pheromone value of the entry (v_D, v_j, φ) by Δ_φ, i.e. this path to the destination is strengthened by the data packet. Likewise, the next hop v_j increases the pheromone value of the entry (v_S, v_i, φ) by Δ_φ, i.e. the backward path to the source node is also strengthened. The evaporation process of the real pheromone is modeled by decreasing the pheromone values according to equation 3.

4.3. Route Failure Handling

The third and last phase of MARA handles routing failures, which are especially caused by node mobility and are therefore very common in mobile ad-hoc networks. The current implementation of ARA assumes IEEE 802.11 on the MAC layer. This enables MARA to recognize a route failure through a missing acknowledgement on the MAC layer. If a node receives a ROUTE_ERROR message for a certain link, it first deactivates this link by setting the pheromone value to 0. Subsequently, the node searches for an alternative link in its routing table. If there is another route to the destination it will send the packet via this path. Otherwise, the node informs its neighbors, hoping that they can forward the packet to the destination. Either the packet can be transported to the destination node or the backtracking continues to the source node. If the packet does not reach the destination, the source node has to initiate a new route discovery process.

4.4. Overhead of MARA

The expected overhead of MARA is very small, because there are no routing tables to be exchanged between the nodes. Unlike other routing algorithms, the FANT and BANT packets do not transmit much routing information. Only a unique sequence number is transmitted in the routing packets. Most route maintenance is performed through data packets. MARA only needs the information in the IP header of the data packets.

5. Performance Evaluation

We evaluate Performance of the proposed algorithm using simulations and compare them with AODV [12]. The algorithm is evaluated in terms of Average end-to-end delay per packet and delivery ratio (i.e. the fraction of successfully delivered data packets). In 5.1 we describe the simulation environment and the test scenarios, and in 5.2 we show and discuss the results.

5.1 Simulation Environment

As simulation software we used GloMoSim is a scalable simulation environment for wireless and wired networks systems developed initially at UCLA Computing Laboratory [19]. It is designed using the parallel discrete-event simulation capability provided by a C-based parallel simulation language, Parsec [20&21]. GloMoSim currently supports protocols for purely wireless networks. All our simulation scenarios are derived from the base scenario used in [10], which is an important reference. In this base scenario 50 nodes are randomly placed in an area of 1500 ×300m$^{2.}$ The area is rectangular in order to have more long paths. Within this area, the nodes move according to the random waypoint model. Each node randomly chooses a destination point and a speed, and moves to this point with the chosen speed. After that it stops for a certain pause time and then chooses a new destination and speed. The maximum speed in the scenario is 20m/s and the pause time is 30 seconds. The total length of the simulation is 900 seconds. Data traffic is generated by 20 constant bit rate (CBR) sources sending one 64-byte packet per second. Each source starts sending at a random time between 0 and 180 seconds after the start of the simulation, and keeps sending until the end. At the physical layer we use a two-ray signal propagation model. The transmission range is 300 meters, and the data rate is 2Mbit/s. At the MAC layer we use the popular 802.11 DCF protocol. The different test scenarios used below were derived from the base scenario by changing some of the parameters. In particular, we varied the pause time, the area dimensions and the number of nodes. For each new scenario, 5 different problems were created, by choosing different initial placements of the nodes a different movement patterns. The reported results are averaged over 5 different runs (to account for stochastic elements, both in the algorithms and in the physical and MAC layers) on each of the problems.

5.2 Simulation Results

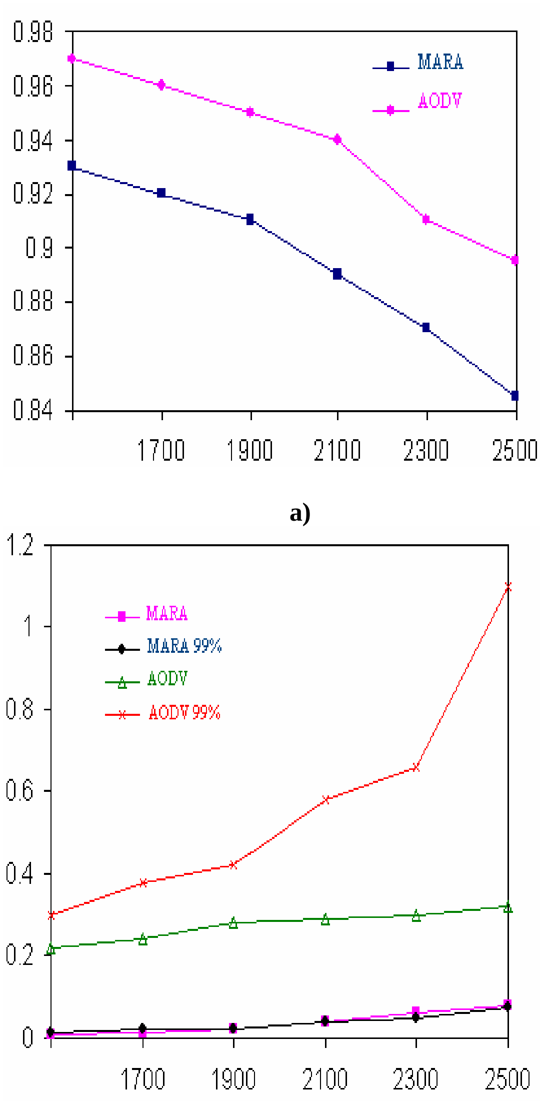
In a first set of experiments we progressively extended the long side of the simulation area. This has a double effect: paths become longer and the network becomes sparser. The results are shown in figure 5. In the base scenario, MARA has a better delivery ratio than AODV, but a higher average delay. For the longer areas, the difference in delivery ratio becomes bigger, and AODV also looses its advantage in delay. If we take a look at the 99th percentile of the delay, we can see that the decrease in performance of AODV is mainly due to a small number of packets with very high delay. This means that AODV delivers packets with a very high delay jitter, a crucial problem in terms of quality of service (QoS). Removing these packets with very high delay could reduce the jitter, but that would mean an even worse delivery ratio for AODV.

