A SWARM INTELLIGENCE ROUTING ALGORITHM FOR MANETS

Zhenyu Liu and Marta Z. Kwiatkowska
School of Computer Science
The University of Birmingham
Birmingham, England B15 2TT
email: {z.liu, mzk}@cs.bham.ac.uk

Costas Constantinou
Electronic Electrical and Computer Engineering
The University of Birmingham
Birmingham, England B15 2TT
email: c.constantinou@bham.ac.uk

ABSTRACT

This paper presents a novel swarm intelligence inspired routing algorithm (EARA) for mobile ad hoc networks. Based on the understanding of the evolutionary cooperation in the biological swarm, which provides an alternative solution resilient against changes in the dynamic environment, We use the principle of swarm intelligence for the reinforcement of optimal routes with only *local* communication (here, local communication is defined as neighbour-to-neighbour one only). The data traffic is influenced at each node, and the communicating nodes observe this influence to update their tables. We also include an evaluation, and the simulation results show that this routing algorithm scales well to a variety of network conditions.

KEY WORDS

MANET, swarm intelligence, routing.

1 Introduction

Mobile ad hoc networks (MANETs) are infrastructureless, multi-hop, wireless mobile networks formed spontaneously. Communication in such a decentralised network typically involves temporary multi-hop relays, with some nodes acting as relay routers without any fixed infrastructure. This kind of network is very flexible and suitable for applications such as temporary information sharing in conferences, military actions and disaster rescues. However, multi-hop routing, random movement of mobile nodes and other features unique to MANETs lead to enormous control overhead for route discovery and maintenance. In some scenarios, the routing maintenance overhead may consume so much resource that it seriously compromises long-term efficiency. Furthermore, compared with traditional networks, MANETs suffer from resource constraints in energy, computational capacities and bandwidth. All of these make routing in MANETs a very challenging problem. The critical question here is how to maintain the routing information efficiently and effectively to deal with the ever-changing topology in a scalable manner with the minimum resource to achieve the best performance of the systems.

To address the routing challenge in MANETs, many approaches have been proposed in the literature. Based on the routing mechanism for traditional networks, the proactive approaches attempt to maintain routing information for each node in the network at all times [1, 2, 3], whereas the reactive approaches only find new routes when required [4, 5, 6]. Other approaches make use of geographical location information for routing [7, 8]. These previous works primarily applied traditional approaches to routing in wired networks to the more volatile network environment experienced in MANETs. While many optimisations to these above algorithms exist, they still suffer from lack of efficiency and scalability with respect both to the network size and to the node movement pattern, due to the exchange of the global or the partial global topology information to maintain the routing information.

Recently, a new family of algorithms emerged inspired by swarm intelligence (SI), which provides a novel approach to distributed optimisation problems. The expression ”swarm intelligence” defines any attempt to design algorithms inspired by the collective behaviour of social insect colonies and other animal societies. SI provides a basis with which it is possible to explore distributed optimisation problems without centralized control or the provision of a global model. Initial studies have unveiled a great deal of matching properties between the routing requirements of ad hoc networks and certain features of SI, such as the ability of ant colony to find a nearly optimal route through indirect communication between the elements. There are some notable algorithms that use ant-like mobile agents to maintain routing and topology discovery for both wired and wireless networks [9, 10, 11, 12]. These algorithms show that the biologically inspired concepts can provide a significant performance gain over traditional approaches. However, they still suffer from scalability problems due to the use of partial global information (record the global information in ant agents) or insufficient use of local information.

