A Position Based Ant Colony Routing Algorithm for Mobile Ad-hoc Networks

Shahab Kamali, Jaroslav Opatrny
Department of Computer Science and Software Engineering, Concordia University, Montreal, Canada
Email: {s_kama,opatrny}@cs.concordia.ca

***Abstract*—Position based routing algorithms use the knowledge of the position of nodes for routing of packets in mobile ad-hoc networks. Previously proposed position based routing algorithms may fail to find a route from a source to a destination in some types of ad-hoc networks and if they find a route, it may be much longer than the shortest path. On the other hand, routing algorithms which are based on ant colony optimization find routing paths that are close in length to the shortest paths. The drawback of these algorithms is the large number of control messages that needs to be sent or the long delay before the routes are established from a source to a destination. In this paper we propose a new reactive routing algorithm for mobile ad hoc networks, called POSANT (Position based Ant Colony Routing Algorithm), which combines the idea of ant colony optimization with information about the position of nodes. In contrast to the other ant colony optimization based routing algorithms, our simulations show that POSANT has a relatively short route establishment time while using a small number of control messages which makes it a scalable reactive routing algorithm.**

***Index Terms*—mobile ad-hoc networks, routing algorithms, position based routing, ant colony optimization**

I. INTRODUCTION

A mobile ad-hoc network consists of nodes which communicate with other nodes using wireless transmitters in the absence of a fixed infrastructure. Since nodes can move, join a network or leave it freely, the topology of a network changes quite often. Furthermore, the transmission range of each node is limited. In order to reach a destination, a message must pass through some intermediary nodes; therefore these networks are also called *multi-hop* networks. One of the challenging problems in this type of networks is how to find routes between nodes. A routing algorithm is called *proactive* if it calculates routes before they are needed and tries to keep routing-information to all nodes every time up-to-date. In contrast, a *reactive routing* algorithm establishes one (i.e. or more) route only when it is needed and does not try to keep routing information to all nodes always up-to-date [8].

The availability of cheap instruments like GPS receivers for estimating the position of nodes in a network motivated some researchers to propose position based routing algorithms. In position-based routing algorithms it is assumed that a node is aware of its position, the position of its neighbors, and the position of the destination. We can mention *GPSR*[12] and *DIR* (also referred to as *Compass routing*)[13] as examples of reactive position based routing algorithms. The drawback of these algorithms is that they may fail to find a route or they may find a non-optimum route in some situations. Algorithms based on *face routing* [1], [3], [12] guarantee to find a route to the destination if it is possible to extract locally a planar subnetwork of a given network. In this paper we consider a more general network model (defined in the next section) which allows the transmission radius of nodes to vary a lot. It has been shown that extracting a planar subnetwork in this type of ad-hoc network is not always possible [1]. Hence face routing algorithms do not guarantee data delivery in our model and they often find a route that is much longer than the shortest path [12].

Another family of routing algorithms for mobile ad-hoc networks is based on *ant colony optimization* (ACO). In general, ant colony optimization meta-heuristic tries to solve a combinatorial problem using the collaboration of a group of simple agents called artificial ants. ACO routing algorithms establish optimum paths to the destination using a number of artificial ants that communicate indirectly with each other by *stigmergy*. Stigmergy is a way of indirect communication between individuals which, in an ad-hoc network case, is done through the modification of some parameters in the nodes of the network. So far, information about the position of nodes has not been used in the algorithms of this family.

ANTNET[5] and *ANTHOCNET*[4] are two well known ant colony based routing algorithms. ANTNET is a proactive and ANTHOCNET is a reactive routing algorithm. They have a very high delivery rate and find routes whose lengths are very close to the length of the shortest path [5], [9]. The drawback of ANTHOCNET is the number of routing messages that needs to be sent in the network for establishing routes to the destination and the disadvantage of ANTNET is the time needed before a system of paths between the nodes of the network is established. This is referred to as the *convergence time*. Regarding the dynamic nature of mobile ad-hoc networks, a long convergence time is a significant drawback.

In this paper we present a new reactive routing algorithm which is based on ant colony optimization and which also uses information about the location of nodes. Since our algorithm is a position based ant colony routing algorithm, we call it POSANT (POSition based ANT colony routing for mobile ad hoc networks). Our simulation results show that POSANT reduces the route establishment time while keeping the number of generated ants much smaller in comparison to other ant colony based routing algorithms. The applications of POSANT may include cases where huge amount of data is transmitted after route establishment and thus it is important to find routes close to optimum (i.e. an example is video and audio streaming) and also in ad-hoc networks where the irregularity of transmission ranges limits the use of other position based routing algorithms.

In the next section we first specify our network model and give a definition of the routing problem. We then explain the typical position based and ant colony based routing algorithms in more details. Section III defines POSANT routing algorithm and Section IV contains the simulation results of POSANT and a comparison with ANTNET, ANTHOCNET and GPSR routing algorithms. Section V contains conclusions.

II. NETWORK MODEL AND RELATED ROUTING ALGORITHMS

We assume that an ad-hoc network is a collection of nodes, each node having a wireless transmitter and receiver of a limited range. The nodes are not assumed to be identical. A node may have different transmission ranges in different directions. Furthermore, due to the possible difference in the available power among nodes, different nodes may have different maximal transmission ranges. We assume that all communications are bi-directional, i.e. a node maintains a link to another node if they are able to exchange a message directly. Hence, our model of mobile ad-hoc networks is more general than *unit disk graph* model [3], [13], in which all nodes have the same transmission ranges in all directions. Clearly the model used in this paper correspond better to real situations where the existence of obstacles or noise can make the transmission radius of nodes irregular and of different reach. Each node is assumed to be aware of its position, the position of its neighbors and the position of the destination node. We represent an ad-hoc network as an undirected graph with an edge between each couple of nodes that can communicate directly. The *routing problem* is to find a route from a given source node to a given destination node in a wireless ad hoc network when a need for such a route arises.

We suppose the time required for an ant to move from one node to a neighbor node is the same for all nodes (i.e. we do not consider congestion, distance between two neighboring nodes and other causes of packet delay in this paper). It means the delay of a packet is a linear function of the number of hops in its way to the destination. So hop count could be used as an evaluation metric for packet delay. Our aim is to make the massage delivery delay as small as possible, so the length of the established routes in hop count should be close to the length of the shortest path.