Figure 5 (a) The delivery ratio (the fraction of sent packets which actually arrives at their destination

and

5(b) The average and 99th percentile of the delay per packet.

(on x-axis the long edge of the area starting from the base scenario of 1500x300 m^{2,} and ending at 2500x300 m^{2}.)

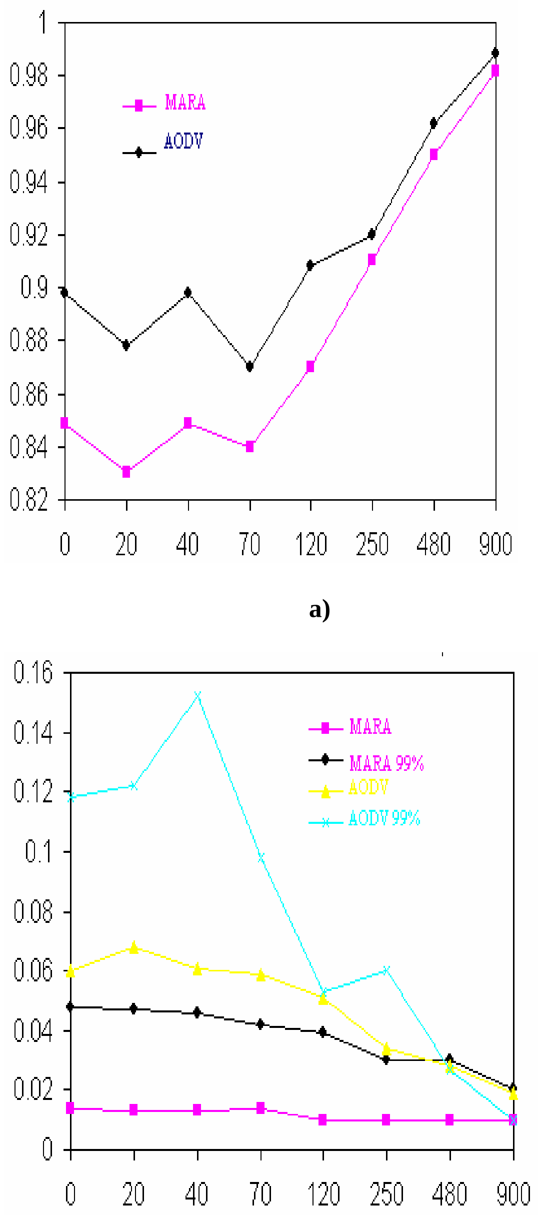


a)

b)

Next we changed the mobility of the nodes, varying the pause time between 0 seconds (all nodes move constantly) and 900 seconds (all nodes are static). The area dimensions were kept on 2500 × 300 m^2, like at the end of the previous experiment (results for 1500× 300m^2 was similar but less pronounced). In figure 6 we can see a similar trend as in the previous experiment. For easy situations (long pause times, hardly any mobility), MARA has a higher delivery ratio, while AODV has lower delay. As the environment becomes more difficult (high mobility), the difference in delivery ratio becomes bigger, while the average delay MARA becomes better than that of AODV. Again, the 99th percentile of AODV shows that this algorithm delivers some packets with a very high delay. Also MARA has some packets with a high delay (since the average is above the 99th percentile), but this number is less than 1% of the packets. In a last experiment we increased the scale of the problem. Starting from 50 nodes in a 1500×500m$_2$ area, we multiply both terrain edges by a scaling factor and the number of nodes by the square of this factor, up to 200 nodes in a 3000×1000m^2 area.

