Improving Network Performance in DSR using Genetic Algorithms

Haridas Kataria[1], Ravinder Singh Sheoran[2]

[1] *Lecturer CSE, Govt. Polytechnic for Women, Sirsa,*
[2] *Lecturer CSE, Govt. Polytechnic for Women, Sirsa,*

***Abstract* – Genetic algorithms (GA's) uses the best approach with a structured yet randomized information exchange to form a search algorithm. High overhead involved in flooding while route creation is a limiting factor of Dynamic Source Routing (DSR) protocol applied for mobile ad hoc networks. During data delivery, it seems that DSR provides best route if the route is not long enough. The GA's efficiently exploiting the historical information to speculate on new search points with the improved performance. GA's have been applied on two parameters in routing in the network hop count and queue length i.e. number of packets queued up on router's lines waiting for transmission or forwarding. For better performance of any network path which is selected for transmission should be with lowest hop count and shortest queue length [8].**

***Keywords* – GloMoSim, hop count, queue length, ECHO packets, fitness function, throughput.**

I. INTRODUCTION

Routing is a fundamental engineering task on the internet which mainly consists of finding a path from source to the destination host. Routing is complex in large networks because of many potential intermediate destinations a packet might traverse before reaching its final destination. To decrease the complexity, network is divided into some smaller domains as considering each domain individually makes the network more manageable. Routing is the function of network layer [6] & the routing algorithm is that part of network layer software responsible for deciding which output line an incoming packet should be transmitted on. Routing algorithms can be grouped into two major categories: Non-Adaptive and Adaptive [3].

i. Non-Adaptive algorithms, also called static routing, do not base their routing decisions on measurements or estimates of current traffic and topology. Instead the choice of the route to use to get from source to destination is computed in advance off-line and downloaded when network is booted.
ii. Adaptive algorithms, also known as dynamic routing, change their routing decisions to reflect changes in the topology as well as traffic. These algorithms differ in where they get their information from, when they change the routes, what metric is used for optimization.

Today's computer network generally uses dynamic routing algorithms rather than the static ones. Two dynamic algorithms in particular, "Distance Vector Routing" and "Link State Routing", are the most popular. Distance vector routing algorithms operates & maintains a table which give the best known distance to each destination and which line to use to get there. These tables are updated by exchanging information with the neighbour routers. In distance vector routing technique, the router measures the distance directly with help of special ECHO packets that the receiver just timestamps and sends it back. All the routers should carry out this process simultaneously, regardless of the router being used or not, all of them connected in the network have to transmit the ECHO packets. Thus apart from the normal data packets, the ECHO packets must also be sent by the router in order refresh its routing table.

The Genetic Algorithms are search algorithms based on mechanics of natural selection and natural genetic. These are the direct, parallel, stochastic methods for global search and optimization, which imitates the evolution of the living beings, described by Charles Darwin (1859). GA's are part of the group of Evolutionary Algorithms [1]. The evolutionary algorithms use three main principles of the natural evolution: "*reproduction", "natural selection"* and "*diversity of the species"*, maintained by the differences of each generation with the previous one.

Genetic Algorithms work with a set of individuals, representing possible solutions of the task. The selection principle is applied by using a criterion, giving an evaluation for the individual with respect to the desired solution. The Section I give the introduction, Section II represents the proposed algorithm, Section III represents the results and analysis and finally Section IV concludes the work done.

II. PROPOSED ALGORITHM

The implementation of proposed technique (GA's) is presented by explaining the type of encoding, reproduction and recombination operators of the genetic algorithm used. The proposed process consists of the followings:

A. Implementation of Genetic Algorithm

For implementation of Genetic algorithm, first of all it is decided the encoding, which is to be used i.e. how chromosomes are represented and used to store data for routing packets and crossover operator and the technique used for crossover in routing while maintaining the feasibility of paths [14,15,20]. By feasible paths means that the paths which actually exist and intermediate routers, which are there in the path, one after another are connected to each other. The terms used in the algorithm are explained as follows:

i. Encoding: Encoding represents transformation of solved problem to N-dimensional space of real numbers. Encoding used here, is list of nodes for reaching from source to destination. The size of chromosomes is not fixed and is variable because number of nodes in a route is not fixed and can vary from two to total number of nodes in the network. The maximum size is obviously equal to total number of nodes. The structure of chromosomes is list of nodes in a sequence in which they are used to reach the destination.

ii. Population: Population is initialized using simple DSR in the beginning and the size of population is taken to be a maximum of number of nodes in the network. Population of chromosomes is initialized by sending normal route request packets to the destination and only after the number of route replies exceeds two, and then GA is applied on them.

