Performance Analysis of an Ant-based Routing Algorithm with Enhanced Path Maintenance for MANETs

Miae Woo
Dept. of Information and Communications Eng.
Sejong University
Seoul, Korea
e-mail: mawoo@sejong.ac.kr

Abstract— **Ant-based routing algorithms belong to a class of ant colony optimization which applies the behavior of ants in nature to routing mechanism. Since ant-based routing algorithms provide high adaptability to the dynamic network topology, it is suitable for routing in mobile ad-hoc network (MANET). In this paper, we introduce a routing method, namely EPMAR (ant-based routing algorithm using enhanced path maintenance), which enhances route selection method and the process upon link failure of EAR. We then compare the performance of ant-based routing algorithms, AntHocNet, EAR, and EPMAR, how they perform as the packet transmission rate varies. The simulation results show that EPMAR provided higher packet delivery ratio and less critical link failure than AntHocNet and EAR.**

Keywords-Routing, Mobile Ad-Hoc Network, Ant Colony Optimization, Performance Analysis

I. INTRODUCTION

An ant-based routing algorithm is inspired by the behavior of ants in nature, which uses swarm intelligence [1]. Ant-based routing is an application area of ant colony optimization (ACO) [2] which uses the behavior of an ant colony to optimize the given problem in a distributed fashion. Ants use chemical substance, called pheromone, to share information with other ants in their colony about the paths between the nest and food sources. Ants choose a path with highest pheromone deposit over other paths to the food source, and lay pheromone on the way back to the nest. Consequently, the shortest path would have the most deposited pheromone among paths to the food source. Such a problem solving method is applied to the routing problems in networks.

Characteristics of ant-based routing methods such as agent systems and capability of multi-path routing make ant-based routing methods suitable for mobile ad-hoc network (MANET) environments [3]. Routing methods based on agent systems allow high adaptability to the dynamic network topology. Since paths in MANETs fail frequently, ant-based routing methods can provide stability in the connections between the source and the destination using multiple paths.

Usually, ant-based routing algorithms consist of four phases: route setup, data transfer, route maintenance, and route recovery phases. Since mobile nodes move around, network topology in MANET changes as time passes. As a result, an optimal path may become non-optimal. To handle such situations, route maintenance is required to discover new or better paths [2].

There are several ant-based routing algorithms proposed especially for MANETs [3-10]. Among them, AntHocNet provides good performance compared to ad hoc on-demand vector routing (AODV) in terms of data delivery ratio and end-to-end delay [7]. Its performance in a realistic urban environment has been analyzed in [8]. However, the overhead of AntHocNet generated by the ants are quite high. To overcome such a drawback of AntHocNet, an efficient ant-based routing algorithm (EAR) was proposed in [9] and its performance was further investigated in [10]. EAR introduced several features in the route set-up phase to decrease the overhead introduced by ants and to efficiently update pheromone values in all the intermediate nodes along the path.

In this paper, we introduce an ant-based routing algorithm, namely EPMAR (ant-based routing algorithm using enhanced path maintenance) [11], which modifies data transfer phase and route recovery phase of AntHocNet and EAR. EPMAR uses procedures of EAR for route setup and route maintenance phases. EPMAR aims to cope with link failure more efficiently than AntHocNet and EAR. The performance of EPMAR is investigated and compared to those of EAR and AntHocNet to see how these algorithms perform when the transmission rates of constant bit rate (CBR) sources vary. The simulation results show that EPMAR provide higher packet delivery ratio and less critical link failure than AntHocNet and EAR.

The remainder of this paper is organized as follows. Section 2 describes the operation of EPMAR. In Section 3, the simulation environment to measure the performance of the EPAMR, AntHocNet and EAR is discussed. The simulation results are analyzed in Section 4. Finally, Section 5 gives the conclusion.

ISBN: 978-1-61208-140-3

II. DESCRIPTION OF EPMAR

EPMAR consists of four phases, which are route setup, data transfer, route maintenance, and route recovery phases. EPMAR uses same route setup and route maintenance procedures of EAR [9] and modifies data transfer phase and route recovery phase of EAR.

