6.1. Introduction

During recent years, new algorithms have been introduced to tackle the combinatorial optimization problems. Some examples of these algorithms are Genetic Algorithms, studied by Hertz et al (1991) [149] and Ant Systems, studied by Dorigo et al (1996) [150]. Behaviour or processes present in nature inspire those algorithms. Such mechanisms have proved to be extremely effective and evolutionary. These algorithms demonstrate adaptive, robust and effective behaviour as nature does; where adaptive means that it improves its goal-achieving competence over time, robust means it is flexible and never breaks down completely and effective means it eventually finds a satisfactory solution.

In this chapter, a complete unique design for MANET routing algorithm called Ant based Routing Algorithm (ARA) has been proposed. ARA is stimulated by the ACO heuristics by combining the ability of reactive routing (AODV) to resolve complicated issues through co-operation while not direct communication. In particular, MANETs have been considered where the network nodes are able to change their position and the communication between the networks nodes have been established over a wireless medium. Also, homogenous networks have been considered with no additional infrastructure. There is no difference between the nodes. This has the following consequences among other things:

- Nodes will leave and be a part of the network at any time,
- There is no centralized control or overview,
- Packets have to be forwarded from node to node.

Routing in MANETs could be the challenge because of the very fact that a good quality path will suddenly become inefficient or maybe an infeasible one. So as to
succeed, a routing algorithm for such an environment has to be accommodative and ready to take care of sharp changes in the topology of the network. These properties can also be found in nature. An insect population shows a robust and efficient way to adapt to the changing environment. This fact inspired to design a routing algorithm based on simple biological agents called ants.


In this section, the overview of ARA has been described. This algorithm uses two types of ant namely forward ants (FANT) and backward ants (BANT). The following reactive approach has been started first in the proposed algorithm. If a node needs to establish a connection between nodes it sends out a FANT that walks randomly in the network. In this algorithm, random walk signifies that on the each node FANT chooses one of its neighbours with equal probability. While moving from one node to following the FANT leaves entries within the routing table that purpose towards the source.

It walks around randomly in the network until it reaches the target it was sent out for. Once it reaches the target, a BANT is generated, which follows the routing table entries back to the source. On its way, the BANT itself leaves entries in the routing table again for its source. The proposed algorithm is fabricated in three phases like ant colony based routing algorithm (ARA); they are route discovery phase, route maintenance, and route failure handling.

6.3. Pheromone table structure

The pheromone table is similar to the routing table in classical routing, except that for the pheromone table is stored information corresponding to only one hop neighbours. This table is updated using forward and backward ants.

Whenever the forward ant visits an intermediate node for the first time an entry is created containing:

- The identifier of the source node that has initiated the route discovery.
- The identifier of the next node used to reach the destination node.
The initial value of the pheromone, which is the same for all newly created entries.

The pheromone table is updated using two ways:

- The first way using forward and backward ants, since at each visit of these ants, the pheromone are increased with a fixed amount such as in real systems.
- Periodically, the value of pheromone is decreased according to time in order to emulate the biological pheromone which loses concentration due to time.

6.4. **Analysis of ARA**

As summarized on top of, ARA offers the only application of swarm intelligence for routing in ad hoc networks as a result of it applies directly the thoughts of ant colony to ad hoc networks while not taking into consideration all the characteristics of ad hoc networks like energy constraints, bandwidth, and mobility, so it is able to differentiate the subsequent shortcomings:

- In ARA, the pheromone table is updated at regular intervals or whenever a forward or backward ants visit nodes over the networks that isn’t adequate since the pheromone should be updated in line with alternative parameters like nodes mobility, battery power yet because the satiate of links between neighbours that have an effect on the routing in ad hoc networks.
- In ARA, the traffic is forwarded according to the concentration of pheromone over edges, therefore the same path may be used by several connections, which consumes the resources of intermediate nodes in this path, so a mechanism must be defined to avoid similar situations by distributing traffic according to the number of connections using the same route and choosing new paths whenever the number of connections over the same path reaches its maximum.
6.4.1 Route discovery phase

Similar to ARA, the route discovery mechanism is intended to find routes over the network, as well as updating pheromone table such as in swarm intelligence based routing. To accomplish the discovery and establishment of routes over the network, two classes of ants are defined which are forward and backward ants.