We suppose the route finding algorithm starts as soon as data packets need to be sent from a source to a given destination (reactive routing). So to reduce the delay of data delivery, the routes should be established in the shortest possible time. Also it is important to guarantee finding a path to the destination when such a path exists. In the remainder of this section we briefly describe some typical position based routing algorithms and ant colony based routing algorithms related to our work.

A. Position Based Routing Algorithms

GPSR[12] and *Compass routing* [13] are examples of position based routing algorithms in which a routing decision is made locally in each node that is reached in the routing process.

In *Compass routing* (Also known as DIR) a node S that receives a packet for destination D, calculates for every neighboring node N the angle between the line segments from S to D and S to N. The packet is forwarded to the neighbor of S for which this angle is the smallest.

Progress is an important concept in many position based routing algorithms. For each node, say S, that receives a message for destination D, the progress of a neighbor N is usually defined as the projection of line segment SN onto line SD. The progress can be positive or negative.

In *GPSR*[12] method, a packet can have two different modes, greedy-mode and perimeter-mode. Upon receiving a greedy-mode packet, a node searches its neighbors to see if there is a neighbor with a positive progress. If such a neighbor exists, it will forward the packet to the neighbor which is closest to the destination, otherwise the packet will be changed into perimeter-mode. GPSR uses a planar graph traversal method to forward perimeter-mode packets. When a packet enters perimeter mode at node S for destination D, it will be forwarded on progressively closer faces of the planar graph which are crossed by line SD. On each face, the traversal uses the right-hand rule to reach an edge that crosses line SD. At that edge, the traversal moves to the adjacent face crossed by SD. When a perimeter-mode packet enters a node which is closer to D than S, its mode will be changed into greedy again.

The above mentioned algorithms do not guarantee to find a shortest path to the destination [8], [2], [12]. Figure 1 shows an example of a case when these algorithms find a path to the destination which is longer than the shortest path.

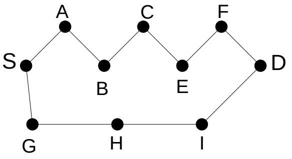

Fig. 1. The route from S to D found by GPSR and DIR is $SABCEFD$ while the shortest path is $SGHID$.

In some cases *DIR* and *GPSR* cannot find a route to the destination although such a route exists. Figure 2 shows an example where both of these algorithms fail to find a route to the destination. As mentioned above, *GPSR* finds

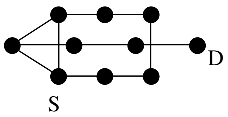

Fig. 2. DIR and GPSR fail to find a route from S to D.

a route to the destination if a planar subgraph could be extracted locally which is not always possible in the ad-hoc model considered in this paper, due to the variability of the transmission ranges [1]. Furthermore, as we said the length of the found route can be much larger than the shortest path [3].

Several other position based routing algorithms have been proposed, but they have similar shortcomings of either not guaranteeing to find a path to the destination or finding a path which is much longer than the shortest path [8], [2] and [10]. Although position based algorithms have the above disadvantages, they have some useful characteristics: They don't need to use memory in the intermediate nodes since they don't have any routing table and there is no need to keep the traffic history of packets. Also no control packet needs to be exchanged. It makes them very simple to implement with little overhead of routing. Moreover failures occur rarely when the network graph is dense [8].

B. Ant colony based routing algorithms

Ant colony optimization(ACO) is a stochastic approach for solving combinatorial optimization problems like routing in computer networks. The idea of this optimization is based on the observation of how ants optimize food gathering in the nature.

Ant colony optimization algorithms [5], [4], [9], [11], [15], [6] use artificial ants to iteratively construct a solution for an optimization problem. We can explain an ant colony optimization algorithm in a graph G informally as follows. A pheromone trail and a heuristic pheromone value is associated with each edge of G. A folk of ants move on the adjacent nodes of G concurrently and asynchronously to find an optimum solution. Each ant selects the next hop by making a stochastic decision using the existing pheromone trails and heuristic information. The solution is built incrementally as the ants move from one node to another node. While moving on the graph and building a solution, an ant evaluates this solution and deposits pheromone on its way. This pheromone trail will be used by the future ants to make a routing decision. Pheromone evaporation is a process that causes the amount of pheromone which is deposited on the links of the graph to be decreased over time. Regarding a set of termination conditions T, an ant stops when one of the conditions in T is satisfied. In most cases an ant retraces its path backward and modifies the pheromone values after it finds a solution.

ANTNET [5] is a proactive ACO routing algorithm for packet switch networks. In this algorithm, a forward ant is launched from the source node at regular intervals. A forward ant at each intermediate node selects the next hop using the information stored in the routing table of that node. The next node is selected with a probability proportional to the goodness of that node which is measured by the amount of pheromone deposited on the link to that node. The probability is calculated using Formula 1.

$$p_i = \frac{\phi_i}{\sum_{j=1}^{k} \phi_j} \tag{1}$$

When a forward ant reaches the destination, it generates a backward ant which takes the same path as the corresponding forward ant but in opposite direction. The backward ant updates pheromone values as it moves on its way to the source node.

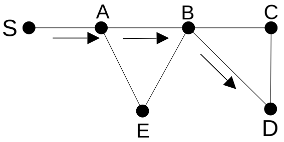

Fig. 3. A forward ant is sent from the source node(S) to the destination node (D).

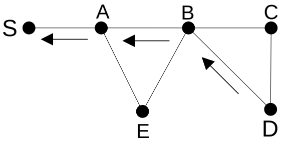

Fig. 4. A backward ant is sent from the destination node(D) to the source node (S).

ARA [9] is a reactive ACO routing algorithm for mobile ad hoc networks. ARA has two phases: route discovery, and route maintenance. In route discovery phase, the sender broadcasts a forward ant. The ant is relayed by each intermediate node until reaches the destination. After receiving a forward ant in the destination, the ant is destroyed and a backward ant is sent back to the sender. The backward ant increases the pheromone value corresponding to the destination in each intermediary node until it reaches the sender. When the sender receives a backward ant, the route maintenance phase starts by sending data packets. Since the pheromone track is already established by the forward and backward ants, subsequent data packets will perform the route maintenance by adjusting the pheromone values.

