A new adaptive routing approach based on Ant Colony Optimization (ACO) for Ad hoc Wireless Networks

Niaz Mosshed Chowdhury, Syed Murtoza Bake, Ershadul H. Choudhury
University of Dublin, Trinity College, East West University
chowdhun@cs.tcd.ie, galib@ewubd.edu, ershad@ewubd.edu

Abstract

This paper proposes the design of a new adaptive routing technique for ad hoc wireless networks. This algorithm works based on the principle of Ant Colony Optimization (ACO). This is a probabilistic adaptive technique that changes its routes with the change of network topology over the period of time by learning its environment. It identifies appropriate paths with the feedback of previously travelled packets and maintains routing table accordingly. A self-made simulator implemented on C++ is used to evaluate performance of this algorithm on the basis of diverse adaptive issues such as Change of Probability, Growth of Pheromone Intensity, Randomness of the Selection and Packet sending rate through different paths.

Keywords: Ant Colony Optimization, Ad hoc Network, Adaptive System, Routing Protocol

1. Introduction

Mobile ad hoc network (MANET) started to become popular during mid 90’s [1]. In between 1994 and 2004, hundreds of routing protocols came though most of them failed to survive due to their weak proactive nature [2, 3, 4]. During this period four Ad hoc routing protocols became extremely popular. In 1994 Charles E. Perkins proposed his first routing protocol Dynamic Destination Sequenced Distance Vector (DSDV) [5]. Two years later in 1996, David B. Johnson proposed his famous Ad hoc routing protocol Dynamic Source Routing (DSR) [9,10, 11]. In 2002 Niklas Beijar proposed another protocol, Zone Routing Protocol (ZRP) [12] though it maintains few restrictions and also lacks in its generalized nature. During this period Perkins was working in Nokea Research Centre and developing a new routing protocol and finally in July 2003 his most famous and most commonly used routing protocol Ad hoc On-demand Distance Vector (AODV) [6] routing protocol got accepted as Request For Comments (RFC) [7]. Since then DSR and AODV have been dominating ad hoc arena with promising success.

Though DSR and AODV are the most popular routing protocols of current time, they are not perfect yet. Biggest drawback of DSR is its packet overhead as it has to send large control information with the packet. On the other hand, AODV broadcasts huge control data (RREQ packet) to its neighbouring router while establishing a route which increases its bandwidth overhead. Since then a number of approaches have been tried to either improve their performances or implement parallel routing protocols.

2. Related study

Due to the NP complex nature of ad hoc routing problem, it is pretty hard to get a perfect solution or even a better solution without increasing the bandwidth or packet overhead. All those problems led the researchers to work with Adaptive protocols for ad hoc routing. In 2001 a routing protocol came named Adaptive Distance Vector Routing [14]. In the same year Adaptive Demand Driven Multicast [15] routing came. There was another algorithm named Fuzzy Sighted Link State Algorithm [15]. In July 2006 M. Roberts Masillamani proposed a GA based routing protocol for conventional distance vector routing and they are currently working to convert this protocol to a GA based adaptive AODV [16].

3. Ant Colony Optimization

Ant Colony Optimization (ACO) [17, 18] is an adaptive technique to find partial solutions for the problems where identifying exact solution is either difficult or impossible (19, 20). ACO is proposed by Dr. Marco Dorigo in 1992 and inspired by the behaviour of ants in finding paths from the colony to food. In 1997, Dr. Marco Dorigo in his work Ant Colony System: A comparative learning approach to the Travelling Salesman Problem [21] hinted that network routing can be implemented based on Ant
Algorithm. Very next year he published another research work [22] where he discussed that packets and ants are typically identical for an adaptive system and routing algorithm can be formed or improved based on ant algorithm. In 1999, in another paper, Ant Algorithm for Discrete Optimization [31] he proposed the same idea. Despite the fact that he always recommended his algorithm to be used in computer networks, neither he nor other researchers showed enough interest in this regard.

4. Design
ACO is a generalized adaptive technique [27, 28, 29, 30] like Fuzzy Logic [25, 26], Genetic Algorithm [16] or Simulated Annealing [24]. It has different phases such as Initialization, Probability Calculation, Selection, Pheromone Deposition, Pheromone Evaporation etc. which are problem dependent [31]. Each of those phases needs to be designed based on the problem demand and nature. Rest of the part of this section will talk about the design issue of those phases for the proposed routing technique.

