ISSN : 2231-0711

NISR: A Nature Inspired Scalable Routing Protocol for Mobile Ad Hoc Networks

Sajjad Jahanbakhsh Gudakahriz [1], Shahram Jamali [2], Esmaeel Zeinali [1]

[1]*Computer, Electrical & IT Engineering Department, Islamic Azad University of Qazvin,Qazvin, Iran*

[2]*Computer Engineering Department, University of Mohaghegh Ardabili ,Ardabil, Iran*

Abstract— Mobile Ad hoc Networks (MANETs) are multi-hop wireless networks consisting of radio-equipped nodes that are mobile. The topology of a mobile ad hoc network changes frequently and this dynamic topology makes it difficult to design an efficient routing protocol. Number of nodes in a MANET can be varied from a few nodes to hundreds and even more nodes. Hence it is important for a MANET routing protocol to keep its performance in an acceptable level increasing number of nodes. Since nature is source of many well-established scalable mechanisms, this paper inspires from nature to design its scalable routing protocol. This protocol is based on TORA routing protocol and borrows some principles from bee and ant colonies. Simulation results signify that NISR is very competitive and outperforms TORA in terms of total delivered data, network life time and system life time.

Keywords— **Ad Hoc Networks, Scalable, Routing, TORA, Bee colony, Ant colony.**

I. INTRODUCTION

While the infrastructure cellular system is a traditional model for wireless network, a network that does not rely on an infrastructure and works in a shared wireless media is called a mobile ad hoc network (MANET) [1]. MANETs which don't have any fixed infrastructure can be used in situations with geographical or time constraints. These networks have many applications involving emergencies, communications, natural disasters and military [2]. In MANETs, each node also can act a router, therefore is a self-organizing and self-configuring multi-hop wireless network. Mobility of nodes results in continuously evolving new topologies and dynamic topology and then the routing protocol algorithms have to adapt the routes according to this dynamic topology [3], [4].

Routing protocols for mobile ad hoc networks have been the subject of extensive research over the past several years [5], [6]. Recently, practical applications such as intelligent sensor networks have focused attention on understanding the issues and tradeoffs in network scalability [7]. Scalability is the ability of a network to support the increase of its limiting parameters. Some of limiting parameters in mobile ad hoc networks are mobility rate and network size. In MANETs scalability is more challenging in the presence of both mobility and network size [8]. In this paper we inspire from principles of bee colony and ant colony in nature to present a new scalable routing algorithm for MANETs that is based on TORA routing protocol. The reminder of paper has been organized as follows. Section II introduces behavior of a bee colony and an ant colony in nature. In section III we will describe our routing protocol. Section IV explains the experimental framework and discusses the results obtained from the simulations. Finally a conclusion is given in section V.

II. PRELIMINARIES: BEE AND ANT COLONIES IN NATURE

Colonies of societal insects such as bees and ants have inherent ability called swarm intelligence. This highly organized behavior enables the colonies of insects to resolve problems further than the capability of individual members by functioning collectively and interacting primitively between members of the group [9]. Since this paper uses bee and ant colonies as source of inspiration to design NISR protocol, in this section we briefly take a look over theses natural colonies.

A. Bee colony

A colony of bee can extend itself over wide distances in order to search for food sources at the same time. The foraging process begins in a colony by scout bees that being sent to search for hopeful flower patches. Scout bees move randomly from one path to another [10]. Flower patches with greater amounts of nectar or pollen that can be collected with less challenge tend to be visited by more bees, whereas patches with less nectar or pollen receive fewer bees. Through the harvesting season, a colony continues its search, keeping a percentage of the population as scout bees. When scout bees return to the hive go to the dance floor to do a dance known as the waggle dance. This dance have three pieces of information about a flower patch: (a) the direction in which flower will be found (b) distance from the hive (c) its quality. This information helps the bee colony to send its bees to flower patches specifically, without using guides or maps. When bees are sent to flower patches, in return to hive, by dance inform other bees in hive about new quality of a flower patch that visited. In this way, information of bees about nature are updated [11].