ANTHOCNET is a hybrid routing algorithm that combines reactive route establishment with proactive route maintenance. When there is a demand for a route, the algorithm first checks the routing table of the source node to see if there is any information about the destination. If there is no routing information for the destination, it will broadcast a forward ant. Due to this broadcasting, each neighbor receives a copy of the ant and checks its routing table. Again, the ant will be broadcasted if there is no routing information for the destination, otherwise it will be sent to the next hop using a stochastic decision. Each forward ant keeps a list of the visited nodes and when reaches the destination, uses it to take the same path back to source and update the pheromone values deposited in the links. To avoid generating a huge number of ants of the same generation, a limit a on the length of a path is defined and when the length of the traveled path by an ant exceeds a, the ant is discarded. Whenever an intermediate node receives two ants of the same generation, checks the length of the traveled path of both ants and discards the ant with longer traveled path if the difference is more than a certain value (α_1). This algorithm uses a stochastic scheme similar to that used in ANTNET for routing data packets. While a data session is running, proactive ants will be sent to perform route maintenance.

In general, ACO routing algorithms guarantee message delivery [5], [9], [14]. They converge to a route which is very close to the optimum route. In most cases ANTNET has a long convergence time. As mentioned above, ANTNET launches an ant at regular intervals. The more the number of ants needed to converge to a route, the more the time elapsed since the start of the algorithm. ANTHOCNET and ARA broadcast an ant in the route discovery phase. Although an ant is a small control packet, broadcasting implies a large overhead on the network especially when the size of the network is too big. In ANTHOCNET, since an ant contains a list of the visited nodes, its size will grow as it goes far from the source and the routing overhead will be increased. As a result ARA and ANTHOCNET are not scalable.

III. POSANT ROUTING ALGORITHM

In this section we define POSANT, a new ant colony optimization based routing algorithm which uses location information to improve its efficiency. POSANT is able to find optimum or nearly optimum routes when a given network contains nodes of different transmission ranges. As stated before, POSANT is reactive, thus a route is searched for only when there is a collection of data packets that are to be sent from a source node, say S, to a destination node, say D. Sending the data packets will start after a route from S to D is established. Before that, only forward and backward ants are being exchanged. In order to minimize the time that POSANT needs to find a route while keeping the number of generated ants as small as possible, information about the position of nodes is used as a heuristic value. In the next subsection we introduce the concept of zones which play an important role in our algorithm.

A. Zones

Consider a destination node D and a network graph G. For each node S (i.e. S is not necessarily the source node) we partition its neighbors into 3 zones called $zone_1, zone_2$ and $zone_3$. Consider a line segment between S to D. For a neighbor H of S, angle θ_H is defined as the angle between line segments SH and SD. Node H belongs to $zone_1$ if $\theta_H \leq \pi/4$, $zone_2$ if $\pi/4 < \theta_H < 3\pi/4$, and $zone_3$ if $3\pi/4 \leq \theta_H \leq \pi$, see Figure 5.

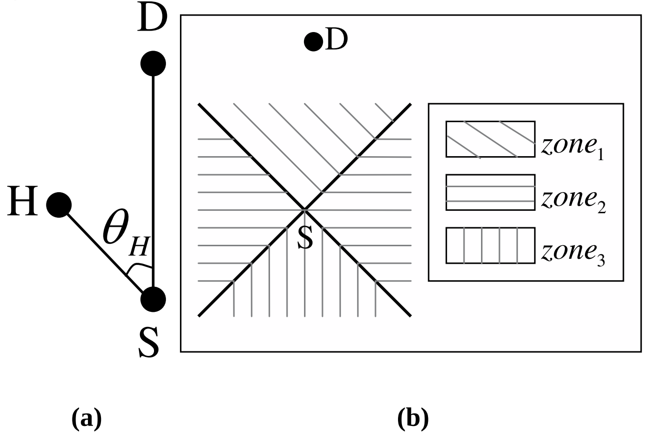

Fig. 5. (a) θ is the angle between SH and SD. (b) Different zones of N for destination node D.

B. Pheromone initialization

To start route establishment step, the amount of pheromone deposited in the edges of the graph must be initialized. For each node like A in the network, a pheromone trail is assigned to each of its outgoing links. Suppose B is one of the neighbors of A. To initialize the value of deposited pheromone on $\vec{AB}$ we follow a greedy policy, the assigned values decrease with the zone number. More formally, having three values ν_1, ν_2 and ν_3 such that $\nu_1 \geq \nu_2 \geq \nu_3$, an amount of pheromone equal to ν_i will be assigned to $\vec{AB}$ if B belongs to $zone_i$. The motivation is that in most cases shortest paths pass through the neighbors whose directions are closer to the direction of the destination. As a result, this greedy initialization causes a faster convergence to a shortest path most of the time. Our experiments show that the value of ν_1, ν_2 and ν_3 has a significant effect on the performance of POSANT.

POSANT assumes that each node maintains a table of the values of pheromone trails assigned to its outgoing links for different destinations. Whenever a node receives a packet for a specific destination, it will check its table to see if there is at least one pheromone trail for that destination. If such pheromone trail exists, it will be used for making a stochastic decision (which will be explained in detail later) to choose the next hop. If it doesn't exist, the pheromone initialization process begins and assigns pheromone trails to all the outgoing links. As just explained, the amount of the deposited pheromone on each link depends on the zone of the corresponding neighbor. The pheromone trails for a specific destination will be deleted from the pheromone trail table of a node if the node doesn't receive any packet pointing to that destination for more than a specific time which is defined to be in the order of seconds. The entries in the *Back Routing* (BR) table which are related to that destination will also be deleted after that time. The role of the BR tables is explained in the next section.