✔ Forward ANT (FANT):

Forward ants are intended to discover routes; it is launched by the source nodes and broadcasted over the entire network until it arrives to the destination node. During its trip over the network, the FANT causes pheromone update because the reception of FANT is the event which launches all kinds of pheromone update. The structure of the forward ant can be described as follow (Table 6.1), where:

- **Packet Type**: This field is 1 byte size; its value describes the purpose of the packet, data, FANT or BANT, in this case, it is fixed to FANT.
- **Source IP Address**: In this field have 4 bytes and describes the IP address of the source node.
- **Destination IP Address**: In this field have 4 bytes and describes the IP address of the destination node.
- **IP list**: This field is an array of four bytes and contains the list of IP addresses followed by the FANT during its broadcasting over the network.
- **Pheromone list**: This field is an array of four bytes and contains the amount of pheromone carried by each link traversed by the FANT.
- **Sequence number**: this field is four bytes and contains a unique sequence number used to avoid route loops, similar to DSR.
- **Time-To-Live (TTL)**: This field is one byte and describes the remaining allowed hop count for the FANT. It is fixed to 255 and decremented at each visited node.
Table 6.1: Forward ANT (FANT) packet structure

- **Packet Type:** Forward ANT (FANT).
- **Source IP:** The address of the source node.
- **Sequence Number:**
- **TTL:**
- **Destination IP:**
- **IP List:**
- **Pheromone List:**

Table 6.2: Backward ANT (BANT) packet structure

- **Packet Type:** Backward ANT (BANT).
- **Source IP Address:** The address of the destination node.
- **Destination IP Address:** The address of the destination node.
- **Reversed IP list:** This field is an array of four bytes and contains the reversed list of the list retrieved from the FANT.
- **Time-To-Live (TTL):** is fixed to the length of the reversed list.
In this phase, new routes are formed. The form of new routes makes utilize of FANTs and BANTs. A FANT message is broadcasted by the sender node and transmitted by the neighbour’s node of the sender. A node getting a FANT message for the first time generates a record in its routing table as shown in figure 6.1.

<table>
<thead>
<tr>
<th>Next hop Destination</th>
<th>Next hop1</th>
<th>Next hop2</th>
<th>. . .</th>
<th>Next hop N</th>
</tr>
</thead>
<tbody>
<tr>
<td>d1</td>
<td>P11</td>
<td>P12</td>
<td>• • •</td>
<td>P1N</td>
</tr>
<tr>
<td>d2</td>
<td>P21</td>
<td>P22</td>
<td>• • •</td>
<td>P2N</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
<td>•</td>
<td>.</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
<td>•</td>
<td>.</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
<td>•</td>
<td>.</td>
</tr>
<tr>
<td>dL</td>
<td>PL1</td>
<td>PL2</td>
<td>• • •</td>
<td>PLN</td>
</tr>
</tbody>
</table>

Figure 6.1: Probabilistic routing tables

When the FANT message reaches the destination node it extracts the data collected by the FANT, destroys it and subsequently creates a BANT that follows the track of the FANT however within the reverse direction. The working of this phase is explained in the flowchart as shown in figure 6.2.
Source node creates FANT and sends to neighbor nodes

BANT received by source node within timeout? per period?

NO

Generate FANT with new sequence number

YES

Send data packet along the path

Increase the hop count value by one and update pheromone value

NO

BANT received by source node within timeout? per period?

YES

Send FANT to nodes in neighbor list

Set type of control packet to BANT with same sequence number as FANT

Reserve resource at current node

Send BANT to node from which it has received FANT

Reserve resource at current node

If current node = destination node

YES

If current node is not source node

NO

BANT reaches source node

When source node receives BANT from the destination node, the path is established and data packets can be sent

Reserve resource at current node

Send BANT to node from which it has received FANT

Figure 6.2: Route discovery phase of ARA

6.4.2 Route Maintenance

In this phase is similar to the original ARA and no changes are made. Hence, whenever a link between two neighbours fails enclosed within the routing process, a route error packet is transferred to the source node that initiates a new route discovery
as summarized above supposed to create a new path between the subsequent nodes. The route error packet structure is similar to ARA:

- *Error Source Address:* The address of the node initiates the route error (node has discovered the link failure).
- *Error Destination Address:* The address of the node to that the route error should be delivered
- *Error Type field:* Always to set NODE_UNREACHABLE.

