

Intelligent Agents for Routing on Mobile Ad-Hoc Networks

Y. Zhou
Dalhousie University
yzhou@cs.dal.ca

A. N. Zincir-Heywood
Dalhousie University
zincir@cs.dal.ca

Abstract

This paper introduces a new agent-based ad-hoc network routing protocol: Mobile Agent Routing protocol (MAR). Mobile agents are simple packets that carry data and explore the network to collect routing information. They communicate with one another to exchange this routing information. By mobile agent exploration, the intelligence is put across the network, which enables the routing to be distributed and adaptive. Comparisons of MAR with another state-of-the-art ad-hoc routing protocol: Destination Sequenced Distance Vector (DSDV) show that MAR performs significantly better than DSDV within 95% confidence interval at speeds greater than 0.1 m/s in terms of delay and the number of received packets.

1. Introduction

Mobile ad-hoc networks developed very quickly and have been the focus of a lot of research recently. A mobile ad-hoc network is a connection of many wireless nodes with short transmission range that communicate with one another generally by multi-hop routes [9]. In wireless ad-hoc networks, every node also acts as a router, so the traffic is distributed across the network instead of on several privileged central nodes as in Mobile IP. In addition, because of the frequent movement of mobile nodes, the topology of an ad-hoc network is generally dynamic. Thus, due to the dynamic multi-hop communication, limited memory and bandwidth, it is critical to develop new routing protocols for mobile ad-hoc networks distinct from the traditional routing protocols.

There are already some mobile ad-hoc routing protocols developed such as Destination Sequenced Distance Vector (DSDV), Ad-Hoc On-Demand Distance Vector (AODV) and Dynamic Source Routing (DSR) [1-2, 8]. They all have their advantages and disadvantages. DSDV tends to provide low delay at the cost of periodic update process overhead, while the others endeavor to reduce excessive overhead at the cost of introducing initial

delay [6]. Furthermore, with contemporary networks becoming larger and more heterogeneous, research on more distributed and intelligent ad-hoc routing protocols becomes critical [4]. That is why mobile agents have become an exciting and promising research area for ad-hoc networks.

This paper presents mobile agents, which are simple packets that explore the network and collect data to be used for routing. They communicate with one another indirectly to exchange routing information (write and read data to /from routing tables) as ants exchange food information with each other. By using mobile agent exploration, we put the intelligence across the network and make the routing distributed. Furthermore, in mobile agent based routing protocols the overhead is proportional to the number of agents [4]. This in return reduces the overhead compared with traditional routing protocols that use frequent updates. In this paper, our objective is to design and implement a naive Mobile Agent Routing protocol (MAR) for ad-hoc networks.

The rest of the paper is organized as follows: Section 2 discusses related research on routing protocols for ad-hoc networks and the usage of mobile agent technology under this context. The proposed approach, MAR, is introduced in section 3. Section 4 presents simulation results. Finally, conclusions are drawn and future work is discussed in section 5.

2. Ad-hoc routing protocols

Current ad-hoc routing protocols can be divided into two groups on the basis of when the route is computed: pre-computed routing protocols and on-demand routing protocols [7]. In pre-computed routing protocols, routes to all destinations are computed and maintained by a periodic update process. In on-demand routing protocols, on the other hand, the route to a specific destination is computed only when necessary, e.g., data packets demand a route to be sent to a destination.

On-demand routing protocols can scale well to large networks. First, because the route is computed only when necessary, there is no permanent entry in the routing table. So generally, there are far fewer

entries in the routing table at a time compared with pre-computed protocols. This can save scarce memory. Second, because there is no periodic update process, there are fewer overheads. However, it also has some disadvantages. On-demand routing starts to compute routes when they are required. This in return introduces initial delay to find necessary routes, which degrades the overall performance, especially for interactive applications. Pre-computed protocols, on the other hand, generally are suitable for interactive applications because they do not have initial delay as on-demand protocols [2]. Pre-computed protocols have their own disadvantages as well. They consume more bandwidth resource by the frequent update process to exchange routing information. Moreover, they do not adapt fast enough to topology changes.