A. Route Set-up Phase

The procedure of route set-up phase of EPMAR is same as that of EAR [9]. Route set-up phase starts when the source node wants to send a data packet to the destination node. If the source does not have any routing information to the destination, it broadcasts a reactive forward ant (RFA) to probe paths. A reactive forward ant contains several fields: source and destination addresses, generation number, trip time, a list of visited nodes, the number of visited nodes, and a flag for reactive backward ant generation at the intermediate node. One broadcasted ant can produce several ants because of the broadcasting mechanism along the route. Those ants have same source address, destination address, and generation number. Such ants are called as same generation ants. Same generation ants may have different values in the fields of trip time, list of visited nodes, number of visited nodes, and flag for reactive backward ant generation at the intermediate node. The values in these fields are updated as the reactive forward ant travels towards the destination.

If an intermediate node receives a reactive forward ant, it checks whether it already received any other same generation ant. If so, it discards the reactive forward ant which it just received. Otherwise, the intermediate node checks whether it has any routing information to the destination in its routing table. If it does not have any information, it saves the values of source address, destination address, generation number, trip time, and list of visited nodes in its routing table. It then updates trip time, list of visited nodes, and number of visited nodes in the reactive forward ant, and broadcasts the ant.

If the intermediate node has routing information to the destination which satisfies some criteria, it can generate a reactive backward ant. The criteria are as follows. First, any previously visited node en route to this intermediate node had not generated a backward ant. The routing information in this intermediate node should be fresh enough. We regard the routing information which was updated within 10 seconds as fresh. Finally, the intermediate node generating a backward ant should be close enough to the source and far enough from the destination. This condition is checked by looking at the hop distances from the source and to the destination. We use 10-hop distance is the maximum hop distance between the source and the intermediate node. Also the minimum hop distance between the intermediate node and the destination is set to 5 hops. If the intermediate node satisfies these conditions, it generates a reactive backward ant (RBA) towards the source. The intermediate node sets the flag for reactive backward ant generation to indicate that the backward ant is not generated from the destination. When the source receives a reactive backward ant, it can decide who sends the backward ant by inspecting the flag for reactive backward ant generation.

The intermediate node unicasts the reactive forward ant to the next hop probabilistically. Before unicasting the ant, the intermediate node updates the fields and sets the flag for the reactive backward ant generation at the intermediate nod in the reactive forward ant.

The destination node can accept several same generation ants and generate reactive backward ants as many as the accepted forward ants to form multi-path. There is a limit for the number of acceptable same generation ants. We only accept at most 3 ants among same generation ants.

The reactive forward ants accepted at the destination pass the list of visited nodes to the reactive backward ants. The reactive backward ants travel towards the sources by backtracking the nodes in the list of visited nodes.

The reactive backward ant calculates the overall trip times from the currently visited node to all the nodes on the path to the destination, and estimates transmission time required at the MAC layer by taking queue length and the average delay into account. It uses the calculated time to update pheromone values in the routing table [7]. Let T_{nj}^i be the pheromone value of the path from node i to node j through the neighbor node n . T_{nj}^i is calculated as a running average using the following equation.

$$T_{nj}^i = \gamma T_{nj}^i + (1-\gamma)\left(\frac{1}{\hat{T}_j^i + hT_{hop}}\right)$$

In the equation, h is the hop distance between node i and node j . T_{hop} is the time required to deliver one packet to the next hop in unloaded condition. $\hat{T}_j^i$ is the estimate of trip time from node i to node j . It is the sum of local estimates $\hat{T}_{i+1}^i$ to reach next hop $i+1$ in each node along the path from node i to node j . $\hat{T}_{i+1}^i$ is calculated using following equation.

$$\hat{T}_{i+1}^i = \alpha \hat{T}_{i+1}^i + (1-\alpha)(Q_{mac}^i + 1)\hat{T}_{mac}^i$$

where Q_{mac}^i is the number of packets in the queue to be sent at the MAC layer and $\hat{T}_{mac}^i$ is the running average of estimate of the average time to send one packet at the MAC layer.

