RESEARCH PAPER
International Journal of Recent Trends in Engineering, Vol 1, No. 1, May 2009

Swarm Based Intelligent Routing for MANETs

S. Prasad[1], Y.P.Singh[2], and C.S.Rai[2]
[1] Centre for Development of Advanced Computing, NOIDA, India
{sunitaprasad@cdacnoida.in}
[2] Faculty of Information Technology, Multimedia University, Malaysia
{y.p.singh@mmu.edu.my}
[3]Guru Gobind Singh Indraprastha University, Delhi, India
{csrai@ipu.ac.in}

Abstract—***The paper presents a novel proactive algorithm to routing called Probabilistic Ant Routing, in mobile ad hoc networks, which is inspired by Ant Colony Optimization (ACO) framework and uses "ants" for route discovery, maintenance and improvement. The algorithm is based on a modification of the state transition rule of ACO routing algorithm that results in maintaining higher degree of exploration along with congestion awareness in the search space. This leads to reduced end-to-end delay and also lowers the overhead at high node density. The comparative experimental results of the proposed algorithm with the state-of-the-art AODV reactive routing algorithm of the MANET are provided keeping mobility and density of nodes as the main consideration. The proposed algorithm is tested for different network sizes and node mobility. The experimental results are very promising as the proposed algorithm exhibits superior performance with respect to reactive AODV routing algorithm in terms of end-to-end delay.***

Index Terms— **Mobile Ad Hoc Networks (MANET), Ant Colony Optimization (ACO), Ad Hoc On-Demand Distance Vector Routing (AODV)**

I. INTRODUCTION

Mobile Ad Hoc Networks (MANETs) are networks in which all nodes are mobile and communicate with each other via multihop wireless connection. A MANET is a self-organizing network consisting of wireless nodes without any central control. A node in the network can communicate with another node only if it lies within its transmission range. The dynamic topology of MANETs makes routing a challenging task, as the existing path is rendered inefficient and infeasible. Mobile ad hoc networks have potential applications in civilian and military environments. The major issues of concern for mobile ad hoc networks are medium access control (MAC), routing, security and quality of service provisioning. This paper addresses the routing problem in a mobile ad hoc network without considering the other issues, i.e., access control, security and the QoS factor. Routing in MANETs can be defined as the directed flow of data from source to destination maximizing the network performance. The dynamic topology of the MANETs places two fundamental requirements on the routing protocol – (i) The protocol should be distributed and (ii) The protocol should be able to compute multiple loop-free routes while keeping the communication overhead to a minimum.

The paper proposes a proactive ant based routing approach for MANETs inspired by the Ant Colony Optimization paradigm. The algorithm proactively sets up multiple paths between the source and the destination. The two factors that affect the performance of a probabilistic algorithm are - exploration and exploitation. In a dynamically changing topology of the mobile ad hoc networks where there are frequent link breakages due to node mobility, an optimal balance between exploration and exploitation is required. More emphasis on exploitation will cause the probabilities of few routes to saturate to 1 and the probabilities of other routes to saturate to 0. As a result new routes will never be discovered. The paper proposes a modification of the state transition rule in ACO to balance exploration and exploitation. According to the modified rule, the ants may be unicast or broadcast at a node depending on the route information. If the route information to the destination is present, the ants are unicast, otherwise it is broadcast.

The paper is organized as follows. Section II briefly describes the routing algorithms used in mobile ad hoc networks. Section III presents the Ant Colony metaheuristics and the related work. Section IV presents a detailed description of the proposed algorithm and Section V reports the experimental settings and the simulation results along with AODV routing algorithm in MANETs. Finally, some conclusions and outlines the direction for future work are derived.

II. ROUTING IN MANET

Routing in MANET is a Dynamic Optimization Problem as the search space changes over time. The routing policy is defined as the rule that specifies what node to take next at each decision node to reach the destination node. Due to the time varying nature of the topology of the networks, traditional routing techniques such as distance-vector and link-state algorithms that are used in fixed networks, cannot be directly applied to mobile ad hoc networks. The constraints of MANETs demand the need of specialized routing algorithms that can work in a decentralized and self-organizing way. The routing protocol of a MANET must dynamically adapt to the variations in the network topology.

