Scalability Issue of Ant-based Routing Algorithms for MANETs

Woo Jong Roh, Ngo Huu Dung, Miae Woo
Department of Information and Communications Engineering, Sejong University
98 Gunja-Dong, Gwangjin-Gu, Seoul, 143-787, Korea
E-mail: {parang, ngohdung ,mawoo}@cnet.sejong.ac.kr

Abstract: Ant-based routing algorithm is a class of ant colony optimization algorithms which are inspired by the behavior of ants in nature. We have been proposed an efficient ant-based routing algorithm (EAR) for mobile ad-hoc networks (MANETs) in order to support multi-path routes and to reduce overheads by managing ants efficiently. In this paper, we analyze the performance of EAR and AntHocNet to show how EAR outperforms AntHocNet in terms of scalability by varying the number of nodes in the network system.

1. Introduction

Finding the shortest path to the food source by the group of ants using stigmergy draws the researchers in the networking areas attention, resulting several ant-based routing algorithms proposed [1, 2]. There are characteristics of ant-based routing methods which make them suitable for MANET environments [3]. Ant-based routing methods are based on the agent systems, allowing high adaptability to the dynamic network topology. Also, ant-based routing methods are capable to support multi-path routing. Consequently, ant-based routing methods can provide stability of the connections between the source and the destination in MANETs despite of frequent path failure.

There are several ant-based routing algorithms proposed especially for MANETs [3-9]. Among them, AntHocNet [7] provides good performance compared with AODV in terms of data delivery ratio and end-to-end delay. 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]. EAR introduced several features in the route set-up phase to decrease the overhead occurred due to ants and to efficiently update pheromone values in all the intermediate nodes along the path. In this paper, we analyze the performance of EAR and AntHocNet in terms of scalability by varying the number of nodes in the network system.

The remainder of this paper is organized as follows. In Section 2 we present the basics of ant colony optimization. Overview of EAR in Section 3. In Section 4, the performance of EAR and AntHocNet is analyzed through simulations. Finally, a conclusion is given in Section 5.

2. Basics of Ant Colony Optimization

In nature, ants use chemical substance, called pheromone. Ants leave pheromone between their nest and food source which they find. Ants trace pheromone deposited by their nest mates to reach the food source and bring the food back to their nest leaving their own pheromones.

Ants spread over a network, wander around to find a path to the destination, or move toward the destination. Depending on the protocol, pheromone deposit by ants can be accomplished either on the way to the destination or on the way back to the source. If any routing algorithm uses a scheme to deposit pheromone on the way to the destination, it is only applicable to the symmetric networks.

By pheromone reinforcement, the path used from the source to the destination attracts more ants and data packets. As time passes, the pheromone concentration on the shorter path will be higher than that on the longer path, because the ants using the shorter path will increase the pheromone concentration faster. The shortest path will thus be identified and will become the only path used by all ants eventually. This leads to a problem, called stagnation. Stagnation occurs when a network reaches its convergence. An optimal path is chosen by all ants and this increases ant's preference of the optimal path recursively. This may lead to congestion of an optimal path and dramatic reduction of the probability of selecting other paths. If an optimal path is congested, then it may become nonoptimal. To mitigate stagnation, some methods, such as pheromone-heuristic control and privileged pheromone laying [2], were proposed.

In MANET, an optimal path may be disconnected due to the movement of nodes. Other nonoptimal paths may become optimal due to changes in network topology. New or better paths may be discovered [2]. To handle such situations, route maintenance is required. Usually, ACO routing algorithms consist of four phases; route setup, data transfer, route maintenance, and route recovery phases.

3. Overview of EAR

EAR [9] is based on the ACO routing algorithm. It is a multi-path algorithm and has both reactive and proactive elements. It is consisted of four phases as other ACO-based routing algorithms. In EAR, pheromone-heuristic control is used for stagnation. EAR introduces several features to the AntHocNet [7], mainly during the route set-up phase. Other phases follow the procedures given in the AntHocNet [7].

3.1 Route Set-up Phase

Route set-up phase starts when the source node wants to send a data packet to the destination node. The source initiates route set-up phase by dispatching a reactive forward ant to the destination. In response to the reception of a reactive forward ant, the destination generates a reactive backward ant towards the source. A reactive backward ant can also be generated by any legitimate intermediate node, which is a unique feature in EAR.

