An Energy-Aware Routing Protocol for Ad-Hoc Networks Based on the Foraging Behavior in Ant Swarms

Sanjay K. Dhurandher
CAITFS, Division of Information Technology
Netaji Subhas Institute of Technology, University of Delhi
New Delhi, India

Sudip Misra
School of Information Technology
Indian Institute of Technology-Kharagpur
West Bengal, India

Mohammad S. Obaidat
Department of Computer Science
Monmouth University
West Long Branch, NJ, USA

Pushkar Gupta, Karan Verma, and Prayag Narula
CAITFS, Division of Information Technology
Netaji Subhas Institute of Technology, University of Delhi
New Delhi, India

Abstract—**Routing in ad-hoc networks can consume considerable amount of battery power. However, as the nodes in these networks have limited power, routing is very much energy-constrained. Continuous drainage of energy degrades battery performance as well. If a battery is allowed to intermittently remain in an idle state, it recovers some of its lost charge due to the charge recovery effect, which, in turn, results in prolonged battery life. In this paper, we use the ideas of naturally occurring ants' foraging behavior [1] and based on those ideas we design an energy-aware routing protocol, which not only incorporates the effect of power consumption in routing a packet, but also exploits the multi-path transmission properties of ant swarms and, hence, increases the battery life of a node. The efficiency of the protocol with respect to some of the existing ones has been established through simulations.**

I. INTRODUCTION

Extending battery life of nodes is a critical issue in ad-hoc networks. Conventional routing algorithms such as AODV [2], DSR [3] and TORA [4] ignore the residual battery of nodes. Data transmission between two nodes is typically done in these protocols through the shortest path routes. The algorithms instrumental in route finding in such protocols often result in faster depletion of battery in the nodes that are incident on the heavily used routes in the network (for example, the ones which lie in the shortest path routes).

The identification of the above-mentioned limitations with using conventional routing protocols for ad-hoc networks is not novel to our work. Quite a few energy-aware/power-aware schemes exist in the literature. Let us discuss only a few notable ones below. The Minimum Total Transmission Power Routing (MTPR) scheme [5] tries to minimize the total transmission power consumption of nodes participating in an acquired route. But because MTPR fails to consider the remaining power of nodes, it does not always necessarily succeed in extending the lifetime of each host. Singh *et al.* proposed the Min-Max Battery Cost Routing (MMBCR) [6] scheme, which considers the residual battery power capacity of nodes as the metric in order to extend the lifetime of nodes. MMBCR tries to choose a path whose weakest node has the maximum remaining power amongst the weakest nodes in other possible routes to the same destination. However, MMBCR does not guarantee that the total transmission power is minimized over a chosen route. The Conditional Max-Min Battery Capacity Routing (CMMBCR) scheme [7] considers both the total transmission energy consumption of routes and the remaining power of nodes. The Minimum Drain Rate (MDR) scheme [8] was also proposed. It introduces a metric, "the drain rate", which is used in conjunction with the residual battery capacity to predict the lifetime of nodes according to the current traffic conditions. A number of other researchers addressed the energy utilization issues for routing in MANETs. References [9], [10] and [11] are examples of a few recent pieces of literature on this topic.

Some of the popular ACO-based routing schemes are AntNet [12], AntHocNet [13] and ARA[14]. The earlier ACO-based routing schemes such as AntHocNet [13] and ARA [14], devised for ad-hoc networks, were not targeted towards energy conservation.

We considered the minimum battery energy remaining from the weakest node of the route and the hop-count of the route as our metric for path discovery. First, we included hop count in our consideration as, the greater the number of hops, the greater will be the transmission, and the more will be the energy consumption. Second, the minimum battery remaining was taken into consideration after getting inspired from the success of MMBCR [6] in minimizing energy usage.

II. ACO & EAAR

A. The Proposed Scheme -- EAAR

Path Discovery

The basic structure of the path discovery mechanism is inspired from AntHocNet [13], but, as will be evident from the following discussions, it is functionally distinct. When a

978-1-4244-3435-0/09/$25.00 ©2009 IEEE

source node *s* starts a communication session with a destination node *d*, and it does not have routing information for *d* available, it broadcasts a reactive forward ant, say F_s^d (here an ant is a representative of a control packet). Due to the initial broadcasting, each neighbor of *s* receives a replica F_s^d, which is F_s^d.k (the notation " .k" refers to indexing – the k^th^ message of a single broadcast will be represented by F_s^d.k) . After the next hop, the next neighboring node will receive F_s^d.k.L and so on where (k, L, … are integers). The set of replicas which originate from the same original ant is also referred to as an ant generation. The task of each ant F_s^d.k.L.m.n … is to find a path connecting *s* and *d*. At each node, an ant is either unicasted or broadcasted, depending on whether or not the current node has routing information for *d*. Since there will be no information initially, all are broadcasted at that point. Also, each packet maintains an array *J* in which its journey information is stored.