The routing scheme in a MANET can be classified into two major categories – Proactive and Reactive. The proactive or table driven routing protocols maintain routes between all node pairs all the time. It uses periodic broadcast advertisements to keep routing table up-to-date. This approach suffers from problems like increased overhead, reduced scalability and lack of flexibility to respond to dynamic changes. The reactive or on-demand approach is event driven and the routing information is exchanged only when the demand arises. The route discovery is initiated by the source. Hybrid approaches combines the features of both the approaches. Destination Sequence Distance Vector (DSDV) is a flat proactive routing protocol [2] whereas Dynamic Source Routing (DSR) and Ad Hoc On Demand Distance Vector (AODV) routing are examples of flat reactive or on-demand protocols [3][4]. The increased latency may render the reactive protocols unsuitable for time-critical traffic. Proactive routing protocols and its hybrid versions may be looked as only option for such applications.

III. Ant colony Metaheuristics

The Ant colony optimization is based on the foraging behavior of ants [6]. When ants search for food, they wander randomly and upon finding food return to their colony while laying a chemical substance called pheromone. Many ants may travel through different routes to the same food source. The ants, which travel the shortest path, reinforce the path with more pheromone that aids other ants to follow. Subsequently more ants are attracted by this pheromone trail, which reinforces the path even more. This autocatalytic behavior quickly identifies the shortest path. Ants are simple autonomous agents that interact via indirect communication known as *stigmergy*. *Stigmergy* is an indirect form of communication where individual agents leave signals in the environment and other agents sense them to drive their own behavior. This form of communication is local wherein simple agents interact locally without having any global information.

A. Exploration and Exploitation

The success of any stochastic search method heavily depends on striking an optimal balance between exploration and exploitation. These two issues are conflicting but very crucial for all the metaheuristic algorithms. Exploitation is to effectively use the good solutions found in the past search whereas exploration is expanding the search to the unexplored areas of the search space for promising solutions. The reinforcement of the pheromone trail by the artificial ants exploits the good solution found in the past. However, excessive reinforcement may lead to premature convergence. To maintain the diversity in the search space the following methods are suggested.

(a) Control the exponent of the pheromone trail.

(b) Introduce additional randomization in the ants' decision process

In this paper, we introduce randomization by modifying the transition rule of the ant at the node.

B. Related Work

Many routing algorithms based on ACO have been proposed for MANETs in the recent past. Most of the proposed algorithms are reactive and use broadcast for exploration. *Accelerated Ants Routing* is a path-based ant routing for mobile ad hoc networks [8] and uses a special kind of ant for exploration called as uniform ant. In this, the ant agents move in the network randomly without any specific destination and update the pheromone entries pointing to their source. The good paths are reinforced by the regular ants, which use the pheromone information at the nodes to move in the network. Ant Colony Base Routing (ARA) is a reactive routing algorithm [9]. The forward ants are broadcast in the network to find a path to the destination. This sets up multiple paths between the source and the destination at the start of the data session. Regular data packets are used to reinforce the good paths discovered during the route discovery session. The Probabilistic Emergent Routing Algorithm [10] is a flooding based on-demand routing algorithm and which reactively establishes route to the destination using delay as the metric. Multiple paths are setup but one with the highest pheromone value is used for the data session. Ant-AODV is a hybrid algorithm that combines ants with AODV [11]. In this, a fixed number of ants keep going around the network, keeping track of the last *n* visited nodes and when they arrive they proactively update its routing table. In general, these algorithms are not very different from the single path on-demand algorithm. AntHocNet uses a hybrid approach- reactive path setup, which is proactively improved [12]. In this, the exploration is done based only on the pheromone value and the proactive path improvement is concentrated around the current path.

The proposed Probabilistic Ant Routing algorithm is a proactive routing protocol and it uses both pheromone and heuristic value for path exploration to maintain unexpired route connectivity.

IV. Probabilistic Ant Routing

The dynamic topology of MANETs categorizes the problem as distributive and non-stationary and makes it suitable to be solved by the multi-agent approach. The learning strategy of the agents is based upon (i) pheromone, which represents the information gained from the past experience and encodes a long-term memory about the whole search process and (ii) heuristics, which represents the run time information. The amount of pheromone deposited represents the quality of the solution found.

A. Data Structures for Routing in ProbabilisticAnt Routing

Forward and Backward Ants

The proposed algorithm uses two kinds of agents – Forward and Backward Ants. The forward ants (FANT) are probabilistic and explore the network to collect the network traffic information. They are routed on normal priority queues, i.e., they use the same queues as normal data packets. When the FANT reach the destination, it is deallocated and the backward ant (BANT) inherits the stack contained in the FANT. The BANT is deterministic and is sent out on high priority queue. The backward ants retrace the path of the FANT and utilize this information to update the routing tables and other data structures periodically**.** These mobile agents are small and light packets containing source IP address, Destination IP address, Packet ID and a dynamically growing stack consisting of Node_ID and the Node_Traversal_Time.