There have already been many experiments performed for the comparison among different mobile ad-hoc network routing protocols. In [1], the ad-hoc network routing protocols are divided into proactive and reactive, namely pre-computed and on-demand as described above. Given this, DSDV, is a proactive protocol, whereas AODV and DSR are reactive protocols. Experiments show that reactive protocols perform better than the proactive protocols specifically under quite dynamic environments [1]. However, the drawback of reactive protocols is the much longer delay they have because of the initial delay they introduce. In the Disaster scenario (dynamic environment) experiment in [1], the average delay of DSDV is 0.196 seconds, while those of AODV and DSR are 0.988 seconds and 1.187 seconds respectively. This kind of long delay may not be acceptable in interactive applications. In Bluetooth type of environments, it is demonstrated that proactive routing protocols are better than reactive routing protocols because of the characteristics of small delay, quick adaptation and limited mobility [2]. Thus, there is a fine balance between overhead reduction and delay reduction in mobile ad-hoc network routing. A proactive routing protocol tends to have relatively small delay but introduces excessive overhead, and therefore decreases the network scalability. On the other hand, a reactive routing protocol has fewer overheads but tends to have too long a delay to be suitable for interactive networks.

In recent research, the “rumor routing” scheme is introduced as an attempt to solve the above problem [3]. In “rumor routing”, mobile agents explore the network to create routes to any event they have witnessed. Events are described as abstractions to identify anything happening on the network such as the node’s processing capability or sensor readings. Each node has an event table. When a node finds an event from its sensor readings, it adds this event with

distance zero in its event table. It then probabilistically generates and sends an agent, and the probability is defined by some strategy. Once the agent is created, it walks across the network and propagates the event to other nodes. The distance metric value increases by one at each hop the agent takes. The agent walks for some predefined time, and then dies. This kind of propagation is similar to the way that ants, use *stigmergy* on the path they follow [5].

Another form of mobile agents is introduced in [4]. This kind of mobile agent is characterized by small size, longevity and mutual invisibility. However, by cooperating indirectly, they enable the network to adapt to the changes happening. The authors of this paper believe that routing protocols with intelligent mobile agents can adapt to the changes on the network more easily since there are always some mobile agents exploring the network. Furthermore, because the agent walks to only one specific node at a time instead of flooding the network with update packets, the overhead in agent based routing is proportional to the number of agents. This in return reduces the overhead compared to the traditional proactive protocols. This way, agent based routing increases the scalability of the network. In addition, by using mobile agents exploring the network to compute routes actively, the delay can be decreased as well. Thus, mobile agent technology can likely be used to solve the problems mentioned above and improve ad-hoc network routing. Motivated by this promising perspective, and based on the study on mobile agents, our objective is to develop a relatively simple, low overhead and adaptive routing protocol for mobile ad-hoc networks.

3. Proposed approach – MAR

Our approach is based on utilizing mobile agents for rumor routing in an attempt to solve the aforementioned problems. To this end, a routing algorithm, which is based on sharing local information to find paths, is employed. By local information, we mean the fact that every node on the network only knows about its neighbors to start with.

In a network based on mobile agents, a mobile agent walks from node to node and collects information during its travels. In other words, the mobile agent explores the network and collects information for routing. The system performance relies on the cooperation of all mobile agents in the network. Agents in a population do not communicate with each other directly; instead, they leave some

information in every node they visit, and then move on, called *stigmergy*.

Our approach (MAR) enables the mobile agent not only to leave information on nodes it has visited, but also to read information left by other mobile agents. Specifically, when an agent arrives at a new node, it first updates the routing table of this node with the information it has collected. Then, it reads routing information from the local routing table. Therefore, mobile agents with this characteristic, like ants, leave information behind on nodes they have visited that other ants can use later. We call this “thorough stigmergy”. In this way, all mobile agents co-operate with others more efficiently and more actively.

In Figure 1, a mobile agent, which carries information about node *D* and node *E*, arrives at node *A*. Thus, this mobile agent has the information of *D* and *E*, whose distances from *A* are 1 and 2, respectively. In node *A*, suppose that the routing table already has information about node *B* and node *C*, whose distances are 1 and 2 respectively. Once the mobile agent leaves its information on node *A*, the number of routing entries in the routing table of node *A* increases from 2 to 4 (*D*, *E*, *B* and *C*). On the other hand, after the mobile agent reads information from the routing table of node *A*, it also acquires the information about node *B* and node *C*, even though it has not visited these nodes yet.

