Optimized TERMITE: A Bio-inspired Routing Algorithm for MANET's

Praveenkumar G Hoolimath, Kiran M, G Ram Mohana Reddy
Department of Information Technology,
National Institute of Technology Karnataka
Mangalore, India
{praveen.hoolimath, kiranmanjappa, profgrmreddy }@gmail.com

***Abstract*— A Mobile Adhoc Network (MANET) is a collection of mobile nodes connected by the Wireless medium and each mobile node is aware of only its neighbours. Due to mobility of these mobile nodes the topology changes dynamically. Such a dynamic network topology makes the task of routing a challenging one. Recently, a new class of routing algorithms based on Swarm Intelligence has emerged. These algorithms are inspired by nature's self-organizing systems like ant-colonies, bird-flocks, honey-bees, school of fish, spiders and fireflies. The characteristics of such algorithms are their capability of self-organization, adaptation to the changing conditions, self healing and local decision making. In this work, a routing protocol inspired by the termite activity in nature, called Optimized-Termite (Opt-Termite), is proposed. Opt-Termite uses concept of stigmergy for self-organization, thereby reducing the control packet overhead. Opt-Termite mainly concentrates on load balancing for optimization. With Opt-Termite, a route with less loaded mobile nodes in terms of traffic will be chosen to reach destination. The routing information at each node gets influenced by the movement of packets and the routing table will be updated accordingly. It also allows the use of multiple paths and each packet is routed randomly and independently. Opt-Termite is implemented in ns-2 and its performance is compared with traditional routing protocol AODV. Opt-Termite's performance has been promising.**

***Keywords*— Bio-inspired routing, Termite, MANET's, load balancing, Stigmergy, Multipath routing**

I. Introduction

Mobile adhoc networks (MANETs) [1] are self-configuring infrastructureless network of mobile devices connected by wireless links. Each device in a MANET is free to move independently in any direction and therefore the links between nodes will be changing frequently. These mobile adhoc networks are very simple to setup and support scalablility, i.e nodes can be added or removed anytime. At the same time, the nodes show a high mobility which implies frequent changes to the network topology. Each node will be forwarding traffic, and therefore each node acts as a router as well. The maintenance of routing information by the nodes is a challenging task as the network topology will be changing frequently in a MANET.

The routing protocols in MANETs can be classified into 3 categories: reactive, proactive and hybrid routing protocols. Proactive routing protocols often need to exchange control packets among mobile nodes and continuously update their routing tables. This has a high overhead congestion of the network, which requires lots of memory. The advantage of proactive protocols is that nodes have correct and updated information. Reactive routing protocols only seek a route to the destination when it is needed. The advantage of these protocols is that the routing tables located in memory are not continuously updated. On the other hand, they have the disadvantage that they cannot establish connections in real time. Hybrids are derived from a mixture of these two protocols, and for this reason, they share some of their advantages. Some of the routing protocols for MANETs are DSDV, Dynamic Destination Sequenced Distance Vector protocol (proactive), AODV, Adhoc On-Demand Distance Vector protocol (reactive).

Biologists observed that biological systems have the ability to self-organise. For example, ants build their nest with a very complex structure that suggests a level of management beyond the capacity of an individual ant, through simple pheromone based communications between individual ants; the colony builds a fully functional nest. The idea here is to derive inspiration from these biological systems and apply them to the mobile adhoc networks. As in the case of biological systems which do not have a pre-defined way of communicating with each other, achieve the same with the help of the concept, stigmergy. Stigmergy is a mechanism of indirect coordination between agents or actions. The principle is that the trace left in the environment by an action stimulates the performance of a next action, by the same or a different agent. This stigmergy exhibited by the biological systems can be used in our networks for defining new routing protocols for MANETs. The MANETs are very similar to the colonies of ants or termites, where these insects work independently like the nodes in a MANET, the interesting feature is the communication that happens between these insects, who don't even know the existence of the other insect. This communication happens through a chemical substance (pheromone) excreted by these insects. These pheromones are used during the nest building process of the ants (trail pheromone), while searching for the food (aggregation pheromone) as well as to inform the fellow colony mates about any threat (alarm pheromone) that might have occurred and they need protect themselves from it. This concept of pheromone can be used to reduce the control traffic in the network. These systems exhibit a positive feedback and

978-1-4673-2014-6/12/$31.00 ©2012 IEEE

negative feedback during their communication. Positive feedback means the deposition of pheromone that indicates other insects to follow the same and thereby achieve the global objective of either food collection or nest building. Negative feedback means the evaporation of the pheromone deposited earlier which eliminates the trails of a route when it is no longer being used and thereby allows for new routes to emerge. The bio-inspired routing has immense potential as it is proved to work in the nature and so it does if incorporated in our network systems.

