ABSTRACT

Swarm intelligence is the collective and organized behavior of decentralized systems, natural or artificial. Ant colony. ACO methods find its applications in problems that need to find paths to goals. In this paper, we propose an algorithm that can improve performance of ACO. The proposed algorithm consists of two agents: the data agent and the routing agent where the routing agent is responsible to prepare the routing table from the given source to destination and the data agent transfers the data along that path. The earlier algorithms do this task with a common agent which also carries the data. This increases the frame size of the packet and hence the time to process the packet at each node for extracting routing information.

Keywords: Swarm intelligence, MANETs, Routing, Ant Colony Optimization, AntNet, Ant Based Control

1. INTRODUCTION

Many social insect societies like ants, fishes, bees and birds, in spite of their simplicity as individuals, depict a complex and often intelligent social behavior [1]. They form distributed decentralized autonomous systems which can accomplish tasks that in some cases far exceed the capabilities of an individual insect. Thus, Swarm intelligence (SI) is artificial intelligence based on the collective behaviour of decentralized, self-organized systems which allows them to coordinate the activities and engage in collective problem-solving and decision-making.

Mobile Ad Hoc Networks [25] are networks which are composed of numerous mobile nodes which can connect to a network or disconnect from the network anytime. These mobile nodes basically make wireless connections to connect to the network. Routing is the task of forwarding the data packets from a source node to a required destination. This task of forwarding is particular hard in Mobile Ad Hoc Networks as the nodes are not fixed. Thus, algorithms used to manage such network should be adaptive and self organizing.

Swarm intelligence aims at simplifying the behavior of this routing task in mobile ad hoc networks by taking inspiration from natural patterns depicted by insect societies like ants etc.

2. PRINCIPLE OF SWARM INTELLIGENCE

The main principle which is used in these interactions is known as stigmergy or communicating using the environment. Ant lays down pheromone on trails followed by it[26]. Pheromone is a potent hormone which can be sensed by ants as they travel along trails. It attracts ants and therefore ants tend to follow trails that have high pheromone concentrations. This causes an autocatalytic reaction, i.e., one that is accelerated by it [1][5]. Ants attracted by the pheromone will lay more of the same on the same trail, causing even more ants to be attracted.
3. SWARM INTELLIGENCE IN MANETS

Routing is the task of forwarding the data packets from a source node to a required destination. Nowadays as the networks are becoming more complicated, algorithms having self organizing and self configuring capabilities are desired. This enables them to adapt to new situations in terms of traffic, services and many more. MANET is one such complex network as discussed in [1]. Mobile Ad Hoc Networks are networks in which all nodes are mobile and communicate with each other via wireless connections and other communication networks as discussed in [2]. Nodes can join or leave at any time. All nodes are equal and there is no centralized control or overview. There are many algorithms like Antnet, Ant based Control that are used for routing.

3.1 TYPES OF MANET ALGORITHMS

MANET algorithms are divided into three types:

1. **Table-driven are purely proactive**: All nodes try to maintain routes to all other nodes at all times[23]. This means that they need to keep track of all topology changes, which can be difficult if there are a lot of nodes or if they are very mobile.

2. **Demand-driven algorithms are purely reactive**: Nodes only gather routing information when a data session to a new destination starts, or when a route which is in use fails[27].

3. **Hybrid**: Combination of proactive and reactive algorithm [7].

3.2 PROCESS OF ROUTING USING SWARM INTELLIGENCE

In routing algorithms, routing information is gathered using a stigmergic process using various agents like forward and backward agent. An ant(forward ant) going from its source s to a destination d collects information about the quality of the path it follows (e.g. end-to-end delay), and, retracing its way back from d to s, uses this to update the routing information(backward ant) at intermediate nodes as discussed in [10].

Routing information is expressed in the form of tables. Each node has its own routing table/Pheromone Table T. Table $T_i$ at node i is a matrix, where each entry $T_{i,j} \in R$ of the table is a value indicating the estimated goodness/Pheromone value of going from i over neighbor j to reach destination d.

These table entries play the role of stigmergic variables in the learning process: an ant agent sampling a path to its destination d makes a stochastic routing decision at each node, giving higher probability to decisions with high goodness, while an ant retracing its way back from d to s updates the table entries influencing the routing of other ants[24].

Data packets are routed more or less in the same way as ant packets are routed stochastically, choosing with a higher probability those links associated with higher pheromone values as discussed in [6].

3.3 DATA-DRIVEN PATH SET UP

When a source node s starts a communication session with a destination node d, and it does not have routing information for d available, it broadcasts a reactive forward ant[22]. If information is available, the ant chooses its next hop n with the probability $P_{n,d}$ which depends on the relative goodness of n as a next hop, expressed in the pheromone variable:
Each forward ant keeps a list $P = [1; 2; \ldots; d]$ of the nodes it has visited. Upon arrival at the destination $d$, it is converted into a backward ant, which travels back to the source retracing $P$. Once the backward ant makes it back to the source, a full path is set up and the source can start sending data. If the backward ant does not arrive for some reason, a timer runs out at the source, and the whole process is started again.