If necessary, the source generates a reactive forward ant and broadcasts it to probe paths to the destination. A reactive forward ant contains source address, destination address, generation number, trip time, list of visited nodes,

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 used in MANETs. Those ants have same source address, destination address, and generation number. Such ants are called as "same generation ants" in this paper. 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.

Upon receiving the reactive forward ant, an intermediate node checks whether any other same generation ant has already visited. If so, the intermediate node discards the currently received ant. 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, then 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 field values of trip time, list of visited nodes, and the number of visited nodes in the reactive forward ant, and broadcasts the ant. If the intermediate node has routing information to the destination, it can generate a reactive backward ant. Then it unicasts the forward ant to the next hop node probabilistically. Among all the neighbor nodes, node i chooses next hop n to let the forward ant reach the destination d with the probability P_{nd}^i:

$$P_{nd}^i = \frac{G_{nd}^i}{\sum_{j \in N_d^i} G_{jd}^i},$$

where G_{nd}^i is the average pheromone value indicating the estimated goodness of the route from node i over next hop n to reach to destination d, and N_d^i is the set of neighbors of i that can be seen as a next hop to reach to d.

Before doing unicast, the intermediate node updates the necessary fields in the reactive forward ant. Only if it generates the corresponding backward ant, it sets the flag for reactive backward ant generation at the intermediate node in the forward ant.

The destination node can accept multiple same generation ants and generate reactive backward ants as many as the accepted forward ants to form multi-path between the source and the destination. There is a limit for the number of acceptable same generation ants. The reactive forward ant accepted at the destination passes the list of visited nodes to the reactive backward ant. The reactive backward ant travels towards the source by backtracking the nodes in the list of visited nodes.

A reactive backward ant can also be generated by an intermediate node. In order to do this, there are several conditions to be met.

- There should be no backward ant generated by any other previously visited node en route to the intermediate node. It can be checked by looking at the flag in the reactive forward ant.
- The routing information to the destination should be fresh enough. The update time of the corresponding routing information should be within some time threshold.
- The hop count between the source and the intermediate node should be within some limited value. Also, the hop count between the intermediate node and the destination should be greater than some limited value. This restriction poses condition that the intermediate node which generating a backward ant should be close enough to the source and far enough from the destination.

If the intermediate node satisfies the above conditions, it generates a reactive backward ant towards the source. In this case, the intermediate node sets the flag for reactive backward ant generation to indicate that the backward ant is not generated from the destination.

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.

The pheromone value distinguishes the goodness of the path. It is calculated from the trip time and number of hops:

$$R_d^i = \left(\frac{T_d^i + hT_{hop}}{2} \right)^{-1},$$

where R_d^i is the pheromone value for destination d at node i, T_d^i is the estimated trip time of a data packet to travel from i to d, h is the number of hops and T_{hop} is the one hop travel time**.** Whenever the pheromone value R_d^i is updated at an entry, the average pheromone value, G_{nd}^i is also updated as follows:

$$G_{nd}^i = \gamma G_{nd}^i + (1-\gamma) R_d^i,$$

where γ is a weighting factor.

In EAR, the number of entries for a specific node in the routing table is controlled in order to reduce the overhead in the routing table. Priorities are given to the entries updated by the backward ant generated from the destination, and entries with fresh pheromone values.

3.2 Data Transfer Phase

After setting-up paths to the destination, data packets are forwarded based on the pheromone values in the routing table. If the node has multiple paths to the destination of the data packet, it selects the next hop stochastically. Such routing strategy leads to data load spreading according to the estimated quality of the paths.

3.3 Route Maintenance Phase

A source node periodically dispatches proactive forward ants at the rate according to the data sending rate to maintain the established paths and to find better or alternative paths [8]. A proactive forward ant can be unicasted probabilistically or be 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. A proactive forward ant can be broadcasted with a small probability at the intermediate node to explore a new or alternative path. If the neighbor node receives a proactive forward ant but does not have any routing information for the destination, it rebrocasts the proactive forward ant. Total number of broadcast allowed through the path toward the destination is limited to control the overhead.