B. Ant colony

Ants are community insects. They live in colonies and their goal is colony survival rather than individuals' survival. Ants when exploring for food source, they initially search the area adjacent their nest in a random manner. When they find food, during the come back to nest put a chemical pheromone trail on the earth. Ants can smell pheromone. When selecting their way, ants tend to choose, paths with more pheromone. Also this ants when are sent to food sources, during come back to nest, put pheromone on the earth. In this way, the indirect communication between the ants via pheromone enables ants to find shortest paths between their nest and food sources [12], [13].

III. NISR ROUTING PROTOCOL

TORA [2] which is the base protocol for the current research, is a reactive multipath routing protocol that discovers all paths between source and destination and finally selects the shortest one. In other words, TORA typically provides multiple routes for any source/destination pair, but it always chooses the route with fewer hops. In this section we enhance this protocol by inspiration from bee and ant colonies and propose a novel nature-inspired routing protocol i.e. NISR.

A. Basic idea: mapping from nature to MANET

Consider a scenario in which ants and bees help each other to find food sources, update quality of paths to these food sources continually, and determine pheromone of paths. Suppose that in this process bees are responsible to discover new food sources and update information about quality of food sources during the time. Also ants determine pheromone of paths. We can describe this collaboration as follows:

- Scout bees travel randomly from one path to another to search for flower patch.
- When scout bees return to the hive, it perform dance to announcement three pieces of information about flower patch: direction, distance from hive and quality of food source.
- By using these information colony sends its bees to food source.
- When bees arrive at food source, an ant and a bee sent to hive, where bee inform new quality of flower and ant update pheromone of path.
- Some scout bees keep on searching and give announcement about new information in the nature. In other words, if new paths exist, these paths are announced by scout bees.

Step 3, step 4 and step 5 are repeated while there is any food source.

We believe that a well-established routing process in a mobile ad-hoc network is so similar to mentioned scenario. In a routing protocol we need to find the best route (the shortest route, the most stable route, the route with highest energy level, etc.) to destination and to monitor its quality during the time. This is similar to mentioned scenario in which bees find paths to food source and ants compute its quality continually. This similarity motivates us to design a new routing protocol, namely NISR, which is inspired from bee and ant colonies and acts as follows:

- QRY packets are broadcasted from source to various paths.
- When QRY packet arrives at destination, UPD packets are sent from destination to source. These UDP packets contain three pieces of information about route: direction of routes, hop counts of routes from source to destination and level of energy in routes.
- By using this information, data packets are sent over routes with high energy level and low hop count and high scalability.
- Every time data packets arrive at destination, energy level and number use of route are update by sending a special packet.
- When new routes created, using UPD packet new routes announced to source.

Step 3, step 4 and step 5 are repeated while there is any data to be sent from source node to the destination node.

B. Details of NISR and its implementation issues

NISR based on TORA, therefore for explain NISR we brought [14] and change it for present NISR. In NSIR, each node has five types of information: H_i, N_i, $HN_{i,j}$, RR_i, $LS_{i,j}$. $H_i = (\tau_i, oid_i, r_i, \delta 0_i, \delta 1_i, ph_i, i)$, in which τ_i, is logical time of a link failure, oid_i is the ID of originator node, r_i is a bit used to divide each of the unique reference into two unique sub-levels, δ_i is a propagation ordering parameter and i is the unique ID of the node. Initially, the height of each node in the network other than the destination node is set to NULL, H_i = (-, -, -, -, -, -, i). The height of the destination node is always ZERO, $H_{did} = (0,0,0,0, did)$, that did is the ID of destination node. N_i is the set neighbors of node i. $HN_{i,j}$ is a height array for each neighbor $j \epsilon N_i$. Initially the height of each neighbor node is set to NULL, $HN_{i,j} = (-,-,-,-,-,-,j)$. If the destination is a neighbor of node i, node i sets the height entry of the destination node to ZERO, $HN_{i,did} = (0,0,0,0,0,0,did)$. RR_i is a route-required flag (RR_i) which is initially un-set. $LS_{i,j}$ is a link-state array with an entry for each link $(i,j) \epsilon L$, where $j \epsilon N_i$. The state of the links is determined by the heights H_i and $HN_{i,j}$. Each link is going from the higher node to the lower node. If a neighbor j is higher than node i, the link called upstream (UP). If a neighbor j is lower than node i, the link called downstream (DN). If the neighbors height entry, $HN_{i,j}$, is NULL, the link is distinct undirected (UN).