The paper is structured as follows, section II briefly discusses some of the bio-inspired routing algorithms. Section III explains the proposed algorithm, section IV gives details of simulation and results. Finally section V Concludes the paper with future work.

II. Related Work

A lot of work has been done on bio-inspired routing algorithms. These algorithms are mostly inspired from the behaviour of the organisms in nature like ants, honeybees, bird flocks, etc. Here, we discuss in brief some of those routing algorithms.

A. Ant colony based routing algorithms

The basic idea of the ant colony optimization (ACO) is taken from the food searching behaviour of real ants [2]. When ants are on the way to search for food, they start from their nest and walk toward the food. When an ant reaches an intersection, it has to decide which branch to take next. While walking, ants deposit a pheromone. The concentration of pheromone on a certain path is an indication of its usage and hence affects the moving decisions of the ants. After a short time the pheromone concentration on the shorter path will be higher than on the longer path, because the ants using the shorter path will increase the pheromone concentration faster. Thus, the ant's collective behaviour leads to global intelligent behaviour and helps in optimizing the path to the food.

The routing algorithms based on ants, mimic this very nature in order to provide an efficient routing mechanism.

(a) AntHocNet

It is a hybrid ACO routing algorithm [2]. It combines reactive route setup with proactive route probing, maintenance and improvement. It also takes into account the dynamic topology and other characteristics of ad-hoc networks. When the network topology changes, then it must be restored quickly and this is achieved through a new route discovery process. The algorithm tries to find paths characterized by minimal number of hops, low congestion and good signal quality between adjacent nodes. It mainly concentrates on path exploration but suffers from large control overhead as it needs to find path frequently and for which it broadcasts forward ants.

(b) ARA

It is Ant-Colony based Routing Algorithm [3]. It is purely a reactive algorithm where the forward and backward ants setup the paths to the nodes, the routing tables are updated by the data packets, reducing the control overhead. Its performance is slightly better than AODV but worse than DSR in highly dynamic environments.

Many routing algorithms have been proposed based on the ant behaviour such as Ant-AODV, Probabilistic Emergent Routing Algorithm (PERA), etc.

B. Bee colony based routing algorithms

The main algorithm here is the BeeAdhoc [4]. It is inspired by the honeybee behaviour in its design of agents and their interaction. It is a reactive algorithm. It has 4 types of agents as in a beehive, packers, scouts, foragers and bee swarms. The packer mimics the food storer and its main task is to find a forager for the data packet at hand. The task of scouts is to find new routes from their launching node to their destination node. These scouts are broadcast with a TTL value; if they don't return within a timeout then new scouts are sent with a greater TTL value. When the scouts return, they assign a forager for the route they just found which is similar to the waggle dance of scout-bees in nature. The foragers are the main workers in BeeAdhoc, they receive data packets from packers and deliver them to their destinations in a source-routed modality. The bee swarms are explicitly used to drop the foragers back to their source nodes in case of unreliable protocols like UDP.

In BeeAdhoc, each node contains a software module called hive at its network layer. It has 3 parts: the packing floor, the entrance floor and the dance floor as shown in Fig. 1.

BeeAdHoc delivers the same or better performance than that of the state-of-the-art algorithms like AODV, DSR and DSDV, but at significantly smaller overall energy expenditure.