B. Data Transfer Phase

After setting-up paths to the destination, data packets are forwarded based on the pheromone values in the routing table. AntHocNet and EAR select the next hop stochastically based on the pheromone value in the routing table to spread data load on multiple paths. However, in the MANET environment where link failures are frequent, it is not confirmed yet how load distribution has effect on the data transfer efficiency. Also, it is uncertain that routes with low pheromone value can actually provide stable paths to the destination.

Thus, EPMAR chooses a path with highest pheromone value among multiple paths in the routing table. Let N_d^i be the set of neighbor nodes of node *i* through which paths exist to the destination *d* in the routing table. Then, a neighbor node which provides the highest pheromone value to the destination is selected as a next hop. In other words, the selection criteria of next hop is

$$\max_{n \in N_d^i} \left\{ T_{nd}^i \right\}.$$

For the data transfer, multiple paths are used as backup paths when link failure along the best path occurs.

C. Route Maintenance Phase

Route maintenance phase of EPMAR is same as AntHocNet [7] and EAR [9]. To maintain the established paths and to find better or alternative paths, a source node periodically dispatches proactive forward ants (PFAs) at the rate according to the data sending rate [7]. A proactive forward ant can be either unicasted probabilistically or broadcasted. Generally, a proactive forward ant chooses the next hop probabilistically to probe an established path. It collects up-to-date information about the established path and updates the pheromone values of the path by the corresponding proactive backward ants (PBAs). A proactive forward ant is broadcasted with a small probability at the intermediate node to explore a new or alternative path. If a node receives a proactive forward ant but it does not have any routing information to the destination, it broadcasts the proactive forward ant. Total number of broadcast allowed through the path toward the destination is limited to control the overhead.

D. Route Recovery Phase

As in AntHocNet and EAR, EPMAR considers two situations as link failures. One is failure in receiving hello messages from a neighbor node and the other is failure in transmission of data packets.

If a node does not receive a hello message from its neighbor for a certain amount of time, the link is considered broken. In this case, the node removes the associated entries in its routing table, and broadcasts a link failure notification message. All the neighbors receiving the notification message update their routing table. If any one of them lost its best or the only path to the destination due to the link failure, it rebroadcasts the notification.

If transmission of a data packet is failed and there is no other path available for the data packet, then the node tries to repair the path locally by broadcasting a forward route repair ant (FRRA). If any backward route repair ant is not received within a certain time period, the node discards all the temporally buffered packets and broadcasts a link failure notification about the lost destination.

The forward route repair ant is perished at the intermediate node if it has any alternative path to the destination. As a result, the source of data packets cannot recognize that the link failure happened in the path to the destination. According to the simulation results obtained in the previous work [9, 10], the ratio of route recovery by using forward route repair ant was low.

EPMAR augments the procedure in route recovery phase by sending unicast link failure message to the source of the data packet which cannot be forwarded due to link failure. The overhead incurred by this message is low because it is delivered by unicating. When the source receives the unicast link failure message, it can recognize that the best path to the destination has been failed. To obtain up-to-date pheromone value of a path to the destination, the source dispatches a proactive forward ant. The procedure for the proactive forward ant in this case is same as that in the route maintenance phase. By doing this, the source node can update the pheromone value, and it can use the route with the highest pheromone value for the data transfer.

III. SIMULATION ENVIRONMENT

To evaluate the performance of EPMAR, we ran simulations using Qualnet. In the area of 3000 × 1000 m^2, 100 mobile nodes were randomly placed. A rectangular space was chosen to force the use of longer routes between nodes than those would occur in a square space with equal node density [12]. Simulation time is set to 300 seconds. By selecting traffic sources and destinations randomly, 30 connections were established. Four different transmission rates were used: 1 packet/sec, 2 packets/sec, 5 packets/sec, and 10 packets/sec. Each source generated 64-byte long CBR packets. Data transmission for each connection was started by selecting random delay from uniform distribution in [0, 60] seconds. Data transmissions were continued till the end of simulation.

For the physical layer, two-ray signal propagation model was used. The radio propagation range of each node was set to 300 meters. For the MAC layer, IEEE 802.11b protocol was used with 2 Mbps bandwidth.

The random waypoint mobility model was used for the node movement model [13]. Random waypoint model defines the mobility pattern of nodes by pause time and the maximum node speed. In Qualnet, each node began the simulation by selecting a random destination in the given space and moved to that destination at a speed distributed uniformly between 0 and some maximum node speed. Upon reaching the destination, the node paused for the specified pause time. It then selected another destination, and proceeded from there as previously described. Each node repeated this behavior for the simulation time. In the simulation, maximum node speed was set 20 m/sec. Also, 5 different pause times, 0, 30, 60, 120, and 300 seconds were used.

IV. PERFORMANCE ANALYSIS

In order to compare the performance of EPMAR with AntHocNet and EAR, we investigated packet delivery ratio, the number of critical link failures, the number of generated forward ants, number of forwarded ants and backwarded ants per node in average and path set-up time.

Figure 1 shows the average packet delivery ratio. As we can see in Figure 1, EPMAR gave the highest packet delivery ratio among three algorithms regardless of packet transmission rate. EPMAR provided about 16% better packet delivery ratio than AntHocNet and 4% better packet delivery ratio than EAR in average. Especially, as the packet transmission rate was increased to 10 packets per second, EPMAR provided 26% and 7% more packet delivery ratio than AntHocNet and EAR respectively.

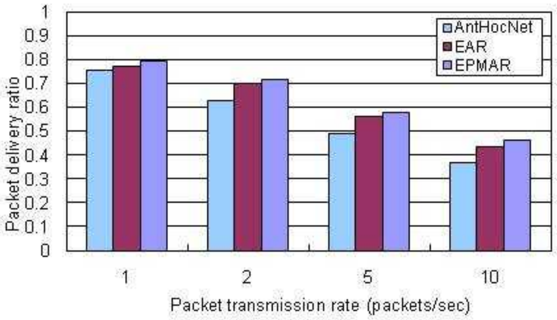

Figure 1. Average packet delivery ratio

The number of critical link failures occurred during the simulation time is shown in Figure 2. Clearly, we can see that least critical link failures occurred when EPMAR was used as a routing algorithm. EPMAR resulted 29% and 10% less critical link failures than in AntHocNet and EAR respectively.

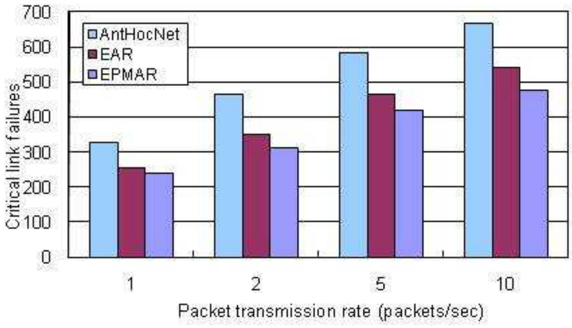

Figure 2. Number of ciritical link failures occurred

Through the results of the packet delivery ratio and critical link failure shown in Figure 1 and Figure 2, it is proven that EPMAR obtained the intended goals. In other words, packet delivery ratio was improved by choosing the path with the highest pheromone deposit. Also, informing link failure situation by sending unicast link failure messages to the source could reduce additional link failures.

However, introduction of unicast link failure message increased overhead in the system. Clearly, delivery of unicast link failure message to the source adds up overhead. Upon receiving the unicast link failure messages, dispatching proactive forward ants is triggered at the source. The destination who receives proactive forward ants should respond with proactive backward ants. These proactive forward ants and backward ants are the added overhead in the EPMAR compared to EAR or AntHocNet. The related results are shown in Figure 3, 4, and 5.

Figure 3 depicted the number of forward ants generated at the 30 sources during the simulation. The counted forward ants include reactive forward ants (RFAs), proactive forward ants (PFAs), and forward route repair ants (FRRAs) which are generated due to link failure. Overall, as we can see in the figure, there is no big difference in the number of forward ants generated among three methods. EPMAR generated the least reactive forward ants and forward route repair ants. However, as we can expect, EPMAR generated the most proactive forward ants.