Thus, we designed a simple structure for agents in MAR. The update operation for the routing information in a mobile agent is based on the agent structure. In order to minimize the network overhead, our MAR agent has a simple structure, figure 2.

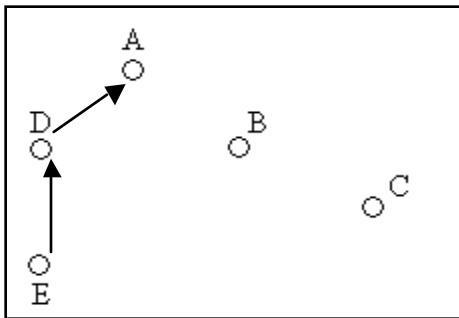


Figure 1. A mobile ad-hoc network

Thus, we designed a simple structure for agents in MAR. The update operation for the routing information in a mobile agent is based on the agent structure. In order to minimize the network overhead, our MAR agent has a simple structure, figure 2.

In this structure, “Type” is used to differentiate these agents from other packets. *ID* is used as a unique id number of every agent. *Last_node* stores

the previous node just visited. *Hist_size* defines how big the *Info_map* is. *Info_map* is the container that stores all gathered routing information. Every time the agent arrives at a node, it updates the *Info_map* based on its gathered routing information about the network. In *Info_map*, *Age* records the sequence of the nodes visited. The larger the sequence value is, the older the information is.

Furthermore, we adopted the “oldest node” strategy. In other words, *Info_map* is similar to a sliding window concept. When an agent selects its next hop, it will select a neighbor of the current node it visited where the neighbor does not exist in its *Info_map*. This means that either the neighbor has never been visited or it has been deleted from the history. If all the neighbors happen to be in its history, which means they have all been visited before, the agent will select the earliest visited node (oldest *Age*) from its history.

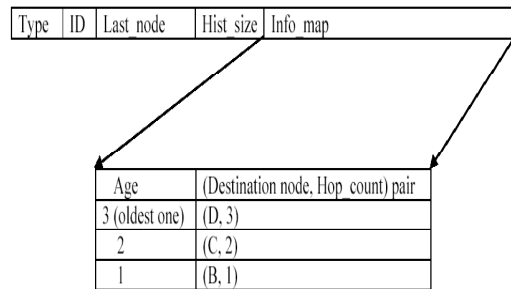


Figure 2. Structure of a routing agent in MAR

Since the agent size should be small to minimize overhead, we set a history size threshold for every agent. Once an agent arrives at a new node, if the history size does not reach the threshold value, the new node entry will be inserted to the history. However if the history size has already reached the threshold value, before the new node entry is inserted, the oldest node, which is the entry with largest *Age* value, will be deleted from the history in order to open up space for the new one. A figure with caption is shown below.

4. Results

The performance of MAR is examined and compared with the DSDV. To evaluate the performance of these two protocols, two sets of experiments (in total 120) are developed. The first set measures the performance under low mobility conditions whereas the second measures the

performance under high mobility. All experiments are conducted using the network simulator, *ns2*, which provides substantial support for simulation of routing, and multicast protocols over wired and wireless (local and satellite) networks including several ad-hoc routing protocols [8].

In all cases, mobile nodes move randomly with a specified average speed (changed from scenario to scenario), where the selection of both the speed and the direction are based on the uniform probability distribution function. Routes are established and broken as nodes move in and out of transmission range respectively. We assume that the radio connection is bi-directional and reliable once established. In order to save power and use bandwidth more efficiently, nodes are characterized by short communication ranges, which in return provides a multi-hop environment.

We evaluate protocol performances under dynamic network environments using different average speeds for the nodes. The average speed is constant during every simulation. We run the experiments with 6 different levels of speed, 6 scenarios (where each scenario run under 20 different initializations, hence 120 experiments in total). We tested “slow-speed” by setting speed to 0.01, 0.05 and 0.1 meters per second. On the other hand, for “high-speed” scenarios it is set to 1, 2 and 3 meters per second, which is similar with the speed of people walking and jogging. The traffic type is constant bit rate (CBR), and the transport layer is UDP. There are 18750 data packets in total in every simulation. Table 1 shows the parameters used in detail.

