ISSN:2229-6093

Arun Biradar et al , Int.J.Computer Technology & Applications,Vol 5 (1),25-30

Intelligent Unipath Routing for MANET Using Genetic Algorithm

Arun Biradar[1] and Dr. Ravindra C. Thool[2]
[1]*East West Institute of Technology, Bangalore*
[2]*SGGS Institute of Engineering & Technology, Nanded*
E-mail: ambiradar@yahoo.com

Abstract

In a mobile ad-hoc network (MANET), the topology of the network may change rapidly and unexpectedly due to mobility of nodes. To support mobility to user generally low configured nodes are in use, so limited resources and link variation are the issues with MANET. Thus, setting up router that meet high reliability is a very challenging issue. Another important problem in MANET is the energy consumption of nodes. When the energy of a node is depleted, it stops working, thus link break.
Hence it is very important to find a route that has sufficient energy level and high stability, and so can obtain reliable routing and data transmission.
Genetic Algorithm (GAs) promise solution to such complicated problems they have been used successfully in various practical applications. Areas in which Genetic Algorithms excel is their ability to manipulate many parameters simultaneously. The use of parallelism enables them to produce multiple equally good solutions to the same problem.

Keywords: Mobile Ad hoc Network (MANET), Genetic Algorithm (GA), link life time (LLT), Route Error (RERR), Route Request (RREQ).

1. Introduction

A Mobile Ad hoc Network (MANET) is a self-configuring (autonomous) system of mobile routers (and associated hosts) connected by wireless links -the union of which form an arbitrary topology. The routers are free to move randomly and organize themselves arbitrarily; thus, the network's wireless topology may change rapidly and unpredictably. Such a network may operate in a standalone fashion, or may be connected to the larger Internet operating as a hybrid fixed/ad hoc network.

A routing protocol is the mechanism by which user traffic is directed and transported through the network from the source node to the destination node.
Here is the basic routing functionality for mobile ad- hoc networks:

Path generation which generates paths according to the assembled and distributed state information of the network and of the application;

· Path selection which selects appropriate paths based on network and application state information;

· Data Forwarding which forwards user traffic along the selected route;

· Path Maintenance which is responsible for maintaining the selected path.

2. Routing Protocols in MANET

All Routing is the act of moving information across an internetwork from a source to a destination. Along the way, at least one intermediate node typically is encountered. For searching of path, a number of MANET routing protocols are available as given below.

A. Table Driven or Proactive Routing Protocols

In this type of protocols, each node maintains individual routing table for every node in the network containing routing information. Each node maintains current routing information by listening and sending control messages periodically among the nodes which update their routing tables. The drawback of this protocol large overhead. Example, DSDV, OLSR.

B. On Demand or Reactive Routing Protocols

In this type of routing protocols, whenever a node wants to send packets to a destination, it starts the route discovery mechanisms to find the route to the destination. It does not maintain any previous route information for long time and no need to maintain up-to-date routing information as in proactive routing protocol. This reduces overhead. It has more delay. Example DSR, AODV, AOMDV.

C. Hybrid Routing Protocols

Hybrid routing protocol, combines the advantages of both proactive and reactive routing protocols. The route is initially established with help of routing tables in which route path is already available and then serves the demand from additionally activated nodes through reactive flooding. Example ZRP, TORA.

3. Genetic Algorithm

The genetic algorithm is a subset of evolutionary algorithms that model biological processes to optimized highly complex cost functions. I. Rechenberg introduced the idea of evolutionary computing in 1960s in his work .Evolution strategies. [1, 8]. Many researchers developed his idea. Genetic algorithms were invented by John Holland over the course of the 1960s and 1970s and finally popularized by his students and colleagues [8]. As simulating the survival of the fittest among individuals over consecutive generation for solving a problem, each generation of the genetic algorithm consists of a population of character strings that are analogous to the chromosome that we see in our DNA. Each individual represents a point in a search space and a possible solution. The individuals in the population are then made to go through a process of evolution. This is motivated by a hope, that the new population will be better than the old one during the evolution.

