Special Issue Published in International Journal of Trend in Research and Development (IJTRD), ISSN: 2394-9333, www.ijtrd.com

Termite Adaptive Routing Algorithm for Mobile Ad Hoc Networks

[1]Anto Ramya. S. I, [2]Naveena. V. A
[1]Assistant Professor, [2]Student
[1,2]Department of Computer Science, St. Joseph's College of Arts and Science for Women, Hosur, Tamil Nadu, India.

Abstract – A Mobile Ad hoc Network (MANET), also called a Mobile Mesh Network, is a self – configuring network of mobile devices connected by wireless links. Contrasting to Traditional Mobile Wireless Networks, Ad Hoc Networks do not rely on any fixed infrastructure. The researches on various issues in Mobile ad hoc networks are becoming popular because of its challenging nature and all-time connectivity to communicate. This paper discusses a solution for finding a path between nodes in MANET. Termite is a novel algorithm for packet routing in communication networks. Termite is an adaptive, distributed, mobile agent-based algorithm which was inspired by recent work on the ant colony metaphor. According to this algorithm, a group of mobile agents also called artificial termites, build paths between a pair of nodes, exploring the network concurrently and exchanging obtained information to update the routing tables. Some of the parameters used to assess its performance are packet delays and throughput. The results of this algorithm prove better throughput as compared to existing algorithms. So, termite algorithm is a hopeful option for routing of data in commercial networks.

Keywords: *Mobile agents, Pheromone, Routing algorithm, Swarm intelligence, and Termite.*

I. INTRODUCTION

Mobile Ad Hoc Networks (MANET) is a self-configuring network of mobile hosts connected by wireless links, the unification of which forms the topology of the network [1]. The advantages of ad hoc networks are the convenience, mobility, productivity, deployment, and expandability. As the nodes in the network are changing, the topology of the network changes unpredictably. Hence, it is difficult to spawn a path between two nodes. This algorithm is adaptive, distributed and is inspired by swarm intelligence. Ant algorithms are the class of optimizing algorithms under swarm intelligence (SI) [2][3]. Routing in an ant algorithm [4][5] is through an interaction of network exploration agents called ants. According to this algorithm, a group of mobile agents builds a path between pairs of nodes by exchanging information and updating routing tables.

II. SWARM INTELLIGENCE

Swarm Intelligence (SI) is a composition for designing systems of simple interacting individuals. It is often coupled with the study of biological systems exhibiting the characteristics of swarm intelligence, such as social insect societies. This methodology is inspired and based upon observations of the behaviour of social insects such as ants, termites, social bees or social wasps. These biological systems embody many of the principles that man-made systems should have. They are composed of a big number of simple and cheap components (workers), co-operating locally and independently to accomplish a global task that any individual could not do alone. Colony behaviour is often tough against a large number of parameters such as individual misbehaviour or loss. Such systems are also able to adapt to the environment as is necessary.

Swarm intelligence is often considered a subfield to Artificial Life (AL). Artificial life is the study of designing machines with traits that mimic those of biological counterparts. This is also referred to as biometrics. SI is primarily concerned with studying large systems of interacting individuals, whereas AL is broader and also looks at the questions of self-replication, evolution, natural system modeling, or borrowing ideas from nature in order to enhance engineering design. Both SI and AL are generally thought to be subsumed by the even broader field of Artificial Intelligence (AI), which is ultimately concerned with understanding intelligent behaviour on all levels.

Swarm intelligent algorithms rely on the multiple interactions of the individuals in the system. Some amount of information that has been gathered locally may be communicated in each interaction. Consequently, information from one portion of the system may be transmitted to another. Multiple local interactions also allow for events to be coordinated on a local scale. In an ad-hoc network, each node may coordinate local routing information with neighbours during each communication.

Another perspective is to insist that swarm intelligence requires a large population of participating individuals. Because individual behaviours are often randomly chosen, it is necessary to have many interactions in order to reliably determine the information necessary to decide on a proper course of action. Having many individuals will increase the interaction rate and make locally sensing system parameters more reliable. Natural SI systems, such as social insects, often contain anywhere from tens to millions of individuals.