When a node receives several ants of the same generation, it will compare the path traversed by each ant to that of the previously received ants of the current generation. For example, let us suppose that an ant A1 has a journey array, a set of nodes, J1, and another previously received ant A2 had J2. If J2 is a subset of J1, then the packet is discarded immediately. If J1 is a subset of J2, the packet is accepted. In the third case, when both do not overlap completely, we use an acceptance factor, λ, if A2 had M hops and A1 currently has N hops, then we will allow only when N ← λ x M. Here, λ is the factor by which each node takes a decision whether to accept or reject a particular control packet that has more hops than the packet which came before it for the same destination. But this decision is taken only if the packet is not the superset of any packet that is already received.

With this approach, we may get a set of paths with similar journey, but only some nodes will be distinct type of nodes. But this will help us to get the "better" paths because our parameter is not limited to hop count only. We attempt to find a path with the maximum of the minimum residual battery energy (MBR) of all the nodes of the journey – together we call it MMBR.

We take λ = 1.5, after experimenting with its different values. It is conjectured that an increase in the number of hops is likely to get us a "good" path in terms of MBR. We just cannot ignore hop count. So we need to set the value of λ by giving due consideration for both the parameters.

Using this scheme, overhead is limited by removing ants which follow "bad" paths, while there is still the possibility to find multiple "good" paths. We also obtain an increased number of paths, which implies that, if needed, in case of link failures, it will help us to divert the packet flow. Each forward ant keeps a list P of the nodes visited.

On the arrival of the first reactive forward ant to the destination, we calculate the end-to-end delay for this first ant. The destination will wait for certain time equal to the product if X (X is an integer factor) and the end-to-end delay calculated for the rest of the reactive forward ants to come and all the ants that came in this time are taken up by the destination.

Meanwhile, all the ants received until that time get converted to backward ants as soon as they arrive and they travel back to the source retracing P (if this is not possible because of the absence of the next hop, for instance, due to node movements, the backward ant is discarded). While moving backwards to node *i* from node *n* and upon reaching there, each backward ant updates or makes an entry into the neighbors table of node *i* about $T_{n,d}^i$ which equals the inverse of the number of hops (H) multiplied by the minimum residual battery energy of all the nodes traveled up to the current node by the current backward ant (MBR):

$$T_{n,d}^i = \frac{MBR}{H} \qquad (1)$$

In Equation (1), $T_{n,d}^i$ is the value that a data packet will check when it arrives at node *i*, as to when it has to chose the next node.

We also drop the packets which have hop count of more than 10. The destination node will wait for a time Y times the end-to-end delay. If no data packet is received by the destination in this time, it will find the *b* next best paths, in addition to the previous one and then repeat the same procedure until a data packet is received in the given time.

Data Session

Once the data session starts, the data packets are sent through the host. The host will either distribute the packet or the packet will choose the next node from the set of neighbors, N_d^i, which have pheromone information in the table with probabilistic condition, P_{nd} , as in AntHocNet [13]:

$$P_{nd} = \frac{(T_{nd}^i)^\beta}{\sum_{j \in N_d^i} (T_{jd}^i)^\beta} \qquad (2)$$

In Equation (2), β is a factor which can take in a set of integer values. The traversal of each data packet increases the pheromone values of each link by a factor π

$$T_{n,d}^i = T_{n,d}^i + \pi \qquad (3)$$

In Equation (3), π lies between 0 and $T_{n,d}^i \frac{1}{2}$. In the results reported in this paper, we took π = $T_{n,d}^i \frac{1}{10}$. The other nodes evaporate the pheromone deposits resulting in the more frequent selection of better paths. Evaporation is done periodically. For every τ time period, node will evaporate the pheromone value $T_{n,d}^i$ automatically.

$$T_{n,d}^i = T_{n,d}^i \times (1 - \rho) \qquad (4)$$

In Equation (4), ρ lies between 0 and 0.5. As soon as the first proactive backward ant is received by the host, it again sends another forward proactive ant analogous to the first one. This eventually leads one to a better path.