C. Route establishment

Consider a destination node D and a source node S with its three zones. To establish a route, S launches n forward ants with unique sequence numbers from each zone at regular time intervals ($3n$ ants each time). In our experiments we set n to be 1. Assigning very large value to n increases the overhead of the algorithm without resulting to a significant improvement. Similar to other ACO routing algorithms, at each node a forward ant makes a stochastic decision which is based on the values of pheromone trails to select the next hop. As mentioned before, the values of pheromone trails are stored in a table at each node. Suppose that a forward ant is currently residing in node N and this node has k neighbors $H_1, H_2, ..., H_k$, and ϕ_i is the amount of pheromone assigned to $\vec{NH_i}$. The forward ant will select H_i as the next node with a probability p_i which is calculated using the following equation.

$$p_i = \frac{\phi_i}{\sum_{j=1}^{k} \phi_j} \tag{2}$$

In addition to the pheromone trial table discussed before, each node maintains another table which we call *Back Routing* (BR) table. Whenever a forward ant enters a node from one of its neighbors, an entry in the BR table will be created that stores the identifier of the neighbor which the forward ant is coming from, the sequence number of the ant and the identifier of the destination. Repeated forward ants will be destroyed. When a forward ant reaches the destination, it is destroyed and a backward ant is sent back to the source. This backward ant has the same sequence number as the corresponding forward ant and traverses the same path to the source using the information stored in BR tables. Moving from node B to node A, the backward ant increases the amount of pheromone stored in $\vec{AB}$ using the following formula.

$$\phi_{\vec{AB}} = \phi_{\vec{AB}} + g(d) \times \omega(\vec{AB}) \tag{3}$$

In the above formula, d is the length of the traveled path from the destination to node B by the backward ant and $g(d)$ is a decreasing function of d. Function ω is a *weight function* and its value is dependent on the zone of A in which B is residing, as given in Formula 4.

$$\omega(\vec{AB}) = \begin{cases} w_1 \geq 1 & \text{if B is in } zone_1 \text{ of } A \\ w_2 = 1 & \text{if B is in } zone_2 \text{ of } A \\ w_3 \leq 1 & \text{if B is in } zone_3 \text{ of } A \end{cases} \tag{4}$$

Using the above weight function yields a faster convergence in most cases because the shortest path usually passes through the nodes which are closer in direction to the destination. The motivation is similar to the one in the greedy initialization.

An evaporation process causes the amount of pheromone deposited in each link to decrease as the time passes on. It is done by multiplying the current pheromone value by a number $\alpha < 1$ at regular time intervals.

$$\phi(\vec{BA}) = \alpha \times \phi(\vec{BA}) \tag{5}$$

Sometimes, especially at the beginning of route establishment process, some ants take non-optimum routes to reach the destination. At their way back to the source they will update pheromone trials. Pheromone evaporation reduces the effect of these updates as the time passes on.

The above stochastic strategy, described in Algorithm 1, establishes multiple paths between the source and destination. As a result, in contrast to regular position based routing algorithms which usually find a single route to the destination, POSANT is a *multipath routing algorithm* (i.e. like other ACO routing algorithms). Reducing the chance of congestion in the network is an advantage of multipath routing algorithms.

Algorithm 1 POSANT routing algorithm

```
{$S$ is the source node, $D$ is the destination node and $C$ is the current node}
for each clock time do
  if $C = S$ then
    if one of the conditions in Section III-D is true then
      send one datapacket from each of the three zones of $C$
    else
      send one forward ant from each of the three zones of $C$
    end if
  end if
  for each message $m$ in $C$'s buffer do
    if (m→type = $ForwardAnt$) or (m→type = $DataPacket$) then
      $NextHop$ = SelectNextHop(n)
      send $m$ to $NextHop$
      if $NextHop = D$ then
        $m$ →type = $BackwardAnt$
      end if
    else if m→type = $BackwardAnt$ then
      find $NextHop$ in $C$'s BackRouting table
      send $m$ to $NextHop$
      IncreasePheromone($NextHop$,$m$)
      if $NextHop = S$ then
        update averages of packet delays for the corresponding zone
        drop $m$
      end if
    end if
  end for
  Evaporate()
end for
```

Algorithm 2 SelectNextHop

```
Input: node $N$
for i = 1 to the number of N's neighbors do
  $p_i = \frac{\phi_{gr_i}}{\sum_{j=1}^{k} \phi_{gr_j}}$
  return neighbor i with probability $p_i$
end for
```

D. Sending data packets

As stated before, to establish routes to the destination, the sender launches n forward ants in each of its three zones at regular time intervals. The modification of pheromone values by backward ants and the evaporation of pheromone cause a higher amount of pheromone to be deposited in some links than the others. It means that the algorithm converges to some routes. After that, sending data packets should be started. It is important to start sending data packets at an appropriate time. If we start sending them too early, they may be lost or follow long routes because optimum routes are not established yet. On the other hand, sending them too late increases data delivery delay. To estimate the best time to start sending data packets POSANT does as follows.

For each of the three zones of the source node for destination D, the average and standard deviation of the delays reported by backward ants is calculated. The number of nodes on a path taken by an ant to reach the destination is the delay of that ant. Each backward ant carries the length of the path passed from the destination to its current hop. Whenever a backward ant is received by the source node, the average and standard deviation of packet delays for the corresponding zone is updated. To reduce the effect of old backward ants, we define a fixed size *window* for each zone that contains recently received backward ants from that zone. The average and standard deviation of delays will be calculated only for the backward ants in the window. When a new backward ant is received we put it in the window of the corresponding zone and discard the oldest ant when the widow size has been reached. Selecting an appropriate window size is important. If the window size is too small, the average delay calculated from the window information would be too far from the real average. If the window size is very big, existence of very old ants would affect the result for a long time. Suppose α_i and σ_i are the average and standard deviation of the delays reported by the backward ants received from $zone_i$ and residing in the window.

- If σ_i is less than a threshold, t, sending forward ants from $zone_i$ is stopped and sending data packets is started.
- If $\sigma_i, \sigma_j < t$ and $\alpha_j > \alpha_i + c$, sending forward ants or data packets from $zone_j$ is stopped. In this formula c is a constant value which determines how different the length of the established routes can be.

If σ_i is small enough, almost all the ants launched from $zone_i$ are following routes with almost the same length to the destination. Hence the algorithm has converged to a route or a group of routes with similar lengths. Thus it is a reasonable time to start sending data packets from $zone_i$.

If the second condition is true, the algorithm won't converge to a route (i.e. or a group of routes with almost the same length) passing through $zone_j$ whose length is shorter than or equal to the average length reported by the ants launched from $zone_i$ plus a constant value, c. In this case the algorithm does not use routes passing through $zone_j$ anymore and stops sending data packets or ants from this zone. It is done by removing all the pheromone trials assigned to the outgoing links in $zone_j$ for destination D from the pheromone trail table of the source node. Assigning big values to c causes the algorithm to establish multiple paths to the destination but choosing very big values may allow non-optimum routes to be used.

In [5], [7] and [14] it is shown that the ant colony policy eventually converges to a route or group of routes with the same length, so the first condition will eventually become true. In the next section we show that sending data packets is started quite soon in practice.

Data packets follow the same process as the forward ants.