The basic genetic algorithm includes the following steps:

1. Start: Define the optimization parameters, the cost function, and the cost.
2. Generate random initial population of *n* chromosomes or produce suitable solutions for the problem.
3. Fitness: Evaluate the fitness *f(x)* of each chromosome *x* in the population
4. New population: Create a new population by repeating the steps from 3- 6 until the new population is complete.

(a) Selection: Select two parent chromosomes from a population according to their fitness. The better the fitness the higher the chance it gets to be selected.

(b) Crossover: With a crossover probability cross over the parents to form a new offspring. If no crossover is performed, then the offspring is an exact copy of the parents.

(c) Mutation: With a mutation probability mutate the new offspring at each locus, which is each position in the chromosome.

(d) Accepting: Place the new offspring generated in the new population.

5. Replace: Use new generated population for a further run of the algorithm
6. Test for convergence: If the end condition is satisfied, stop the execution and return the best solution in the current population.
7. Loop: Go to step 2 again.

4. Design of GA Based Intelligent Routing Protocol

To present a genetic algorithmic approach variable length chromosomes and their genes have been used for encoding the problem, the crossover operation exchanges partial chromosomes at appositionally independent crossing site and mutation operation maintained the genetic diversity of the population. The objective of a 'GA' in our work is to assist in the Qos routing by doing some processing in the information.

It selects a routing path based on two following parameters.

1.Stability of paths from the source node to the destination node. For calculating stability of a path we use the link life time (LLT) between two connected nodes using mobility information of nodes obtained by global positioning system (GPS).

2. Residue energy of paths found from the source node to the destination node. For determining residue energy of a path, we find node with minimum residue energy of a path. Then we use three parameters: the route life time, the residue energy and the number of hops to select a routing path with high stability, sufficient energy level and low latency , and consequently high quality.

The way GA works is by combining existing routes which share a common node and destination in order to generate new routes and selecting the routes which are more efficient in terms of achieving the QoS goal. This processing is done only at the source node. Here the following figure 1 represents the design model of our work.

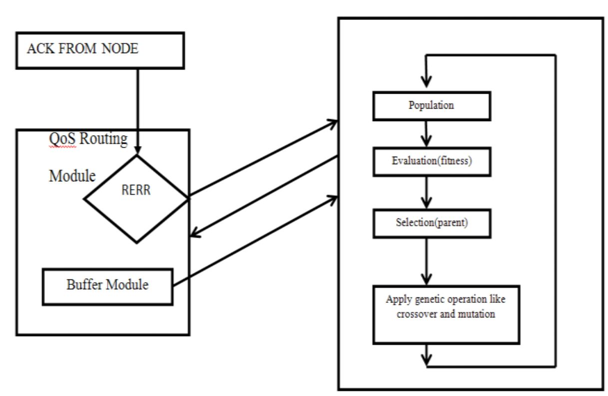


Figure 1. Design model of a GA based QoS Routing

When the Qos Routing module receives a request from a program to send data to a given destination, it starts by checking the routing table, if a path is already in buffer (routing table) then it follows it, but if none is available then it fires a smart agent to try to discover the new route to the destination just as of any traditional on-demand routing protocol.

When the acknowledgement (ACK) packet – Route Reply (RREP) is sent back to source (Src) it follows the reverse route and update the routing table of IN if needed, but ACK can be in form of Route Error (RERR), the RERR message is sent whenever a link break causes one or more destination to become unreachable from some of these nodes neighbors.

So when RERR is available to Src then GA module is utilized. Here, by providing the available entries in buffer as the initial population, and applying evaluation function (fitness function) which shows availability of destination by fulfilling QoS requirements.

The data exchange operations between the QoS routing module and the GA module is bidirectional. As the Routing module passes measurements to the GA module and in return the GA module updates the module's route repository with the paths that have the fitness at that time. Whenever there is Route Error (RERR), the GA module is always the initiator of the data exchange so that the kernel module does not need to keep track of the presence or absence of the GA module to route packet traffic.

Whenever there is RERR GA module provides new route to QoS module and it is stored in buffer of QoS module, and if there is no RERR then path from buffer is followed . For the sake of efficiency, at some periodic time, route from buffer is again evaluated by GA module to manage modularity concepts, the QoS routing module only has a minimum knowledge of the internals of the GA module and data is exchanged in the form of simplified individual (paths).