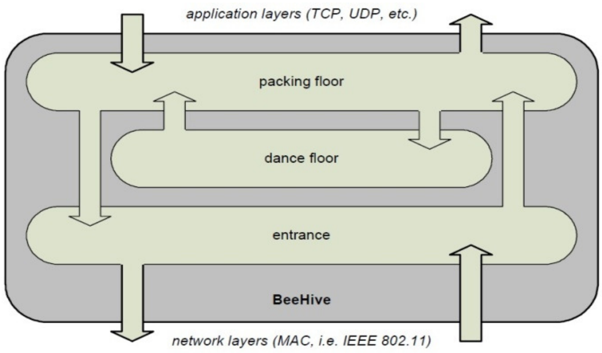

Fig. 1 Different floors in a beehive

C. Bird flocking based routing algorithms

Birds travel long distances in flocks which are '**V**' shaped where all the burden is on the sphere head of the flock which reduces the up thrust required by the rest of the birds in the flock. Thus, it reduces the amount of energy required by the birds behind the wing of the sphere head to fly. This is the reason why birds travel long distances by loosing minimum amount of energy [5]. As the energy of the sphere head reduces, the sphere head is replaced by the other bird to take the burden of the V shape No specific bird directs the movement of flock. Instead, each bird takes its cue to turn in one direction or another from those immediately surrounding it. BFBR [5] is a hybrid routing algorithm and uses Encounter Search algorithm for routing.

III. Proposed Algorithm

The Termite algorithm [7-11] is based on termites' activity of hill building. Termites' colonies use decentralized, self-organized systems of activity guided by swarm intelligence to exploit food sources and environments that could not be available to any single insect acting alone. The hill building nature of termites is briefly explained below for the convenience. Consider a flat surface upon which termites and pebbles are distributed. The termites would like to build a hill from the pebbles. Termites act independently of all other termites, and move only on the basis of an observed local pheromone gradient. A termite is bound by these rules:

1) A termite moves randomly, but is biased towards the locally observed pheromone gradient. If no pheromone exists, a termite moves uniformly randomly in any direction.
2) Each termite may carry only one pebble at a time.
3) If a termite is not carrying a pebble and it encounters one, the termite will pick it up.
4) If a termite is carrying a pebble and it encounters one, the termite will put the pebble down. The pebble will be infused with a certain amount of pheromone.

Termite retains most of the main features of the general ACO meta-heuristic such as pheromone tables, probabilistic decisions, pheromone evaporation, etc. In Termite, forward ants (RREQ) are unicast and follow a random walk. Backward ants (RREP) do not necessarily follow the forward path backwards, but are also routed stochastically. Most importantly it overcomes the disadvantage with the ant based algorithms by having a low control packet overhead. The paths identified with the above approach Termite [7] may not necessarily be optimal, load balancing has not been dealt with, resulting in the use of a single path, which may result in the selection of a single route all the time irrespective of the number of hops and the kind of load on that route, which gets congested leading to packet drops.

The algorithm Opt-Termite proposes certain modifications to Termite that lead to a better route selection. Opt-Termite is explained in the following sub section where first 3 subsection explains the data structures used followed by the route establishment and maintenance mechanism.

1. *Pheromone table(P)*

Pheromone table is an MxN matrix containing pheromone values, M destinations and N neighbor nodes, which will be considered while determining the nexthop for a packet as shown in TABLE I. Along the rows will be the destination nodes and along the columns will be the neighbor nodes. An entry $P_{d,n}$ gives the amount of pheromone that the neighbor *n* has for the destination *d*. The neighbor having a better pheromone value has a higher probability of being chosen as the nexthop for a packet while routing.

Consider a topology shown in Fig.2 and the corresponding pheromone table entry for node 0 is shown in TABLE 1. Node 0 has 5,3,6,7 neighbor nodes and 2,4,1 destination nodes. The Table is read as follows. The pheromone entry for the destination node 2 at node 0 through neighbor node 5 is 87.989 i.e, $P_{2,5} = 87.989$

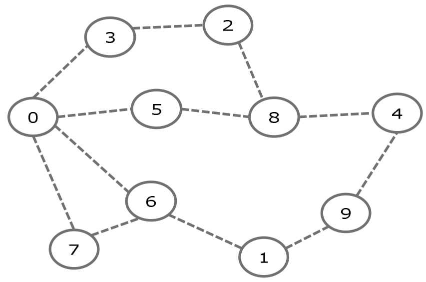

Fig. 2 Network topology of MANET

TABLE I Pheromone Table for node 0

		Neighbors (N)			
	0	5	3	6	7
Destinations (d)	2	87.989	54.99	987.09	255.76
	4	1098.98	87.772	509.28	198.99
	1	56739.08	1067.09	123.08	830.72

2. *Neighbor table(nt)*

The neighbor table stores an extra field of information about every node, the queue status (values are fractions between 0 and 1, that is percentage) as shown in the TABLE II, which is the percentage of the queue filled in those nodes. This gives the information about the load on these nodes which is used in the forwarding function.