A precise selection of the parameters of POSANT is discussed in Chapter IV

E. Failure recovery

If a link between two nodes say A and B breaks while a connection is running between a source S and a destination D, POSANT does as follows. For each node, POSANT defines a mode that can have two values, the *regular mode* or the *broken mode*. Initially each node is in the regular mode. Whenever a link to one of the neighbors breaks, the mode of this node changes to the broken mode. Suppose A realizes that the link to B is broken and there is a pheromone trial corresponding to link $\vec{AB}$ for D in the pheromone table of A. In this case the stochastic data routing will continue but if there is no pheromone trail for D corresponding to any of the other outgoing links of A, node A sends a message to its neighbors to inform them that there is no route to D from A. Upon receiving this message, these neighbors do the same as if the link to A is broken. To avoid a loop, if A receives a packet for the second time while it is in broken mode, it sends a message to its neighbors to inform them that there is no route to the destination from A. If the source node has only one outgoing link that contains a pheromone trail for D, and this link breaks or a message from this link is received that states there is no route to D, a new route establishment process will begin and sending data packets will be suspended until a new route is found. The mode of A will be changed to regular mode after a specific time, which is defined to be in the order of milliseconds, is passed from the moment that the link failure is detected.

IV. PERFORMANCE EVALUATION

In this section we first study the influence of different parameters of POSANT on its performance and then we carry out a comparison of POSANT with ANTNET, ANTHOCNET and GPSR by simulating these algorithms on a set of randomly generated networks. In our experiments, we used network graphs with 80 nodes randomly and uniformly spread over a square of length 500 units. The transmission range of each node is randomly selected for different directions and its value is between 25 and 45 units. Two nodes are connected if each one is in the

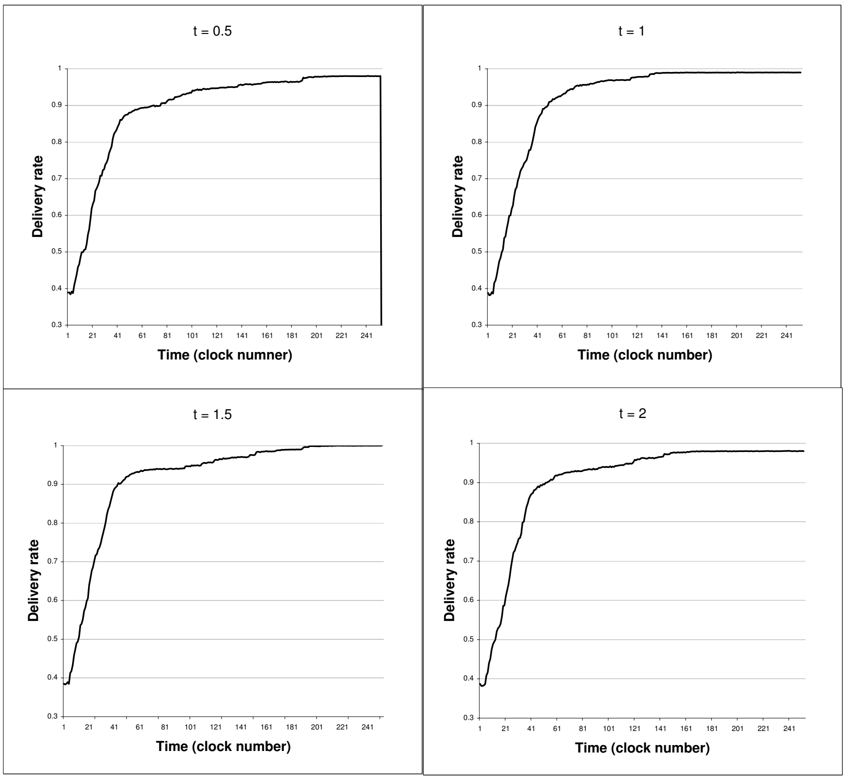

Fig. 6. Average delivery rate of POSANT with different values of threshold t.

TABLE I
PARAMETER VALUES USED IN POSANT.

Parameter	Value	Description
t	1	standard deviation threshold
(ν_1, ν_2, ν_3)	(20,1,1)	used in pheromone initialization
μ	0.95	used in pheromone evaporation
$window\ size$	50	
$g(d)$	$\frac{1000}{d^2}$	used in equation 3

TABLE II
PARAMETER VALUES USED IN ANTHOCNET.

Parameter	Value	Description
μ	0.95	used in pheromone evaporation
α_1	2	
a	20	used in pheromone evaporation

transmission range of the other one. If the network graph is not fully connected, the source and destination may reside in different partitions which means that there is no route to the destination. In our experiments we tried the routing algorithms only on connected graphs which are generated as mentioned. To estimate the performance, we ran the algorithms on 200 networks and averaged the results. For each network, one source-destination pair is selected randomly.

The algorithms are evaluated in terms of *delivery rate*, *packet delay* and *overhead*. The delivery rate is defined as the ratio of the number of packets received by the destination node to the number of packets sent by the source node. We use the hop count to measure packet delays, and the overhead is measured as the number of control messages created and exchanged in the network by a routing algorithm.

In the next subsections we give the results of the simulation experiments.

A. Selection of the parameters of POSANT

We tried POSANT with different parameter settings on the set of randomly generated graphs. Among the parameter discussed in Section III, we study the effect of pheromone evaporation rate μ and the standard deviation treshold t on the performance of POSANT (see Section III for their definitions). In the following comparisons, except the parameters whose effect is studied, the other parameters have the values listed in Table I.

In Figure 6, 7 the effect of *t*, the threshold of the standard deviation of packet delays, on the performance of POSANT is studied. As these figures show, for $t = 1$ and $t = 1.5$ the performance of POSANT is the best. For $t = 0.5$ and $t = 2$, the graphs have relatively high fluctuations.

Figure 8, 9 show the effect of μ on the performance. The performance of POSANT with $\mu = 95\%$ is the best.

The values of ν_1, ν_2 and ν_3 have a significant effect on the performance of POSANT. In the next subsections we ran POSANT with different values assigned to these parameters to compare the results. POSANT(a, b, c) denotes a POSANT algorithm with ν_1, ν_2, ν_3 set to a, b, c.

Based on the above experiments, we set the parameters of POSANT as in Table I. The parameters of ANTHOCNET and their corresponding values are listed in Table II.