iii. Crossover: Crossover used in the problem is one point crossover and after the parents are selected randomly for crossover, it is checked whether they have some node in common. If they have some node in common then only crossover is applied, otherwise crossover is not done on those chromosomes. After selection of common node the parts after that node are swapped and two new offsprings are created and stored in the population.

iv. Hop Count: It is the number of intermediate routers between source and destination nodes.

v. Queue Length: It is the number of packets queued up on router's lines waiting for transmission or forwarding.

vi. *Fitness Function:* Fitness function or objective function is that function which needs to be optimized i.e. either minimized or maximized according to the problem. Here, Fitness function depends on hop count and queue length.

Fitness function=1 / (hop count + queue length)

As fitness function is inversely proportional to the sum of hop count and queue length, if for any path the number of nodes between source and destination is large hop count is also large and so the value of fitness function is small. Similarly a large queue length means lesser value of fitness function.

vi. Reproduction: For selection of chromosomes for next generation, roulette wheel selection is used. The chromosomes with high value of fitness function have more probability of selection than the chromosomes with lesser value of fitness. So those chromosomes are selected more times whose hop count and queue length both are low.

B. Genetic Algorithm Used

The Genetic Algorithm is applied on the DSR protocol. Route reply module is altered mainly so that all replies are handled by genetic algorithm. When a node has data to be send to another node, it checks its route cache table to check for the presence of any path to that node. If path is present it is used for sending packet otherwise a route request is sent to all neighbours. If neighbours have a path to that node, they send that information to the source node else they forward the packet. This process is repeated until either a path is found or the destination is reached. If the packet reaches the destination, it see that the route request is for it only so it collect the path information from the packet and sends back a reply to source telling the route followed by the request path as route to it. Destination may receive multiple route request packets by multiple links. It sends back all replies to the source. After source receives the replies sent by the destination, the genetic algorithm comes into picture. For the time number of replies is less than two, simple reply handling procedure is carried out. After the number of replies exceeds one genetic algorithm is applied, selection of chromosomes for next generation is done using the recombination technique. Crossover is then applied to selected chromosomes as per the crossover probability. Fitness is then calculated based on the hop count values and queue length. For the next generation, which are more fit i. e. whose fitness is more are selected more number of times for next generation. This process is repeated for many times unless the solution converges to a best solution. After the convergence the route whichever is best is selected and copied to the route cache table and then that route is used as a path from source to destination.

C. Simulation Tool Used

The simulation eases the analyzing and the verification of the protocols, mainly in large-scale systems and offers flexible testing with different topologies, mobility patterns, and several physical and link-layer protocols. There are two well-known simulators that are used for distance vector routing simulations:

i. NS-2
ii. GloMoSim

Here GloMoSim is used, as it provides a scalable environment especially to simulate large wireless networks. It supports thousands of nodes, using parallel and distributed environment.

GloMoSim Description

GloMoSim (**Glo**bal **Mo**bile information systems **Sim**ulation) [14,15,19,21] is designed as a set of library modules, each of which simulates a communication protocol in the protocol stack. The library uses the OSI layer approach and supports multiple protocols in each layer.

GloMoSim provides a scalable simulation environment for large wireless and wired communication networks. GloMoSim uses a parallel discrete-event simulation capability provided by Parsec. Parsec is a C-based simulation language, developed by the Parallel Computing Laboratory, for sequential and parallel execution of discrete-event simulation models. It can also be used as a parallel programming language

GloMoSim simulates networks with up to thousand nodes linked by a heterogeneous communications capability that includes multicast, asymmetric communications using direct satellite broadcasts, multi-hop wireless communications using ad-hoc networking, and traditional Internet protocols. The specification lists the GloMoSim models at each of the major layers.

Structure of GloMoSim: After downloading GloMoSim-“2[1].03.tar.gz.tar” the files will be extracted from it and obtain a folder GloMoSim-2.03 which contains the following directories [3,15,16]:

- /application contains code for the application layer.
- /bin for executable and input/output files.
- /doc contains the documentation.
- /include contains common include files.
- /java_gui contains the visual tool.
- /mac contains the code for the MAC layer.
- /main contains the basic framework design.
- /network contains the code for the network layer.
- /tcplib contains libraries for TCP.
- /transport contains the code for the transport layer.