3.4 Route Recovery Phase

There are two mechanisms for detecting link failure. One is not receiving hello messages from the neighbors and the other is the failure of data packet transmission.

If a node does not receive hello messages from its neighbor for a certain amount of time, it considers that the link is broken. The node takes a sequence of actions. First, it removes the corresponding neighbor of the broken link from its neighbor list and the associated entries in its routing table. Then, the node broadcasts a link failure notification message, containing a list of destinations to which the node lost its best path, the new best estimated end-to-end delay, and the number of hops to this destination. All its neighbors receiving the notification message update their pheromone values in the routing table. If any one of them lost its best or the only path to the destination due to the failure, it will broadcast the notification until all concerned nodes are notified about the new situation.

If the link failure is detected by the failed transmission of a data packet, and there is no other path available for this packet, then the node tries to repair the path locally by broadcasting a forward route repair ant. The route repair ant travels to the specified destination using the same way taken by the reactive forward ant. Difference is that the route repair ant has a maximum number of broadcasts. If any backward route repair ant is not received within a certain time period, then the node concludes that route repair failed. As a following action, it discards all the temporarily buffered packets and broadcasts a link failure notification abut the lost destination.

4. Scalability Evaluation

For the scalability evaluation, Qualnet was used as a simulation tool. The scalability of EAR and AntHocNet is investigated by varying the number of nodes in the network system; 100, 150 and 200 nodes. The simulation time for each experiment was set to 300 seconds.

In the area of 3000 × 1000 m^2, mobile nodes were randomly placed. 20 connections were formed by choosing randomly 20 nodes as traffic sources and 20 nodes as destinations. Each source generated 64-byte long constant bit rate (CBR) packets in every second. Data transmission for each connection was started by selecting random delay time from uniform distribution in [0, 60] seconds and 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 movement model. Two maximum node speeds were used; 1 m/sec and 20 m/sec. Also, 5 different pause times, 0, 30, 60, 120, and 300 seconds were used for each maximum node speed.

Scalability issue was investigated in terms of packet delivery ratio, average end-to-end delay of data packets, and the number of forward ants delivered in the system. When the maximum node speed was 1 m/sec, there were not much difference in packet delivery ratio and average end-to-end delay between EAR and AntHocNet. But EAR showed better performance than AntHocNet when the maximum node speed was 20 m/sec. For the number of forward ants delivered in the system, EAR provided fewer numbers of forward ants delivered in the system than AntHocNet regardless of maximum node speed.

The packet delivery ratios when the maximum node speed was 20 m/sec are shown in Figure 1. As we can see from the figure, EAR consistently provided about 1.8~2.5% better packet delivery ratio than AntHocNet.

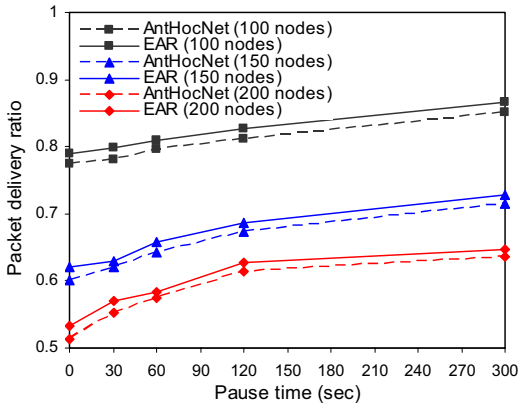


Fig. 1 Average packet delivery ratio when maximum node speed was 20 m/sec

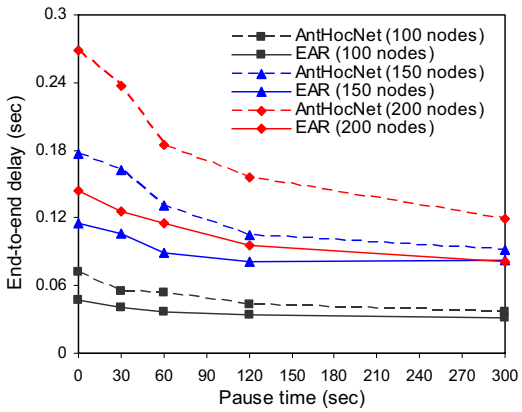
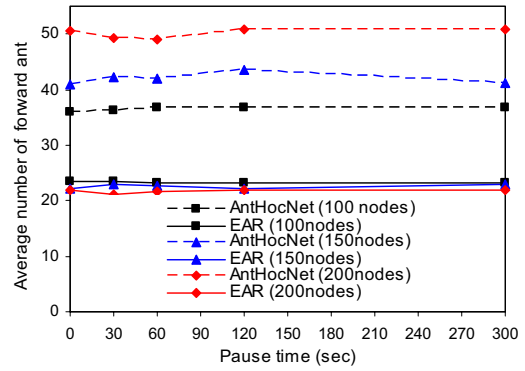