In this paper we explore the SI approach to address the routing problem in ad hoc networks with the goal to reduce routing overhead. Based on the transfer of ant colony social behaviour to the context of ad hoc networks, we propose a self-organised Emergent Ad hoc Routing Algorithm (EARA) that uses the concept of *pheromone* trails to reinforce optimal/sub-optimal paths, using only *local* (neighbour-to-neighbour) communication without knowing the global topology, and the concept of *stigmergy* to reduce the amount of control traffic, especially in highly dynamic networks. Local communication implies that all the information a node can know is from its neighbours. Thus, the resulting paths may be sub-optimal over a short-time span, while the global routing optimisation evolves eventually by the indirect communication of the stigmergy paradigm. Considering the dynamic nature and the resource constraints confronted by MANETs, this evolutional approach is sensible in two aspects. Firstly, for a highly dynamic network, it is acceptable to trade off some accuracy for resource efficiency, since accurate computation of routing information in real time is uneconomical. Secondly, for a stable network, the global optimisation can evolutionally emerge. Because this algorithm does not depend on the periodic advertisement and global dissemination of connectivity information, the routing overhead is substantially less than those in the protocols that necessitate such advertisements. The results from simulation of mobile ad hoc networks confirm that the performance of this algorithm scales well over a variety of environmental conditions, such as network size, nodal mobility and traffic loads.

EARA uses symmetric links between neighbour nodes. It does not attempt to follow paths between nodes when one of the nodes cannot hear the other. The mechanism to prevent the use of asymmetric links is described in Section 3.3.

2 Foraging Strategies in Ants

One famous example of biological swarm social behaviour is the ant colony foraging [13]. Many ant species have trail-laying trail-following behaviour when foraging: individual ants deposit a chemical substance called *pheromone* as they move from a food source to their nest, and foragers follow such pheromone trails. Subsequently, more ants are attracted by these pheromone trails and in turn reinforce them even more. As a result of this autocatalytic effect, the optimal solution will emerge rapidly. In this food searching process a phenomenon called *stigmergy* plays a key role in developing and manipulating local information. It describes the indirect communication of individuals through modifying the environment.

From self-organisation theory point of view, the behaviour of the social ant can be modelled based on four elements: positive feedback, negative feedback, randomness and multiple interactions. This model of social ants using self-organisation theories provides powerful tools to transfer knowledge about the social insects to the design of intelligent decentralised problem-solving systems, swarm-intelligence systems.

3 The Emergent Ad Hoc Routing Algorithm (EARA)

EARA is an on-demand multipath routing algorithm. Inspired by the ant foraging intelligence, this algorithm uses positive feedback originated from destination nodes to reinforce the existing pheromone on good paths. Ant-like packets, analogous to the ant foragers, are used to locally find new paths. Artificial pheromone is laid on the communication links between nodes and data packets are biased towards strong pheromone but the next hop is chosen probabilistically. To prevent old routing solutions from remaining in the current network status, exponential pheromone decay is adopted as the negative feedback.

Each node using this algorithm maintains a *probabilistic routing table*. This routing table serves in a probabilistic fashion. Each route entry for the destination d is associated with a list of neighbour nodes j. A probability value $P_{i,j,d}$ in the list expresses the goodness of node j as the next hop to the destination d. For each neighbour, the shortest hop distance to the destination and the largest sequence number seen so far are also recorded.

The routing table is updated as follows. First, in addition to the routing table, each node also possesses a *pheromone table*. This table tracks the amount of pheromone on each neighbour link. The table may be viewed as a matrix with rows corresponding to neighbourhood and columns to destinations. There are three threshold values controlling the bounds on pheromone in the table. They are the *upper pheromone* $\mathcal{U}$ that prevents extreme differences in pheromone, the *lower pheromone* $\mathcal{L}$, below which data traffic cannot be forwarded, and the *initial pheromone* τ_0 that is assigned when a new route is found.