TABLE II Neighbor Table

Neighbor node	Queue status (%)
5	30
3	25
6	70
7	10

3. *Forwarding function*

At every node all routing decision will be made using the forwarding function shown in (1). The forwarding function uses the pheromone table and neighbor table to choose the next hop.

$$p_{d,n} = \frac{(P_{d,n} * (1 - K_n))^F}{\sum_{i \in N} (P_{d,i} * (1 - K_i))^F} \qquad (1)$$

where, $p_{d,n}$ is the probability for the neighbor node *n* to get selected as the nexthop towards the destination *d*. $P_{d,n}$ is the pheromone value on neighbor node *n* to reach destination *d*, A new variable K (*load factor),* the percentage of queue full on the neighbor node *n*, is introduced in the forward function for load balancing. And *F* is a smoothing factor as like basic Termite algorithm. Nominal value for *F* is 2. The neighbor node having the highest probability will be the nexthop for the packet.

4. *Opt-Termite*

The routing algorithm is explained in the form of the actions taken by a node every time it receives a packet. It consists of the following phases

4.1 Route discovery

The route discovery starts with the broadcast of route request (RREQ) packets. The Fig. 3 below explains what happens when a RREQ packet is received. The route request is replied by a node at one hop distance from the destination or any node having routing information (pheromone) about the destination. All nodes will be aware of their neighbors and this neighbor responds for the route request, by sending the route reply. Fig. 4 explains what happens when a node receives a route reply.

On receiving a route request *rreq* for a destination *d*, a node *n* does the following,

1. Update its neighbor table *nt* for queue status of the previous hop of *rreq* as follows

   $$n \rightarrow nt[rreq \rightarrow prevhop] \rightarrow queuestatus = rreq \rightarrow prevQueue$$

2. Update the pheromone table *P* for the source of the *rreq* using the formula

   $$P[SrcID][prevhop] = \left(\frac{1}{rreq \rightarrow hopcount} + \frac{rreq \rightarrow hopcount}{rreq \rightarrow QueueSum}\right) * Ph$$

   hopcount is the no of hops from the source of *rreq* and *QueueSum* is the sum of the percentages of queue filled at each node encountered by the *rreq* in its journey so far

3. If the pheromone table *P* has an entry for the destination *d* of *rreq* then

   create a *rrep* with pheromone value, *hopcount*=0 and *QueueSum*=current node's queue status, and send it back.

   $rrep \rightarrow ph = P[d][n]$
   $rrep \rightarrow hopcount = 1$
   $rrep \rightarrow QueueSum = n \rightarrow queueLength$

Else, Update the *hopcount* and *QueueSum* fields of *rreq* and forward it.

$rreq \rightarrow hopcount{+}{+}$
$rreq \rightarrow QueueSum{+}{=}\ n \rightarrow queueLength$
$rreq \rightarrow prevQueue = n \rightarrow queueLength$

Fig. 3 Node's behavior on receipt of Route Request

The route request and route reply packets carry the information about the number of hops and the amount of load on their way towards their destinations. The load is indicated by taking sum of the percentage of packet queue filled at each node on the route. The expression for pheromone updation during route discovery takes into account both, the number of hops as well as the load. This contributes for the selection of a route with lesser number of hops and lesser load on the nodes of the route.

4.2 *Route maintenance*

Route maintenance is taken care of by the data packets. Every time a data packet is received the pheromone for its source is updated by incrementing by a constant δ. Nominal value for δ = 1. This increment helps the optimal path to exist for longer time.

If the current node is the destination of the DATA packet then process it else Forward it according to the forwarding function given earlier in Eq (1). This reduces a lot of control packet overhead that would have been if separate control packets were used for route maintenance, which contributes for the better performance of the algorithm.