Figure 6(a) The delivery ratio.
and
6(b) The average and 99th percentile of the delay.
(on the x-axis the node pause time in seconds)

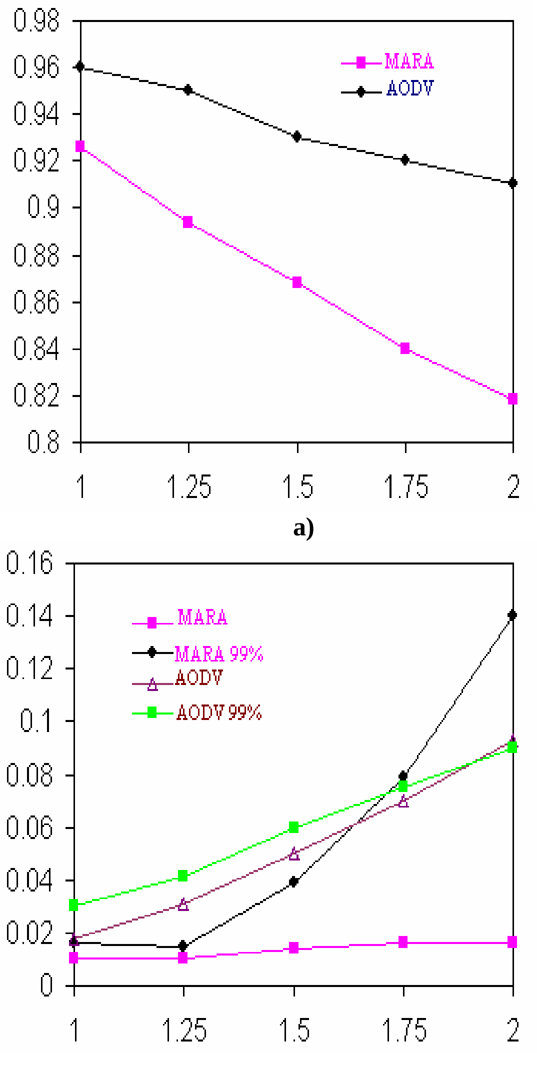


a)

b)

The results, presented in figure 3, show again the same trend: as the problem gets more difficult, the advantage of MARA in terms of delivery ratio increases, while the advantage of AODV in terms of average delay becomes a disadvantage. Again this is due to a number of packets with a very high delay. The experiments described above show that MARA has some clear advantages over AODV.First of all, MARA gave a better delivery ratio than AODV in all scenarios. The construction of multiple paths at route setup, and the continuous search for new paths with proactive ants ensures that there are often alternative paths available in case of route failures, resulting in less packet loss. Second, MARA has a higher average delay than AODV for the simpler scenarios, but a lower average delay for the more difficult ones. The average delay of AODV increases sharply in each of the difficult scenarios, and the 99th percentile figures indicate that this is mainly due to a fraction of packets, which is delivered with an abnormally high delay. Moreover, the 95th percentile (not shown in the figures) is usually lower for AODV than for line with the multipath nature of MAR. Since it uses different paths simultaneously, not all packets are sent over the shortest path, and so the average delay will be slightly higher. On the other hand, since AODV relies on just one path, delays can become very bad when this path becomes inefficient or invalid. This is especially likely to happen in difficult scenarios, with longer paths, lower node density or higher mobility, rather than in the dense and relatively easy base scenario. Delivering packets with low variability and low maximum delay is an important factor in QoS routing.

Figure 7(a) The delivery ratio
and
7(b) The average and 99 th percentile of the delay.
(on the x-axis the scaling factor for the problem)



a)

b)

6 Conclusions and future work

We have presented Multi agent Ants based Routing Algorithm (MARA), a new algorithm for routing in mobile ad hoc networks. It is a hybrid algorithm, combining reactive route setup with proactive route probing and exploration. In simulation experiments we show that MARA can outperform AODV in terms of delivery ratio and average delay, especially in difficult scenarios. Also in terms of delay jitter, MARA shows better results. In future work we want to improve the exploratory working of proactive ants. By extending the concept of pheromone diffusion, more information about possible path improvements will be available in the nodes, and this information can guide proactive ants. This should lead to better results with less overhead. Further investigations will include experiments with high network load and multimedia data. Also, we would like to try out a virtual circuit based approach. This could result in better control over paths, so that data delivery can be made more reliable.

Acknowledgment

The authors would like to express their cordial thanks to Dr.S.K. Srivatsa, Former Professor, Department of Electronics, M.I.T. Campus, Anna University, India for his valuable advice.

References:

[1] S. Das, C. Perkins, E. Royer, “Ad hoc on demand distance vector (AODV) routing”,

Internet Draft, draft-ietf-manetaodv-11.txt, work in progress, 2002.