Next, the routing probability value $P_{i,j,d}$ is computed by the composition of the pheromone values and the local heuristic values as follows:

$$P_{i,j,d} = \frac{[\tau_{i,j,d}]^{\alpha}[\eta_{i,j}]^{\beta}}{\sum_{l \in \mathcal{N}_i}[\tau_{i,l,d}]^{\alpha}[\eta_{i,l}]^{\beta}}, \qquad \tau_{i,j,d} > \mathcal{L} \tag{1}$$

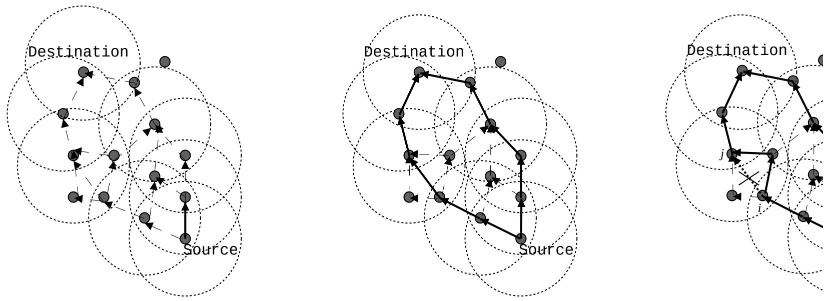
where α and β ($\alpha, \beta \geq 0$) are two tunable parameters that control the relative weight of pheromone trail $\tau_{i,j,d}$ and heuristic value $\eta_{i,j}$, $\mathcal{N}_i$ is the neighbourhoods as a next-hop to some destination d. With $\tau_{i,j,d} > \mathcal{L}$, data traffic can only be forward following the valid route.

The heuristic value $\eta_{i,j}$ is a measure of congestion in a node. Incorporating the heuristic value in the routing computation makes this algorithm possess the congestion awareness property. Based on the probabilistic routing table, data traffic will be distributed according to the probabilities for each neighbour in the routing table. The routing algorithm exhibits load balancing behaviour. Nodes with a large number of packets in the buffer are avoided.

The EARA algorithm consists of several elements. Multiple routes are found with the *route discovery* procedure, good quality routes are reinforced and maintained with *positive* and *negative* feedback, and route failures are handled with the *local connectivity management*.

3.1 The Route Discovery

On initialisation, a neighbourhood for each node is built using the single-hop HELLO messages. The HELLO packet contains *source IP address* and *hop count* (set to 0). This packet can be replaced by the link or network layer mechanism, such as CTS and RTS in IEEE 802.11 or ICMP Echo



(a) Initial Pheromone Setup (b) Path Reinforcement (c) Local Re|

Figure 1: Illustrating Working Mechanism of EARA

Request packet, if available. Whenever a traffic source s needs a route to a destination d, it broadcasts *route request* packets (RQ) across the network. The RQ packet contains *destination IP address*, *source IP address*, and *broadcast ID*. A straightforward solution to route discovery would be through flooding.

During the course of flooding RQ packets to the destination d, the intermediate node j receiving a RQ packet first sets up reverse paths to the source by recording the source address and the previous hop node in the *message cache*. This cache records the incoming packet information of destination address, source address, previous-hop neighbour, the shortest hop distance from the source and the number of incoming packets. If a valid route to the destination d is available, that is, there is at least one link associated with the pheromone trail greater than the lower pheromone bound $\mathcal{L}$, the intermediate node j generates a *route reply* (RP). This packet contains *source IP address*, *destination IP address*, *sequence number*, *hop account*, and *life time*. The RP is routed back to the source s via the reverse paths. Otherwise, the RQ is rebroadcast.

Observing the fact that the flood of RQ packets across the network would potentially constitute multiple paths to the destination d, we make effort to make use of this underlying information as much as possible. Rather than just establishing a single forward path, when the destination node d receives RQs it will send a RP to *all* the neighbours from which it sees a RQ. In addition, in order to maintain multiple loop-free paths at each intermediate node j, node j must record all new forward paths that possess the latest sequence number but hold a lower hop-count in its routing table, and also send a RP to all the neighbours from which it saw a RQ. During the course of the RP tracking back to the source s, it assigns an initial pheromone value τ_0 to the corresponding neighbour node, which indicates a valid route to the destination d. This procedure is illustrated in Figure 1(a).