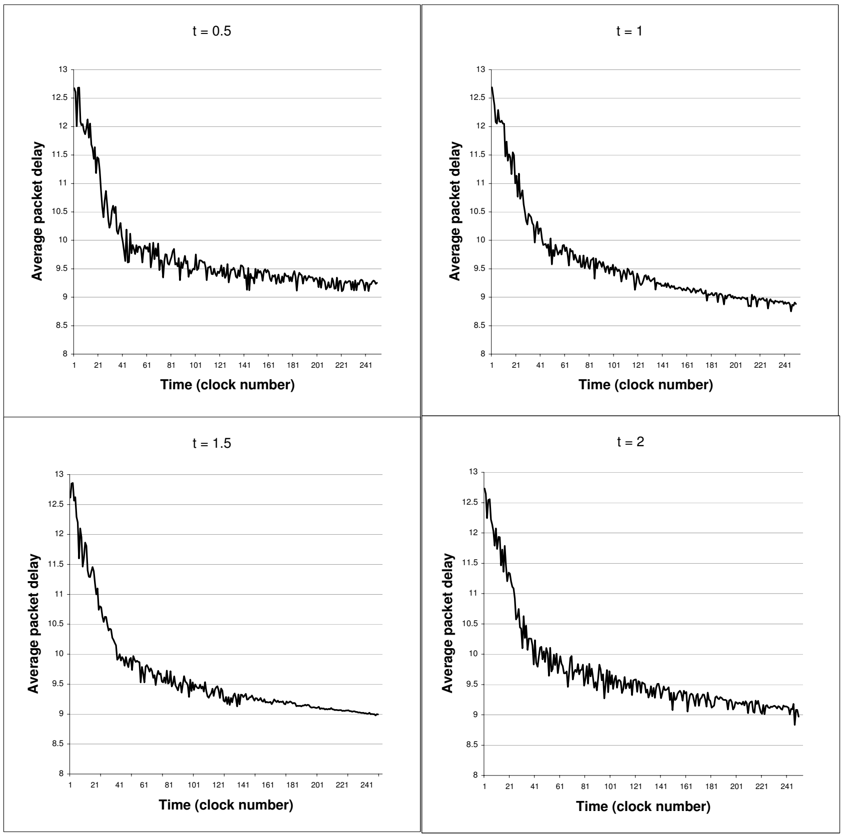

Fig. 7. Average packet delay in POSANT with different values of treshold t.

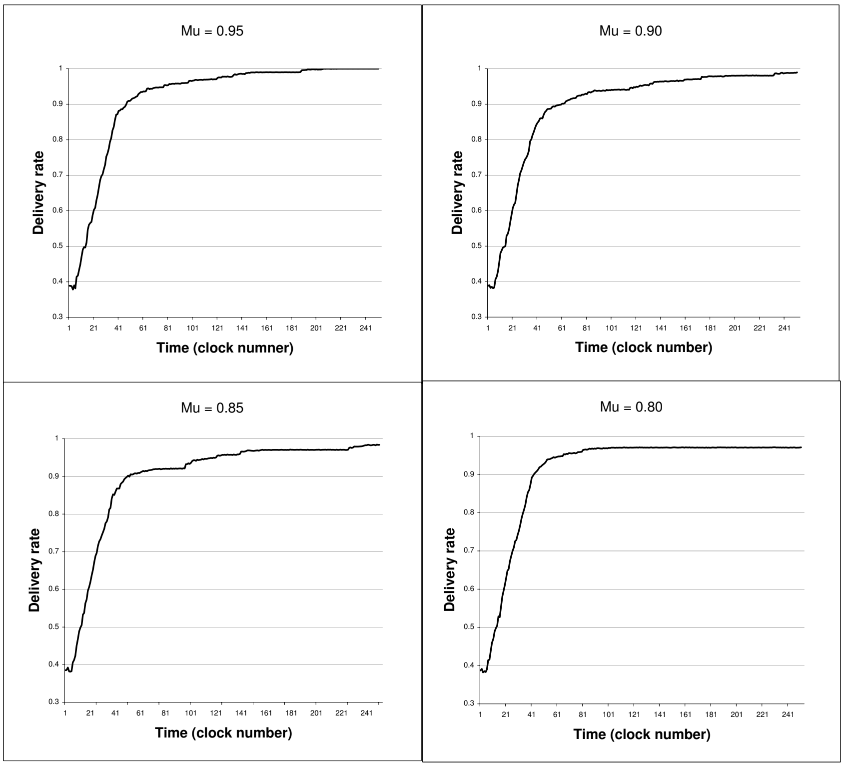

Fig. 8. Average delivery rate of POSANT with different values assigned to parameter μ.

For ANTNET algorithm, the evaporation rate was set to 0.95, the same as for POSANT.

B. A comparison of POSANT to other routing algorithms

1) Delivery rate: If a routing algorithm fails to deliver a packet to the destination, the sender must somehow

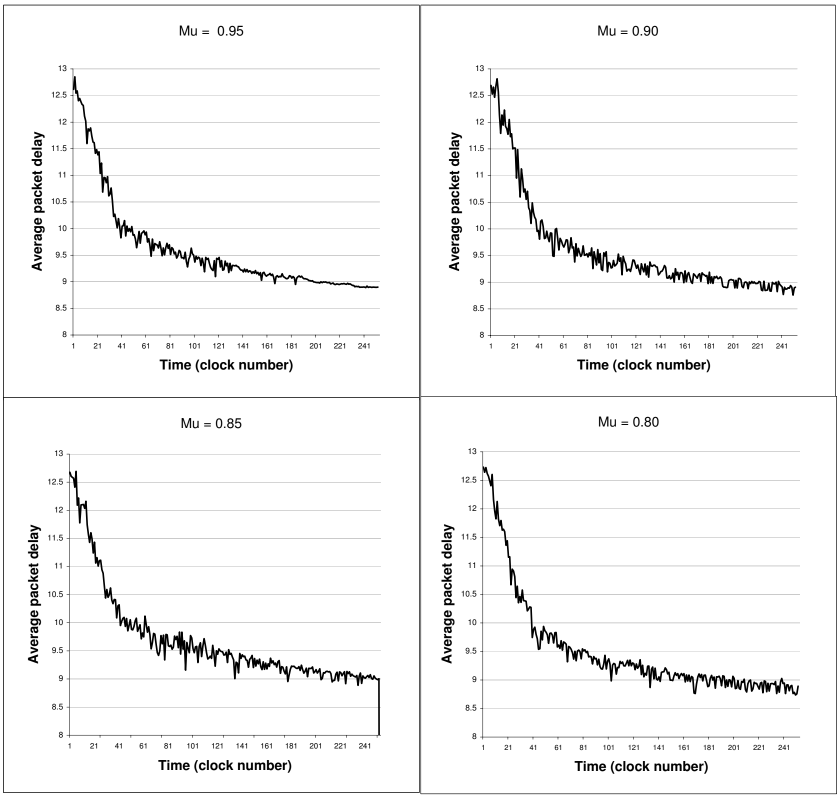

Fig. 9. Average packet delay in POSANT with different values assigned to parameter μ.

