A Biologically Inspired Congestion Control Routing Algorithm for MANETs

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

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

Abstract

This paper presents EARA-CG, an improved version of a swarm-intelligence ad hoc routing algorithm EARA introduced in [10]. Based on the understanding of the evolutionary cooperation in the biological swarm, we use the principle of swarm intelligence to reinforce good quality routes with only local communication. The concept of stigmergy is adopted to reduce the amount of control traffic. By locally monitoring the network transmission queue and other MAC layer information, this algorithm can forward data traffic through paths that avoid network hotspots. The data traffic influences routing table updates at each node. We also discuss an evaluation methodology to simulate ad hoc networks, and the simulation results show that this novel routing algorithm scales well to a variety of network conditions.

1. Introduction

Mobile ad hoc networks (MANETs) are wireless mobile networks formed spontaneously. Unlike in the traditional wireless networks, communication in such a decentralised network is typically multi-hop, with the nodes using each other as relay routers without any fixed infrastructure. This kind of network is very flexible and suitable for applications such as temporary information sharing in a conference, 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 longterm efficiency. Furthermore, compared with the traditional networks, MANETs suffer from the resource constraints in energy, computational capacities and bandwidth. All of these make routing in MANETs a very challenging problem.

To address the routing challenge in MANETs, many approaches have been proposed in the literature. Based on the routing mechanism for the traditional networks, the proactive approaches attempt to maintain routing information for each node in the network at all times [7, 11, 13], whereas the reactive approaches only find new routes when required [8, 12, 14]. Other approaches make use of geographical location information for routing [2,9]. In most of the previous works, the number of hops is the most common criterion to determine routing. However, in ad hoc networks, some areas of the network are likely to have higher data loads than other areas. This can make certain areas prone to congestion, thus decreasing the overall network performance. In that case, criteria based on the number of hops do not suffice as a metric for making routing decisions.

Another concern of this paper is that in previous works researchers worked mainly at the network layer. While many optimisations to these above algorithms exist, [15] indicates that optimising a particular single layer might improve the performance of that layer locally, but might produce non-intuitive side-effects that will degrade the overall system performance.

In this paper, we propose a new improved version of self-organised Emergent Ad hoc Routing Algorithm (EARA) [10] that uses a light-weight mechanism for determining network congestion, based on information from the MAC layer, to improve algorithm performance. This algorithm, which we call EARA-CG, is inspired by the foraging behaviour of biological ants when discovering paths to food sources. EARA-CG 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. Consequently, 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.

Our algorithm adopts the cross-layer design approach, utilising parameters from different layers to achieve overall system optimisation. The parameters for measuring local network congestion around a node depend largely on the MAC layer. In this paper, we focus on the IEEE 802.11 DCF mode, since it is the most widely used wireless LAN standard. By using the wireless medium information from the MAC layer, EARA-CG prevents the discovery of routes over which it is undesirable to carry additional data traffic since the wireless medium over those hops is already very busy. The results from simulation of mobile ad hoc networks show that the performance of this algorithm scales well over a variety of environmental conditions, such as network size, nodal mobility and traffic loads.

EARA-CG 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 Backgrounds

In this section, we will give a brief introduction to swarm intelligence and the parameters from the MAC layer.

2.1 Social Insects

One famous example of biological swarm social behaviour is the ant colony foraging [3]. 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.

2.2 Congestion Metrics

The major parameters that are used to measure local network congestion reside in the MAC layer. We adopt two metrics to reflect the congestion level around a node.

The first metric is the average MAC layer utilisation ϕ_i for a node i. This metric provides the node with a view of current status of the shared wireless medium. The instantaneous MAC layer utilisation at a node is either 1 (busy) or 0 (idle). In our algorithm, we average the time T_{busy} when the medium is busy over a period of time T_{mac} to reflect the usage of the wireless medium around that node as follows:

$$\phi_i = \frac{\sum T_{busy}}{T_{mac}}, \qquad T_{busy} \subseteq T_{mac} \tag{1}$$

We deem the wireless medium around a node to be busy when the MAC layer at that node either detects physical carrier present or is deferring due to virtual carrier sensing, interframe spacing, or backoff. In addition, we also consider the medium is busy at any time when the node has at least one packet in the transmission queue.

The second metric is a heuristic value η that is calculated with the network interface transmission queue length in the current node. Apart from the media status, the transmission queue length is also a key factor that can affect the packet latency or packet drop due to the size limit on the queue length. The heuristic value is set with the following rule:

$$\eta_{i,n} = 1 - \frac{q_{i,n}}{\sum_{l \in \mathcal{N}_i} q_{i,l}} \tag{2}$$

where $q_{i,n}$ is the length (in bits waiting to be sent) of the outgoing queue of the link $l_{i,n}$ in node i and $\mathcal{N}_i$ denotes the neighbourhood of node i as a next-hop to some destination.

3 Description of EARA-CG

EARA is an on-demand multi-path 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.

Parameters from the MAC layer are used to attempt to make routing decisions taking into account the status of the local wireless medium, which helps to avoid choosing routes through congested areas of the network.

Each node that uses this algorithm maintains a *probabilistic routing table*, according to which routes are selected 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. In addition to the routing table, each node also possesses a *pheromone table*. This table tracks the amount of pheromone on each neighbour link. There are three threshhold values controlling the bounds on pheromone in the table. They are the *upper pheromone* $\mathcal{U}$, the *lower pheromone* $\mathcal{L}$, and the *initial pheromone* τ_0.

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

$$P_{i,j,d} = \frac{[\tau_{i,j,d}]^{\alpha}[\eta_{i,j}]^{\beta}}{\sum_{l \in \mathcal{N}_i}[\tau_{i,l,d}]^{\alpha}[\eta_{i,l}]^{\beta}}, \qquad \tau_{i,j,d} > \mathcal{L} \tag{3}$$

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 neighbourhood as a next-hop to some destination d. With $\tau_{i,j,d} > \mathcal{L}$, data traffic can only be forward following a valid route.

As explained before, the heuristic value $\eta_{i,n}$ 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 for each neighbour. The routing algorithm thus exhibits load balancing behaviour. Nodes with a large number of packets in the buffer are avoided.

The EARA-CG algorithm consists of several components. They are the *route discovery* procedure, the positive and negative *reinforcement* , and the *local connectivity management*.

3.1 Route Discovery

We use a similar route discovery procedure as described in [10]. On initialisation, a neighbourhood for each node is built using the single-hop HELLO messages. Whenever a traffic source s needs a route to a destination d, it broadcasts *route request* packets (RQ) across the network. Rather than simply flooding the RQ packets, we adopt the probabilistic broadcast scheme explored in [6] combined with the MAC layer utilisation ϕ. When a node first receives a packet, with probability p' it broadcasts the packet to its neighbours, and with probability $1-p'$ it discards the packet. The probability value p' is calculated as

$$p' = e^{-\sigma\phi} \tag{4}$$

where σ ($\sigma > 0$) is the coefficient. This broadcast scheme helps to discover new routes avoiding congestion areas, but at the cost of missing potential routes to the destination.

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*. 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). The RP is routed back to the source s via the reverse paths. Otherwise, the RQ is rebroadcast.

Other 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 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.

3.2 Route Reinforcement

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. 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 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 $\mathcal{U}$,

$$\Delta\tau^{\pi}_{n,d} = \mathcal{U} \cdot e^{-(\alpha'\Delta h_s + \beta'\hat{n})} \tag{5}$$

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,

$$\Delta\tau^{\pi}_{i,j,d} = \max(\tau_{j,k,d}) \cdot e^{-(\alpha'\Delta h_{s,j} + \beta'\hat{n}_{j,d})} \tag{6}$$

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:

$$\Delta h_{s,j} = h^{i}_{s,j} - h^{min}_{s,j} \tag{7}$$

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:

$$\hat{n}_{j,d} = 1 - n^{i}_{j,d}/n_{j,d} \tag{8}$$

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 reinforcement is generated for the corresponding link.

On receiving the RS from a neighbour n, node i needs to positively increase the pheromone of the link towards node d. if the sequence number in the RS is greater than the one recorded in the pheromone table, node i updates its corresponding pheromone $\tau_{i,n,d}$ with the value of $\Delta\tau^{\pi}_{i,n,d}$ carried on the RS π:

$$\tau_{i,n,d} := \Delta\tau^{\pi}_{i,n,d} \tag{9}$$

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

$$\tau_{i,n,d} := \begin{cases} \Delta\tau^{\pi}_{i,n,d}, & \text{if } \tau_{i,n,d} < \Delta\tau^{\pi}_{i,n,d} \\ \tau_{i,n,d}, & \text{otherwise} \end{cases} \tag{10}$$

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

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. This process will continue until reaching the source node s. The same procedure can apply to any intermidate node as long as it has pheromone value that is greater than the lower bound $\mathcal{L}$.

There is also an implicit negative reinforcement for the pheromone values. Within every time interval T_{decay}, if there is no data towards a neighbour node, its corresponding pheromone value decays by a factor ρ as follows:

$$\tau_{i,n,d} := (1-\rho)\cdot\tau_{i,n,d}, \qquad \rho \in (0.1] \tag{11}$$

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 will take a random walk from its original node.

During the course of its walk, if the LFA detacts congestion around a node (the average channel utilisation ϕ is greater than a predifined threshhold value ϕ'), then the LFA dies to avoid increasingly use the wirless medium. Otherwies, the LFA pushes the address of the nodes that it has travelled into its memory stack $\mathcal{S}$. To avoid forming of loops, LFA will not choose to travel to 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.

3.3 Local Connectivity Management

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.

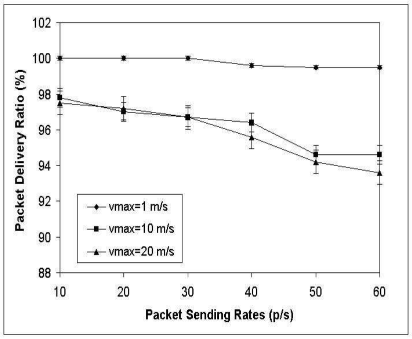
In case of a route failure occuring at node i, i cannot forward a data packet to the next hop for the intended destination d. Node i sends a RS message that sets ROUTE_RERR tag to inform upstream nodes of the link failure. This RS signal assigns to the corresponding links the lower bound $\mathcal{L}$. 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.

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.

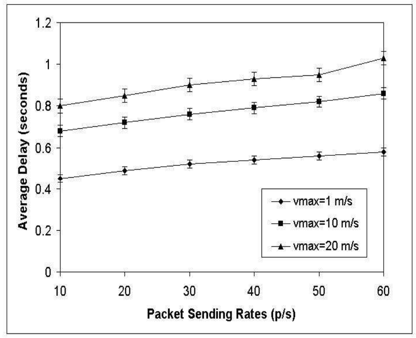
4 Simulations

To evaluate the performance of the EARA-CG protocol, we performed simulations using simulator *ns*-2 [1]. We use the IEEE 802.11 Distributed Coordination Function (DCF) [5] as the MAC layer protocol. The radio model simulates Lucent's WaveLAN with a nominal bit rate of 2 Mbps 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* [4]. In our simulations, we define the v_{min} as 1 m/sec and the v_{max} as 20 m/sec. Pause time is set to 0 seconds representing the continuous nodal motion throughout the simulation. We performed two sets of simulations to study the performance of the routing algorithm in different load regimes.