The pheromone table entries are reduced by certain amount after every second. This is called pheromone decay. The pheromone increment and pheromone decay are the positive and negative feedback mechanisms respectively. The pheromone increment and pheromone decay at regular intervals, are important for the better routes to stay alive and at the same time eliminating the less optimal routes from the pheromone table. If a pheromone value goes below a threshold (PH_FLOOR) then that entry will be removed from the pheromone table, thereby eliminating the less optimal route.

On receiving route reply *rrep*, a node *n* does the following,

1. Update the neighbor table *nt* for queue status of the previous hop of *rrep*

   $$n \rightarrow nt[rrep \rightarrow prevhop] \rightarrow queuestatus = rrep \rightarrow prevQueue$$

2. Update the pheromone table *P* for the source of the RREP using the formula

   $$P[SrcID][prevhop] = \left(\frac{1}{rreq \rightarrow hopcount} + \frac{rreq \rightarrow hopcount}{rreq \rightarrow QueueSum}\right) * Ph$$

   hopcount is the no of hops from the source of *rrep* and *QueueSum* is the sum of the percentages of queue filled at each node encountered by the *rrep* in its journey so far

3. If the current node itself is the destination of *rrep* then,

   rrep is discarded and data transmission is started

   Else,

   Update the *hopcount* and *QueueSum* fields of *rrep* and forward it.

   $rrep \rightarrow hopcount{+}{+}$
   $rrep \rightarrow QueueSum{+}{=}\ n \rightarrow queueLength$
   $rrep \rightarrow prevQueue = n \rightarrow queueLength$

Fig. 4 Node's behavior on receipt of Route Reply

On receiving *DATA* pkt *data*, a node *n* does the following, Update the pheromone table *P* for the source of the *DATA* pkt *data* according to the equation (2), where δ is a constant. The nominal value for δ used in the simulation is 1.

$$P_{data \rightarrow prevHop, data \rightarrow SrcID} {+}{=} \delta \quad (2)$$

At regular intervals of time, 1 second, a node does the following, Updates all the pheromone table entries according to the equation (3), where τ is the decay rate. The nominal value for τ used in the simulation is 0.105.

$$\forall\, n, d \in P, P_{d,n} = P_{d,n} * e^{-\tau} \quad (3)$$

4.3 *Route failure*

In case there is a route failure, which may be caused due to the node moving out of transmission range or a node getting overloaded, the packets can be easily switched to an alternate route, since multiple paths are maintained in the pheromone table.

IV. Simulation and Results

The simulation was carried using the network simulator ns2.34, with the simulation parameters as mentioned in the TABLE III. The simulation constants like PH_FLOOR, decay rate and decay period are same as the one mentioned in [7]. The random waypoint mobility model is used for random motion of nodes. The simulation results are compared with traditional routing protocol, AODV against throughput and control packet overhead. Since data packets are used to update the pheromone table, a lot of control packets are avoided, reducing traffic in the network.

TABLE III Simulation Parameters

Simulation area	2500 x 200
Number of nodes	100
Initial pheromone	10
Pheromone ceiling	10000
Pheromone floor(PH_FLOOR)	0.1
RREQ timeout	2 [seconds]
τ(decay rate)	0.105
Decay period	1 [second]
Data TTL	32 [hops]
RREQ TTL	32 [hops]
RREP TTL	32 [hops]
RREQs per Route Request	2

The results are analysed by taking the parameters, time (x axis) and throughput, control packet overhead (y axis) as shown in Fig. 5 and Fig. 6 respectively.

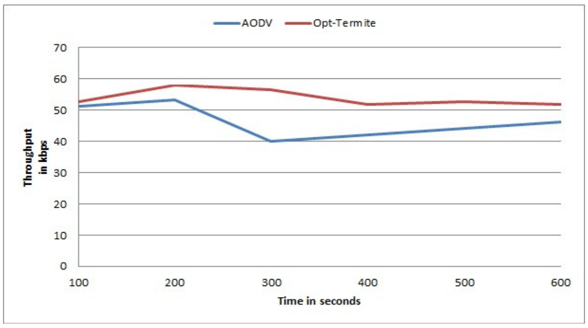

Fig. 5 Graph comparing the throughput v/s time of two protocols, Opt-Termite and AODV.

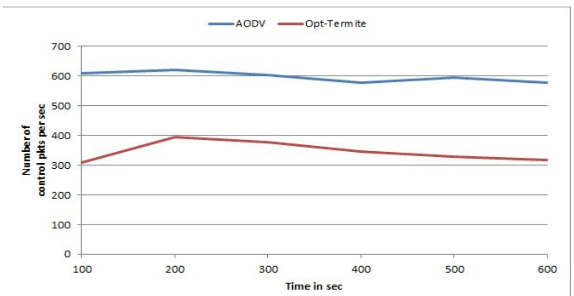

Fig. 6 Graph comparing the control packet overhead v/s time, Opt-Termite and AODV

Fig. 5 describes behaviour of Opt-Termite and AODV under the scenario for various different topologies, considered for simulation. Initially both have same characteristic but as the time goes on Opt-Termite tends to be more stable than AODV. Fig. 6 describes the control packet overhead experienced by Opt-Termite and AODV. From the graph it is very clear that Opt-Termite generates lesser control traffic as compared to AODV. Both throughput and control packet overhead are the average values taken from the simulations run for various different topologies.

V. Conclusion

The bio-inspired routing algorithms perform better in frequently changing conditions like, mobile adhoc network, as they are adaptive to the situations. Opt-Termite is one of the routing protocol inspired by the behaviour of termites in their hill building process. It has a low control overhead as well as it offers multiple paths. It also takes care of the congestion that could be caused by the usage of the same route for a long period of time. The simulation of the proposed framework and comparison with AODV protocol indicates a better performance of Opt-Termite in heavy traffic networks. In near future, authors like to give an analytical proof for the proposed method.

References

[1] IETF Mobile ad-hoc networks (MANET) working group, http://www.ietf.org/html.charters/manet-charter.html

[2] G.A. Di Caro, F. Ducatelle, and L.M. Gambardella. "AntHocNet: an ant-based hybrid routing algorithm for mobile ad hoc networks." In Proceedings of PPSNVIII, volume 3242 of LNCS, pages 461–470. Springer, 2004.

[3] MesutG¨unes¸, UdoSorges, ImedBouazizi "ARA the Ant-Colony Based Routing Algorithm for MANETs", Proceedings of the International Conference on Parallel Processing Workshops,2002.

[4] Horst F. Wedde, Muddassar Farooq, Thorsten Pannenbaecker, Bjoern Vogel, Christian Mueller, Johannes Meth and Rene Jeruschkat, "BeeAdhoc : An energy efficient routing algorithm for mobile adhoc networks inspired by bee behavior", GECCO '05 Proceedings of the 2005 conference on Genetic and evolutionary computation

[5] T. Srinivasan, T. NiranjanBabu, V. Mahadevan, M.Nivedita, G. Sahitya, A.Meyyappan, J.V.Sahana: "BFBR A novel Bird Flocking Behavior based Routing for Highly Mobile Ad hoc Networks" . In International Conference on Computational Intelligence for Modelling Control and Automation,and International Conference on Intelligent Agents,Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)2006, IEEE.

[6] Xu Hao, Ruth Falconer, David Bradley and John Crawford, "FUNNet – A Novel Biologically-Inspired Routing Algorithm Based on Fungi", CTRQ '09. Second International Conference on Communication Theory, Reliability, and Quality of Service, 2009.

[7] Martin Roth and Stephen Wicker. "Termite: Emergent Ad-Hoc Networking" This research is sponsored by the Defence Advance Research Projects Agency (DARPA), and administered by the Army Research Office under Emergent Surveillance Plexus MURI Award No. DAAD19-01-1-0504.

[8] Roth M. and Wicker S. 'Termite: Emergent ad-hoc networking', Second Mediterranean Workshop on Ad-Hoc Networks, Mahdia,Tunisi,2003.

[9] Martin Roth & Stephen Wicker, "Asymptotic Pheromone Behavior in Swarm Intelligent MANETs: An Analytical Analysis of Routing ", Behavior, Sixth IFIP IEEE International Conference on Mobile and Wireless Communications Networks (MWCN), 2004.

[10] Roth M, "The Markovian Termite: A Soft Routing Framework", Swarm Intelligence Symposium, 2007 SIS 2007, IEEE. 1-5 April, 2007.

[11] "The Network Simulator NS-2" http://www.isi.edu/nsnam/ns/