The route maintenance phase is to blame for the development of routes throughout the communication. Once the FANT and also the BANT have created the pheromone tracks for the source and destination nodes; consequent information packets are used to maintain the path. Whenever a node relays a packet through a link the pheromone concentration is strengthened. When the BANT visits a node it provides a positive reinforcement by incrementing the probability of choosing the particular link as the next hop and also provides negative reinforcement by decrementing the probabilities of the other links. Thus, the best route is not congested and the traffic is sent through the backup routes. The working of this phase is explained in the flowchart as shown in figure 6.3.

### 6.4.3 Route Failure Handling

A heuristic approach is used for handling failures. This is aggravated through node mobility. Missing ACK recognizes route failure. To handle such failures the link should be deactivated and pheromone content is set to zero. The node that detected the link failure will search for an alternate route in its cache. Else, it backtracks and checks with the predecessor. The worst case is the source re-initiating the route discovery process. The working of this phase is explained in the flowchart as shown in figure 6.4.
Current node receives the data packets

If current node = destination node

NO

Extract data

Send acknowledge packet to the previous node

Decrement pheromone value by $\alpha$

If pheromone value $= 0$ for any destination node

YES

Call route discovery phase

NO

Get pheromone value of all links using neighbor list

Compute probability for all nodes in neighbor list

Send packets to that link which has highest probability value

Increment pheromone value for the highest probability link

If acknowledge is not received before timeout

NO

Send route error to the previous node

YES

Refresh route

Figure 6.3: Route maintenance of ARA
6.5 Proposed Solution

In this previous section, existing algorithm detect the black hole attack. ARA and AODV are evaluated by so many authors and identified ARA is always better than AODV. In this section, the proposed AODV is modified to identify and avoid black hole attack by using ant colony algorithm such as ARA. Pheromone updates play a significant role in the performance of the ant algorithm. In ARA algorithm, initial pheromone value is calculated by number nodes during the route discovery process. The working principles of the algorithm are given below:

1. Establish a network with N number of nodes.
2. Specify the properties of the network.
3. Define the source and also the destination node over the network.
4. Place the ant at every node within the network.
5. Define “m” is a malicious nodes over the network.

Figure 6.4: Route failures handling of ARA
6. Route discovery process: Source node broadcast the RREQ message to neighboring nodes using FANT forward technique and hop count is initialized. It is an agent to establish pheromone value to the source node.

7. Collecting replies:
   i. Collecting the neighboring nodes information stored in the routing table.
   ii. Neighboring nodes receive the request then it will check whether the node is the destination node or not.
       If yes then
       FANT is sent to only that neighbor
       else
       it’s forwarded to all the neighbors.

A node is receiving a FANT for the first time, can produce a record in its routing table and fields like destination address, next hop, and pheromone value.

8. For each FANT (currently in node i)
   Do
   – Choose the neighbor node, probability value will be high that route/neighbor needs to be considered.
   – Add that node pheromone value to neighboring pheromone table with the node, pheromone value between these nodes until the ant has reached the destination.
   End

9. The full process is mention above to get repeated until the Forward Ant (FANT) reaches the destination node.

10. Once FANT destroys its reaches to the destination node and create BANT send to on the path to the source node. It’s an agent that establishes the pheromone value to the destination node.

11. Route maintenance: Once FANT and BANT have established route path between source to destinations and data packets are send along the same path. The pheromone track value is strengthened means path is the shortest path between these two nodes.
Assumptions

The complete methodology is based on the following assumption to evaluate the network performance with and without the effect of the malicious node at distributed levels.

- Malicious node does not acknowledge with the data packet in the network.
- Black hole node will receive the packet but instead of forwarding the packet it will drop all the received to lower the packet delivery ratio and network efficiency.