As data exchange with the QoS Routing module is done through a buffer, the exchange format consists of a header bearing timestamp, source node, number of nodes and a byte for flag followed by a succession of node measurement pairs.

The data structure sorting the pair of nodes which constitute the hop and the corresponding parameter is referred to as gene. The GA individuals each correspond to a route and their internal representation of this route is a linked list of pointer to genes. The individuals are arranged in subpopulation, one for each destination and within these subpopulations they are periodically ordered by fitness. This is done just before the subpopulation are checked for size, so that when the size limit is reached the less efficient paths are the ones that get dropped.

The diversity in the individual population is highly desirable both in order to have a set of alternative routes. If the current route become saturated and in order to enhance the chances of crossover producing interesting new individuals. The GA module treats path back by intelligent route in a different fashion.

So using GA based approach intelligent routing find alternate solution which is better at that time and using crossover function of GA we are having alternate path.

A. Link Life Time:

The link life time prediction is a method ; in-order to be employed each mobile device should be equipped with a GPs receiver for obtaining its longitude and latitude. Using this geographical information and considering the network area map we can determine the position of each node. For calculating the nodes direction and speed, the position information of them should be updated continuously. Assuming two mobile nodes A and B are within the radio transmission range of each other let us consider;

(X_A, Y_A): coordinate of a mobile node A;

(XB, Y_B) : coordinate of mobile node B;

V_A: mobility speed of mobile node A;

V_B : mobility speed of mobile node B;

I_A : direction of motion of mobile node A (0 <IA <2π);

I_B : direction of motion of mobile node B (0 <IB <2π);

Using the afore mentioned parameters, we can define the link life time equation as follows (Rappaport, 1995);

$$LLT = \frac{-(ab+cd)+\sqrt{(a^2+c^2)r^2-(ad-cb)^2}}{(a^2+c^2)} \quad \text{-----------} (1)$$

Where, the link life time is calculated at each hop during the route request packet is traversing the path. Each node calculates the life time of the link between itself and previous hop. If node A is the previous hop of the packet for node B, it appends its position and the movement information to the route request packet. When node B receives this packet, it calculates the life-time of the link A -> B.

B. Route discovery:

When a source node wants to transmit some data to a destination node, it needs an active route from itself to the destination. If the source node has no active routing information regarding the destination, it initiates the route discovery process. Like AODV, our protocol neither maintains any routing table nor exchange routing table information periodically. When a source node requires a communication, it broadcast a route request (RREQ) packet to its neighboring nodes until they reach to the destination node. In our protocol, each RREQ packet records information of all links- status and nodes along traversed. Link-status information is delivered from the source node to the destination node. The destination node may receive link-status information from different RREQ packets : it means that there are different feasible paths. Finally, required computation for route selection is accomplished at the destination node and result is backward to the source node in order to decide route.

We define RREQ packet format as follows:
The type of packet:
S : The source node address;
D : The destination node address;
SEQ : The packet sequence number;
PMI : The position and the movement information of mobile nodes that consists of location, velocity and direction;
RLT: Minimum of life time that RREQ packet traversed them;
RME: Minimum of residue energy level of nodes that RREQ packet traversed them;
HC: Hop count;
TTL : The maximum hop length of the path is constructing ;

C. Route discovery process

Representation of Network state: To start we should describe four parameters. Calculate the LLT, LLT denotes the link life time between two nodes. Each mobile node identifies its position and movement information. First , when a source node broadcasts a RREQ packet , it appends its position and movement information (such as location, velocity and direction) to the PM1 field of RREQ packet. Upon a node receives the RREQ packet, it calculates the link life time by using PM1 field of the RREQ packet and its own position and movement information.
Next, calculate RLT i.e. the minimum link life time along a routing path. Therefore, the RLT is equal to the minimum of LLTs for a route.
Next calculate the RME i.e. the minimum of the residue energy level of nodes along a routing path.
Hop count (HC): is number of links of a path.

