Improved antnet routing algorithm with link probability evaporation over the given time window

F. Tekiner\textsuperscript{a}, Z. Ghassemlooy\textsuperscript{a} and S. Al-khayatt\textsuperscript{b}

\textsuperscript{a}: Optical Communications Research Group, School of Engineering and Technology, Northumbria University, Newcastle upon, Tyne, NE1 8ST, UK.

\textsuperscript{b}: School of Computing and Management Sciences, SHU, Sheffield, S1 1WB, UK.

Email(s): ftekiner@ieee.org, fary.ghassemlooy@unn.ac.uk

Abstract - Antnet is a software agent based routing algorithm that is influenced by the unsophisticated and individual ants emergent behaviour. In real life, ants deposit some kind of chemical substance to mark the path that they used. Then on their way back they choose the path with the most pheromones which becomes the shortest path. Ants (nothing but software agents\textsuperscript{1}) in antnet are used to collect traffic information and to update the probabilistic distance vector routing table entries. In the original antnet algorithm the only feedback provided to the system by the software agents is the trip time observed on their way to source to destination. Then this feedback signal is reinforced by updating the related probability entry in the distance vector table. In this work link usage information is used as the second feedback to the system. For every link the usage information is held at every node that they are connected in a given time window. Routing table entries then reinforced based on the calculated evaporation information based on the link usage statistics. This is influenced from the real life scenario where chemical substance deposited by the ants evaporates over time. Simulation results show that average delay experienced by data packets is reduced by \(~7\%\) by evaporating link probabilities under non-uniform traffic model. However, there was no performance gain on the uniform traffic models.

I. INTRODUCTION

In today’s fast growing Internet traffic conditions changes and failures occurs at some parts of the network from time-to-time, in an unpredictable manner. Therefore, there is a need for an algorithm to manage traffic flows and deliver packets from the source to the destination in a realistic time.

Routing algorithm is the key element in networks performance and reliability, thus it can be seen as the ``brain” of the network. An ideal routing algorithm should be node and link independent, and be able to deliver packets to their destination with the minimum amount of delay, regardless of the network size and the traffic load. The only way to achieve this would be by employing an intelligent and distributed routing algorithm.

The routing algorithms currently in use lack intelligence, and need human assistance and interpretation in order to adapt themselves to failures and changes. Routing is considered to be NP-Hard Optimization problem, therefore widely used optimization problems have been applied in the literature. To name a few, Genetic Algorithms, Neural Networks, Simulated Annealing, Software Agents and Reinforcement Learning.

In recent years, agent based systems and reinforcement learning have attracted researchers interest. This is because these methods do not need any supervision and are distributed in nature. Swarm intelligence particularly ant based systems [1], Q-learning [2] methods and hybrid agent based Distance Vector algorithms [3] have shown promising and encouraging results. In this paper, the focus will be on the antnet routing algorithm.

The ant-based approach applied to routing problem was first reported in [4]. This work was influenced from the work done on the software agents used for control in telecommunication networks [5]. Improved version of the algorithm in [4] was applied to the connection oriented systems [6]. Then for the first time ant based routing was applied to the packet based connection-less systems [7]. This was followed by a mobile agents approach to adaptive routing in [8].

Although ant based routing algorithms have shown some interesting results, they are still far away from being ideal. One of the major problems is that the network freezes and consequently the routing algorithm gets trapped in the local optima and is therefore unable to find new improved paths. This is also called stagnation [9], which has been studied by many researchers. A number of methods have been proposed to overcome this problem. Such as: evaporation, aging, pheromone control and hybrid algorithms. However, most of these are rather complex and requires other heuristics. Here the focus is on evaporation, since it is believed that it is one of the simplest methods and adds negligible overhead to the node itself. This paper proposes a simple method by using the idea of evaporating the probabilities assigned to the every link based on a statistical variables held at every node.