Route Maintenance and Link Failures

While a data session runs, the routes are maintained through pheromone reinforcement and evaporation techniques. Our scheme does not waste energy by sending proactive ants.

When a link fails to transmit the packet and the node having it has no more neighboring node in the pheromone table to send, it sends a control packet to all the neighboring nodes, which have this node in their data to remove the current node's entry. The node having the data then initiates a route request packet for the required destination and sends the data on the first arrival of its reply, without waiting. In the other cases of link failure, no action is taken.

B. Algorithm

Algorithm EAAR

Input:

- The following blank tables of all nodes are input:
  (1)The neighbor table- a table containing all nodes in the neighborhood of a node.
  (2)Seen table - a table containing all packets received by a node and their paths.
  (3)Routing table- a table containing next hop to transfer packets.
- Initial pheromone for all nodes = 0.

Output:

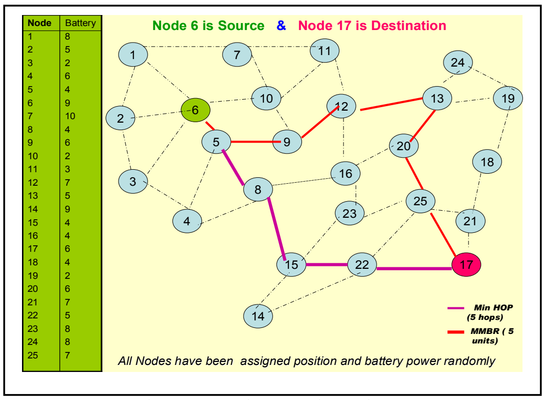
- Updated tables with all the values required to transmit data.
- Pheromone value for selected nodes.

Steps:

```
1. Broadcast all the request packets and initialize a "seen" set S of every node as NULL.
2. On receiving any route request:
   for all routes Ri in S of node check:
      if the route traveled by the request is not a superset of the Ri
      if the route is subset of Ri OR the hop count is less than 1.5 times the highest in the set S:
         add route in set S and rebroadcast it.
      else
         discard it
3. On reaching the destination, the route request is converted to the route reply, the path traveled is returned to, and the pheromone PH in the routing table of each node of path is added.
      PH= MBR/HOPS
4. When the source receives the first reply, the delay of the first packet is made 5 times in order to receive more packets and the routing table is updated.
5. Data transmission is initiated with each packet, selecting next hop with probability $P_{nd}$ from all available, by taking the pheromone values from routing table from Equation 2.
6. On each transmission, the pheromone is reinforced and others are evaporated.
7. On link failure, Step 1 is repeated from the node that has data to send, but no neighbors available.
```

C. Example

In Fig. 1, intuitively, our most preferable paths should be the two shaded paths. One having minimum hop and other having highest minimum battery, i.e., the path in which the node with the minimum battery, among others in the same path, is highest as compared to minimum of other paths. The red shaded route has all nodes having energy ≥ 5; we say that the minimum battery of that route is 5. Similarly, for other routes every node has energy greater than or equal to some positive number, which comes out to be less than 5. So, the route with the highest minimum battery (MMBR) is the red colored one having MMBR as 5 units and the route with the minimum hops is the purple colored one.



Node	Battery
1	8
2	5
3	2
4	6
5	4
6	9
7	10
8	4
9	6
10	2
11	3
12	7
13	5
14	9
15	4
16	4
17	6
18	4
19	2
20	6
21	7
22	5
23	8
24	8
25	7

Fig. 1. Example network topology.

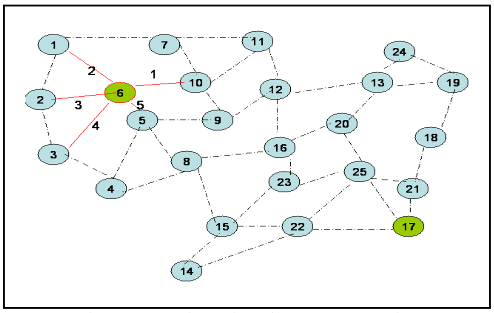


Fig. 2. Example to show how EAAR works.

Referring to Fig. 2, we broadcast $F_s^d \triangleright k$, where $k \in \{1,2,3,4,5\}$. On reaching the destination, the return ants are created and they carry pheromone to the path listed below. As we can see, many paths are discovered, but if we look closely, it is just the next hop which matters the most, and in all cases it is around 2 or 3. The bold ones have maximum pheromone which is our objective in starting the shaded paths have got best pheromone.