Routing and Traffic Statistics Table

Each node maintains a probabilistic routing table. The probability value $P_{i,j,d}$ expresses the desirability of node *i* of taking node *j* as the next hop to reach the destination *d*. The routing probability value is computed as given in Equation (1). In addition, each node also maintains a statistics table for each destination *d* to which the forward ant has been previously sent. The statistics table contains the mean and the variance (μ_{id}, σ_{id}^2) for the routes and estimates the average trip times from the current node *i* to the destination node *d*.

B. Overview of Proposed Algorithm

The sequence of steps required in the proposed algorithm is summarized as follows.

- A node builds a neighbor list by using single hop HELLO message packets. These packets are transmitted at regular intervals (periodically). Each node is assigned a probability of *1/N* where *N* is the number of neighbors of node *i*. This initializes the routing table at every node. The identical probabilities indicate that nothing is known about the state of the network.
- The forward ants are generated by the source *s* at regular intervals. The FANT may be unicast or broadcast depending on the availability of a route to the destination *d* at node *i*. If the route to *d* is available at the node, the ant is unicast as given in Equation (1). If the route to *d* is not available, then the ant is broadcast.
- While moving towards the destination, the FANT pushes the Node_ID and the Node_Traversal_Time of every intermediate node, it visited into its stack *S* until it reaches the destination *d*.

When the FANT reaches the destination, it passes all its information about the route to the Backward Ant (BANT) created by the node and dies.

The high level description of the algorithm is given in Figure 1.

C. Proactive Path Setup

The Probabilistic Ant Routing sets up a path proactively. The FANTs are either unicast or broadcast at each node depending on the availability of the routing information for destination *d*. If the routing information is available, the ants stochastically choose the next hop for the destination. This scheme helps limit the route maintenance overhead. If the pheromone information is available at the node *i*, then the routing probability $P_{i,j,d}$ of choosing neighbor node *j* as the next hop for destination *d* is given as follows.

$$P_{i,j,d} = \frac{[\tau_{i,j,d}]^{\alpha}[\eta_{i,j}]^{\beta}}{\sum_{l \in N_i}[\tau_{i.l.d}]^{\alpha}[\eta_{i.l}]^{\beta}} \qquad (1)$$

where α and β are the relative weights for pheromone trail $\tau_{i,j,d}$ and the heuristic value $\eta_{i,j}$. For optimal performance, normally α is set to 1 and β is set to 2 [6]. N_i are the neighbors of node *i*. The heuristic value $\eta_{i,j}$ is calculated with the network interface queue length $(q_{i,j})$ on the outgoing link connecting node *i* and neighbor j given as

$$\eta_{i,j} = 1 - \frac{q_{i,j}}{\sum_{l \in N_i} q_{i,l}} \qquad (2)$$

The algorithm is made congestion aware by the incorporation of the local heuristic value at each node, $q_{i,j}$, since it reflects the instantaneous state of each node's queue, and gives a quantitative measure associated with the queue waiting time.

D. Pheromone Update Scheme

The ant algorithm is a variant of reinforcement learning. The rules for update are as follows.

1. Increase the probability of the hop of the node from where the ant packet has immediately come from.
2. Decrease the probabilities of other hops.

Case (1) provides the positive feedback for ant routing and case (2) provides the negative feedback. The positive feedback quickly identifies the best paths. The elapsed trip time T_{id} from node *i* to destination *d*, is the reinforcement signal which indicates the goodness of the route. Every discovered path by the forward ant receives a positive reinforcement in its selection by the BANT.

$$P_{fd} = P_{fd} + r_{+} \qquad (3)$$

where P_{fd} is the previous routing table probability values assigned to the neighbor *f* of the current node for the destination *d* and r_+ is the positive reinforcement. At the same time, a negative reinforcement occurs for the other neighbors' by normalization. Their probabilities are reduced so that the sum of the probabilities always remains 1.

$$P_{nd} = P_{nd} + r_{-} \qquad (4)$$

where P_{nd} is the previous routing table probability values assigned to neighbors *n* of the current node for the destination *d* and r_{-} is the negative reinforcement.