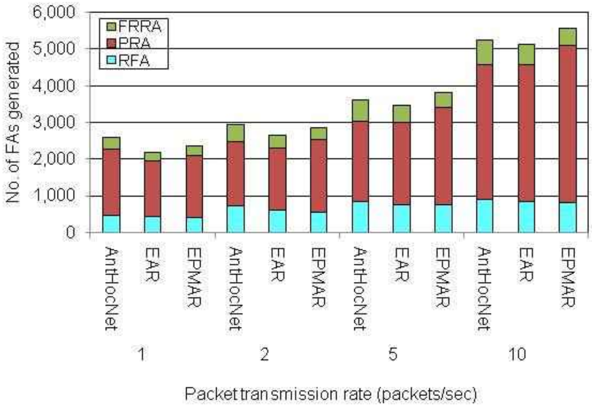

Figure 3. Number of generated forward ants

The generated forward ants are delivered to the destination through intermediate nodes either by unicasting or by broadcasting depending on the type of ants. Figure 4 shows the average number of forwarded ants per nodes. The shown number summed all the number of generations at the sources and forwards at the intermediate nodes and averaged out by the total number of nodes in the system. Although, EPMAR generated the most forward ants among three methods when the transmission rate were 5 packets/sec and 10 packets/sec, it provided the least forwarding in the overall system. The average number of forwarded ants per node of EPMAR was 31% of AntHocNet and 97% of EAR.

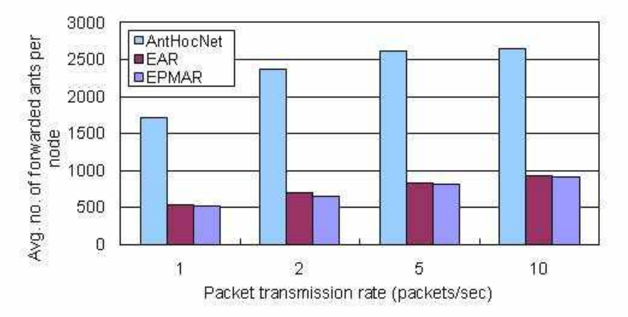

Figure 4. Average number of forwarded ants per node

On the other hand, EPMAR produced the most backwarded ants per node as shown in Figure 5. EPMAR forwarded 57% more backward ants than EAR and 21% more backward ants than AntHocNet. Reactive forward ants and proactive forward ants trigger dispatching the corresponding backward ants after they arrived at the destinations. Since the objective of forward route repair ants is recovery of local paths, the time-to-live value of the forward route repair ants is set to relatively smaller than those of reactive and proactive forward ants. As a result, there might be many cases which did not result the corresponding backward ants. Actually, local route repair ratio was less than 20% in the simulation results. Consequently, EPMAR produced more backward ants because it generated more proactive forward ants as shown in Figure 3. When compared with EAR, EPMAR forwarded at most 70 more backward ants and 20 less forward ants. Thus, the increased overhead was not big.

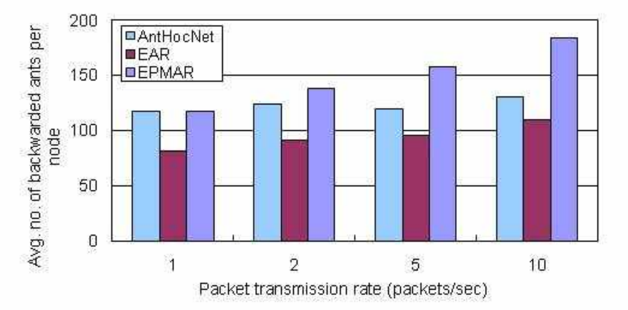

Figure 5. Average number of backwarded ants per node

Figure 6 shows the path set-up time of three routing methods. Since EPMAR and EAR uses the same route set-up procedure, path set-up time of these methods is almost same. Compared with the path set-up time of AntHocNet, EPMAR and EAR reduced the path set-up time about 3/10 by generating RBA at the intermediate nodes.