detect this failure and try to resend the packet hoping that this time the algorithm delivers it. This increases the traffic in the network and also the delay experienced by the receiver. Thus it is very important to guarantee high delivery rate. In Figure 10 the average delivery rate of ANTNET, ANTHOCNET and POSANT is compared. This figure shows how the average delivery rate varies with time. As the time progresses, the delivery rate increases and eventually becomes almost 100% for all algorithms except GPSR. As the graph shows, ANTHOCNET reachs to 100% delivery rate faster than the others. This is the result of broadcasting ants in this algorithm. GPSR always has a relatively low delivery rate. It is because this algorithm fails in some cases as a result of varying transmission ranges of the nodes. POSANT(20,1,1) has the highest delivery rate among the others. ANTNET and POSANT(20,20,20) reach to 100% delivery rate slower than the other ant based algorithms.

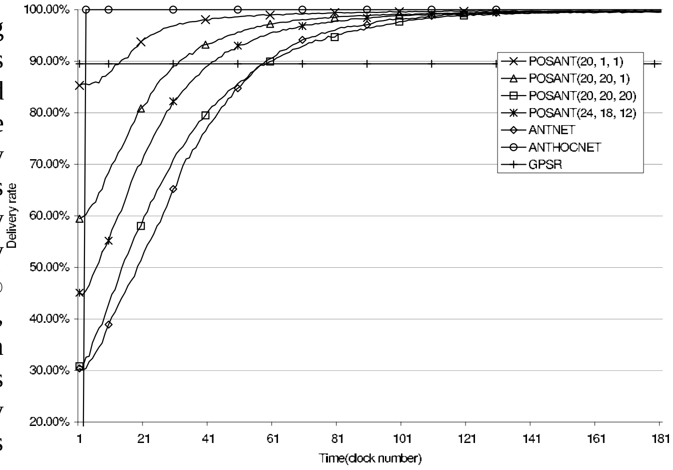

Fig. 10. Average delivery rate of ANTNET, POSANT, ANTHOCNET and GPSR.

2) Convergence time: In this part the average packet delay of POSANT is compared with that of ANTNET, ANTHOCNET and GPSR. The results of this comparison is presented in Figure 11 where hop count is used as the metric to measure packet delays. Figure 11 shows how the average packet delays for the different algorithms vary with time. In the beginning, the average packet delay of ANTHOCNET is smaller than the others. GPSR has a relatively small average delay at the beginning but it is still longer than that of ANTHOCNET. It is because GPSR doesn't find the shortest path in many cases. The average delay of POSANT(20,1,1) reaches to its minimum faster than the others. As the time passes on, all the algorithms except GPSR converge to the paths with almost the same lengths and the packets experience almost the same delay.

POSANT decides that a convergence to a route has been achieved when the standard deviation of packet delays is less than a threshold value. Since POSANT is reactive, the convergence time is important. Even for a proactive routing algorithm like ANTNET the convergence time should be small because the topology

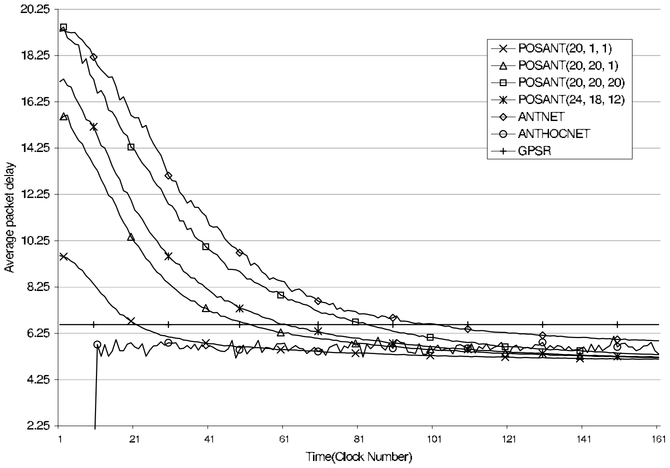

Fig. 11. Average packet delay of ANTNET, POSANT, ANTHOCNET and GPSR.

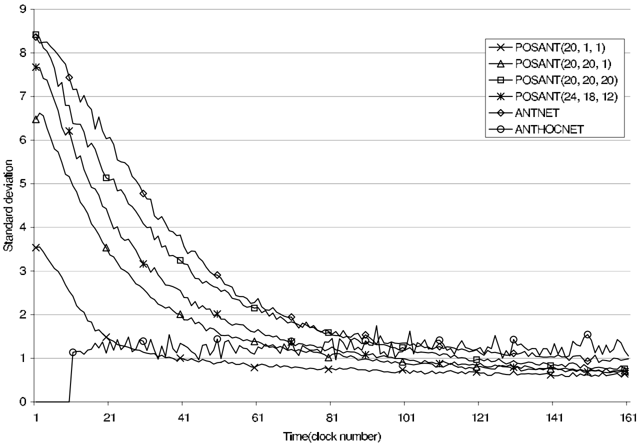

Fig. 12. The standard deviation of the packet delays in ANTNET, POSANT, ANTHOCNET and GPSR.

of the network is changing very often and a new route to the destination should be found as fast as possible. In Figure 12 the average standard deviation of the packet delays is shown. The average standard deviation of POSANT(20,1,1) and ANTHOCNET is smaller than the others in the beginning. A small standard deviation means that the packets are traveling paths with almost the same length.

3) Algorithm overhead: It is very important that the overhead of a routing algorithm be as small as possible. A big overhead makes a routing algorithm less useful and sometimes even useless by increasing the traffic, reducing the scalability and so on. One of the parameters that should be kept small is the amount of control traffic exchanged in the network when a routing protocol is running. In this subsection the number of control messages created and exchanged in the network by each routing algorithm is compared.