4.1 Initialization
ACO is a meta-heuristic method that requires to be initialized at the beginning of the procedure [23]. Conventionally in ACO, all the ant tracks (here routing paths) are initialized with numerical value zero because it is assumed that initially there is no pheromone on the tracks. But proposed algorithm initializes its paths in a different way; with numerical value of one instead of zero. The rational behind this modification is quite straightforward – in this algorithm a non-zero value of intensity represents a connection which means pheromone intensity also indicates connectivity of nodes and paths having zero intensity indicate there is no connection between two particular nodes.

**Figure 1:** A network topology with a stable pheromone intensity of the paths.

4.2 Probability Calculation
Probability calculation is one of the most important issues for any adaptive algorithm and success of the proposed technique is highly influenced by the proper calculation of the path probability as while sending a packet from one node to another node, which path should get selected is determined by this probability. Proposed technique calculates this parameter based on the pheromone intensity of different paths. Probability of a path is defined as,

$$ P_i = \frac{I_i}{I_t} $$

**Equation 1:** Path probability

Here,

- $P_i$ is the probability of a particular path $i$ connected with the respective node.
- $I_i$ is the intensity of a particular path $i$ connected with the respective node.
- $I_t$ is the summation of intensity of all paths connected with the respective node.

4.3 Selection
Selection is done by single spin roulette wheel method. Space allocation for each path on the wheel is proportional to its probability. Say for example, there are three paths A, B, and C connected with a router having probability $PA=0.50$, $PB=0.25$ and $PC=0.25$. Arrangement on the wheel for this particular case is shown in Figure 3.

**Figure 2:** Changed topology when a new router G joined the network.

In Figure 1, a stable network topology has been shown where pheromone intensities of the paths are $I_{AB}$, $I_{CD}$, $I_{DB}$, $I_{DF}$ and $I_{GE}$. Let us assume that a new node G has just joined the network (shown in Figure 2), hence creates three new paths GB, GF and GE. Conventional approach of ACO would initialize these three paths with zero intensity as those new paths have not been travelled by any ant (packet) yet. But, as per our proposed algorithm initialization will be done with one in order to indicate that there is a path available between B and G, F and G and E and G.
Figure 3: Roulette Wheel arrangement for PA=0.50, PB=0.25 and PC=0.25.

Another important issue in selection is generation of pseudo random number as weakness of random number generation may create initial fluctuation problem [31]. In real life, most of the built-in random number generator functions return close value if they have been called with in a short duration of time (such a 1 millisecond) as they take current time in second as seeds. A method for well distributed random number generator has been proposed with this algorithm. A C++ code segment has been given bellow. This function takes a value as a parameter and returns a well distributed random number in between zero and value.

Random Number Generator Function

```cpp
int random(int highest) {
    srand( (unsigned)time(0));
    int random_integer;
    int lowest=0;
    int range=(highest-lowest)+1;
    random_integer = lowest + int(range*rand()/(RAND_MAX + 1.0));
    return random_integer;
}
```

4.4 Pheromone Deposition

In this proposed algorithm, pheromone deposition is done by the backward packets. After a successful delivery, each packet increases the intensity of each path that it has travelled to deliver the packet based on the following formula.

\[ I_{\text{new}} = I_{\text{old}} + \frac{Q}{\text{Length Ratio}} \]

**Equation 2:** Pheromone Deposition Formula

Here, \( Q \) is a unit quantity of pheromone to be deposited. But, all the paths wont get the full portion of \( Q \) rather deposition of \( Q \) will be inversely proportional to the travelled Path.

Here Length Ratio varies from node to node. For each node on the path it is measured as the path length between the source and the calculated node.

In this technique, backward packet will deposit more pheromone on the shorter path compare to the longer path while coming back from the destination. As the probability calculation is dependent of the pheromone density which ultimately increases the chance of best path to be selected.

A pictorial example is shown here to describe the above literature. Consider the following network topology shown in Figure 4 and 5. In Figure 4, it shows that A has made a successful delivery of a packet to C. During this travel, this packet marked the paths that it has travelled.

![Figure 4: A is sending a packet to C.](image)

Let us consider \( Q = 10 \). Based on the **Equation 2** pheromone of each track will be updated as follows:

\[ I_{AB}^{(\text{new})} = 10 + \frac{10}{3} = 13 \]
\[ I_{DB}^{(\text{new})} = 87 + \frac{10}{2} = 92 \]
\[ I_{CD}^{(\text{new})} = 20 + \frac{10}{1} = 30 \]

Figure 5 shows backward packet sent by C to A. This backward packet will increase the intensity of the paths (underlined) that it has travelled while coming from A to C.