Fig. 2 Average end-to-end delay of data packets when maximum node speed was 20 m/sec

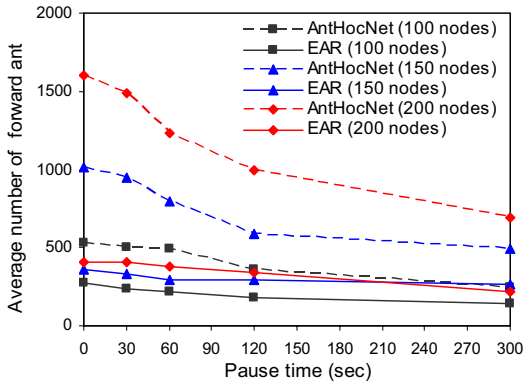
As shown in Figure 2, when the maximum node speed was 20 m/sec, EAR provided 27%, 29% and 42% reduced

end-to-end delay than AntHocNet as the number of nodes changed from 100 to 200.

The result of the average number of forward ants delivered in the system is given in Figure 3. When the maximum node speed was 1 m/sec, EAR generated 36%, 46%, and 56% fewer forward ants compared to AntHocNet as the number of nodes changed 100, 150, and 200 respectively. On the other hands, as shown in Figure 3 (b), the performance difference between EAR and AntHocNet got bigger when the maximum node speed was 20 m/sec. EAR provided less forward by 51%, 59%, and 70% compared to AntHocNet as the number of nodes changed 100, 150, and 200 respectively. For EAR, pause time did not affect the number of forward ants significantly, but the maximum node speed did. However, for AntHocNet, the number of forward ants delivered in the system increased as the pause time shortened when the maximum node speed was 20 m/sec. As a result, it can be stated that EAR was not susceptible to the node mobility in terms of control overhead by maintaining fairly stable number of forward ants.



(a)



(b)

Fig. 3 Average number of forward ants (a) when the maximum node speed was 1 m/sec (b) when the maximum node speed was 20 m/sec

5. Conclusion

In this paper, we analyzed the performance of ant-based routing algorithms for MANETs, namely EAR and AntHocNet in order to see how they are scalable. To do that, we varied the number of nodes in the system from 100 nodes to 200 nodes and ran simulations. The simulation results showed that EAR is more scalable than AntHocNet by providing consistently better performance in terms of packet delivery ratio, end-to-end delay and overhead occurred by the forward ants. Especially improvements in the end-to-end delay and overhead generated by the forward ants were biggest when the number of nodes was 200. Consequently, it can be concluded that EAR is more scalable than AntHocNet in terms of node density in the system.

Acknowledgement

This research is supported by Foundation of ubiquitous computing and networking project (UCN) Project, the Ministry of Knowledge Economy(MKE) 21st Century Frontier R&D Program in Korea and a result of subproject UCN 08B3-B2-10M.

References

[1] M. G. Hinchey, R. Sterritt, and C. Rouff, "Swarms and Swarm Intelligence," IEEE Computer, Vo. 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, Vol. 33, No, 5, Sep. 2003, pp. 560-572.

[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] T. Maekawa, et. al. "An Ant-based Routing Protocol using Unidirectional Links for Heterogeneous Mobile Ad-Hoc Networks," in Proc. of ICWMC '06, pp. 43 - 43, July 2006.

[9] Miae Woo, Ngo Huu Dung, Woo Jong Roh, "An Efficient Ant-based Routing Algorithm for MANETs," in Proc. of ICACT 2008, Feb. 17-20, 2008.