D. Collection of information about the route:

First source node S sets up TTL to number of the entire nodes in the network. Then it broadcasts a request packet to its neighboring nodes.

Secondly, when the intermediate node $N_t$receives the RREQ, it updates some fields of the RREQ packet such as PM1, RLT and RME as earlier explained. Then it decrements the TTL value and increments hop count. If the node $N_{t\ is}$ not zero, then it forwards the RREQ packet to all value is not zero, then it forwards the RREQ packet to all its neighboring nodes. If TTL value is zero, it drops the RREQ packet and does and does not reforward that to any node.

E. Evaluate the fitness function:

If node N receive RREQ packet and it is destination node, it first appends the RREQ packet information to its own routing table and after that uses the following equation to calculate route quality factor. i.e. R

$$R = \frac{RLT * RME}{HC} \quad \text{------------------------ (2)}$$

It appends this value to the route reply (RREP) packet and backwards the RREP packet to the source node through same path. The destination node does this operation for all received RREQ packets. The R value is the main parameters for route selection.
Clearly, in order to maximize the R value, RLT and RME should be maximized and HC should
be minimized as possible. RREP packet include the following fields;
The type of packet:
S : Source node address;
D : Destination node address;
SEQ : The packet sequence number;
R : The ratio of path;
TTL : The limitation of hop length.

4. Simulation and Result

The entire simulations were carried out using NS-2 /Qualnet network simulator.
The prime goal of NS-2 is to support Research and Education in networking . It is suitable for designing new protocols, comparing different protocols and traffic evaluations.

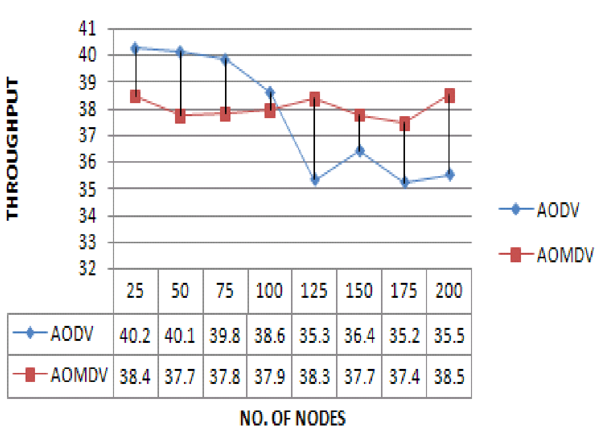
QualNet is a commercial software that runs on all common platforms (Linux, Windows, Solaris, OS X) and is specialized in simulating all kind of wireless applications. It has a quite clear user interface compared to other solutions while also offering an easy to use command line interface.

A. Performance Metrics

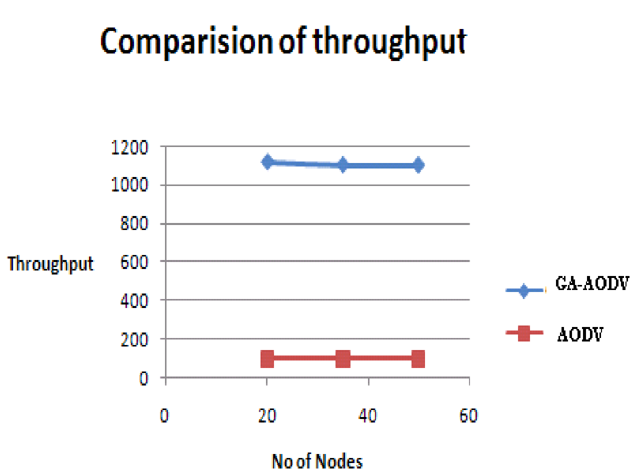
We evaluate the following performance metrics:

