ADAPTIVE ROUTING ALGORITHM FOR MANET: TERMITE

[1]Sharvani G S, [2]Cauvery N K, [3]Dr.Rangaswamy.T

Department of Computer Science & Engineering, RVCE, Bangalore, India
[1]sharvanim.@yahoo.com, [2]cavi_giri@yahoo.co.in
[3] Department of IEM, RVCE, Bangalore, India
dr.tmrswamy@rediffmail.com

Abstract

Mobile Ad hoc network consists of a set of mobile nodes. This network does not have any infrastructure or central administration, hence it is called infrastructure less network. As the nodes are mobile, it is very difficult to find the path between two end points. This paper presents a solution for finding path between nodes in mobile ad hoc network. Termite is an innovative algorithm for packet routing in communication networks. Termite is an adaptive, distributed, mobile-agents-based algorithm which was inspired by recent work on the ant colony metaphor. According to this algorithm, a group of mobile agents (or artificial termites) build paths between pair of nodes; exploring the network concurrently and exchanging obtained information to update the routing tables. Some of the parameters used to evaluate its performance are packet delays and throughput. The results of this algorithm shows better throughput as compared to existing algorithms. So, Termite algorithm is a promising alternative for routing of data in commercial networks.

KEYWORDS

Routing algorithm, Swarm intelligence, Mobile agents, Pheromone.

1. INTRODUCTION ON MOBILE AD HOC NETWORK

Mobile Ad hoc network is self configuring network of mobile hosts connected by wireless links, the union of which forms the topology of the network [1]. The advantages of ad hoc networks are the convenience (no central administration), mobility, productivity, deployment and expandability. As the nodes in the network are mobile, the topology of network changes unpredictably. Hence it is difficult to generate path between two nodes. This paper deals with the development of on-demand ad-hoc network routing which can achieve load balancing for packet switched network. The 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 ant algorithm [4][5] is through interaction of network exploration agents called ants. According to this algorithm, a group of mobile agents builds path between pairs of nodes by exchanging information and updating routing tables.

2. 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 the 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 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 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 the inherent behavior as seen in biological insects

3. Description of Termite algorithm

A simple example of the hill building behavior 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:[8]

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 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 trail 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].

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

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. Equation (1) describes the pheromone update procedure when a packet from source s is delivered from previous hop r. $P_{rs} = P_{rs} + 1$ ----(1)

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. Equation (2) describes pheromone decay. $P_{n,d} = P_{n,d} - P_{n,d} * 0.2$ ---- (2)

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 neighbor links of b. A packet is never forwarded to the same neighbor from which was received, p. If node b has only one neighbor, i.e. the node that the packet was just received from, the packet is dropped.

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 cannot be forwarded, it is dropped. Route request is done up to maximum trials. If the RREQ packet is not able to deliver the destination then the destination is declared as not reachable. Once a 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 reply packet updates the pheromone table based on the number of hopes in the path from the source to destination. This type of update takes care about the multiple paths and favors load balancing. The reply packet is routed by following the trail content of the route request packet. Intermediate nodes on the return path automatically discover the requested node.

Route Maintenance: Route maintenance has been 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 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 of routing table and maintaining the path. At the destination the data is extracted and given to the node. In the intermediate node calculates the best path to send the data packet. A 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.

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 (2sec) 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 path for the node which are actively involved in the process of routing.

3. Design

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 number of hops i.e. the optimal distance between S and D will be the one with minimum no of hops. Network topology is established by sending the hello packets. These hello packets are broadcasted in case of wireless node. The node which is in the transmission region receives the hello packets and replies through the hello reply packet. The node which are replied through the hello reply are 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 a entry to the destination field is created. Due to the dynamic nature of the routing table it is implemented as two dimensional array of linked list. Three types of nodes are identified header_node, root_node and pheromone_node.

header_node: It consists of fields name, nextpheremone and next header. Name- is used to store the IP address of the node. Nextpheromone - this is a pointer to the pheromone_node which is used to store the pheromone values of the node pointed by header node. Nextheader - this is the pointer to the next header_node used to store the another node details.

root_node: It contains name, time, row and column. name: same as root node. time: This is of type time_t used to store the last accessed time of the routing table. This is field is used to

proper decaying of routing table. row and column: These are of type struct header_node. Column is a pointer to the neighbouring nodes, row is a pointer to the destination nodes

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

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 overhead 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 decayed

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. a) 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 time stamp

- *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 from where the packet has been received and *nextHop* is used to where the packet has to be transferred.

- *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

4. 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.

Route discovery module: Route discovery is responsible for generating route between source and destination. Working of this module is as follows:
- At source node create RREQ packet and sends through the neighbors in the increasing order of cost value
- At any node, when it receives RREQ, it 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 update the pheromone table according to the hop count   }
  else
  {   add the current node address into the trail field & 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 normal queue.   }
else
{   update the pheromone table according to the hop count and forward the packet to
    neighbour following the trail content   }

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 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 neighbor table, compute probability for all
        nodes in neighbor table and send packets to that link which has 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.

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 (10 s) 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 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 route discovery phase. The algorithm provides multipath routing hence favors 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 time out 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 internet, as a future option.

Future Enhancement

The application developed can be further enhanced to include following features:
• It can utilize different QoS (Quality of Service) parameters in order to determine the
    optimal path.
• It can be made to route the packets of audio, video application.
• It can be made to transfer the packets of larger length by fragmenting at the sender side and
    defragmenting at the receiver side.

References

[1]   Andrew S Tannenbaum, "Computer Networks", 4th Edition, Prentice-Hall of India

[2]   E. Bonabeau, M. Dorigo, and G. Théraulaz, 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 Behavior Hewlelt-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

F.A.Sharvani.G.S has completed her M.Tech from Visveswaraya technological University in the year 2008 in the field of Computer Network and Engineering. She is pursuing her Ph.D in Avinashi Lingam University for women; Coimbatore .She has industrial experience of 2.5 in Macrosoft Ltd and 4.6 Years of academic experience in R.V.C.E. She is a member of CSI and ISTE. Her area of research includes Storage Area Networks, Artificial Intelligence and cloud computing

.B Cauvery N.K has completed her ME from Bangalore University in the year 1999 in the field of Computer Science and Engineering. She is pursuing he Ph.D from VTU, Belgaum. She has an academic experience of 10 years in R V College of Engineering, Bangalore. She has two publications in International Conference. She is a member of CSI and ISTE. Her area of research include Swarm Intelligence routing and Genetic Algorithms

C T. Rangaswamy is working as Prof in the Department of Industrial Engineering & Management. He obtained his PhD from Bangalore University, PGDM from IIM Bangalore, DIIT from IIT Madras and B E in Mechanical Engineering from UVCE, Bangalore University. He has worked in public sector undertakings BEML and HAL and has Industrial Experience of 10 years. He has been teaching for the past two decades. He is member of several academic bodies. He has worked in the World Bank project (TEQIP) as nodal officer. He has conducted techno cultural symposium, carrier guidance programs and state level work shop on Academic autonomy of TEQIP institutions. He has attended several seminars, work shops at state and national level at NIFM Faridabad and MDI Gurgaon. His area of interest is Management of change.