Moreover, we evaluate the performance in terms of the number of received packets and average delay. Here, the number of received packets means the number of data packets sent from source nodes and received in destinations successfully, and delay is the total trip time it takes for a data packet to move from its source to its destination. All the experiments are conducted on a PC (Intel Celeron 1.0 GHz, 256M memory), running Linux Mandrake 8.2.

In the Mobile Agent Routing protocol, there are two key parameters that are critical for the protocol performance. These are the number of agents and the history size. In theory, the higher the agent number and history number are, the better the performance is expected. However, more routing agents and long histories also introduce more overheads into the network. In this work, we controlled the number of agents using a global registry to be able to conduct the experiments and measure the performance.

A mobile network is a dynamic network, and the routes are established and broken frequently as nodes move in and out of the transmission range. It is

possible that a node can move out of the transmission range of all others and becomes isolated. If a mobile agent happens to reside at such a node, it becomes stranded. Thus its contribution to the population as a whole is lost. Also mobile agents may be lost during transmission. If we do nothing about such cases, the effective number of mobile agents will drop as time passes. As we know, in mobile agent based networks, system performance relies on the cooperation of all mobile agents [4]. Thus, a drop in the number of agents will degrade the network performance.

Transmission Range	10m
Bandwidth	2 Mbps
Simulation Time	250 sec.
Number of Nodes	50
Pause Time	2 sec.
Environment Size	40m by 40m
Traffic Type	CBR
Packet Rate	5 packets/sec
Packet size	64 bytes
Number of Traffic Flows	15

Table 1. Parameters used for experiments

To solve this problem, a global registry system is defined for the mobile network. This is used as a simple and quick solution to be able to coordinate the generation of agents and measure their performance. All mobile agents register when they are created, and re-register each time a node receives a mobile agent. We also set a threshold time and an agent’s entry will “time out” in the registry if it does not re-register within this threshold time. The system counts the total number of mobile agents currently registered in the global register, and introduces new agents if it detects that an agent is “dead”. Future work will concentrate on different strategies to avoid global registry but maintain the average number of agents on the network stable.

In all experiments reported here, for slow-speed scenarios, history size is set to 5, and the number of agents is set to 10. For high-speed scenarios, history size is set to 7, and the number of agents is set to 50. These values are chosen empirically, after comparing different combinations of agent number and history size values. We ran the simulation for each scenario, and each time nodes moved with random direction and distribution. However, as described above, the average speed is constant during the simulation.

We measured the average delay and the number of packets received, Figures 2 and 3. As shown in these figures, MAR generally performs as well as DSDV.

In terms of the number of packets received, MAR has almost the same performance as DSDV. In addition it has much less delay than DSDV under high-speed scenarios. It should be noted here that it takes some time for MAR to converge. This means that initially all the routing tables are empty, with agents exploring the network to acquire routing information, the network gradually converges. It takes 2-10 seconds for the network to converge to its equilibrium state, i.e. propagate routing information [10]. Statistics given in this work include this convergence time as well.

Finally, a T-test analysis is performed to evaluate the significance of the simulation results between the different protocols for each scenario. The results show that MAR can perform as well as DSDV and even better in more dynamic scenarios, i.e. high-speed, within 95% confidence interval, tables 2 and 3.