\textsuperscript{1} Term ant and software agents will be used interchangeably throughout the text.
II. ANTNET ROUTING ALGORITHM

Individual ants are unsophisticated and simple insects, but when they act in a collaborative way that they are capable of performing a variety of complex tasks [10]. Real ant behaviour has been observed by many researches and a set of collective behaviour of several species has been agreed [1, 4, 10]. Ants in real life,

- explores particular areas for food
- builds and protecting nests
- sorts brood and food items
- cooperates in carrying large items
- immigrates as a colony
- leaves pheromones on their way back
- preferentially exploits the richest available food source
- stores information on the nature (uses world as a memory)
- makes decision on a stochastic way
- finds the shortest paths to their nests or food source
- are blind, can not foresee the future
- has very limited memory.

It is believed that these behaviours emerge from their interactions when in a large colony and environment they are in. A notion of stigmergy, which is a form of indirect communication through the environment, is used by ants [1,4]. They lay pheromone, some kind of chemical, to mark the path used for finding food source, which is then evaporates over certain time.

Antnet [8, 11] is an agent based routing algorithm that is influenced from the real ant’s behaviour. In antnet ants explores the network to find the optimal paths from the randomly selected source destination pairs. Moreover, while exploring the network ants update the probabilistic routing tables and construct a statistical model of the nodes local traffic. Ants make use of these tables to communicate with each other. The algorithm uses two types of ants namely, forward ants and backward ants to collect network statistics and to update the routing table. In each node there are two types of queues, low priority and high priority. The data packets and the forward ants use low priority queues, whereas the backward ants use the high priority queues. Later forward ants do also use the high priority queues [8, 11].

![Fig. 1. Ants laying pheromones.](image)

For example, in Fig. 1(i) ants don’t know where to go initially, and choose one of the paths randomly. It is expected on average, half of the ants will choose the shorter path and the other half will choose the longer path, since they have no information or knowledge of the paths. It can be observed from Fig. 1(ii) that the ants that have chosen the shorter path is about to reach their destinations, whereas others have still some way to go. The ants that have chosen the shorter path will return to their nest before the others. In Fig. 1(iii), the number of ants that choose the shorter path will be increased as more pheromones are laid on the shorter path. Thus it represents the optimal solution.

![Fig. 2. Probabilistic routing table and traffic statistic.](image)

Ants communicate through two data structures stored in every node (see Fig. 2) as outlined below [8, 11]:

i. A distance vector routing table $T_k$ with distance metric defined with probabilistic entries where for each possible destination $d$ and neighbour node $n$ there is a probability value $P_{nd}$ that reflects the goodness of the link (path), given in equation 1.

$$
\sum_{n \in N_k} P_{nd} = 1, d \in [1, N] \cup N_k
$$

ii. An array $M_k (\mu_d, \sigma_d^2, W_d)$ defines a simple statistical traffic model experienced by the node $k$ over the network. Where $W_d$ is the observation window used to compute the estimated mean $\mu_d$ and the variance $\sigma_d^2$ parameters given as, respectively:

$$
\mu_d = \mu_d + \eta(\mu_{k \rightarrow d} - \mu_d)
$$

$$
\sigma_d^2 = \sigma_d^2 + \eta((\mu_{k \rightarrow d} - \mu_d)^2 - \sigma_d^2)
$$

$\eta$ weights the number of samples that affect the average and set to 0.05[10].
Where \( o_{k \rightarrow d} \) is the new observed trip time experienced by the agent while travelling from node \( k \) to destination \( d \). Algorithm description can be found in detail in [8, 11, 12].

### III. EVAPORATION

When a network reaches to its equilibrium state stagnation occurs. This is an undesirable property also caused by software ants recursively choosing the same path in antnet routing algorithm. Therefore routing optimisation may get stranded in the local optima and may not be able to discover new paths that became optimal due to the changes in network traffic and topology (i.e. due to link/node failures or deleting/adding new nodes). Moreover, ants finding the shortest path is a statistical process [13]. In the beginning if ants choose a non-optimal path, the probability of other ants following them increases. Several methods such as aging, limiting amount of pheromone laid, evaporation and using other heuristics beside ant colony optimisation have been used and studied in the literature [9].