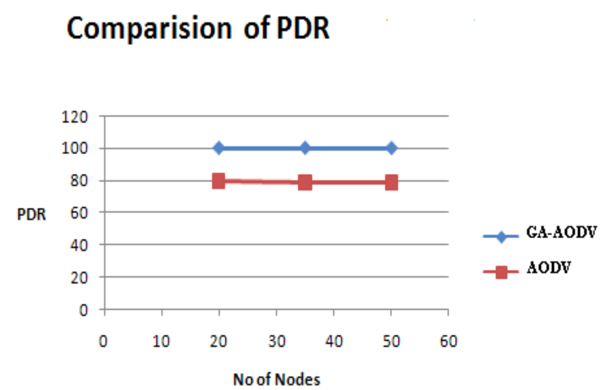
1.Packet delivery fraction — ratio of the data packets delivered to the destination to those generated by the CBR sources; or a related metric *received*
2. throughput in Kb/sec received at the destination;
3. Average end-to-end delay of data packets —this includes all possible delays caused by buffering during route discovery, queuing delay at the interface, retransmission delays at the MAC, propagation and transfer times;
4. *Normalized routing load* — the total number of routing packets "transmitted" for each delivered data packet. Each hop-wise transmission of these packets is counted as one transmission.

Comparison of AODV and GA-AODV



	25	50	75	100	125	150	175	200
AODV	40.2	40.1	39.8	38.6	35.3	36.4	35.2	35.5
AOMDV	38.4	37.7	37.8	37.9	38.3	37.7	37.4	38.5





5. Conclusion

A number of research efforts have attempted to design selforganizing mobile networks based on Genetic Algorithms. Great deals of successful research in the field of Mobile Ad Hoc Networks have been inspired by Genetic Algorithms. Yet, we believe genetically inspired mobile ad hoc networking still has much room to grow. In particular, there are great opportunities in exploring a new approach.
We have proposed a reliable QoS routing protocol which bases on the route life – time that is obtained using mobility information and residue energy and low latency using genetic algorithm.

6. References:

[1] D. E. Goldberg, *Genetic algorithms in search, Optimization and Machine Learning*, Addison Wesley, 1989.
[2] E. M. Royer and C. K. Toh, *A review of current routing protocols for ad hoc mobile wireless networks*, IEEE Personal Communications, 6 (1999).
[3] C. Siva Ram Murthy and B. S. Manoj, Ad Hoc Wireless Networks Architectures and Protocols, 2009, Pearson Education.
[4] Perkins CE, Belding-Royer E, Das SR. Ad hoc on-demand distance vector (AODV) routing, http://www.ietf.org/rfc/rfc3561. txt, July 2003.
[5] Shahram Jamali1*, Bita Safarzadeh2 and Hamed Alimohammadi, SQR-AODV: A stable QoS-aware reliable on-demand distance vector routing protocol for mobile ad hoc networks, Scientific Research and Essays Vol. 6(14), pp. 3015-3026, 18 July, 2011
[6] Arun Biradar, Dr. Ravindra C. Thool, Performance Evaluation of Mobile Ad-hoc Networks Routing Protocols to Employ Genetic Algorithm, 3rd World Congress on Information and Communication Technologies (WICT 2013), 15-18 December 2013, Hanoi, Vietnam, IEEE.

[7] Arun Biradar, Dr. Ravindra C. Thool, Reliable Genetic Algorithm Based Intelligent Routing for MANET,World Congress on Computer Applications and Information Systems, WCCAIS'2014, Hammamet, January 17-19, 2014.
[8] D. Lawrence, *Handbook of genetic algorithms*, Van Nostrand Reinhold, New York, 1991.
[9] The network simulator – ns-2, http:// www.isi.edu/nsnam/ns
[10] Qualnet Developer's Guide Book, www.ee.iitb.ac.in/prakshep/Qualnet

About Author's:

Arun Biradar has received B.E, M.Tech. in Computer Science & Engineering and Pursuing Ph.D. in Computer Science & Engineering. His main research interests are Wireless Ad-hoc Networks, Computer Networks and Genetic Algorithms.

Dr. Ravindra C. Thool received his Ph. D. in Electronics and Computer Engineering. He is working as a Professor and Head, Department of Information Technology, Shri Guru Gobind Singhji Institute of Engineering and Technology (SGGSIE&T), Nanded, Maharashtra, India. He is a member of IEEE, ISTE and ASABE (USA). His main research interests are Wireless Ad-Hoc Networks, Computer Networks, Genetic Algorithms and Computer Vision.