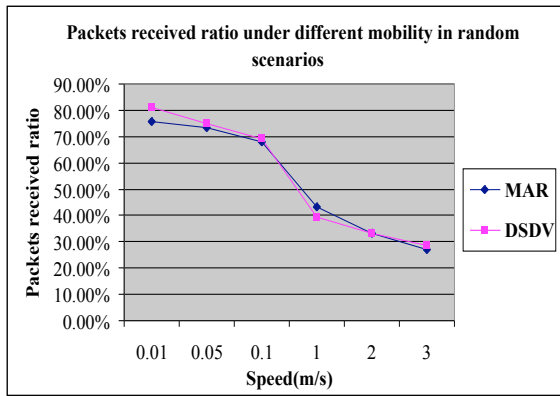


Figure 3. Percentage of average packets received under different mobility

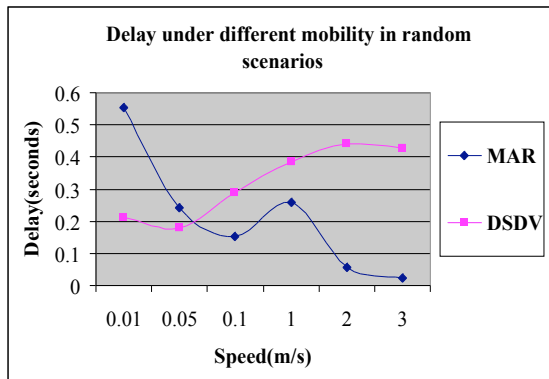


Figure 4. Average delay under different mobility

Mobility	Mean for DSDV	Mean for MAR	T-Test Results
0.01 m/s	80.96%	75.74%	DSDV is better by 6%
0.05 m/s	74.77%	73.37%	No significant difference
0.1 m/s	69.05%	68.07%	No significant difference
1 m/s	39.59%	43.14%	MAR is better by 8%
2 m/s	33.27%	33.27%	No significant difference
3 m/s	28.50%	27.12%	No significant difference

Table 2. T-test results for packets received

Mobility	Mean for DSDV	Mean for MAR	T-Test results
0.01 m/s	0.211366	0.553960	DSDV is better by 62%
0.05 m/s	0.179969	0.240598	DSDV is better by 15%
0.1 m/s	0.289699	0.152245	MAR is better by 49%
1 m/s	0.385568	0.259218	MAR is better by 33%
2 m/s	0.440061	0.056125	MAR is better by 86%
3 m/s	0.427426	0.022500	MAR is better by 95%

Table 3. T-test results for delay

5. Conclusion

In this paper, we introduced a simulation environment where we compare our MAR protocol to the DSDV protocol for mobile ad-hoc network routing. The results show that MAR achieves better performance under more dynamic conditions and achieves substantial improvement in terms of delay and the number of received packets compared to DSDV. A distributed approach to generate and control agents is under development to employ only local information. Tests for the performance of dynamic generation of agents is an issue that needs to be studied further as well as employing comparisons under different scenarios.

Acknowledgements

This work was supported in part by Discovery grant, of the second author from the Natural Sciences and Engineering Research Council of Canada.

References

- [1] Johansson P., Larsson T., Hedman N., Mielczarek B., Degermark M., Scenario-based Performance Analysis of Routing Protocols for Mobile Ad-hoc Networks, ACM MOBICOM, 1999.
- [2] Groten D., Schmidt J. R., Bluetooth-based Mobile Ad Hoc Networks: Opportunities and Challenges for a Telecommunications Operator, The IEEE Vehicular Technology Conference, 2001.
- [3] Braginsky D., Estrin D., Rumor Routing Algorithm for Sensor Networks, The 1st ACM Wireless Sensor Networks and Applications, 2002.
- [4] Minar N., Kramer K. H., Maes P., Cooperative Mobile Agents for Dynamic Network Routing, in Software Agents for Future Communications Systems, Chapter 12, Springer-Verlag, 1999, ISBN: 3-540-65578-6.
- [5] Di Caro G., Dorigo M., AntNet: Distributed Stigmergetic Control for Communications Networks, Journal of Artificial Intelligence Research, 9, pp. 317-365, 1998.
- [6] Amin K., Mayes J., Mikler A., Agent Based Distance Vector Routing, Proceedings of the Third International Workshop, 2001.
- [7] Zou X, Ramamurthy B., Magliveras S., Routing Techniques in Wireless Ad Hoc Networks – Classification and Comparison, Proceedings of the Sixth World Multi-conference on Systemics, Cybernetics, and Informatics, 2002.
- [8] The Network Simulator - ns2. Retrieved April 2002, from: <http://www.isi.edu/nsnam/ns/>.
- [9] Corson, S., & Macker, J., Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations. IETF RFC 2501. Retrieved April, 2002, from: <http://www.ietf.org/rfc/rfc2501.txt>.
- [10] Morrison, D., Agent-based ad-hoc network routing: The approach to equilibrium, Honours Thesis, Dalhousie university, Faculty of computer Science, 2003.