NISR do the routing process in four phases: (1) creating routes, (2) maintaining routes, (3) erasing routes, (4) updating energy level and pheromone of routes.

1) Creating routes: This phase requires use of the QRY packets and UPD packets. A QRY packet consists of an ID of destination ($\boldsymbol{did}$), which distinct the destination node that algorithm is running. An UPD packet consists of a $\boldsymbol{did}$, and the height of the node $\boldsymbol{i}$ which is broadcasting the packet, $\boldsymbol{H_i}$. When a node that has undirected links and its route required flag is 0, need a route to the destination node, it broadcasts a QRY packet and set its route-required flag with 1. When a node $\boldsymbol{i}$ receives a QRY packet, it operates as follows: (a) if it has no downstream links and its route required flag is 0, it re-broadcasts the QRY packet and sets its route-required flag with 1. (b) If it has no downstream links and its route-required flag is set with 1, it rejects the QRY packet. (c) If it has at least one downstream link and its height is NULL, this node sets its height to $H_i = (\tau_j, oid_j, r_j, \delta 0_j + 1, \delta 1_j +$

$energy\ of\ this\ node,\ ph_i, i)$ assuming $HN_{i,j} = (\tau_j, oid_j, r_j, \delta0_j, \delta1_j, ph_i, j)$ is the minimum height of its neighbours that are non-NULL, then broadcasts an UPD packet. (d) If it has at least one downstream link and its height is non-NULL, it first compares the time the last UPD packet was broadcast to the time the link over which the QRY packet was received became active. If an UPD packet has been broadcast since the link became active, it rejects the QRY packet; otherwise, it broadcasts an UPD packet. If a node has the route-required flag with 1 when a new link is established, it broadcasts a QRY packet. When a node i receives an UPD packet from a neighbour$j \in N_i$, node i first updates the entry $HN_{i,j}$ with the height contained in the received UPD packet and then operates as follows. (a) If the route required flag is set with 1, node i sets its height to $H_i = (\tau_j, oid_j, r_j, \delta0_j + 1, \delta1_j + energy\ of\ this\ node,\ ph_i, i)$, that j is its non-NULL neighbour and height of it is minimum in between neighbour nodes, then updates all the entries in its link-state arrayLS, un-sets the route-required flag and then broadcasts an UPD packet which contains its new height. (b) If the route-required flag is 0, node i simply updates the entry $LS_{i,j}$ in its link-state.

2) *Maintaining routes:* Routes Maintenance is only performed for nodes that their height is non-NULL. Furthermore, any neighbor's height which is NULL is not used for the computations. A node i is said to have no downstream links if H_i is lower than $HN_{i,j}$ for all non-NULL neighbors $j \square N_i$. This will result in one of five possible actions depending on the state of the node and the preceding event. Each node (other than the destination) that has no downstream links modifies its height, , $H_i = (\tau_i, oid_i, r_i, \delta0_i, \delta1_i, p\square_i,\ i)$ as follows.

Case 1: Node i has no downstream links due to a link failure. In this case:

$$(\tau_i, oid_i r_i) = (t, i, 0)$$
$$(\delta0_i, \delta1_i, \mathrm{ph}_i, i) = (0, 0, ph_i, i) \quad (1)$$

In equation (1), t is the time of the failure

Case 2: Node i has no downstream links due to a link reversal following reception of an UPD packet and the ordered sets (τ_j, oid_j, r_j) are not equal for all $j \epsilon N_i$. In this case:

$$(\tau_i, oid_i, r_i) = max\{(\tau_j, oid_j, r_j) \big| j \epsilon N_i\}$$
$$(\delta0_i, \delta1_i, ph_i, i) = \left(min\left\{ \delta0_j \middle| \begin{array}{l} j \in N_{i\ with\ (\tau_j, oid_j, r_j)} \\ = \max\{(\tau_j, oid_j, r_j)\} \end{array} \right\} - 1, \delta1_i - energy\ of\ this\ node,\ ph_j, i \right) \quad (2)$$