Evaporation is a real life scenario where pheromone laid by real ants evaporates in time due to natural circumstances. In this paper, evaporation concept is applied to the antnet routing algorithm to improve the performance in a non-ideal network and changing conditions. Evaporation rate \( E(x) \) defined in equation 4 represents the reduction in the probability from a link to a neighbour \( x \). It is the proportion of number of forward ants destined to the node \( x \) over total ants received by the current node in the given time window. These variables that kept track of the number of ants forwarded to the neighbours are cleared after a specific time period.

\[
E(x) = \left[ 1 - \frac{\text{ant}_\text{send}(x)}{\sum_{i=0}^{\text{ant}_\text{send}(i)}} \right] xP(x)
\]

Having found \( E(x) \), it is then subtracted from its probability (eq. 5).

\[
P(i) = P(i) - E(x), i = x
\]

Since, sum of all probabilities for all neighbours \( N \) is 1, then probability of evaporation from node \( x \) distributed equally to the other neighbours is given by:

\[
P(i) = P(i) + \frac{E(x)}{N-1}, i \neq x
\]

### IV. SIMULATION RESULTS

Evaporation criteria are applied to the modified antnet routing algorithm [12] and are implemented in the following environment.

- Algorithms are implemented in the C language in a parallel environment by using Parallel Virtual Machine (PVM).
- Parallel behaviour is simulated by assigning every process to a different node both on the same machine and different machines.
- Poisson traffic distribution is used on non-uniform data traffic where 33% of the traffic is forwarded to node 1 and node 21.
- No packet creation destined to the immediate neighbours.
- 5 seconds are given to discover all the paths and to initialise the probabilistic routing table entries.
- Ant creation rate is set to 10 seconds per node.
- Random network topology with each link having equal cost is used, see Fig. 3.
- For each simulation, packet generation is stopped after creation of 2500 packets per node and simulation is stopped after all packets are arrived to their destinations or detected and deleted from the network.
- Every simulation is run 15 times and the average of the results is used for accuracy. It is assumed that the packet size is fixed and there is no packet loss.
- All experiments are implemented for varying link evaporation rates, since it has a significant effect on the performance of the algorithm.

Average packet delay is used for evaluating the performance of the algorithm, which is the average delay experienced by the packet while it is being routed from source to the destination.

Results for the average delay against the evaporation rate are presented in Fig. 4 and are compared with the case with
no evaporation. It can be seen from Fig. 4 that for all cases the algorithm with evaporation has performed better than the non-evaporation (E(NA)) version. Results also show that, with the evaporation algorithm the lowest average delay is achieved when the evaporation window is set to 1 second (E(1)), which is ~7% better than the non-evaporating algorithm. However, while conducting the tests it was observed that evaporation algorithm was not as stable as the non-evaporation algorithm. There was not much difference on the 15 tests conducted by non-evaporation configuration. On the other hand, there is around 0.2 seconds difference among the best and worst cases observed with the evaporation case. It is believed that, this is due to the two main reasons. Firstly, due to the non-uniform traffic model used in the tests, and secondly the algorithm capturing the wrong time window as that is the only factor on choosing the time to evaporate.

V. CONCLUDING REMARKS

In the original antnet routing algorithm the only feedback signal used was the reinforcement applied by using the agents trip times. Here, in addition the link usage statistics are used as a feedback signal and reinforced to the solution by evaporation parameter to improve the performance of the algorithm. Evaporation parameter is calculated based on the link usage statistics held by every node. Simulation results showed that under non-uniform traffic conditions the algorithm with evaporation displayed reduced average delay per packet compared with algorithm with no evaporation. Moreover, the simple rules and unsophisticated variables used in the improved version have not added a significant overhead to the network and to the node. On the other hand, further tests showed that evaporating path probabilities has no effect on uniform and non-problematic traffic conditions. This was an expected outcome as there is no need to change the solution found if traffic stays static all the time. When network conditions do not change, there are no new improved paths are available. Thus, the solution found is the optimal solution. However, inconsistency observed in the average delay is the main drawback observed in the simulations. It is thought that this can be overcome by using dynamic evaporation window variable instead of using a constant variable. This is the subject of our current and future research.

REFERENCES