From the above directories, the “*main*” directory is important. Directory *main* contains the basic framework to execute GloMoSim, which includes “driver.pc” file that defines the driver entity. Every Parsec program must include an entity called *driver* which serves a purpose similar to the main function of a C program. This file reads the configuration file, initiates the simulation and writes the final statistics results file “*glomo.stat”* from some temporary file “(.STAT.x)”. The sequence of events at run time is as follows:

i. The main function in “*driver.pc*” is run. This is the C main function, where GloMoSim starts.
ii. The main function calls parsec main () to start the Parsec simulation engine, initialize the simulation runtime variables and create the driver entity. The “*parsec main”* function is used when the user wants to write his own *main* and is found at “*PCC DIRECTORY/include/pc api.h”* (since the function is part of the Parsec runtime system, it is not possible to access the source for it).
iii. When the simulation ends, parsec main () returns, and the rest of the main function is executed.

III. RESULTS AND ANALYSIS

Since the system was developed to operate in the environment of the network simulator, the evaluation was carried out with the same tool by developing several experiments that illustrate the performance of the system. The simulation parameters like number of nodes, terrain range etc. that use GloMoSim as the basic tool for development and evaluation.

Various parameters are analyzed for the evaluation of the network.

1. Experimental Setup

The experimental setup describes the strategy with all the parameters of network used for simulation. The parameters taken for the study are given below:

Table 1: Simulation Parameters

Parameter	Value	Description
Simulation time	10M	Maximum execution time
Terrain Dimensions	1200, 1200	Physical area in which the nodes are placed
Number of Nodes	20-250	Nodes participating in the network
Traffic Model	CBR	Constant Bit Rate link used
Node Placement	Uniform	Node placement policy
Mobility	0-20 (m/s)	Speed of node with which they are moving
MAC-Protocol	CSMA	MAC layer protocol used
Routing Protocol	DSR	Routing protocol used

The values of simulation parameters are used to examine the performance of the network. These parameters are available in the “*config.in*” file present in the GloMoSim. The values can be adjusted according to requirements in this file. After adjusting the values in this file, this file is executed. An output file “*glomo.stat*” is used to check the various parameters to analyze the performance of network.

2. Performance Evaluation Metrics

The following performance metrics are chosen to evaluate the impact of the genetic algorithm on the performance of DSR.

- *Packet Delivery Ratio (PDR):* It is the ratio of number of data packets actually sent to the data packets actually received by the destination. It is selected to evaluate the percentage of delivered packets without and with attack mechanism. This is the best parameter to evaluate the performance of a network.

$$\text{PDR (\%)} = \frac{\text{Total number of Packet Received}}{\text{Total number of Packet Sent}}$$

- *Average End to End Delay:* The delay is the total latency experienced by a packet to traverse the network from the source to the destination. At the network layer, the end-to-end packet latency is the sum of processing delay, packetization, transmission delay, queuing delay, and propagation delay. The end-to-end delay of a path is the summation of the node delay at each node plus the link delay at each link on the path.

- *Throughput:* Throughout is the dimensional parameter which is defined as percentage of successful transmission attempts. Throughput is calculated as:

Throughput = Packet received / Simulation time

- *Hop Count:* It is the number of intermediate routers between source and destination. Path with lower hop counts are preferred over paths with more hop count. It is selected to evaluate the fitness of the route.

- *Queue Length:* The number of packets queued up any line of a router waiting to get forwarded to some line other than it came from is called queue length. Routers with larger queue lengths take more time for processing packets and packet delay is increased. So a path with low average queue length is given more priority over the high queue length route.

3. Evaluations of Results

The genetic algorithm is applied to DSR. Then the effect of applying GA on DSR is analyzed. Here the performance of network is analyzed by changing the various parameters like number of nodes, queue length etc. Then the effect of genetic algorithm is on performance of DSR is analyzed.

- *Hop Count v/s throughput:* Figure1 shows that as number of hops between sender and receiver increases average throughput decreases. For less number of nodes between sender and receiver the number of packets transmitted per unit of time is more than if number of hops is less that is both are inversely proportional to each other.

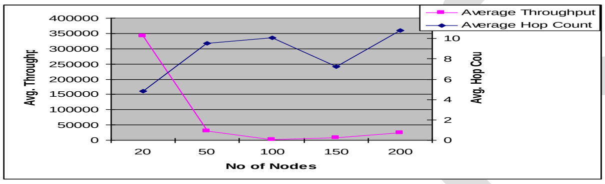

Figure1: Hop Count v/s throughput

- *Throughput with varying Number of Nodes:* As shown in figure2 the addition of genetic algorithm to DSR improves the performance of DSR by increasing its throughput. As the number of nodes increase the DSR with GA has more throughput than the DSR without GA. For lesser number of nodes the performance of both algorithms i.e. with and without using GA is nearly same, the difference comes when number of nodes is more than 50.After 50 numbers of nodes the performance of DSR with GA keeps on getting better. So it can be concluded that GA is helpful in case of more number of nodes.

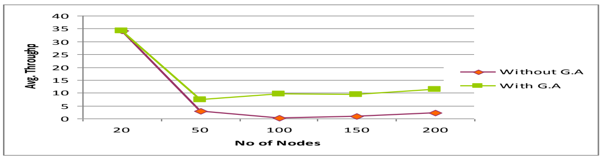

Figure2: Average Throughput vs No. of Nodes

- *Hop Count with varying Number of Nodes:* In any network as the number of nodes increase the average hop count of routes selected will increase. As shown in figure3 as the number of nodes increase the difference between average hop count for with GA and without GA increase. For less number of nodes the average hop count is nearly same in both cases, but as the number of nodes increase the average hop count of DSR with GA decrease that is in this case the route with less hop count are generated and selected for transmission of packets.

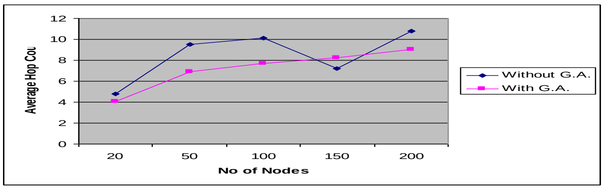

Figure3: Average Hop Count vs No. of Nodes

- *Average Queue Length And Average Throughput Vs Number Of Nodes:* As can be seen from figure4 throughput is more for less number of nodes because number of nodes are less which make the path length less and thus increases throughput.

For queue length also for lesser number of nodes queue length is less because less number of nodes will transmit fastly on shorter paths. It is also seen from the graph that as the queue length increases the throughput decreases. It is because for longer queues most of the packets to be transmitted are queued up waiting to be transmitted and thus more time needed for processing packets on intermediate routers and hence the decreased throughput.

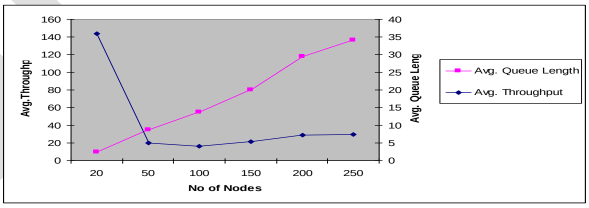

Figure4: Average Throughput and Average Queue Length

- *Average Throughput Vs Average Queue Length:* Figure5 shows the comparison of throughputs between DSR with GA and DSR without GA. As the average queue length is less the throughput is high whereas for more number of packets in queues average throughput is less.

In DSR without GA the throughput is less than in case of DSR with GA. It means that GA decreases the probability of selection of a route with greater average queue length whereas in case DSR without GA the route with more average queue length more number of times for transmission before selecting the best possible path.

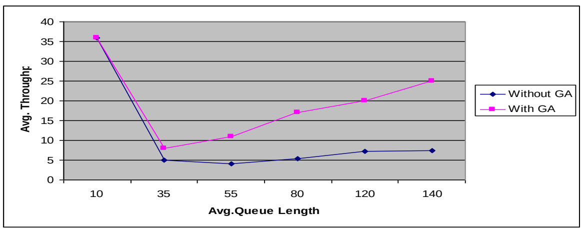

Figure5: Average Queue Length Vs Average Throughput

IV. CONCLUSION

Genetic Algorithms are used to solve the problems of network with large search space and which are difficult to solve. It concludes that the GA's have many advantages and these are:

(i) Genetic Algorithms have the capacity of solving complicated solutions in practical systems.
(ii) Hardware implementations of GA are very fast and it is not sensitive to network size.
(iii) GA's are very flexible as quality of solution can be adjusted as a function of population size.
(iv) GA hardware scale well to network that may not even fit in memory.
(v)

So due to the above reasons GA offers very promising solutions to many problems. Here, an explanation of GA's, their implementations, Distance Vector Routing technique and their advantages and disadvantages were presented. Then, various encoding, crossover and reproduction operators to be used in the proposed system were discussed. It is found that by using GA in DSR performance of network has improved if number of nodes in the network is large. For lesser than 100 nodes the performance of simple DSR is either better or equal to DSR with GA.

REFERENCES

[1]. A, Selamat and M. H. Selamat, "Routing algorithm of mobile agents for query retrieval using genetic algorithm," Malaysian Journal of Computer Science, Vol. 17 No. 2 December 2004.