Paths	Pheromone
6-1-7-11-12-16-23-25-17	(3/8)
6-1-7-11-12-16-20-25-17	(3/8)
6-1-7-11-12-13-19-18-21-17	(2/9)
6-3-4-8-15-23-25-17	(2/7)
6-3-4-8-15-22-17	(2/8)
6-3-4-8-16-20-25-17	(2/7)
6-3-4-8-16-23-25-17	(2/7)
6-5-9-12-19-18-17	(4/6)
6-5-9-16-20-25-17	(4/6)
6-5-9-16-23-25-17	(4/6)
6-5-8-15-23-25-17	(4/6)
6-5-9-12-13-20-25-17	**(5/7)**
6-5-8-15-22-17	**(4/5)**
6-5-8-15-14-22-17	(3/7)
6-10-9-12-13-19-18-21-17	(2/8)
6-10-9-16-23-25-17	(2/6)
6-10-9-16-20-25-17	(2/6)

III. SIMULATION RESULTS

In this Section, we report the simulation experiments we conducted and the results we obtained for comparing the performance of EAAR with a selection of benchmark algorithms. Simulation was done using GloMoSim.

The following parameters were set for the simulations performed:

(a) Simulation Time: 1000 seconds
(b) Terrain dimensions: (2000, 2000) meters square
(c) Number of Nodes : 30
(d) Node Placement : Uniform
(e) Mac Protocol: 802.11
(f) Initial energy of Nodes : All Nodes were initiated with Equal Energy.

$$\beta = 1, \ \pi = T^{i}_{n,d}\frac{1}{10}, \ \rho = 0.1.$$

Traffic types used are as follows (using the format CBR <src> <dest> <items to send> <item size> <interval> <start time> <end time>):

(a) CBR 1 26 100 1536 1S 0S 0S,
(b) CBR 12 18 100 1536 1S 250S 0S,
(c) CBR 23 9 100 1536 1S 500S 0S,
(d) CBR 14 27 100 1536 1S 750S 0S.

The following benchmark protocols were used for comparing the performance of our proposed protocol, EAAR: AODV [2], MMBCR [6], AntHocNet [13]. AODV is one of the most popular and widely used proactive routing protocols for routing in ad-hoc networks. MMBCR is very popular for energy-aware routing and AntHocNet is perhaps the most popular ACO-based ad-hoc network routing algorithms. This justifies the choice of these benchmark algorithms.

The simulation experiments were done under the following six scenarios. In the context of the Random Waypoint Mobility Model, WP-PAUSE refers to the amount of time the nodes will pause and WP-MIN-SPEED and WP-MAX-SPEED refer to the minimum and the maximum speeds, respectively, with which the nodes will move.

1) Data Size = 100 times Control Packet Size.
   Mobility: NONE.
2) Data Size = 125 times Control Packet Size.
   Mobility: NONE.
3) Data Size = 100 times Control Packet Size.
   Mobility speed: 10 m/s, random way-point model.
4) Data Size = 125 times Control Packet Size. Mobility same as previously.
5) Data Size = 100 times Control Packet Size. Mobility same as previously.
6) Data Size = 125 times Control Packet Size. Mobility same as previously.

The following performance metrics were used for the sake of comparison of our proposed protocol, EAAR, with the benchmark protocols discussed earlier: (a) *Number of dead nodes*, (b) *Number of packets dropped*, (c) *Total energy consumed*, (d) *Number of packets delivered*, (e) *Energy per packet delivered*. Several simulation experiments were performed, of which only a few key results are reported below due to space limitation.

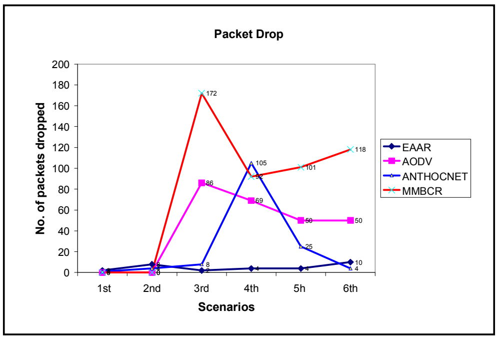


Fig 3. Number of packets dropped.

Fig. 3 shows the relative performance of the four protocols with respect to the number of packets dropped. As we can see, EAAR has very less number of packets lost compared to the others. This observation is very prominent for the 3^{rd}, 4^{th} and the 5^{th} scenarios. Let us take these scenarios one by one.