[2] J. Broch, D. Johnson, and D. Maltz, "The Dynamic Source Routing Protocol for Mobile Ad hoc Networks," http://www.ietf.org/internet-drafts/draft-ietfmanet- dsr-01.txt, Dec 1998, IETF Internet Draft.

[3] A. Nasipuri and S. R. Das, "On-Demand Multipath Routing for Mobile Ad Hoc Networks," Proceedings of the 8th Int. Conf. On Computer Communications and Networks (IC3N), Boston, October 1999.

[4] V. D. Park and M. S. Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks," Proceedings of IEEE INFOCOM'97 Conf., April 1997.

[5] M. Abolhasan, T. Wysocki, E. Dutkiewicz, "A review of routing protocols for mobile ad hoc networks", Journal on Ad Hoc Networks, Elsevier Computer Science, 2: 1–22, 2004.

[6] S. Das, C. Perkins, E. Royer, "Ad hoc on demand distance vector (AODV) routing", Internet Draft, draft-ietf-manetaodv-11.txt, works in progress, 2002.

[7] Marco Dorigo and Gianni Di Caro. The ant colony optimization meta-heuristic. In David Corne, Marco Dorigo, and Fred Glover, editors, New Ideas inOptimization, pages 11. 32. McGraw-Hill, London, 1999.

[8] G. Di Caro and M. Dorigo, "AntNet: distributed stigmergetic control for communications networks". Journal on Artificial Intelligence Research, 9: 317-365, 1998.

[9] R. Schoonderwoerd, O. Holland, J. Bruten, L. Rothkrantz, "Ant-based load balancing in telecommunication networks", Adaptive Behavior, vol. 5, pp. 169-207, 1996.

[10] M. Günes, U. Sorges, I. Bouazizi, "ARA - the ant-colony based routing algorithm for manets", ICPP Proc. of the 2002 Workshop on Ad Hoc Networks, pp. 79-85, 2002.

[11] J.S. Baras, H. Mehta, "A probabilistic emergent routing algorithm for mobile ad hoc Networks", WiOpt03 Proc. of Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, 2003.

[12] S. Marwaha, C.K. Tham, D. Srinavasan, "Mobile Agents based Routing Protocol for Mobile Ad hoc Networks", IEEE Proc. of the Global Telecommunications Conference – GLOBECOM'02, Taipei, Taiwan, 2002.

[13] Charles E. Perkins. Ad Hoc Networking. Addison-Wesley, 2001. ISBN 0-201-30976-9.

[14] C. E. Perkins and P. Bhagvat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. Computer Communications Rev., pages 234. 244, October 1994.

[15] C. E. Perkins, E. M. Royer, and S. R. Das. Ad hoc on-demand distance vector (AODV).

[16] Chai-Keong Toh. Ad hoc mobile wireless networks: protocols and systems. Prentice Hall, 2002. ISBN: 0-13-007817-4.

[17]. J. Broch, D. Maltz, D. Johnson, Y.-C. Hu and J. Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In Proc. of 4th Annual ACM/IEEE Int. Conf. on Mobile Computing and Networking, 1998.

[18]. D. B. Johnson and D. A. Maltz. Mobile Computing, chapter Dynamic Source Routing in Ad Hoc Wireless Networks, pages 153{181. Kluwer, 1996.

[19] L. Bajaj, M. Takai, R. Ahuja, K. Tang, network simulation environment. Technical Report 990027, UCLA Computer Science Department, May 1999.

[20] R. A. Meyer. PARSEC User Manual. UCLA Parralel Computing Laboratory, ttp://pcl.cs.ucla.edu.

[21] http://pcl.cs.ucla.edu/projects/parsec/samples/ and

http://pcl.cs.ucla.edu/projects/parsec/

[22] MasutGunes and Otto Spaniol, Routing algorithm for mobile multi hop Adhoc Networks, International workshop NGNT

Biographical notes

D. Siva Kumar received the B.E. degree in Electronics and Communication Engineering from University of Madras, India, in 1995.He received the M.E. degree in Process control and instrumentation from Annamalai University, India, in 2002. Currently he is Research scholar in the Department of Electrical and Electronics Engineering in Anna University, India. His research interests include Mobile Computing, Adhoc networks, Distributed systems and Multicast Distribution.

Raghuvel S. Bhuvaneswaran received Master of Technology in Computer Science and Engineering from Pondicherry University, India, in 1996 and Ph.D in Computer Science and Engineering from Anna University, India, in 2003.Presently, he is with Anna University as Assistant Professor and He is on lien to Nagoya Institute of Technology from 2004 to 2006 under JSPS fellowship. His research interests include distributed systems, wireless networks and fault tolerant systems.