4.5 Pheromone Evaporation

![Figure 5: Intensity of the paths is being changed by the backward packet.](image)
Pheromone evaporation is another issue in ACO. It decreases pheromone from less travel paths. In this algorithm pheromone evaporation is done based on a parameter called evaporation parameter, \( \lambda \). Value of \( \lambda \) depends on the traffic density of the network. If Intensity of any path at a junction crosses the value of \( \lambda \), intensity for all the paths of that junction will be changed as follows:

\[
I_{i\text{(new)}} = \frac{I_{i\text{(old)}}}{\gcd(I_{1\text{(old)}}, I_{2\text{(old)}}, \ldots, I_{k\text{(old)}})}
\]

Equation 3: Pheromone Evaporation Formula

Here,

- \( I_{i\text{(new)}} \) is the Intensity of \( i \) path after the evaporation.
- \( I_{i\text{(old)}} \) is the Intensity of \( i \) path before the evaporation.
- \( k \) is the number of connected paths with that router.

5. Operation

A routing technique has to perform three basic operations, namely Route Establish, Route Discovery and Route Maintenance [13]. Nature and performance of a routing algorithm is determined on how does it deal with these three tasks. Followings are the brief operational summary of the proposed technique.

5.1 Route establish

Route establishment is a prime and foremost task for any routing procedure. In our proposed technique, at the beginning of the network or when a new node enters into the network, it finds its neighbouring node and initializes local intensity value of the corresponding paths with unit value (i.e. 1). Initially it also assumes that all the nodes of the network are reachable through any of those paths. It maintains a vector table like distance vector routing where it puts the intensity of different paths. This table is arranged in the conventional form of “from A to B via C” [13]. But instead of distance value, it holds intensity of a particular link. As time passes, gradually best outgoing link for a particular node becomes stable in the intensity table.

5.2 Route discovery

This routing technique does not send any route request packet like AODV or any service discovery like DSR. Still it maintains a reactive on demand nature. Before sending any packet to a particular destination, it finds the path based on a probabilistic selection mentioned earlier in section 3. Path having been travelled most to reach a destination will get higher priority in the selection. This route discovery is an adaptive method based on Ant Colony Optimization and it becomes stable eventually by learning the environment.

5.3 Route maintenance

Proposed technique maintains its route in an adaptive manner. When a node gets down from the network, no packet can successfully be delivered through the path that goes over that particular node. As a result pheromones of that path will be evaporated and other paths will start to get the priority in selection. If the down node gets back to the network then this path will be established again otherwise due to periodical pheromone evaporation this path will become a dead link.

6. Result and Evaluation

A self made simulator implemented on C++ is used to evaluate performance and behaviour of the proposed technique. This simulation mainly concentrates in measuring and evaluating adaptive natures of the algorithm. A 50 node scenario is considered taking two nodes as source and destination where three different and complex paths are available to reach the destination from the source node. Among the paths, Path 2 is the best one while Path 3 and Path 1 are the second and third best respectively. We measured following four issues to evaluate our proposed technique:

1. Change of Probability
2. Growth of Pheromone Intensity
3. Randomness of the Selection
4. Packet Sending Performance

6.1 Change of Probability

Change of probability is measured over the period of 900 packet send time from the source node to the destination. Initially when the very first packet was sent, the probability of each path was equal, i.e. 0.33. As time increased, gradually best path was established and the worst two got almost down. Figure 8.1 shows change of probability of different paths within an interval of 100 packet send period.

6.2 Growth of Pheromone Intensity

In this part of the evaluation once again 900 packets were sent and within an interval of each 100 packet send time measurement was taken. Initially pheromone intensity of all three paths was 1. Figure 8.2 shows change of intensity due to pheromone deposition on the experimental
paths. Result clearly shows that best path got more pheromone deposition and its intensity increased. Intensity of other two paths was increasing with an extremely slow rate.

6.3 Randomness of the Selection
Randomness of the selection has been measured by this experiment maintaining the same network topology over five different runs. Figure 8.3 shows the result where it is clearly noticeable that the variation is very marginal in different runs.

6.4 Packet Sending Performance
In this final evaluation total number of packet sent through each path is measured for the same topology over the period of 100 packet sent time. Figure 8.4 shows, as time increased, Path 2 (best path) got selected more frequently for sending packets from the source node to the destination.

7. Conclusion and Future Work
An ant navigation based adaptive routing technique is proposed and evaluated in this paper and we are confident that it will work effectively in the real ad hoc network scenario. This work mainly concentrates on designing different adaptive parameters of the technique rather than concentrating in the network layer issues. Our future objective is to implement this algorithm using OPNET [32] for mobile ad hoc network and compare its performances with AODV and DSR.

Reference: