

AntNet-R: Ant Colony Optimization Based Routing Algorithm

K.Kotecha* and P. Kotak**

** Computer Dept A.V.Parekh Tech. Institute, Saurashtra University, Ahmedabad , India

** Nirma Institute of Technology, Nirma University, Ahmedabad , India

drketankotech2@gmail.com, kotakp2003@yahoo.com

Abstract - In this paper we present AntNet-r, a new algorithm for routing in mobile ad hoc networks. Due to the ever changing topology and limited bandwidth it is very hard to establish and maintain good routes in such networks. Especially reliability and efficiency are important concerns. AntNet-r is based on ideas from Ant Colony Optimization. AntNet-r is even closer to real ants's behavior that inspired the development of the ACO met heuristic than ACO algorithms for N-P hard problems , Here we have also focus on the network routing problem and survey swarm intelligent approaches or its solution , and during the course of the communication session, ants proactively test existing paths and explore new ones. In simulation tests we show that AntNet-r can outperform AODV, one of the most important current state-of-the-art algorithms, both in terms of end-to-end delay and packet delivery ratio.

I. INTRODUCTION

In this work we describe AnNe-r, a routing algorithm for mobile ad hoc networks (MANETs) [3]. AntNet's design, which combines both proactive and reactive components, is based on the shortest path behavior observed in ant colonies and on the related optimization framework of Ant Colony Optimization (ACO) [1]. It has been experimentally observed that ants in a colony can converge on moving over the shortest among different paths connecting their nest to a source of food [2, 1]. The main catalyst of this colony-level shortest path behavior is the use of a volatile chemical substance called pheromone: ants moving between the nest and a food source deposit pheromone, and preferentially move in the direction of areas of higher pheromone intensity. Shorter paths can be completed quicker and more frequently by the ants, and will therefore be marked with higher pheromone intensity. These paths will therefore attract more ants, which will in turn increase the pheromone level, until there is convergence of the majority of the ants onto the shortest path. Local intensity of the pheromone field, which is the overall result of the repeated and concurrent path sampling experiences of the ants, encodes a spatially distributed measure of goodness associated with each possible move. This form of distributed control based on indirect communication among agents which locally modify the environment and react to these modifications is called stigmergy [4].

Here AntNet-r, the routing algorithm discussed, is extension of Ant Colony Optimization (ACO) In this paper we focus on the network routing problem . The aim of intelligent network routing is to detect dynamic traffic

and topology events, thus identifying network bottlenecks, addressing them in an adaptive, intelligent manner.

II. ANTNET-R : INTRODUCTION

In the AntNet-r algorithm, routing is determined through complex interactions of network exploration agents, called ants. These agents are divided into two classes, the forward ants and the backward ants. Ants in each set possess the same structure, but they are differently situated in the environment, that is , they can sense different inputs and they can produce different , independent outputs. Ants communication in an indirect way, according to the stigmergy paradigm[[4], through the information they concurrently read and write on the network nodes they visit.

In AntNet-r, artificial ants move on the construction graph $G_c = (C, L)$ with the constraint of never using the set of links that do not belong to the network graph. Like all ACO algorithms, AntNet-r exploits pheromone trails. These are maintained in an artificial pheromone matrix T_i associated with each node i of the data network. AntNet maintains at each node i a sample parametric statistical model M_i of the traffic situation over the network as seen by node i . Here T and M , illustrated in fig 1 can be seen as memories local to nodes capturing different aspects of the network dynamics. The mode M maintains absolute distance/time estimates to all the nodes, while the pheromone matrix gives relative goodness measures for each link-destination pair under the current routing policy implemented over all the network . Here it is summarized as:

1. Each network node launches forward ants to all destinations at regular time intervals.
2. The ants find a path to the destination *randomly* based on the current routing tables.
3. The forward ants creates a *stack*, pushing in trip times for every node as that node is reached
4. When the destination is reached, the backward ants *inherit* the stack
5. The backward ants pop the stack entries and follow the path in reverse
6. The routing tables of each visited node are updated based on trip times

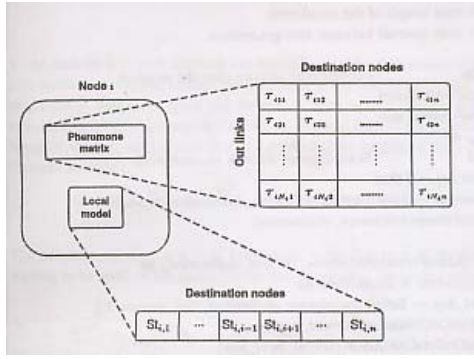


Figure 1. Data structure used by ants in AntNet for the case of a node i with $N_i = |N_i|$ neighbors and a network with n nodes.

III. ANTNET : DATA STRUCTURE

In AntNet-r artificial ants move on the construction graph $G_c = (C, L)$

Here the AntNet-r algorithm, whose high-level description in pseudo-code is explained.

```

procedure AntNet-r( $t, t_{end}, \Delta t$ )
  input  $t$  % current time
  input  $t_{end}$  % time length of the simulation
  input  $\Delta t$  %time interval between ants generation
  foreach  $C \in C$  do
     $M \leftarrow \text{InitLocalTrafficModel}$ 
     $T \leftarrow \text{InitNodeRoutingTable}$ 
    While  $t \leq t_{end}$  do
      In_parallel
        If ( $t \bmod \Delta t$ ) = 0 then
          Destination  $\leftarrow \text{SelectDestination}$ 
          (traffic_distribution_at_source)
          LaunchForwardAnt(source, destination)
        End-if
      End-parallel
    End-while
  End-foreach
  foreach(ActiveForwardAnt[Source,current,destination] )
    do
      while(current # destination) do
        next_hop  $\leftarrow \text{SelectLink}$  (current,
          destination,link_queues,T)
        PutAntOnLinkQueue(current, next_hop)
        WaitOnDataLinkQueue(current, next_hop)
        CrossLink(current, next_hop)
        Memorize(next_hop, elapsed_time)
      End-while
    End-foreach
  End-while

```

```

      current  $\leftarrow$  next_hop
    End-while
  LaunchBackwardAnt(destination, source,
    memory_date)
End-foreach

foreach (ActiveBackwardAnt[source, current,
  destination] )
do
  while(current # destination) do
    next_hop  $\leftarrow$  PopMemory
    WaitOnHighPriorityLinkQueue(current,
      next_hop)
    CrossLink(current, next_hop)
    from  $\leftarrow$  current
    current  $\leftarrow$  next_hop
    UpdateLocalTrafficModel(M, current_from,
      source_memory_data)
     $r \leftarrow \text{GetNewPheromone}$ (M, current_from,
      source_memory_data)
    UpdateLocalRoutingTable(T, current, source, r)
  End-while
End-foreach
End-in_parallel
End-while
End-foreach
End-procedure

```

Informally, the AntNet-r algorithm and its main characteristics can be summarized as follows:

- 1) At regular intervals and concurrently with data traffics, from each network node artificial ants are asynchronous launched towards destination nodes selected according to the traffic distribution.
- 2) Artificial ants act concurrently and independently, and communicate in an indirect way through the pheromones they read and write locally on the nodes.
- 3) Each artificial ant searches for minimum cost path joining its source and destination node.
- 4) Each artificial ant moves step by step towards its destination node. At each intermediate a greedy stochastic policy is applied to choose the next node to move to. The policy makes use of (1) node-local artificial pheromones, (2) node-local problem-dependent heuristic information, and (3) the ant's memory.

- 5) While moving, the artificial ants collect information about the time length, the congestion status, and node identifiers of followed path.
- 6) Once they have arrived at the destination, the artificial ants go back to their source node by moving along the same path as before but opposite direction.
- 7) During this backward travel, node local models of the network status and the pheromones stored. On each visited node are modified by the artificial ants as a function of the path they followed and of its goodness.
- 8) Once they have returned to their source Node, the artificial ants are deleted from system.

IV. CONCLUSION

We have described AntNet-r, an ACO algorithm for routing in MANETs. It is an algorithm, combining reactive route setup with proactive route probing and exploration. The algorithm seems to benefit a lot from situations in which there are some regularities and correlations which can be learned and exploited for data transport and path discovery. The algorithm also shows good scalability. In future work we will improve the exploratory working of proactive ants. By extending the concept of pheromone diffusion, more information about possible path improvements will be available in the nodes, and this information can guide proactive ants. This should lead to better results with less overhead. We also want to make the generation and forwarding of proactive ants adaptive to the Network situation.