In the first scenario when the mobility is zero and the packet size is small, the network acts as an ideal one. As a result every protocol behaves similarly. But in practical terms, this is not possible all the time. In the second scenario too the difference is very minute. But in the third scenario, when we introduce some mobility, we see the difference. The packet drop for the ant based protocols (AntHocNet and EAAR) is found to be minimum. This is due to multi – path routing. But even then EAAR performs better than AntHocNet, which is designed for the better delivery of packets. In the fourth scenario for large data sizes, the difference increases and EAAR is a clear favorite. This shows that the performance of EAAR improves for bigger data sizes.

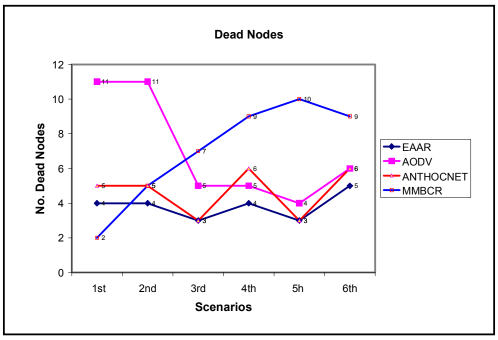


Fig. 4. Number of dead nodes

Fig. 4 shows that EAAR, in general, has a relatively very less number of dead nodes compared to the benchmark algorithms. This is perhaps related to the existence of multi-path routing in EAAR. For the same reason, AntHocNet also shows less number of dead nodes, but it cannot beat EAAR because EAAR is energy aware. As AODV and MMBCR route using the unipath mechanism, they lose more nodes because of the faster depletion of energy from these selected nodes. Let us discuss each scenario.

In the first scenario MMBCR performs best followed by EAAR. In this scenario the packet size is small with no mobility. This gives MMBCR an edge as it does not consider mobility much in its approach. And due to smaller data sizes, the energy lost is not substantial. This is the reason MMBCR has performed relatively well. In the second scenario, we can see the consistency in the ant based routing protocol as they are multi- path and offer better performance. But still EAAR performs better. In the third scenario, when we introduce mobility, we can see that there is still less number of dead nodes in EAAR as compared to other protocols. This goes to show how well EAAR adapts itself in the mobile environment. In the fourth scenario too, we see a similar result. So is the case in the 5th and the 6th scenario as well.

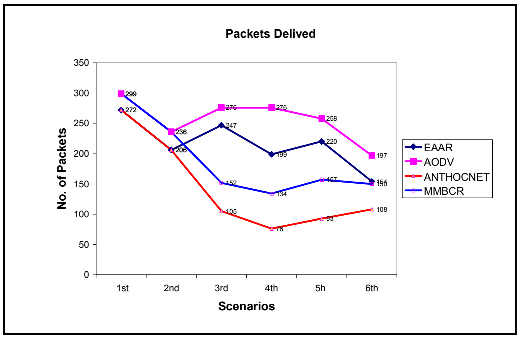


Fig. 5. Number of packets delivered.

AODV was devised to get the best throughput. Consequently, it has a greater number of packets delivered compared to the other protocols. This is shown in Figure 5. As our scheme waits for some time to get more routes and then transmit, the packets delivered are relatively less compared to AODV, but it is still better than AntHocNet, in which, the mobile network loses more links, thereby impairing the ability to deliver packets.

In the first scenario with no mobility, AODV and MMBCR perform better than the multi- path routing protocols. So is the case in the second scenario too. This is due to the fact that in a static environment, once a connection is established, it hardly breaks. So AODV and MMBCR perform well in these scenarios. But as we introduce mobility in the environment, the performance of these protocols decreases substantially, whereas EAAR performs as good as previously. In scenarios 1, 3 and 5, where the data size is small, we see a slight variation for EAAR. So is the case for the scenarios 2, 4 and 6. But the other protocol's performance degrades as we increase mobility. AODV being the exception, for the reason mentioned above, it delivers more packets throughout all scenarios.

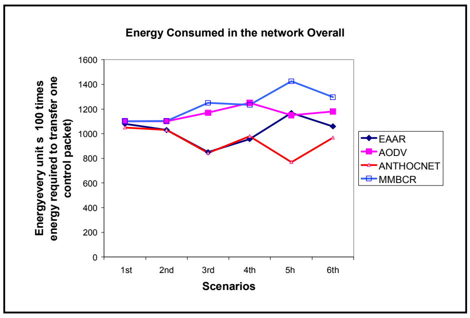


Fig. 6. Energy consumed in the overall network.