### 3.4 TABLE-DRIVEN PATH SET UP

Proactive forward ants are used to update the information about currently used paths and to try to find new and better path [8].

The pheromone diffusion is implemented using short messages, broadcast periodically and asynchronously by the nodes to all their neighbors. In these messages, the sending node $n$ places a list of destinations it has information about, including for each of these destinations $d$ the best pheromone value $T^m_{n,d}$: $m \in N^d$, which $n$ has available for $d$[11].

A node $i$ receiving the message from $n$ first of all registers that $n$ is its neighbor. Then, for each destination $d$ listed in the message, it derives an estimate of the goodness of going from $i$ to $d$ over $n$, combining the cost of hopping from $i$ to $n$ with the reported pheromone value $T^m_{n,d}$ [28].

### 3.5 LINK FAILURE

When a node $i$ discovers the disappearance of a neighbor $n$, it takes a number of actions:

1. In the first place, $i$ registers that $n$ is no longer a neighbor, and removes all associated entries from its pheromone table [13].

2. Next, $i$ broadcasts a link failure notification message. This message contains a list of the destinations to which $i$ lost its best path, and the new best pheromone to this destination (if it still has entries for the destination).

### 4. ALGORITHM 1: ANTNET

In the AntNet algorithm, routing is determined by means of very complex interactions of forward and backward network exploration agents ("ants") [16]. The idea behind this sub-division of agents is to allow the backward ants to utilize the useful information gathered by the forward ants on their trip from source to destination.

Based on this principle, no node routing updates are performed by the forward ants. Their only purpose is to carry data and report network delay conditions to the backward ants, in the form of trip times between each network node. The backward ants inherit this raw data and use it to update the routing table of the nodes as discussed in [4].

### 5. ALGORITHM 2: ANT BASED CONTROL

This algorithm shares many key features with AntNet, but has important differences.

#### 5.1 SIMILARITY

1. The basic principle shared is the use of a multitude of agents interacting using stigmergy [19]. The algorithm is adaptive and exhibits robustness under various network conditions.

2. The routing table of every node is the same as in AntNet algorithm.

#### 5.2 DISSIMILARITY

1. The update philosophy of the routing table is slightly different though[20]. There is only one class of ants, which is launched from the sources to various destinations at regular time intervals. The ants perform the table updating simultaneously.

So, although network traffic is less the delay increases as the forward ant which carries the data has to update the tables also.

### 6. PROPOSED ALGORITHM: TWO PROCESS BASED ROUTING

We propose a two process based policy where we introduced two agents.

**DATA AGENT:** Carries the data.

**ROUTING AGENT:** Consists of forward and backward. These ants controls and guides the data agents. It ensures better performance[12]. Thus reducing congestion in network and improving network bandwidth utilization.

#### 6.1 INFORMATION BOOTSTRAPPING

**MONTE CARLO:** Ants try out complete paths from source to destination. The estimated goodness values for routing decisions recorded are the result of the direct experiences of the ants [17].

**BOOTSTRAPPING:** Nodes estimate the cost-to-go of a path by combining the cost estimates made by neighboring nodes and the cost to go to those neighboring nodes, rather than by the direct sampling of a full path [18].
We attempt to combine the best of both learning methods.

6.2 ALGORITHM AND FLOWCHART

If(routing agent)
{
  if (forward ant)
  {
    1. Get the next node based on the current node value
    2. if (the link is available and no loop caused) then
       {  
         2.1 Update forward ant with network status(stack)
         2.2 Send forward ant to the next node
       }
    3. else if (no such link exist)
       {  
         1.1 Create backward ant and load contents of forward ant to backward ant.
         1.2 Send backward ant towards source along the same path as forward ant.
       }
  }
  //end-if for forward ant
  if (backward ant)
  {
    1. if (current node is source node)
       {  
         1.1 Store path and kill backward ant
         1.2 Update routing table
       }
    2. else
       {  
         2.1 Forward backward ant on to link available on backward ant stack.
         2.2 Update routing table.
       }
    3. if (next node is not available)
       {  
         3.1 Kill backward ant
       }
   //end-if for backward ant
  } //end-if for routing agent

If(data agent)
{
  forward the data packet to the destination using routing table
}

Fig 2: Flowchart of two agent based routing
7. CONCLUSION

Current kind of network scenarios demands complex routing techniques like in MANETs etc.

Table1. Comparison of different Ant Colony optimization Algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Carry data</th>
<th>Calculate network parameters</th>
<th>Update routing table</th>
</tr>
</thead>
<tbody>
<tr>
<td>Antnet</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Forward Ant</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Backward Ant</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ant Based Control</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Forward Ant</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Two Process Based</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Routi ng Agent</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Forward Ant</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Backward Ant</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Agent</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

With these kind of features, two process based routing technique is the best routing technique amongst the other two, as it has separate agent to prepare the routing table with the help of forward and backward ant, and a separate agent to carry the data[21]. When the entry in the routing table already exists, data process will simply carry the data to the destination without making the use of routing process. This therefore reduces the network congestion and so helps in fast routing. Therefore, two process based routing is most optimal for routing [30].

REFERENCES


