AntHocNet Routing Algorithm for Mobile Ad Hoc Networks

Arockia Panimalar.S [1], Prateeksha.S.R[2], Ansu Rodrigues[3], Selshya.S[4]

[1] *Assistant Professor, Department of BCA & M.Sc SS, Sri Krishna Arts and Science College, Coimbatore, India*
[2,3,4] *III BCA, Department of BCA & M.Sc SS, Sri Krishna Arts and Science College, Coimbatore, India*

Abstract - *The paper provides a new routing algorithm named AntHocNet for Mobile Ad hoc Networks (MANETs). It is a kind of networks where all nodes are mobile and they communicate with each other via wireless connections. It is based on analysis of AntHocNet routing algorithm and Ant Colony Optimization Routing (ACOR) algorithm. It consists of both reactive and proactive components. In a reactive path setup phase, multiple paths are set up between the source and destination of a data session, and during the course of the communication session, ants proactively test existing paths and explore new ones. The main aim is to design an ant based routing algorithm for Ad hoc network. The work is completed partially and still new additions are required to make it a complete routing protocol.*

***Key Words*: Ad hoc, MANETs, AntHocNet, Routing Protocol**

1. Introduction

MANETs are a cluster of mobile nodes in communication without using fixed infrastructure. Figure 1 shows the Real ant's search for food from Source (S) to Destination (D). Forward Ants (FA) search for food from S to D whereas, Backward Ants (BA) searches from D to S. When ants travel they discharge a chemical called pheromone. They find the shortest path using the high probability of pheromone. FA and BA are control packets used to amend the routing tables and disperse information about the traffic in the network.

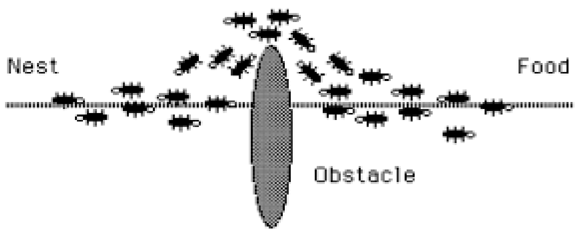

Fig 1: Ants take the shortest path from source to destination

In recent years several ant based routing protocols are implemented, such as Probabilistic Emergent Routing Algorithm (PERA), AntHocNet, and Ant-Colony-Based Routing Algorithm (ARA). Ant based routing algorithm for MANETs can react to dynamic changes and solves problems locally. It finds the shortest path from S to D. It provides high number of routes, reliable connections, and controls updating and broadcasting. It combines a Reactive Route Discovery (RRD) [1] with proactive route maintenance and recovery process. Routing data is saved in pheromone tables and routing tables. In RRD [2] the source node creates a reactive forward ant. This is a control packet that has a goal to find a path from S to an assigned D. Ant contains the address of S and D. It proceeds through the network. It collects a list of nodes it has visited from S to D. If routing data is available, the node chooses a next hop for the ant probabilistically, based on the different pheromone values associated with next hops for D. At the destination the reactive FA is transformed into the BA, if at least one copy of the FA is received. The first Ant is accepted and converted in to BA. Remaining copies are discarded [3]. Proactive path maintenance serves to update and extend the availability of routing data. It has two sub processes, pheromone diffusion and proactive sampling. Pheromone diffusion periodically disperses quality of information. This field of pheromone is indicated in the virtual pheromone values in the pheromone tables of the nodes.

Section 2 briefs about the various routing algorithms.
Section 3 gives the approach of implementing a new routing protocol and
Section 4 concludes the paper.

2. Various Routing Approaches

A. Demand Routing Approach

Demand routing approach for multi hop Ad hoc networks is proposed. It is based on swarm Intelligence and Ant Colony Optimization (ACO) metaheuristic. Route discovery generates optimal path for data packets. Path maintenance exhausts periodically and retains optimal path when the topology is changed in Ad hoc Networks. Existing routes may fail or new paths are generated.

B. Ant Routing Algorithm

Ant routing algorithm for mobile Ad hoc networks (ARAMA) is proposed. It solves the routing problem in MANET and it is a dynamic routing algorithm with controlled overheads in the network.

C. Ant Based Ad hoc Routing Protocols

Ant Based Ad hoc Routing Protocols is the best solution for the routing algorithm for solving the routing problems in

MANETs with controlled overheads. This algorithm can use different quality of service parameters in optimization. Exploring and tables updating is done on demand, which minimize the overheads. This technique provides the redundant paths between source and destination.

D. AntHocNet and ACO routing algorithm on MANETs

AntHocNet and ACO routing algorithm on MANETs are discussed. It associates both reactive setup and proactive maintenance. In simulation tests authors prove that AntHocNet has a performance gain over the Ad hoc On-Demand Distance Vector (AODV) routing protocol. Increase in Delivery ratio, average end-to-end delay, and average jitter are observed.

3. Implementing a New Protocol for NS2

Initially, we create a current sample MANET routing protocol called Antrouteproto. Consider that Antrouteproto is a proactive routing protocol that requires sending out some control packets periodically. In the routing table, information is stored in each entry in the form of destination and next hop addresses. Then we configure the Antrouteproto routing protocol similar to AntHocNet. We add and modify the delay table in Antrouteproto in header and source files.
Antrouteproto routing protocol is compiled successfully but when we run a test simulation the packets are not successfully transmitted. The trace file is generated but the routing table is empty throughout the simulation. The same simulation file works on AntHocNet, AODV, and Dynamic Source Routing (DSR). Various modules of implementation and changes are shown as below.

A. File Structure

1Antrouteproto.h defines all needed timers and routing agents that perform protocol's functionality.