E. Goodness Value Estimation

The local statistics table records the mean and the variance of the trip time to every destination. The mean is the estimated average time to go from node *i* to node *d* and σ_{id}^2 is its associated variance. The ratio of the variance to the mean (σ/μ) is used as a measure of the consistency of the trip times and accordingly, it alters the effect of the trip time on the routing table. The update of the routing table is done using the quantity r', which is derived according to [5].

$$r' = \begin{cases} \frac{T_{id}}{c\mu_{id}} & c \geq 1 \text{ if } \frac{T_{id}}{c\mu_{id}} < 1 \\ 1 & \text{otherwise} \end{cases} \quad (5)$$

where T_{id} is the trip time from the current node i to the destination d, μ_{id} is the average of the trip time and c is the scaling factor usually set to 2. The out-of-scale values of r' are saturated to 1. The reinforcement equations are given as:

$$r_+ = (1 - r')(1 - P_{fd}) \quad (6)$$

$$r_- = -(1 - r') P_{nd} \qquad n \neq f \quad n \in N_i \quad (7)$$

P_{fd} and P_{nd} are the previous routing table probabilities, f is the node from which the backward ant comes from, N_i is the neighbor of node i (current node) and d is the destination node.

F. Routing Data Packets

The data packets are routed through the path that has the best pheromone concentration but the next hop is chosen probabilistically. The path having higher probability is the primary path and other paths are available for backup. All the data packets are routed through a single path and no load balancing is provided.

V. EXPERIMENTAL RESULTS

In the first scenario, mobile ad hoc network consisting of 100 nodes placed randomly using uniform distribution in an area of 1000 x 600 m^2 is considered for simulation study. The nodes in the network have the transmission range of 300*m* and a channel capacity of 2 Mbps. The data traffic consists of 30 CBR sources sending four 512 bits packet per second. The mobility model used is Random Waypoint. In this, each node is randomly placed in the simulated area and remains stationary for a specified pause time. It then randomly chooses a destination and moves there at a velocity chosen uniformly between a minimum velocity v_{min} and a maximum velocity v_{max}. Each node independently repeats this movement pattern through the simulation. The experimental setup defines v_{min} as 0 m/s and v_{max} as 20 m/s and varies the pause time as the independent variable. The ant generation rate at each node was set to 0.3 seconds. OMNeT++, a discrete event simulator, is considered to implement the proposed routing protocol.

The performance of the proposed routing algorithm is gauged in terms of packet delivery ratio, average end-to-end delay and normalized routing overhead. The results presented here are the average of 10 runs obtained for the same simulation configuration of 30 active sources. The results obtained after simulation are compared with the well known reactive protocol AODV. Figure 2 shows the packet delivery ratio compared with AODV. The packet delivery ratio is higher for the ant algorithm as compared to AODV. At high mobility, the AODV has to reinitiate the route discovery process again. This leads to lower packet delivery ratio. The ant algorithm proactively maintains the path to the destination, which leads to better performance. At lower mobility, the performance is comparable as expected.

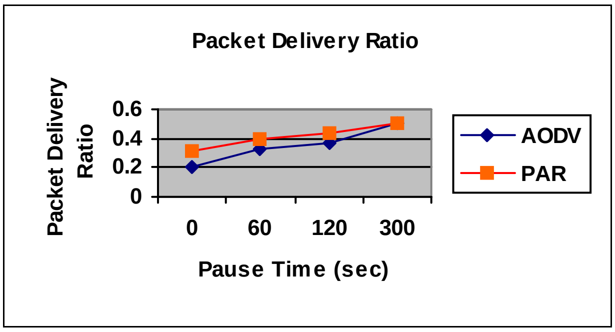

Figure 2 Comparison of Packet Delivery Ratio

The quality of service of the network is defined by the end-to-end delay. The average delay decreases at low mobility for both the protocols. The average delay is higher for AODV at high mobility as route failure occurs very frequently. The proposed algorithm maintains connectivity at all times leading to better performance. Figure 3 gives the comparison for the average end-to-end delay for AODV and Probabilistic Ant Routing.