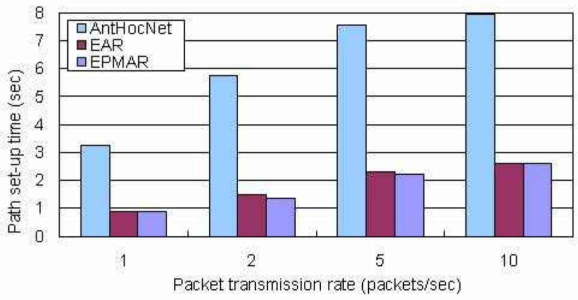

Figure 6. Path set-up time

V. CONCLUSION

In this paper, we introduced an ant-based routing algorithm with enhanced path maintenance, namely EPMAR. The objective of EPMAR was set to increase the performance by choosing the best path for the data delivery and to reduce the critical link failures. The performance of EPMAR was compared with that of AntHocNet and EAR. The simulation results showed that EPMAR provided better packet delivery ratio and less critical link failures than AntHocNet and EAR. Though EPMAR introduced a new message called unicast link failure message to reduce critical link failures, the control overhead was comparable to that of EAR and less than AntHocNet.

ACKNOWLEDGMENT

This research is supported by the ubiquitous Computing and Network (UCN) Project, the Ministry of Knowledge and Economy(MKE) Knowledge and Economy Frontier R&D Program in Korea as a result of UCN's subproject 11C3-C2-10M.

REFERENCES

[1] M. G. Hinchey, R. Sterritt, and C. Rouff, “Swarms and Swarm Intelligence,” IEEE Computer, Vol. 40, No. 4, pp. 111-113, April 2007.

[2] K. M. Sim and W. H. Sun, “Ant Colony Optimization for Routing and Load-Balancing: Survey and New Directions,” IEEE Trans. on Systems, Man, and Cybernetics-Part A: Systems and Humans, Vol. 33, No, 5, pp. 560-572, Sep. 2003.

[3] M. Güneş, U. Sorges, and I. Bouazzi, “ARA-the ant-colony based routing algorithm for MANETs,” in Proc. of IWAHN 2002, pp. 79-85, August 2002.

[4] H. Matsuo and K. Mori, “Accelerated Ants Routing in Dynamic Networks,” 2nd International Conf. on Software Engineering, Artificial Intelligence, Networking & Parallel/Distributed Computing, pp. 333-339, 2001.

[5] S. Marwaha, C. K. Tham, and D. Srinivasan, “Mobile agents based routing protocol for mobile ad hoc networks,” in Proc. of IEEE ICON, pp. 27-30, August 2002.

[6] J. S. Baras and H. Mehta, “A Probabilistic Emergent Routing Algorithm for Mobile Ad Hoc Networks,” in Proc. of WiOpt03, 2003.

[7] G. Di Caro, F. Ducatelle, and L. M. Gambardella, “AntHocNet: An Adaptive Nature-Inspired Algorithm for Routing in Mobile Ad Hoc Networks,” Tech. Rep. No. IDSIA-27-04-2004, IDSIA/USI-SUPSI, Sep. 2004.

[8] F. Ducatelle, G. Di Caro and L. M. Gambardella, "An Evaluation of Two Swarm Intelligence MANET Routing Algorithms in an Urban Environment," in Proc. of IEEE Swarm Intelligence Symposium, Sep. 2008.

[9] Miae Woo, Ngo Huu Dung, Woo Jong Roh, "An Efficient Ant-based Routing Algorithm for MANETs," in Proc. ICACT 2008, Feb. 2008.

[10] Myungjun Oh, Ngo Huu Dung, and Miae Woo, “Effect of Transmission Rates on the Performance of Ant-Based Routing Algorithms for MANETs,” in Proc. UBICOMM 2009, pp. 238-243, Oct. 2009.

[11] M. Woo, “An Ant-based Routing Method using Enhanced Path Maintenance for MANETs,” Journal of Korea Information and Communications Society, Vol. 35, No. 9, pp. 1281-1286, Sep. 2010.

[12] Broch, D. A. Maltz, D. B. Johnson, Y. Hu and J. Jetcheva, “A Performance Comparison of Multi-Hop Ad Hoc Network Routing Protocols,” in Proc. MobiCom’98, Oct. 1998.

[13] D. B. Johnson and D. A. Maltz, “Dynamic source routing in ad hoc wireless networks,” in Mobile Computing, edited by T. Imielinski and H. Korth, Kluwer Academic Publishers, pp. 153–181, 1996.