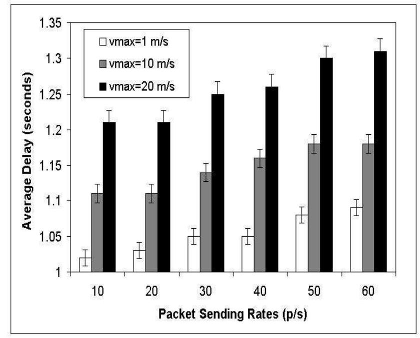
The first set was performed using 50 nodes in a rectangular field of 1500m$\times$300m, and the other set was performed using 100 nodes in an area of 1000m$\times$1000m. Both of them use CBR (constant bit rate) traffic with 512-byte packets. The first set has 10 CBR flows and the other has 20 flows. All traffic flows are generated randomly. We varied data



(a) Delivery Ratio Vs. Load



(b) Delay Vs. Load



(c) Path Opti. Vs. Load

Figure 1. Simulation of 50 nodes in an area of 1500m$\times$300m

load by changing the packet sending rate. For each set, we varied the packet rate from 10 packet/second to 60 packets/second at a step of 10 packets/second. 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.

We evaluate the performance of our algorithm with three metrics, namely

- *Packet delivery ratio* that is the fraction of packets sent by a source node to that received by the corresponding destination node.
- *Average ETE delay* that reflects the total time needed to successfully deliver a packet by a source node till it is received by the corresponding destination node.
- *Path optimality* that is the fraction of delivered data packets that were routed by the protocol over routes of various lengths relative to the shortest optimal route (the shortest optimal route is determinded by the simulator).

4.1 Results and Discussion

The results from the experiments are shown in Figure 1 and Figure 2. In these graphs, the error bars represent the standard deviation of the mean out of 20 runs.

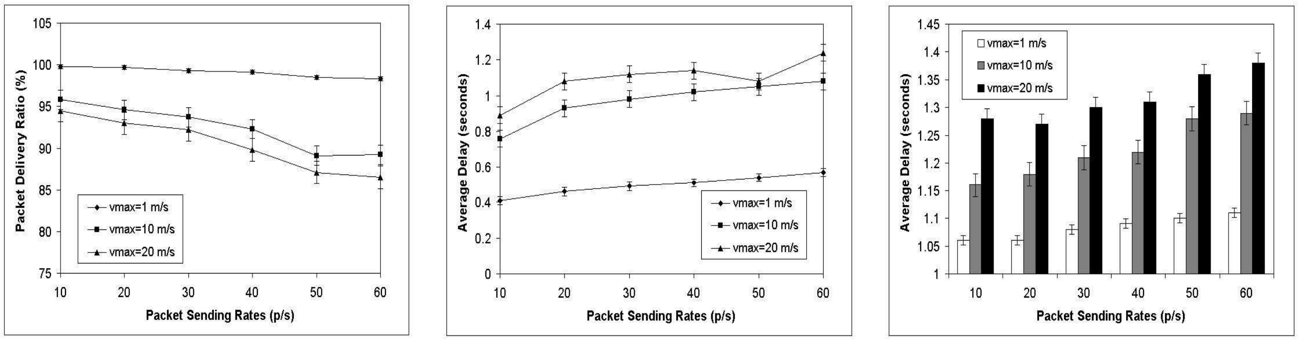
In the first set of simulations, when the offered load is low, EARA-CG can deliver almost all the packets for low mobility scenarios, and nearly 97% packets for high mobility scenarios. As the offered load increases, the packet delivery ratio drops. But the differences between the low load and the high load are fairly small. The packet delivery ratio remains almost stable for low mobility and only drops about 4% for high mobility. This good delivery is due to the multi-path nature of the algorithm and its effort to forward all the packets when link errors happen. The average delay remains fairly stable for all the mobility scenarios as offered load varies. These results indicate that the performance of the algorithm is affected mainly by the nodal motions and is nearly stable for various offered loads. In the experiments, we observed that the congestion built up at some nodes at the packet rate of 50 p/s. However, the congestion alleviated over a short time and the performance of the algorithm falled slightly.

The average path length is about 25% longer than the optimal one for the high nodal mobility and about 6% for the low nodal mobility. This indicates that, as nodal motion increases, the link errors increase. The local repair of the error links resulted in significantly sub-optimal routes. The path optimality slightly increases as the offered load increases. The average actual route distance for the high offered load is about 10% longer than the optimal one. This can be explained by the fact that the algorithm tries to build longer routes to avoid congestion areas.