Stigmergy is a method of indirect communication through effects on the environment of the behaviour of an individual, which another will use in deciding what to do, it is the use of an outcome of previous work to guide current work. This term was first introduced by Grasse in 1959 in order to describe a communication mechanism of termites engaged in nest construction. For instance, while ants can communicate directly by feeling each other with their antennae, they also share information via effects on their shared environment. This is often done through the use of an excreted volatile chemical compound called pheromone. Different types of pheromone are used, varying concentrations of which will elicit different behaviours. Another example is with bees, who may decide to gather nectar or store food based on statistical information on the availability of food or food storers.

III. LITERATURE SURVEY

Some of the algorithms used for routing in ad-hoc networks are destination-sequenced distance vector routing, wireless routing protocol, ad hoc on-demand distance vector routing and dynamic source routing protocol [6][7]. The algorithm

presented in this paper has the following difference compared to existing ones:

- All the above-mentioned algorithms have a lot of overhead involved as they have to transfer their routing tables to other nodes over the network. They either transfer them on a time-based approach or event-based approach. This problem does not exist with the termite as there is no need for the transfer of the routing tables in the network.
- Some of the currently used algorithms do not support multiple paths and hence there is no possibility of load balancing in case the optimal path is heavily congested. The termite algorithm supports a generation of multiple paths and it also supports load balancing if the optimal path is congested.
- The above algorithms also require special packets for the purpose of the route maintenance. This is not the case with the termite algorithm where the route maintenance is done by the data packets themselves and the termite algorithm is more robust and scalable than the current algorithms due to inherent behaviour as seen in biological insects.

IV. TERMITE ALGORITHM

A simple example of the hill building behaviour of termites provides a strong analogy to the mechanisms of termite routing algorithm. Consider a flat surface upon which termites and pebbles are distributed. The termites would like to build a hill from the pebbles. A termite is bound by these rules:

- 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.
- Each termite may carry only one pebble at a time.
- If a termite is not carrying a pebble and it encounters one, the termite will pick it up.
- If a termite is carrying a pebble and it encounters one, the termite will put the original pebble down. The pebble will infuse with a certain amount of pheromone.

Analogous to the example above, each network node is a termite hill. Termite carrying pebble is the data packet and termite without pebble is the control packet. Pheromone is laid on the communications links between nodes. Packets are biased towards strong pheromone gradients. While laying source pheromone along the same trial increases the likelihood of packets following that reverse path to the source. This is positive feedback. In order to prevent old routing solutions from remaining in the collective network memory, pheromone decay is introduced as negative feedback [8].

A. Pheromone Table: Each node maintains a table tracking the amount of pheromone on each neighbour link. The table may be visualized as a matrix with neighbour nodes listed along the side and destination nodes listed across the top. Rows correspond to neighbours and columns to destinations. An entry in the pheromone table is referenced by P_n, d where n is the neighbour index and d denotes the destination index. In other words, P_n,d is the amount of pheromone from node d on the link with neighbour n.

B. Pheromone Update: When the acknowledgement packet arrives at a node, the pheromone for the source of the packet is incremented by a constant, γ. The nominal value of γ is one. Only packets addressed to a node must be processed. A node is said to be addressed if it is the intended next-hop recipient of the packet.

$$P_{rs}=P_{rs}+1 \qquad \text{----------(1)}$$

Equation 1 describes the pheromone update procedure when a packet from source s is delivered from previous hop r.

C. Pheromone Decay: To account for pheromone decay, each value in the pheromone table is periodically subtracted by the 20% of the original pheromone. A high decay rate will quickly reduce the amount of remaining pheromone, while a low value will degrade the pheromone slowly. The nominal pheromone decay interval is one second, this is called the decay period. Following equation describes the pheromone decay:

$$P_{n,d}=P_{n,d}-P_{n,d}*2 \qquad \text{----------(2)}$$

D. Routing: Upon arrival to a node b, an incoming packet with destination d is routed randomly based on the amount of d's pheromone present on the neighbour links of b. A packet is never forwarded to the same neighbour from which was received p. If node b has only one neighbour, i.e. the node from which the packet was just received, the packet is dropped.