Case 3: Node i has no downstream links due to a link reversal following reception of an UPD packet and the ordered sets $(\tau_j, oid_j, r)_j$ are equal with $r_j = 0$ for all $j \epsilon N_i$. In this case:

$$(\tau_i, oid_i, r_i) = (\tau_j, oid_j, 1)$$
$$(\delta0_i, \delta1_i, ph_i, i) = (0{,}0, \mathrm{ph}_i, i). \quad (3)$$

Case 4: Node i has no downstream links due to a link reversal following reception of an UPD packet, the ordered sets (τ_j, oid_j, r_j) are equal with $r_j = 1$ for all $j \epsilon N_i$ and $oid_j = i$. In this case:

$$(\tau_i, oid_i \square r_i) = (-,-,-)$$
$$(\delta0_i, \delta1_i, ph_i, i) = (-,-,-,i) \quad (4)$$

Case 5: Node i has no downstream links due to a link reversal following reception of an UPD packet, the ordered $(\tau_j, oid_j, r)_j$ are equal with $r_j = 1$ for all $j \epsilon N_i$ and $oid_j \neq i$. In this case:

$$(\tau_i, oid_i, \square r_i) = (t, i, 0)$$
$$(\delta0_i, \delta1_i, ph_i, i) = (0{,}0, ph_i, i). \quad (5)$$

3) *Erasing routes*: If case 4 of maintaining routes phase are detected, node $\boldsymbol{i}$ sets its height and the height entry for each neighbour $j \epsilon N_i$ to NULL unless the destination is a neighbor, in which case the corresponding height entry is set to ZERO, updates all the entries in its link-state array LS, and broadcast a CLR packet. The CLR packet containing of a did and the reflected level of node i. When a node i receives a CLR packet from a neighbour it performs as below. (a) If the reference level in the CLR packet matches the reference level of node i it sets its height and the height entry for each neighbour $j \epsilon N_i$ to NULL unless the destination is a neighbor, in which case the corresponding height entry is set to ZERO, updates all the entries in its link-state array LS and broadcasts a CLR packet. (b) If the reference level in the CLR packet does not match the reference level of node i , it sets the height entry for each neighbour $j \epsilon N_i$ to NULL and updates the matching link-state array entries. Thus the height of each node in the part of the network which was partitioned is set to NULL and all invalid routes are erased. If (b) causes node i to lose its last downstream link, it perform as in case 1 of maintaining routes.

4) *Updating energy level and pheromone of routes:* In this phase, energy level and pheromone of routes is updated. For this purpose, when a data packet arrives to destination, NISR start from destination node toward source node and update the energy level of node i by equation (6):

$$\delta1_i = \delta1_j + energy\ of\ this\ node \quad (6)$$

In equation (6) j is neighbor of node i that is earlier than node i in the route from destination to the source, and update the pheromone of node i by equation (7):