[2]. C. W. Ahn and R. S. Ramakrishna, "A genetic algorithm for shortest path routing problem and the sizing of populations" IEEE Transaction on Evolutionary Computation, VOL. 6, NO. 6, December 2002.

[3]. C. Davies and P. Lingras, "Genetic algorithms for rerouting shortest paths in dynamic and stochastic networks," European Journal of Operational Research, vol.144 pp. 27–38, 2003.

[4]. H. Zlatokrilov and H. Levy, "Navigation in distance vector spaces and its use for node avoidance routing"

[5]. H.Huang, G. Chen, F.C.M. Lau and L. Xi, "A distance vector routing protocol for networks with unidirectional links," HKU CSIS Tech report TR-99-03, March 1999.

[6]. J .J. G. L Aceves, "A unified approach to loop free routing using distance vectors or link states," ACM, vol.89, No. 9, pp.212-223, 1989.

[7]. L. Zhu, R. L. Wainwright and D. A. Schoenefeld, "A genetic algorithm for the point to multipoint routing problem with varying number of requests," O-7803-4869-9/98 IEEE, 1998

[8]. M. Gen, A. Kumar and J. R. Kim, "Recent network design techniques using evolutionary algorithms," International Journal of Production Economics vol.98 pp.251–261, 2005.

[9]. M. R. Masillamani, A. V. Suriyakumar, R. Ponnurangam, and G.V.Uma, "Genetic algorithms for distance vector routing technique," AIML 06 International Conference, pp.160-163, Sharm El Sheikh, Egypt.15 June 2006

[10]. T. Wan E.Kranakis and P.C. van Oorschot," Securing the destination-sequenced distance vector routing protocol (S-DSDV)," *Proceedings of the 6th International Conference on Information and Communications Security*, LNCS vol. 3269, pp.358-374. Malaga, Spain. October 27-29, 2004.

[11]. T. Wan, E. Kranakis and P.C. van Oorschot, "S-RIP: A secure distance vector routing protocol," *Proceedings of Applied Cryptography and Network Security* (academic track), Yellow Mountain, China.. LNCS vol. 3089, pp.103-119. Springer-Verlag, June 8-11 2004

[12]. Z. Xu , S. Dai and J .J. G.L. Aceves," A more efficient distance vector routing algorithms," 0-7803-4249-6/97 IEEE,1997

[13]. Niaz Morshed Chowdhury, Syed Murtoza Baker and Ershadhul H. Choudhury "Ant Navigation-based Adaptive Routing Technique for Mobile Ad hoc Networks ", 2006.

[14]. H. Kumar, "Effectiveness of Distance Vector Routing Using Genetic Algorithm", Dissertation, Computer Dept., Chaudhary Devi Lal University, Sirsa, 2011.

[15]. K. Sunita, "Performance Analysis of Distance Vector Routing Using Genetic", Dissertation, Department of Computer Science & Engineering Guru Jambheshwar University of Science & Technology, Hisar, 2007.

[16]. M. Gen, R. Cheng, D. Wang, Genetic algorithms for solving shortest path problems, in: Proceedings of 1997 IEEE International Conference on evolutionary Computing, 1997, pp. 401–406.

[17]. J.H. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, 1975.

[18]. B.P. Buckles, F.E. Petry, Genetic Algorithms, IEEE Computer Press, Los Alamitos, CA, 1994

[19]. J. Inagaki, M. Haseyama, and H. Kitajima, "A genetic algorithm for determining multiple routes and its applications," in *Proc. IEEE Int. Symp. Circuits and Systems*, 1999, pp. 137–140.

[20]. Y. Leung, G. Li, and Z. B. Xu, "A genetic algorithm for the multiple destination routing problems," *IEEE Trans. Evol. Comput.*, vol. 2, pp. 150–161, Nov. 1998.

[21]. Z. Xiawei, C. Changjia, and Z. Gang, "A genetic algorithm for multicasting routing problem," in *Proc. Int. Conf. Communication Technology (WCC-ICCT 2000)*, 2000, pp. 1248–1253.

[22]. Q. Zhang and Y.W. Leung, "An orthogonal genetic algorithm for multimedia multicast routing," *IEEE Trans. Evol. Comput.*, vol. 3, pp. 53–62,Apr. 1999.

[23]. H. Pan and I. Y. Wang, "The bandwidth allocation of ATM through genetic algorithm," in *Proc. IEEE GLOBECOM'91*, 1991, pp. 125–129.