3.2 The Route Reinforcement

In the scheme we have described so far, the route discovery procedure sets up the initial pheromone trails for the destination node d. After the destination node d receives the data traffic sent by the source node s, it begins to *reinforce* some good neighbour(s) n in order to "pull" more data traffic through the good path(s) by sending *reinforcement signal* packets (RS) whenever it detects new good paths. The RS packet contains *destination IP address*, *sequence number* and *pheromone value* $\Delta\tau$. When node n receives a RS, it knows it has an outgoing link toward the destination d, which is currently deemed a good path. Subsequently, node n updates the corresponding pheromone table entry with the value $\Delta\tau$ and forwards a RS packet to (at least one) selected neighbour locally based on its message cache, e.g. the neighbour(s) that saw the least hops of the incoming packets. The amount of the pheromone $\Delta\tau$ used to positively reinforce the previous hop neighbour should be calculated based on the empirically estimated metrics.

In our implementation, the pheromone value $\Delta\tau^\pi$ carried on the RS packet π is computed as follows. If the RS packet π is sent by the destination to node n, then $\Delta\tau^\pi_{n,d}$ is calculated using the upper bound pheromone value $\mathcal{U}$,

$$\Delta\tau^\pi_{n,d} = \mathcal{U} \cdot e^{-(\alpha'\Delta h_s + \beta'\hat{n})} \tag{2}$$

If the RS packet π is sent by an intermediate node j towards node i, the $\Delta\tau^\pi_{i,j,d}$ is calculated using the current largest pheromone value max($\tau_{j,k,d}$) in node j with the next hop k to the destination d in the pheromone table,

$$\Delta\tau^\pi_{i,j,d} = \max(\tau_{j,k,d}) \cdot e^{-(\alpha'\Delta h_{s,j} + \beta'\hat{n}_{j,d})} \tag{3}$$

where α' and β' are two parameters that control the relative weight of the relative source hop distance $\Delta h_{s,j}$ and the relative packet number $\hat{n}_{j,d}$.

The relative source hop distance $\Delta h_{s,j}$ is calculated as follows:

$$\Delta h_{s,j} = h^i_{s,j} - h^{min}_{s,j} \tag{4}$$

where $h^i_{s,j}$ is the shortest hop distance from the source s to the current node j through node i, and h^{min}_s is the shortest hop distance from s to j. This parameter is used to ensure that paths with shorter hop distance from the source node to the current node are reinforced with more pheromone.

The relative packet number $\hat{n}_{j,d}$ is calculated as follows:

$$\hat{n}_{j,d} = 1 - n^i_{j,d}/n_{j,d} \tag{5}$$

where $n^i_{j,d}$ is the number of incoming packets from neighbour i to the destination d, and $n_{j,d}$ is the total number of incoming packet towards the destination d. This parameter is used to indicate that the data forwarding capacity of a link also affects the reinforcement. The more data arrives, the stronger the reinforcement for the corresponding link.

On receiving the RS, node i needs to positively increase the pheromone of the link towards node d. Node i also has to decide to reinforce (at least) one of its neighbours by sending the RS message based on its own message cache. Now, to node i's neighbours, this RS message appears to originate from the sending node i, although this information intrinsically comes from the destination node d. This is an example of *local* interaction. This process will continue until reaching the source node s. Consequently, good quality routes emerge from this procedure as shown in Figure 1(b). In order to enable the reinforcement scheme above, it must be possible to distinguish individual neighbours. All local unique neighbour identifiers may be used, e.g. 802.11 MAC address.

So far, we have discussed the situations in which reinforcement is triggered by the destination node d. However, as long as an intermediate node has pheromone value that is greater than the lower bound $\mathcal{L}$, it can also proactively apply the same local reinforcement rule to its neighbours. This is useful to enable the local repair of degraded links. For instance, if an intermediate node i detects a link failure from one of its upstream links $l_{i,j}$, it can apply the reinforcement rules to discover an alternative path as shown in Figure 1(c). Instead of seeking global optimality, this local interaction of nodes only results in sub-optimality.