Figure 13 shows the number of generated ants by ANTHOCNET in different clock times. As the graph shows, a burst of ants is generated at the beginning of route establishment when the algorithm broadcasts an ant. The number of generated ants in each clock time is constant for POSANT and ANTNET. For ANTNET this value is 1 and for POSANT the average value is 2.6 (it is less than three because in some cases there is no neighbor in some zones so less than three ants will be generated in each clock). Table III compares the total number of ants generated by POSANT, ANTHOCNET and ANTNET. The total number of ants generated in ANTHOCNET grows exponentially at the beginning while it is a linear function of time in POSANT and ANTNET. The big number of generated ants in ANTHOCNET when it establishes a new route is a severe drawback of this algorithm. Each ant in ANTHOCNET contains a list of the visited nodes, so its size can be relatively large which makes the overhead even worst. The number of generated ants in POSANT and ANTNET is very small and doesn't affect the network's traffic.

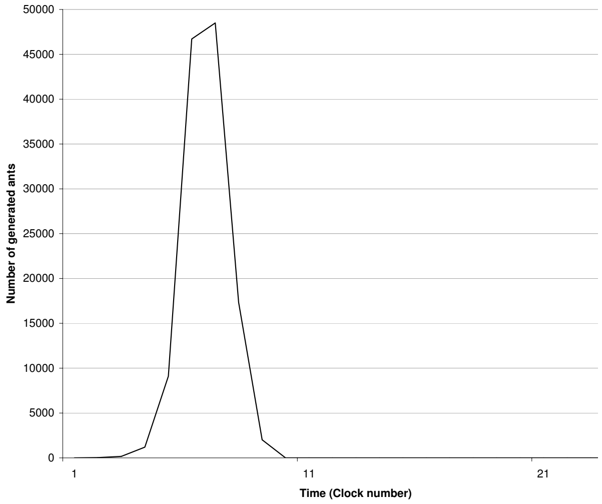

Fig. 13. Number of generated ants in ANTHOCNET algorithm at each clock time.

Clock time	0	5	10	300
POSANT	2.53	12.67	25.35	756.4
ANTNET	1	5	10	300
ANTHOCNET	1	10488	125048	125048

TABLE III
TOTAL NUMBER OF GENERATED ANTS.

V. CONCLUSIONS

In this paper we proposed POSANT, a new ant colony based routing algorithm that uses the information about the position of nodes to increase the efficiency of ant-routing. In contrast to other position based routing algorithms, POSANT does not fail when the network contains nodes with different transmission ranges. Unlike the previously defined position based routing algorithms which are single path, POSANT is a multipath routing algorithm. While in some cases regular position based routing algorithms find a route which is much longer than the shortest path, POSANT converges to routes which are close in length to the shortest path. The use of location information as a heuristic parameter resulted in a significant reduction of the time needed to establish routes from a source to a destination which is important for a reactive routing algorithm. In addition to having a short route establishment time, POSANT reduces greatly the number of generated control messages, unlike some ACO routing algorithms, e.g. ANTHOCNET, which use flooding to reduce the route establishment time, making them unscalable. Our simulations show that in our network model, POSANT has a higher delivery rate with a shorter average packet delay than GPSR. POSANT also reaches a stable behavior faster than ANTNET. Thus, POSANT is a robust scalable reactive routing algorithm suitable for mobile ad hoc networks with irregular transmission ranges.

REFERENCES

[1] L. Barrière, P. Fraigniaud, L. Narayanan, and J. Opatrny. Robust position-based routing in wireless ad-hoc networks with irregular transmission ranges. *Wireless Communications and Mobile Computing Journal*, pages 141–153, 2003.

[2] P. Bose and P. Morin. Competitive online routing in geometric graphs. *Theoretical Computer Science*, pages 273–288, 2004.

[3] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. *Wireless Networks*, 7:609–616, 2001.

[4] G. D. Caro, F. Ducatelle, and L. M. Gambardella. Anthocnet: An adaptive nature-inspired algorithm for routing in mobile ad hoc networks. *Special Issue on Self-Organisation in Mobile Networking*, 16:443–455, 2005.

[5] G. Di Caro and M. Dorigo. Ant colonies for adaptive routing in packet-switched communications networks. In *Proceedings of the 5th ACM International Conference on Parallel Problem Solving from Nature*, pages 673–682, 1998.

[6] G. Di Caro and M. Dorigo. Two ant colony algorithms for best-effort routing in datagram networks. In *Proceedings of the Tenth IASTED International Conference on Parallel and Distributed Computing and Systems*, pages 541–546, 1998.

[7] M. Dorigo and T. Stutzle. The ant colony optimization meta-heuristic: Algorithms, applications, and advances. In *Handbook of Metaheuristics*, volume 57 of *International Series in Operations Research and Management Science*, pages 251–258. Kluwer, 2003.

[8] S. Giordano, I. Stojmenovic, and I. Blazevic. Position based routing algorithms for ad hoc networks: A taxonomy. *Ad Hoc Wireless Networking*, 2003.

[9] M. Gunes, U. Sorges, and I. Bouazizi. Ara - the ant-colony based routing algorithm for manets. In *Proceedings of the 2002 International Conference on Parallel Processing Workshops*, pages 79–89, 2002.

[10] I. Haque, C. Assi, and J. Atwood. Randomized energy aware routing algorithms in mobile ad hoc networks. In *Proceedings of the 8th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pages 71–78, 2005.

[11] P. Jeon and G. Kesidis. Pheromone-aided robust multipath and multipriority routing in wireless manets. In *Proceedings of the 2nd ACM international workshop on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*, pages 106–113, 2005.

[12] B. Karp and H. T. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. In *Proc. ACM/IEEE MobiCom conference*, pages 243–254, 2000.

[13] E. Kranakis, H. Singh, and J. Urrutia. Compass routing on geometric networks. In *Proc. of 11th Canadian Conference on Computational Geometry*, pages 51–54, August 1999.

[14] J.-H. Yoo, R. La, and A. Makowski. Convergence results for ant routing. In *Proceedings of CISS University of Princeton, Princeton, NJ*, 2004.

[15] S. Ziane and A. Mellouk. A swarm intelligent multi-path routing for multimedia traffic over mobile ad hoc networks. In *Proceedings of the 1st ACM international workshop on Quality of service and security in wireless and mobile networks*, pages 55–62, 2005.