As we can see from Fig. 6, the energy consumed in the overall network is minimal in EAAR, although the packets delivered remain very similar. This affirms that, in all conditions, EAAR is most energy-aware compared to the existing protocols.

In scenarios 1 and 2, all the protocols perform equally as there is no mobility and hardly any connection breaks. But still multi-path routing schemes perform better. But in a mobile environment, where nodes are moving randomly, multi-path routing outshines single routing algorithms.

IV. CONCLUDING REMARKS

It is, thus, evident that EAAR is better compared to AODV, AntHocNet and MMBCR in most mobility and energy conservation scenarios. The superior energy conservation attribute of EAAR, in turn, helps in extending the life of the nodes. The energy consumed in the network, the energy per packet and the number of packets lost in case of EAAR are less compared to others in small and medium mobility scenarios. The packet delivery ratio is, therefore, much superior for EAAR. Of course, the packet delivery rate and the energy per packet in high mobile conditions are relatively not as superior for EAAR as the other parameters. This is because energy awareness increases the time to judge the best route for transmission. The results get better for larger data packets (more than 1536 bytes s we have seen in scenarios 2, 4 and 6). In the future, we plan to: (a) use other mobility models, (b) use networks having an extremely large number of nodes and varying network density, (c) develop test-beds for observing whether the results remain the same for actual networks.

REFERENCES

[1] M. Dorigo and T. Stuetzle, Ant Colony Optimization, Prentice Hall, 2004.

[2] C. E. Perkins, E. M. Belding-Royer, and S. Das, "Ad Hoc On-demand Distance Vector (AODV) Routing", IETF Internet Draft, 2001.

[3] B. Johnson, D. A. Maltz, Y.-C. Hu and J. G. Jetcheva, "The Dynamic Source Routing for Mobile Ad Hoc Wireless Networks",http://www.ietf.org/internet-drafts/draft-ietfmanet-dsr-06.txt, IETF Internet Draft, Nov. 2001.

[4] V. Park and S. Corson, "Temporally-Ordered Routing Algorithm (TORA) Version 1", IETF Internet Draft, July 2001.

[5] K. Scott and N. Bambos, "Routing and Channel Assignment for Low Power Transmission in PCS", Proc. Intl. Conf. Universal Personal Communications (ICUPC'96), Cambridge, MA, 1996, pp. 498-502.

[6] S. Singh, M. Woo, and C. S. Raghavendra, "Power-Aware Routing in Mobile Ad Hoc Networks", Proc. 4th Annual ACM/IEEE Intl. Conf. Mobile Computing and Networking, 1998, Dallas, TX, pp. 181-190.

[7] C.-K. Toh, "Maximum Battery Life Routing to Support Ubiquitous Mobile Computing in Wireless Ad Hoc Networks", IEEE Communications Magazine, Vol. 39, No. 6, June 2001, pp. 138-147.

[8] Kim, J. J. Garcia-Luna-Aceves, K. Obraczka, J.-C. Cano and P. Manzoni, "Power-Aware Routing Based on the Energy Drain Rate for Mobile Ad Hoc Networks", Proc. 11th Intl. Conf. Comp. Comm. Netw., 2002, pp. 565-569.

[9] W. Liang and Y. Yuansheng, " Maximizing Battery Life Routing in Wireless Ad Hoc Networks", Proceedings of the 37th Hawaii International Conference on System Sciences, Hawaii, USA, 2004.

[10] A. Srinivas and E. Modiano, "Finding Minimum Energy Disjoint Paths in Wireless Ad-Hoc Networks", Wireless Netw., Vol.11, 2005, pp. 401-417.

[11] W.H. Yuen and C.W. Sung, "On Energy Efficiency and Network Connectivity of Mobile Ad Hoc Networks", Proc. 23rd IEEE Intl. Conf. Distrib. Comput. Sys. (ICDCS'03), Rhode Island, May, 2003, pp. 38-45.

[12] M. Dorigo, V. Maniezzo and A. Colorni, The Ant System: An Autocatalytic Optimizing Process, TR91-016, Politecnico di Milano, 1991.

[13] G. Di Caro, F. Ducatelle and L. M. Gambardella, "AntHocNet: An Adaptive Nature-Inspired Algorithm for Routing in Mobile Ad Hoc Networks", Telecommunications (ETT), Vol. 16, No. 2, 2005.

[14] M. Guenes, U. Sorges and I. Bouazizi, "ARA: The Ant-Colony Based Routing Algorithm for MANETs", Proc. of ICPPW'02, 2002, pp. 79- 85.