E. Route Discovery: Route request (RREQ) packets are sent when a node needs to find a path to an unknown destination. A route request (RREQ) packet selectively flooded through the network until it reaches the destination node. If a route request (RREQ) cannot be forwarded when it is dropped, Route request is performed to the maximum number of trials. If the RREQ packet is not able to deliver the destination, then the destination is declared as not reachable. Once the route request packet is received by the destination node, a route reply (RREP) packet is constructed and returned to the requestor. The RREP message is created such that the source of the packet appears to be the requested destination and the destination of the packet is the requestor. The route's reply packet revises the pheromone table based on the number of hops in the trail from the source to destination. This type of update takes care of the multiple paths and favours load balancing. The reply packet is routed by following the trail content of the route request packet. Intermediate nodes on the return path will automatically discover the requested node.

F. Route Maintenance: Route maintenance is done by the data packets. If the data packets are not sent through the discovered path then the path will be lost by the decaying process. This algorithm makes use of the reactive approach to discover the path since decay rate is high. This module used to extract the information from the packet and is used for updating the routing take and for maintaining the path. The data is extracted at the destination and given to the node. The intermediate node calculates the best path to send the data packet. The best path is calculated by the node having the largest pheromone for the destination entry in the pheromone table and decay route is the module used to remove the bad solution from the memory of the router by periodically decaying the routing table.

G. Route Failure Handling: If the link fails, it will be detected by the periodic sending of "hello" packets. If the hello reply is not obtained within the timeout period (2 Sec) then that nodes neighbour entry is deleted from the pheromone table. Hence, during the next route request process newer path is generated. If all the entries in the destination column reached the minimum pheromone then that destination field is removed from the pheromone table thinking that node has moved to the

different location. Hence every node will maintain the path for the nodes which are actively involved in the process of routing.

V. DESIGN

The network that is considered in this work consists of n nodes. It is required to send data packets from source node S to Destination node D, where S ϵ n and D ϵ n. Links connecting the nodes can be wired or wireless. The metric of optimization is the number of hops i.e. the optimal distance between S and D will be the one with a minimum number of hops. A network topology is established by sending the hello packets. These hello packets are broadcasted in case of a wireless node. The node which is in the transmission region receives the hello packets and replies through the hello reply packet. The node which is replied through the hello reply is added to the neighbouring node list and is initialized to value 1 in the pheromone table. Round trip time of hello packet is considered as the cost for the neighbouring link. When the packet is received from the unknown source, an entry to the destination field is created. Due to the dynamic nature of the routing table, it is implemented as a two-dimensional array of linked list. Three types of nodes are identified header node, the root node, and pheromone node.

A. Header Node: It consists of a field name, next pheromone and next header. Name is used to store the IP address of the node. Next Pheromone is used to store the pheromone values of the node pointed by the header node. Next header is the pointer to the next header node used to store another node details.

B. Root Node: It contains name, time, row and column, Name: same as the root node. Time: This is of type time_t used to store the last accessed time of the routing table. This field is used to proper decaying of the routing table. Row and Column: These are type struct header_node. Column is a pointer to the neighbouring nodes, row is a pointer to the destination nodes.

C. Pheromone Node: This structure is used to store the pheromone value. It contains a pheromone, row, and bottom. Pheromone: this field stores the content of the pheromone value. Row and Bottom: this is a pointer to the pheromone node which is used to traverse the pheromone table.

D. Routing Table Effective Decay: Current application needs the routing table to be decayed periodically as it is derived analogy from the termite pheromone trail that will be evaporated in the environment as time passes. Effective decay is applied to the application as a substitution to periodic decay. That is, decay is done only when it is accessed – effective decay. This has reduced an extra thread to the application and overheard of synchronization of this table between these threads. Effective decay is implemented using timestamps. The difference between two access times is used to find the period to which it has to be delayed.

E. Packets: Packets are used for sending the data to the other nodes. These packets are also used for the maintenance of the route by increasing the pheromone concentration of the link. No special packets are needed for route maintenance. The different packet formats used are a) Data Packet contains type, source address, destination address, previous hop, next hop, data length, data, and trail. b) Route request/ Reply/ DataAck contains type, source address, destination address, previous hop, next hop, and trail. c) Hello/ Hello Reply contains type, source address, destination address, and timestamp.

- Type: It is used for defining the type of the packet.

  Type=0 hello packet type=1 hello reply packet
  Type=2 route request packet (RREQ) type=3 route reply packet (RREPLY)
  Type=4 Data Packet type=5 Data acknowledgement packet
- **sourceAddr:** It is used to store the IP address of the source node and destAddr is used to store the address of the destination node.
- **prevHop:** This field stores the source and nextHop is used to store the destination.
- **msgId:** This field is used to eliminate the duplicate packets and dataLen is used to store the length of the data packets and data field stores the data content.
- **Trail:** This field is of type character array, used to store the path that packet has traversed.
- **Timestamp:** This field is used to measure the round trip time of the hello packets.