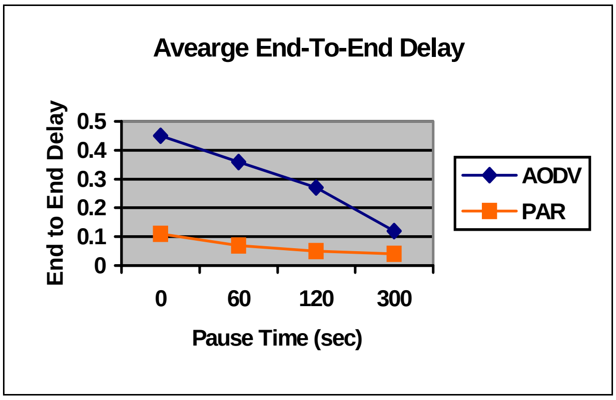

Figure 3 Comparison of Average End-to-End Delay

The routing overhead is shown in Figure 4 gives the number of control packets per data packet to perform routing. The routing overhead is more for the proposed algorithm. The performance of AODV is relatively stable at lower mobility leading to a decrease in routing overhead.

Since the proposed algorithm is stochastic in nature, so the number of times the best solution found is also analyzed. This indicates the consistency of the algorithm. The results are given in Figure 5. At high mobility, due to frequent link breakage the consistency of the good solutions is less. However, at low node mobility the results are encouraging.

Impact of varying node density

In the second scenario, the node density is varied for a fixed network parameter. Scalability in ad hoc networks can be broadly defined as whether the network is able to provide an acceptable level of service even in the presence of large number of nodes in the network. The results show that the packet delivery ratio increases with increasing network size. Table 1 summarizes the results.

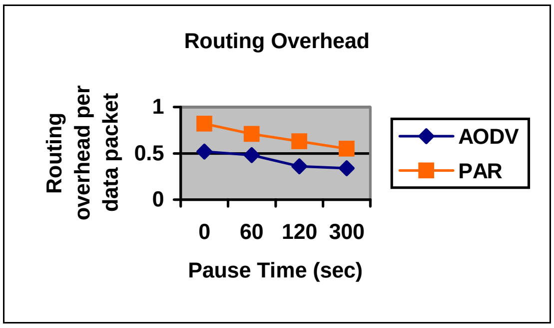

Figure 4 Comparison of Routing Overhead per Data Packet

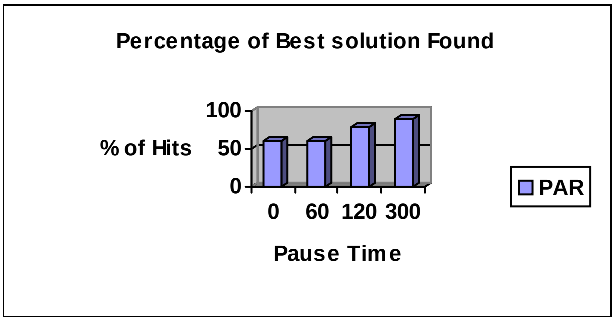

Figure 5 No. of Best Solutions for 10 runs for a fixed network setting

Impact of ACO Parameters

The selection pressure is provided by the equation that governs the decision of the ants' transition at each node by transition probability. The ACO parameter α and β determine the degree of the selection. If α is set to 0, then ACO corresponds to a classical greedy algorithm and if β is set to 0, then only amplification (reinforcement) is at work resulting in suboptimal solutions. We examined the effect of these two parameters on two networks, N1 consisting of 50 nodes and N2 consisting of 300 nodes. The best solutions found by both the networks were equal in quality. However, the larger network was found to be more sensitive to the parameter choices. In N2, the best and the average value for packet delivery ratio were 0.73 and 0.67 respectively for smaller values of the parameters. The fluctuation rate between the best and the average values was relatively higher. For higher values of the parameter ($\alpha > 2$ for fixed value of β), the results were more consistent for the network N1.

Table 2 gives a qualitative comparison of the performance between AODV and the proposed routing protocol. The results indicate the competitiveness of proactive probabilistic ant based routing algorithm with the conventional reactive AODV algorithm

CONCLUSIONS

The paper has proposed a routing algorithm based on ACO metaheuristic for mobile ad hoc networks that enhances the diversity in ACO by modifying the transition rule. When selecting the next hop to the destination, the decision whether the ant will be unicast or broadcast depends on the pheromone information. When compared with AODV, a popular reactive routing protocol for the mobile ad hoc network, the proposed algorithm showed good potential in producing better quality solution in presence of appreciable mobility in the networks. This was tested for different node mobility and node density through simulation results. The proposed algorithm was found to be consistent in producing good quality solutions. The routing overhead was high for small networks but it is actually low for large network scenario. This ensures the scalability of the network without affecting the performance of the network. The future work could be to investigate methods to further limit the overhead and compare the performance of the algorithm for different loads for other proactive routing algorithms. Hybrid extension of the proposed algorithm is currently under investigation.