3.2.1 Local Foraging Ants

In a dynamic network like MANET, the changes of the network topology create chances for new good paths to emerge. In order to make use of this phenomenon, this algorithm launches *local foraging ants* (LFA) with a time interval T_{ant} to locally search new routes whenever all the pheromone trails of a node towards some destination drop below the threshhold τ_0. The LFA packet contains *source IP address* (the node that sent LFA), h_{min} (the shortest hop distance from the source to the destination), *stack* $\mathcal{S}$ (recording intermediate node IP addresses), ant_ttl (we define ant_ttl as 5), and *hop count*. The LFA will take a random walk from its original node. During the course of its walk, the LFA pushes the address of the nodes that it has travelled into its memory stack $\mathcal{S}$. To avoid forming of loop, LFA will not choose the node that is already in $\mathcal{S}$. Before reaching the maximum hop, if LFA can find a node with pheromone trails greater than τ_{ant} and the hop distance to destination not greater than the one from its original nest, it returns to its 'nest' following its memory stack $\mathcal{S}$ and updates the corresponding paths with τ_0. Otherwise, it simply dies.

3.2.2 Pheromone Table Update

The update of the pheromone table includes two aspects: positive reinforcement and negative reinforcement. In the scheme described above, whenever node i receives the RS from its neighbour j, if the sequence number in the RS is greater than the one recorded in the pheromone table, node i updates its corresponding pheromone $\tau_{i,j,d}$ with the value of $\Delta\tau^{\pi}_{i,j,d}$ carried on the RS π:

$$\tau_{i,j,d} := \Delta\tau^{\pi}_{i,j,d} \tag{6}$$

If the sequence number is equal to the current one, then:

$$\tau_{i,j,d} := \begin{cases} \Delta\tau^{\pi}_{i,j,d}, & \text{if } \tau_{i,j,d} < \Delta\tau^{\pi}_{i,j,d} \\ \tau_{i,j,d}, & \text{otherwise} \end{cases} \tag{7}$$

If the sequence number in RS is less than the current one in the pheromone table, then this RS is just discarded.

There is also a mechanism to time out the existing paths, which plays a role of implicit negative reinforcement. Within every time interval T_{dec}, if there is no data towards a neighbour node its corresponding pheromone value decays by a factor ρ as follows:

$$\tau_{i,j,d} := (1-\rho)\cdot\tau_{i,j,d}, \qquad \rho \in (0,1] \tag{8}$$

During the course of pheromone decay, if all the pheromone for a particular node are equal to the lower pheromone bound $\mathcal{L}$, the corresponding row and/or column is removed from the pheromone table, which means that no data packets have been received for some time. If a particular destination is also a neighbour node, it cannot be removed unless all the entries in that neighbour row are also decayed. If a neighbour is determined to disappear from the communication range, the corresponding row can be simply removed from the pheromone table.

3.3 Local Connectivity Management

When link failures happen, the MAC layer has to go through multiple transmissions to conclude that a link has failed, which results in time gap between the occurence of link failure and its detections. We use periodic beacons in the routing layer to reduce this time gap. Nodes maintain their local connectivities in two ways. Whenever a node receives a packet from a neighbour, it updates its local connectivity information to ensure that it includes this neighbour. In the event that a node has not sent any packets to its neighbours within a time interval T_{hel}, it has to broadcast a HELLO packet to its neighbours. Failure to receive packets from the neighbourhood in T_{hel} indicates changes in the local connectivity. If HELLO packets are not received from the next hop along an active path, the node that uses that next hop is sent notification of link failure.

In case of a route failure occuring at node i, i cannot forward a data packet to the next hop for the intended destination d. Node i sends a RS message that sets ROUTE_RERR tag to inform upstream nodes of the link failure. This RS signal assigns to the corresponding links the lower bound $\mathcal{L}$. Here, RS plays the role of an explicit negative feedback signal to negatively reinforce the upstream nodes along the failure path. This negative feedback avoids causing buffer overflow due to caching on-flight packets from upstream nodes.