REFERENCES

- [1] M. Dorigo, G. Di Caro, and L. M. Gambardella. Ant algorithms for distributed discrete optimization. *Artificial Life*, 5(2):137–172, 1999.
- [2] S. Goss, S. Aron, J. L. Deneubourg, and J. M. Pasteels. Self-organized shortcuts in the Argentine ant. *Naturwissenschaften*, 76:579–581, 1989.
- [3] E. Royer and C.-K. Toh. A review of current routing protocols for ad hoc mobile wireless networks. *IEEE Personal Communications*, 1999.
- [4] R. Sutton and A. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.
- [5] G. Di Caro and M. Dorigo, “AntNet: a mobile agents approach to adaptive routing”, *Tech. Rep. IRIDIA/97-12, Université Libre de Bruxelles, Belgium*.
- [6] G. Di Caro and M. Dorigo, “AntNet: distributed stigmergetic control for communications networks”, *Journal of Artificial Intelligence Research*, vol. 9, pp. 317–365, 1998.
- [7] G. Di Caro and M. Dorigo, “Extending AntNet for best effort quality-of-service routing”, *Proc. ANTS’98 – 1st Intl. Workshop on Ant Colony Optimization*, Brussels, Belgium, October 15-16, 1998.
- [8] G. Di Caro and M. Dorigo, “AntNet: A Mobile Agents Approach to Adaptive Routing in Communication Network”, *9th Dutch Conf. on Artificial Intelligence (NAIC ’97)* Antwerpen (BE), November 12-13, 1997.
- [9] G. Di Caro and M. Dorigo, “Ant colonies for adaptive routing in packet-switched communications networks”, *Proc. PPSN V – 5th Intl. Conf. on Parallel Problem Solving from Nature*, Amsterdam, Holland, September 27-30, 1998.
- [10] G. Di Caro and M. Dorigo, “An adaptive multi-agent routing algorithm inspired by ants behavior”, *Proc. PART’98 – 5th Annual Australasian Conference on Parallel and Real-Time Systems*, Adelaide, Australia, September 28-29, 1998.
- [11] G. Di Caro and M. Dorigo, “Two ant colony algorithms for best-effort routing in datagram networks”, *Proc. 10th Intl. Conf. on Parallel and Distributed Computing and Systems*, Las Vegas, Nevada, October 28-31, 1998.
- [12] G. Di Caro and M. Dorigo, “Ant colony routing”, *PECTEL 2 Workshop on Parallel Evolutionary Computation in Telecommunications*, Reading, England, April 6-7, 1998.
- [13] G. Di Caro and M. Dorigo, “Adaptive learning of routing tables in communication networks”, *Proc. Italian Workshop on Machine Learning*, Torino (IT), December 9-10, 1997.
- [14] G. Di Caro and M. Dorigo, “Distributed reinforcement agents for adaptive routing in communication networks”, *3rd European Workshop on Reinforcement Learning*, Rennes, France, October 13-14, 1997.
- [15] G. Di Caro and M. Dorigo, “Mobile agents for adaptive routing”, *Proc. 31st Hawaii Intl. Conf. on System Sciences*, IEEE Computer Society Press, Los Alamitos, CA, pp. 74-83, 1998.
- [16] M. Dorigo and G. Di Caro, “Ant colony optimization: a new meta-heuristic”, *Proc. 1999 Congress on Evolutionary Computation*, July 6-9, 1999, pp. 1470-1477.
- [17] R. Schoonderwoerd, “Collective intelligence for network control”, M.S. Thesis, *Delft University of Technology, Faculty of Technical Informatics*, May 1996.
- [18] R. Schoonderwoerd, O.E. Holland, J. Bruten, L. Rothkrantz, “Ant-based load balancing in telecommunications networks”, *HP Labs Technical Report*, HPL-96-76, May 21, 1996.
- [19] R. Schoonderwoerd, O.E. Holland, J.L. Bruten, “Ant-like agents for load balancing in telecommunications networks”, *Proc. 1st ACM International Conference on Autonomous Agents*, February 5-8, 1997, Marina del Rey, CA, USA, pp. 209-216.
- [20] A. Bieszczad, B. Pagurek, and T. White, “Mobile agents for network management”, *IEEE Communication Surveys*, Fourth Quarter 1998, vol. 1, no. 1, 1998.
- [21] E. Bonabeau, F. Henaux, S. Guerin, D. Snyers, P. Kuntz, and G. Théaulaz, “Routing in telecommunications networks with “smart” ant-like agents”, *Proc. Intelligent Agents for Telecommunications Applications ’98*.