In the second set, as both the network size and the data flows are bigger than those in the first set, the average performance of the algorithm is slightly worse in terms of packet delivery ratio and transmission delay. When both the nodal motion and the offered load are low, the results of the two sets are close. As the offered load increases, the performance of the second set drops more than that of the first set. As the network size increases, the path optimality also increases slightly.

Table 1. Protocol Parameters

σ	1	$\mathcal{L}$	0	$\mathcal{U}$	50	τ_0	1
α	1	β	1	α^0	1	β^0	1
T_{mac}	5s	T_{hello}	5s	T_{decay}	50s	T_{err}	60s
τ_{ant}	1	ρ	0.5	ant_ttl	5	ϕ'	0.8



(a) Delivery Ratio Vs. Load

(b) Delay Vs. Load

(c) Path Opti. Vs. Load

Figure 2. Simulation of 100 nodes in an area of 1000m$\times$1000m

From the experiments we can see that the performance of this algorithm is stable as the offered load increases. As the nodal motion increases, more sub-optimal routes are built. Although the actual routing path length is often longer than the optimal one, the transmission delay is still low. This is mainly because, although EARA-CG attempts to find alternative routes that might be longer than the optimal one, it avoids the congestion area to save the transmission time.

5 Conclusions

In this paper we present a biologically inspired routing algorithm for mobile multi-hop ad hoc networks. By using the concept of stigmergy, optimal or sub-optimal routes can emerge without the system-wide dissemination of connectivity information. Also, by incorporating the wireless medium information from the MAC layer into the routing decision procedure, this algorithm can prevent forward data traffic across congested areas. The results of experiments show that this cross-layer routing architecture with swarm intelligence inspiration performs well under situations of various nodal mobility, network density and data loads.

References

[1] A Collaboration between UC Berkeley, LBL, USC/ISI and Xerox PARC. *The ns Manual*, 2002. The VINT Project.

[2] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward. A distance routing effect algorithm for mobility (dream). In *Proceedings of the IEEE/ACM international Conference on Mobile Computing and Networking (MOBICOM'98)*, pages 76–84, 1998.

[3] 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*, 159:397–415, 1992.

[4] J. Broch, D. A. Maltz, D. B. Johnson, Y. C. Hu, and J. Yetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM'98)*, 1998.

[5] I. C. S. L. M. S. Committee. Wireless lan medium access control (mac) and physical layer (phy) specifications, 1997.

[6] Z. J. Hass, J. Y. Halpern, and L. Li. Gossip-based ad hoc routing. In *Proceedings of the IEEE Conference on Computer Communication*, 2002.

[7] P. Jacquet, P. Muhlethaler, and A. Qayyum. Optimised link state routing protocol. IETF MANET, Internet Draft, 1998.

[8] D. Johnson and D. Maltz. Dynamic source routing in ad hoc wireless networks. *Mobile Computing*, 353, 1996.

[9] Y. B. Ko and N. H. Vaidya. Location-aided routing (lar) in mobile ad hoc networks. In *Proceedings of the IEEE/ACM International Conference on Mobile Computing and Networking (MOBICOM'98)*, pages 66–75, 1998.

[10] Z. Liu, M. Z. Kwiatkowska, and C. Constantinou. A swarm intelligence routing algorithm for manets. In *Proceedings of the IASTED International Conference on Communications Internet and Information Technology*, St. Thomas, USA, 2004.

[11] S. Murthy and J. J. Garcia-Luna-Aceves. A routing protocol for packet radio networks. In *Proceedings of the 1st annual international conference on Mobile computing and networking*, pages 86–95. ACM Press, 1995.

[12] V. D. Park and M. S. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, pages 1405–1413, 1997.

[13] C. E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. *ACM SIGCOMM: Computer Communications Review*, 24(4):234–244, 1994.

[14] C. E. Perkins and E. Royer. Ad-hoc on-demand distance vector routing. In *the Second IEEE Workshop on Mobile Computing systems and Applications*, pages 90–100, 1999.

[15] E. M. Royer, S. J. Lee, and C. E. Perkins. The effects of mac protocols on ad hoc networks communication. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, volume 2, pages 543–548, 2000.