2. Antrouteproto.cc implements all timers, routing agent and Tcl hooks.

3. Antrouteproto_pkt.h defines all packets. Antrouteproto protocol needs to exchange among nodes in the MANET.

4. Antrouteproto_rtable.h defines our own routing table.

5.Antrouteproto_rtable.cc shows the routing table implementation.

6. Delay_table.h defines generation of our own delay table.

7. Delay_table.cc is delay table implementation.

B. Class Structure

This is the main class to implement the routing protocol. Agent's shows endpoints at the network layer. Agent class connects with Tcl script, which is adequate to handle our routing protocol. We create an agent to implement a routing protocol by inheriting from Agent class of ns2.34. Routing agent consists of internal state and a routing table. Internal state maintains a new class or as a collection of attributes. Routing table is a new class, Antrouteproto_rtable. Timer Class is everything related to timing like some periodic updating of routing table or discarding any routes after some time using timer class. Trace Class is writing to a trace file about what happened in the simulation. Debug function is used to print a debug messages.

C. Packet Types

This defines the packet structure. Packet class stores all the defined packets as an array of unsigned chars where packets fields are saved. To access our defined packet, we define in our header, an offset property which will find where the packet is.

D. The Routing Agent

A new Antrouteproto class having attributes and functions is defined. Header files included packet header, Agent base class, Packet class, and Timer Handler base class. A port classifier object is a node that consists of an address classifier and a port classifier. Port Classifier class, used for passing packets up to upper layers, recv to call whenever agent receives a packet either from upper layer agent like UDP or TCP or from some other node. A Port classifier object consists of an address classifier and a port classifier. Another important attribute is the Trace object. It is used to produce logs to be store in the trace file.

E. Timer

It has expire method performing two tasks
i. Sends a new control packet.
ii. Reschedule the timer itself

F. Constructor

The constructor implementation and calling the constructor is started as a base class passing PT_ANTROUTEPROTO as an argument. It is used to classify control packets sent and received by this routing agent. We bind accessible_var as a Boolean attribute that can be read and written from Tcl. To bind this variable as an integer, we must use the bind function instead of bind bool. We save the address of agent using the accessible_var_ from Tcl interface. Antrouteproto obtained from Agent base class, has two functions recv and command. recv is called at any time the agent receives a packet. This can occur when the node itself is generating a packet or when it is receiving one from another node. The command function is invoked from Tcl.

recv_Antrouteproto_pkt method

The recv_Antrouteproto_pkt is invoked by recv when a Antrouteproto packet is received and gets the IP header and the Antrouteproto's header as defined in Antrouteproto_pkt.h. RT_PORT as defined in common/packet.h is the routing port with value 255. We check the source and destination port to be RT_PORT process. These packets according to protocol specifications release the resources Packet::free (p).

send_Antrouteproto_pkt method

It is called by custom timer function expire. When it expires we allocate the packet using allocpkt function. This function is defined for all agents.

reset_Antrouteproto_pkt_timer method

It consists of packet sending timer. This can reschedule itself.

forwarddata ()

The forward data function chooses whether a packet has to be delivered to the upper layer agents or to be forward to other node. In case if packet is broadcasted, then next hop will be field accordingly. The implementation returns IP_BROADCAST when there is no route to destination address and we drop the packet with a debug message

G. The Routing Table

The routing table can be implemented as a different class or as any other data structure. For each entry in routing table, there exists information to store like destination address, next hop, cost, etc. etc. Here, we are storing only the destination address and the corresponding next hop in the routing table using Hash Table. The routing table class determines the routing table according to the need and any data structure can be used to define functions like add entry, rm_entry, clear or print.

H. The Delay Table

We can implement the delay table in Antrouteproto routing protocol as a different class or any other data structure. For each entry in delay table, there might be packet information like destination address, source address, flow id and delay.

4. Conclusion

In this paper we have presented some of the steps in the design of the ant based routing protocol for ad hoc networks. The routing protocol is based on the Anthocnet routing protocol. The implementation is not complete but we have presented some of the changes required to different files. By simulations we analyze that the Antrouteproto can create the routing table and the trace file is generated, but routing table is printed empty throughout the simulation.

5. Future Enhancement

In future, the routing table for Antrouteproto routing protocol can be implemented using delay table without empty routing table. The work includes making Antrouteproto a hybrid routing protocol which reduces the overhead of Anthocnet.

6. References

[1] Hussein, Osama H., Tarek N. Saadawi, and Myung Jong Lee. "Probability routing algorithm for mobile ad hoc networks' resources management."

[2] Gunes, Mesut, Udo Sorges, and Imed Bouazizi. "ARA-the ant-colony based routing algorithm for MANETs." International Conference on. IEEE, 2002.

[3] Al Maghayreh, Eslam, et al. "Bees_ Ants Based Routing

Algorithm." International Conference on Intelligent Systems, Modelling and Simulation.

[4] Garg, Dweepna, and Parth Gohil. "Ant Colony Optimized Routing for Mobile Ad Hoc Networks (MANET)." International Journal of Smart Sensors and Ad Hoc Networks

[5] Hussein, O., and T. Saadawi. "Ant routing algorithm for mobile ad-hoc networks (ARAMA)."

[6] Ladhake, S. A., and Shradha A. Thakare. "A review: ant based ad hoc routing protocols." International Journal of Emerging Technology and Advanced Engineering .

[7] Di Caro, Gianni, Frederick Ducatelle, and Luca Maria Gambardella. "AntHocNet: an ant-based hybrid routing algorithm for mobile ad hoc networks." International Conference on Parallel Problem Solving from Nature. Springer Berlin Heidelberg, 2004.

[8] Ros, Francisco J., and Pedro M. Ruiz. "Implementing a new manet unicast routing protocol in ns2." Dept. of Information and Communications Engineering University of Murcia (2004).