$$ph_i = ph_i + 1 \quad (7)$$

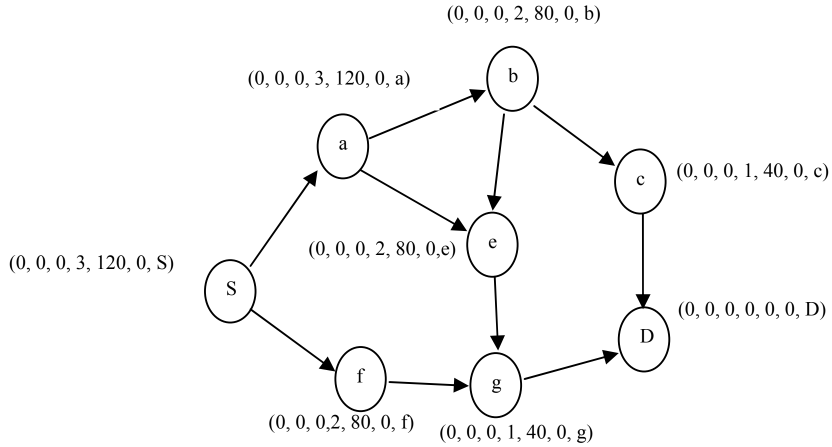

Fig. 1 Routes creating process in NISR

Fig. 1 shows an example in which node S wants to send data to node D. Then, all routes from source to the destination discovered using NISR routing protocol. In this example assume energy of all nodes is 40J. In this way, when routes between source node and destination node discovered, we have direction of routes in link-state LS, distance of routes in $\delta 0_i$, quality of routes in $\delta 1_i$. When the 4 items (direction, distance, quality, stability) determined, the data send via a route with maximum value in: $\left(\frac{\delta 1_i}{\delta 0_i}\right) \times 0.75 + \left(\frac{1}{\delta 0_i} \times 5\right) \times 0.25$. If some routes have maximum value, data send via a route with maximum stability (maximum ph_i).

IV. Performance evaluation and simulation results

To investigate the impact of our enhancements, we implement NISR by making some modifications over TORA module of ns-2 [15] simulator. Three performance metrics that are used to evaluate performance of NISR are network life time, system life time and total delivered data [16]. The network lifetime is defined as the time when a node finished its own battery for the first time. The system lifetime is defined as the time when 20% of nodes finish their own battery. The total delivered data is the total number of data packets that delivered during system lifetime.

In this simulation the network traffic was modeled as CBR sources with data sent in 512 byte packets with rate 4 packet/sec. One movement pattern was produced based on a random waypoint model [15] with pause time 10 seconds. The primary energy of all nodes is 40J. The IEEE 802.11 is used as the medium access protocol. The interface queue is 50-packet drop-tail priority queue. The simulation time is 500 seconds and network space is 1000m$\times$1000m. To study about capability of NISR to support the increase of number of nodes, we consider various numbers of nodes in our simulations. It dynamically is set to 10, 20, 30, 40 and 50 nodes. We simulate this network for each node count separately, once under TORA protocol and then under NISR protocol. Simulation results are shown in Figs. 2-4.