Each node that encounters a route failure should try to forward data that arrives before the route failure through alternative paths that are associated with valid pheromone trails. If there is no valid outgoing link at the moment, the data should be held for some time T_{err} and the node waits for alternative routes emerging from the reinforcement procedure or new routes found by local foraging ants (LFA). After T_{err}, if there is still no proper path emerging, then this node can just drop the data. This is essential to guarantee good performance of this algorithm.

Moreover, the use of HELLO packets can also help to ensure that only nodes with bidirectional connectivity are deemed as neighbours. For this purpose, the HELLO packet sent by a node has an option to list the nodes from which it has heard HELLO packets, and nodes that receive the HELLO check to ensure that it uses only routes to neighbours that have sent HELLO packets.

4 Evaluation Methodology

To evaluate the performance of the EARA protocol, we carried out a series of simulations with the simulator *ns*-2.

4.1 The Simulation Configurations

We use the IEEE 802.11 Distributed Coordination Function (DCF) as the MAC layer protocol. The radio model simulates Lucent's WaveLAN with a nominal bit rate of 2Mbps and a nominal transmission range of 250 meters. The radio propagation model is the two-ray ground model. The mobility model we use is the *Random Waypoint Model*. In our simulation, we define the minimum velocity v_{min} as 1 m/sec and the maximum one v_{max} as 20 m/sec. We use the pause time of the node mobility as the independent variable that reflects the degree of the node mobility.

We performed two sets of experiments. One set of these experiments was performed using 50 nodes in a rectangular field of 1500m$\times$300m, and the other set was performed using 150 nodes in a rectangular area of 2000m$\times$700m. All experiments use CBR (constant bit rate) traffic with a sending rate of 4 512-byte packets/second. For each set, the network consists of 5, 10, 15 and 30 traffic flows. The results of this simulation reflect the impact of different nodal mobility, network density and traffic loads. Each scenario uses the same simulation parameters as listed in Table 1. The simulation time is 600 seconds and we took 20 runs of the simulations. The metrics we used to evaluate the performance of our algorithm are the packet delivery ratio, the average end-to-end delay, path optimality and the transmission optimality.

Table 1: Protocol Parameters

$\mathcal{U}$	50	$\mathcal{L}$	0	τ_0	1	ρ	0.5
α^0	1	β^0	1	α	1	β	1
T_{err}	60s	T_{ant}	10s	T_{dec}	50s	T_{hel}	5s

4.2 Results and Discussion

The results from the two sets of experiments are shown in Figure 2 and Figure 3. In these graphs, the error bars shown represent the confidence interval of 95% out of 20 runs.

Firstly, we discuss the robustness aspect of the routing algorithm. Figure 2(a) and Figure 3(a) show the packet delivery ratio under different conditions. When both data loads and nodal motion are low, EARA can deliver almost all the packets. As nodal motion increases, the delivery ratio drops. With the data load increasing, the system slightly degradtes. The results of the two sets of experiments are close, but the overall performance of the second degrades by about 3%. From the results, we can see that EARA data delivery ratio scales fairly well to both the nodal mobility and network size. This is because, instead of buffering data packets waiting for a new route to be found, EARA forwards the data packets through alternative routes whenever route errors happen due to its multiple paths nature. This mechanism ensures data delivery as much as possible.

Next, we consider the quality of service provided by the protocols using the average delay as shown in Figure 2(b) and Figure 3(b). EARA shows low transmission delay in all cases. When either the nodal motion or the network size increases, the average delay slightly increases. More data traffic can also result in longer delay. The transmission delay can be affected by several factors such as nodal mobility, network size and data loads. EARA presents a reasonable level of scalability with respect to nodal mobility and network size. The reason for this scalable low transmission delay is that EARA periodically explores new potential paths, and hence does not suffer the long transmission delay under the highly dynamic environments as traditional ad hoc routing protocols do.