REFERENCES

[1] S. Corson and J. Macker, MANET: Routing Protocol Performance Issues and Evaluation Considerations, RFC 2501, IETF Working Group, 1999.
[2] C. E. Perkins and P. Bhagwat, Highly Dynamic Destination Sequenced Distance Vector Routing (DSDV) for Mobile Computers, *Proc. of SIGCOMM '94 Conf. On Communications Architecture, Protocols and Applications*, pp 234-244, 1994.
[3] D. Johnson, D. Maltz and J.Jetcheva , The Dynamic Source Routing Protocl for Mobile Ad hoc networks, *Internet Draft, draft-ietf-manet-dsr-07.txt*, 2002.
[4] C.E. Perkins, E.M. Royer and S.R. Das, Ad Hoc On Demand Distance Vector (AODV) Routing, *Proc. IEEE Workshop on Mobile Computing Systems and Applications*, pp 90-100, 1999.
[5] E. M. Royer and C.K. Toh, A review of current routing protocols for ad hoc mobile wireless networks, *IEEE Personal Communication 6(2)*, pp 46-55,1999.
[6] M. Dorigo and T. Stützle, Ant Colony Optimization, MIT Press, 2004.
[7] D. Subramanian, P. Druschel and J. Chen, Ants and Reinforcement Learning: A Case Study in Routing in Dynamic Networks, *Proc. of IJCAI-97, International Joint Conference on Artificial Intelligence, Morgan Kaufmann,* 1997, pp 832-839,1997.
[8] H. Matsuo and K. Mori, Accelerated Ants Routing in Dynamic Networks, *2nd Intl. Conf. On Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed computing*, 2001.
[9] M. Günes, U. Sorges and I. Bouzazi, ARA- Ant Colony Based Routing Algorithm for MANETs, *Proc. of ICPP Workshop for Ad Hoc Networks, IEEE Computer Society Press*, pp 79-85, 2002.
[10] J. S. Baras and H. Mehta, A Probabilistic Emergent Routing Algorithm for Mobile Ad Hoc Network, *Proc. of Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, 2003.
[11] S. Marwaha, C. K. Tham and D. Srinivasan, Mobile Agents based Routing protocol for Mobile Ad Hoc

Networks, Proc. of Annual Workshop on Mobile Ad Hoc Network Computing, 2003.

[12] G. Di Caro, F. Ducatelle and L. M. Gambardella, AntHocNet: An Ant Based Hybrid Routing Algorithm for Mobile Ad Hoc Networks, *Proceedings of Parallel Problem Solving from Nature (PPSN VIII), LNCS Springer-Verlag* 2004.

[13] G. Di Caro and M. Dorigo, AntNet: Distributive Stigmertic Control for Communication Networks, Journal of Artificial Intelligence Research, vol 9, 371-365, 1998.

TABLE1
Performance Comparison of PDR for varying nodes

# of nodes	Packet Delivery Ratio	
	Probabilistic Ant Routing	AODV
50	0.30	0.24
100	0.48	0.35
200	0.60	0.41
300	0.72	0.50

TABLE 2
Performance Comparison of Routing Protocols

Performance parameters	AODV	Probabilistic Ant Routing
Packet delivery ratio	Low	High
End-to-End Delay	High	Low
Connectivity	Low	High
No. of paths for routing	Single	Multipath
Routing overhead	Low	High

```
/* Initialization */
Initialize the Routing Table at each node with equal probability
t = current time
Δt = time interval between ants generation
S_k = Φ
/* Main Loop */
For each node do  /* Concurrent activity at each node */
While t ≤ t_max do
   If (( t mod Δt) == 0) do
     Select destination node
     Launch forward Ant
   End If
   For each forward ant do
     While (current node ≠ destination node ) do
        If pheromone_info_available do
              j ← select next hop according to
                   probability given in (1)
         Else
              j ← broadcast the ants and select the
                  node which is first to reply back
       End If
       S_K ← S_k ∪ j
      End While
      Launch Backward Ant
      Pass the Stack Information and Die
   End For
   For each backward ant do
     While (current node ≠ destination node ) do
           Choose next hop by popping the stack
           Update the Traffic model
           Update the Routing Table
     End While
   End For
End while
End For
```

Figure 1. High Level Description of Probabilistic Ant Routing