6.6 Results and Analysis

The proposed methodology is compared with the existing algorithm of safe route method based upon the ant colony based routing algorithm on the basis of throughput, packet delivery ratio, end-to-end delay and so on. The performance and results of the routing algorithm as below:

✓ Throughput:

The throughput is the number of bytes transmitted or received per second. The throughput is denoted by T,

\[ T = \frac{\sum_{i=1}^{R} N_{r}^{i}}{\sum_{i=1}^{R} N_{s}^{i}} \times 100\% \]  

Where, \(N_{r}^{i}\) = average receiving node for the \(i^{th}\) application, \(N_{s}^{i}\) = average sending node for the \(i^{th}\) application, and \(n = \) number of applications. In figure 6.5 shows that the proposed algorithm improved good throughput compared to AODV with black hole attack.
Packet Delivery Ratio:

It can be measured as the ratio of the received packets by the destination nodes to the packets sent by the source node.

\[
PDR = \frac{\text{number of received packets}}{\text{number of sent packets}} \times 100
\]

\[
PDR = \frac{\sum_{i=1}^{n} (N_i^s - N_i^f)}{\sum_{i=1}^{n} N_i^s} \times 100\% \tag{6.2}
\]

Where, \(N_i^s\), \(N_i^f\) node sent by the sender and the number of application data node received by the receiver, respectively for the \(i^{th}\) application, and \(n\) is the number of applications. In figure 6.6 shows that packet delivery ratio of the proposed algorithm is more than AODV routing algorithm with black hole attacks. Black hole stimulates packet dropping, the original AODV decreases packet delivery ratio with an increase in a number of nodes.
End-to-End Delay:

It represents the time required to move the packet from the source node to the destination node.

\[
E-2-E \text{ delay } [\text{packet}_i] = \text{received time } [\text{packet}_i] - \text{sent time } [\text{packet}_i]
\]

\[
E2E = \frac{\sum_{i=1}^{n} d_i}{n}
\]  \hspace{1cm} (6.3)

The average end-to-end delay can be calculated by summing the times taken by all received packets divided by its total numbers. Where, \( d_i \) = average end to end delay of node of \( i^{th} \) application and \( n \) = number of application. In figure 6.7 shows that the proposed algorithm provided minimum end-to-end delay compared with original AODV with black hole attack.
Figure 6.7: End-to-end delay vs number of nodes

✓ Dropped Packets:
It represents the number of packets that sent by the source node and fail to reach the destination node.

\[
\text{Dropped packets} = \text{sent packets} - \text{received packets}
\]

\[
T = \sum_{i=1}^{n} (N_r^i - N_f^i) - \sum_{i=1}^{n} N_s^i
\]

(6.4)

Where, \(N_r^i\), \(N_f^i\) node sent by the sender and the number of application data node received by the receiver, respectively for the \(i^{th}\) application, and \(n\) is the number of applications. In this proposed system, get better performance to deliver the data packets. It easy to analysis packet dropped rate in the routing process.

6.7 Summary
In this section, the paper summarized for a study about mobile ad hoc networks; to initiate that most repeated attack is a black hole in MANETs. To discover a resolution for that various algorithms is available. But to decide security and performance issues some improvements on the routing technique is implemented. In this chapter, it has analyzed the effects of black hole attack in the light of network load, throughput and end-to-end delay in MANETs and simulating the black hole attack using reactive routing protocols (e.g. AODV). Compared and determined that AODV while not attack offers better result in all situations. When observing the results it’s
found that under attack case system has a lot of packet drop ratio it’s forever greater to the threshold. Design and implement a security algorithm for detection of black hole attack supported on Ad hoc On-demand Distance Vector routing protocol and Ant Colony Optimization algorithm.

Implementation of proposed method is quite efficient for network and able to detect the attack. In addition, the performance of the network is improved effectively. The summary of performance is packet delivery ratio, end-to-end delay and throughput can be improved. The proposed protocol can able to improve two main problems such as security and performance, into one place, but this concept is able to detect only one attack and effective for black hole. In future a framework for security is required, where more than one attack is handled.