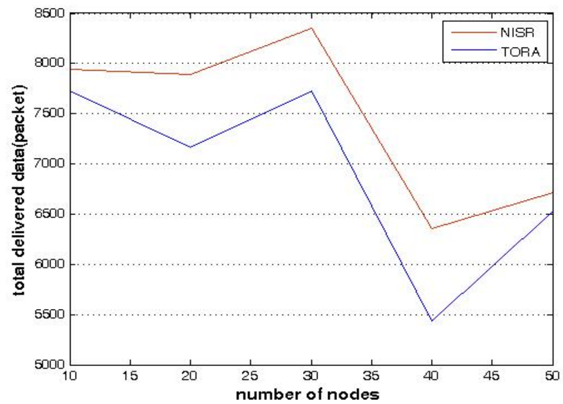

Fig. 2 Total delivered data versus number of nodes

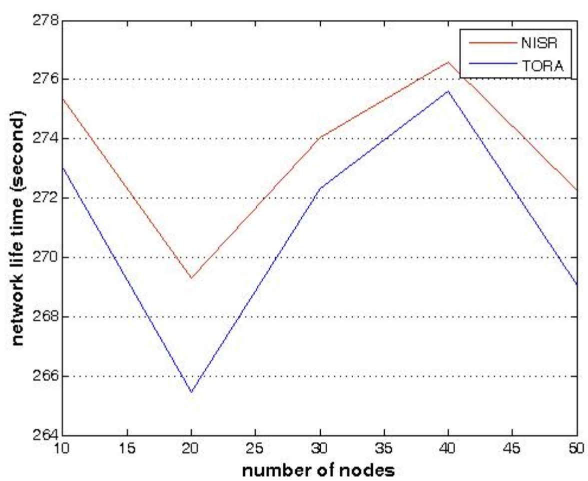

Fig. 3 Network life time versus number of nodes

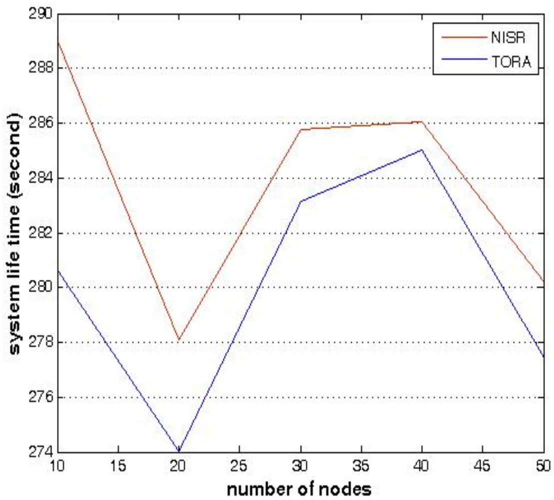

Fig. 4 System life time versus number of nodes

Fig. 2 shows that NISR increases delivered data in compare with TORA protocol. On the other hand Fig. 3 and Fig. 4 show that NISR outperforms TORA in terms of network life time and system life time. Better performance of NISR is due to this fact that it uses those routes that have higher level of energy or equally high stability. This leads to balanced consumption of energy in various routes and nodes; hence NISR experiences less route breakages and achieves better performance.

V. Conclusions

In this paper we proposed NISR a scalable routing protocol for MANETs that has been developed by improving TORA routing protocol. This improvement came from inspiration from bee and ant colonies. In this way we drew an analogy between the routing in MANETs and finding source of food in ant and bee colonies. The simulating results indicated that the total delivered data, network life time and system life time in NISR are better than TORA routing protocol for a wide range of number of nodes.

References

[1] W. Kiess and M. Mauve,“A survey on real-world implementations of mobile ad-hoc networks”, Ad Hoc Networks, 2007.
[2] R. Ramanathan and J. Redi, “A Brief Overview of Ad Hoc Networks: Challenges and Directions”, IEEE Communications Magazine, 2002.
[3] J.A. Torkestani and M.R. Meybodi, “Mobility-based multicast routing algorithm for wireless mobile Ad-hoc networks: A learning automata approach”, Computer Communications, 2010.
[4] O. Souihli, M. Frikha and M.B. Hamouda, “Load-balancing in MANET shortest-path routing protocols”, Ad Hoc Networks, 2009.
[5] O. Liang and O. Ren, “Energy and mobility aware geographical multipath routing for wireless sensor networks”, IEEE WCN, 2005.
[6] M. Tarique and K.E. Tepe, “Minimum energy hierarchical dynamic source routing for Mobile Ad Hoc Networks”, Ad Hoc Networks, 2009.
[7] A. Iwata, Ch. Ch. Chiang, G. Pei, M. Gerla and T.W. Chen, “Scalable Routing Strategies for Ad Hoc Wireless Networks ”, IEEE journal on selected areas in communications, 1999.
[8] H. Li and M. Singhal, “A Scalable Routing Protocol for Ad Hoc Networks”, IEEE ICINB, 2005.
[9] Z. Z. Abidin, M.R. Arshad and U.K. Ngah, “A Survey: Animal-Inspired Metaheuristic Algorithms”, EEPC 2009, 2009.
[10] H.F.Wedde, M. Farooq, T. Pannenbaecker, B. Vogel, C. Mueller, J. Meth and R. Jeruschkat, “BeeAdHoc: An Energy Efficient Routing Algorithm for Mobile Ad Hoc Networks Inspired by Bee Behavior”, ACM, 2005.
[11] K. Von Frisch, “The Dance Language and Orientation of Bees”, Harvard University Press, 1967.
[12] Ch. Blum, “Ant colony optimization: Introduction and recent trends”, Physics of life Review, 2005.
[13] M. Dorigo, Ch. Blum, “Ant colony optimization theory: A survey”, Theoretical Computer Science, 2005.
[14] V.D. Park and M.S. Corson, “A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks”, INFOCOM, 1997.
[15] NS-2, “The Network Simulator”, http://www.isi.edu/nsnam/ns/.
[16] F.Yu, Y.Li, F.Fang and Q.Chen, “A New TORA-based Energy Aware Routing Protocol in Mobile Ad Hoc Networks”, 2007.