In order to evaluate the effectiveness of the routing algorithm, we explore the path optimality and transmission optimality. Figure 2(c) and Figure 3(c) show the average route length used in sending a data packet relative to the optimal route length. This metric shows the degree to which the algorithm finds and maintains optimal routes. As shown here, for the moderate nodal motion, the optimality is within a factor of 1.2. For very short pause times, the maximum ratio is about 1.5 of the optimal one. When either the network size or data load increases, the path optimality slightly increases. From the results, the main factor on path optimality is the nodal motion. At the highest nodal motion, the length of used path can be almost two times of the optimal one. This indicates that most paths evolved by EARA are sub-optimal. Figure 2(d) and Figure 3(d) show the total number of packet transmissions performed relative to the optimal number of transmissions. The ratio of 1 indicates a perfect algorithm without any routing overhead. At a moderate nodal motion, the average optimality is a

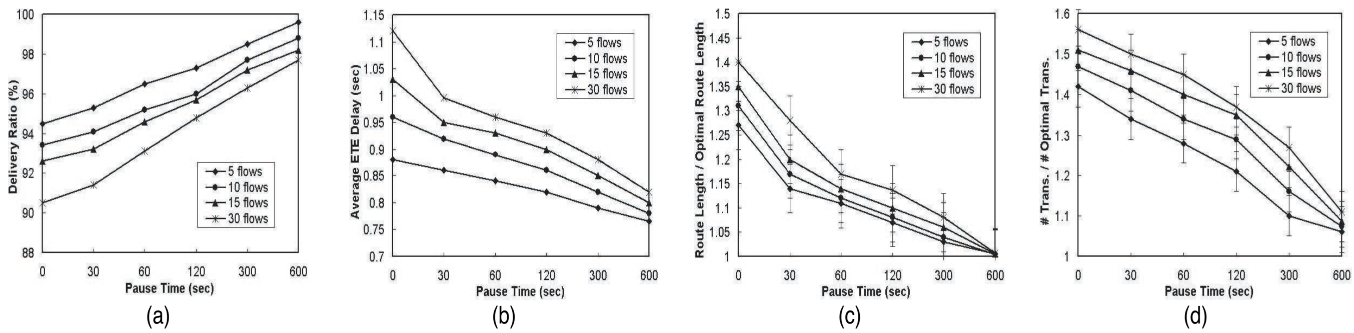


Figure 2: Simulation in an area of 1500m×300m with 50 mobile nodes

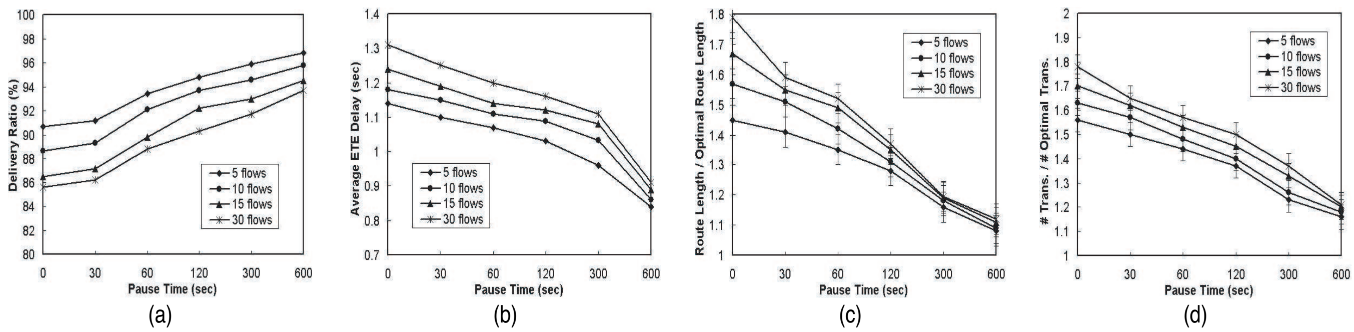


Figure 3: Simulation in an area of 2000m×700m with 150 mobile nodes

ratio of 1.15. At the highest nodal motion, the overhead can reach a value of 1.42 in the first set of experiments. With the network size and the data loads increasing, the algorithm overhead also increases. The average overhead of the second set of experiments is about 8% higher than that of the first one. From the results we can see that the overhead difference of EARA between different scenarios is much smaller than that of the traditional protocols. This is mainly because EARA uses the concept of stigmergy to implement local communication to maintain routing status, which is independent of nodal motion. This feature is a significant advantage of EARA over other routing protocols.