VI. MODULES

The termite algorithm for Mobile Ad Hoc network is designed into three modules viz. Route discovery module, Route maintenance module, and Route failure handling.

A. Route Discovery Module: Route discovery is responsible for generating a route between source and destination. Working of this module is as follows:

- At the source node, create RREQ packet and sends it through the neighbours in increasing order of cost value.
- Any node that receives RREQ does the following:
  if current_addr=dest_addr
  {
  Construct the RREPLY packet and copies the trail content of RREQ packet, send RREPLY packet to node prevHop from which it has received RREQ and updates the pheromone table according to the hop count.

  }

  Else

  {

  Add the current node address into the trail field and forward the RREQ packet to the neighbour nodes.

  }

- At any node, when it receives RREPLY it does the following:
  If current_addr=dest_addr
  {
  Update the pheromone table according to the hop count and retrieve the packet from the data queue and insert into a normal queue.
  }
  Else
  {
  Update the pheromone table according to the hop count and forward the packet to the neighbour following the trail content
  }

B. Route maintenance module: Route maintenance module is responsible for the maintenance of the path generated during the discovery phase.

- At each node, when it receives a data packet, it does the following
  If current_addr=dest_addr
  {
  Extract data and set type=ack in data packet, remove the data content and send acknowledge packet to prev_id.
  }
  Else
  {
  Get pheromone values of all links using neighbour table, compute the probability for all nodes in neighbour table and send packets to that link which has the highest probability.
  }
- Decay the table whenever accessed. If the pheromone value=0.1 for any destination then delete the destination entry from the routing table.

C. Route failure handling module: This module is responsible for generating alternative routes in case the existing route fails. At any node, if acknowledgement packet is not obtained within the timeout period (10secs) then destination field is deleted from the routing table and fresh route request has been carried out. This new route request obtains the newer path and is maintained by the data packet.

CONCLUSION

In this work, termite routing algorithm was studied, it is a novel adaptive routing algorithm technique for data networks based on mobile agents. Nodes in a mobile ad hoc network involve a high degree of mobility, therefore the network topology may change frequently. If the topology changes and the optimal path may change, then the packets take alternate paths discovered by the route discovery phase. The algorithm provides multipath routing, hence favours load balancing. Crashing a node can be handled by the dynamicity of the algorithm. Also, the loss of termite can be handled by using a timeout mechanism. Therefore, it can be inferred that the commercial implementation of this algorithm may be feasible. It can even be considered its use in large networks, such as the internet, as a future option.

Acknowledgment

First of all, I am glad to thank THE LORD ALMIGHTY for giving me the spirit in completing this paper. I would thank my family for the constant support they provided throughout my preparation.

References

[1] Andrew S. Tannenbaum, "Computer Networks", 4th Edition, Prentice- Hall of India
[2] E. Bonabeau, M. Dorigo and G. Theraulaz, Swarm intelligence: from natural to artificial systems, Oxford University Press, 1999.
[3] T. White, "Swarm intelligence and problem-solving in telecommunications", Canadian Artificial Intelligence Magazine, Spring 1997.
[4] G. Di Caro and M. Dorigo, "Mobile agents for adaptive routing", Proc. 31st Hawaii International Conference on System Sciences, IEEE Computer Society Press, Los Alamitos, CA, pp. 74-83, 1998.
[5] Schoonderwoerd R, Holland O, Bruten J, Rothkrantz L. "Ant based load balancing in telecommunications networks, Adaptive behaviour Hewlett-Packard Laboratories, Bristol- England, pp 162 – 207, 1996.
[6] Nader F Mir, "Computer and Communication Networks", Pearson Education, 2007.
[7] Liang S, Zincir Heywood A N, Heywood M I, "The effect of Routing under local information using a Social insect Metaphor", IEEE International Congress of Evolutionary Computation, pp, 1438 – 1443, May 2002.
[8] Martin Roth and Stephen Wicker, "Termite: Emergent Ad-Hoc Networking", Wireless Intelligent Systems Laboratory School of Electrical and Computer Engineering Cornell University Ithaca, New York 14850 USA.