5 Conclusions and Future Work

In this paper we present a novel routing algorithm for mobile multi-hop ad hoc networks. Through the concept of stigmergy, optimal/sub-optimal routes emerge without the system-wide dissemination of connectivity information. Moreover, this algorithm is able to adapt to changes of the network topology without invoking high routing overhead. The results of experiments show that our algorithm performs fairly well under various network situations.

Since this algorithm uses the hop distance as the metric to determine routing, it encounters some performance degradation under heavy data traffic due to contention in the MAC layer and congestion in the transport layer. In our furture investigations, we will explore the use of metrics from other network layers to manipulate the pheromone concentration on the edges, which can influence the performance of this algorithm.

References

[1] C. E. Perkins and P. Bhagwat, Highly Dynamic Destination-Sequenced Distance-vector Routing (DSDV) for mobile computers, *ACM SIGCOMM: Computer Communications Review*, 24(4), 1994, 234-244.

[2] P. Jacquet, P. Muhlethaler and A.Qayyum, Optimised Link State Routing Protocol, IETF MANET, Internet Draft, 1998.

[3] S. Murthy and J. J. Garcia-Luna-Aceves, A Routing protocol for Packet Radio Networks, *Proc. 1st ACM Conf. on Mobile Computing and Networking*, Berkeley, USA, 1995, 86-95.

[4] D. Johnson and D. Maltz, Dynamic Source Routing in Ad hoc Wireless Networks, *Mobile Computing*, 353, 1996.

[5] C. E. Perkins and E. Royer, Ad-hoc On-Demand Distance Vector Routing, *Second IEEE Workshop on Mobile Computing systems and Applications*, New Orleans, USA, 1999, 90-100.

[6] V. D. Park and M. S. Corson, A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks, *Proc. IEEE Conf. on Computer Communications* , Kobe, Japan, 1997, 1405-1413.

[7] Y. B. Ko and N. H. Vaidya, Location-Aided Routing (LAR) in Mobile Ad Hoc Networks, *Proc. IEEE/ACM Conf. on Mobile Computing and Networking*, Dallas, USA, 1998, 66-75.

[8] S. Basagni, I. Chlamtac, V. R. Syrotiuk and B. A. Woodward, A Distance Routing Effect Algorithm for Mobility (DREAM), *Proc. IEEE/ACM Conf. on Mobile Computing and Networking* , Dallas, USA, 1998, 76-84.

[9] M. Güneş, U. Sorges, and I. Bouazisi, ARA - the Ant-Colony Based routing Algorithm for MANETs, *Proc. ICPP Workshop on Ad Hoc Networks*, Vancouver, Canada, 2002, 79-85.

[10] J. S. Baras and H. Mehta, A Probabilistic Emergent Routing Algorithm for Mobile Ad Hoc Networks, *Proc. Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, Sophia-Antipolis, France, 2003.

[11] D. Câmara and A. Loureiro, A Novel Routing Algorithm for Hoc Networks, *Proc. 33rd Hawaii Conf. on System Sciences*, Hawaii, USA, 2002, 8022.

[12] M. Heissenbüttel and T. Braun, Ants-based Routing in Large Scale Mobile Ad-hoc Networks, Institue of Computer Science and Applied Mathematics, University of Berne, Tech. Rep. CH-3012, 2003.

[13] R. Beckers, J. L. Deneubourg and S. Goss, Trails and U-turns in the Selection of the Shortest Path by the Ant Lasius Niger, *Journal of Theoretical Biology*, 159, 1992, 397-415.