Routing Optimization using Genetic Algorithm in Ad Hoc Networks

Mustafa AL-GHAZAL Hajr for Technical Education and Training Co., Al Ahsa, KSA. must@hajr.net Ayman EL-SAYED (IEEE Member) Faculty of electronic engineering, Menoufiya University, EGYPT ayman.elsayed@free.fr Hamedy KELASH Faculty of electronic engineering, Menoufiya University, EGYPT

Abstract -- An Ad-Hoc network is a collection of wireless mobile nodes forming a temporary network without the aid of any established infrastructure or centralized administration. The topology of connections between nodes in ad hoc networks may be quite dynamic. Ad hoc networks require a highly adaptive routing scheme to deal with the frequent topology changes. In this paper we propose algorithm for improving routing in clustering algorithm based on both clusterhead gateway switching protocol (CGSR) and the mechanisms of a genetic algorithm (GA). We use GA's because GA mechanisms allow for self-configuration systems and maintain state information about the neighboring network better than traditional MANET routing mechanisms. GA mechanisms allow a node to change routing information quickly and efficiently to adjust an ever changing local topology, initiating fewer link breakages and increasing lower MAC layer overhead. Also our proposed algorithm shows that GA's are able to find, if not the shortest, at least a very good path between source and destination in ad-hoc network nodes.

I. INTRODUCTION

A mobile ad hoc network (MANET) is an autonomous network that consists of mobile nodes that communicate with each other over wireless links. This type of networks is suited for use in situations where a fixed infrastructure is not available, not trusted, too expensive or unreliable. A few examples include: a network of notebook computers or PDAs in a conference or campus setting, rescue operations, and headquarters industry. In the absence of a fixed infrastructure, nodes have to cooperate in order to provide the necessary network functionality. Routing is one of the primary functions each node has to perform in order to enable connections between nodes that are not directly within each others send range. The development of efficient routing protocols is a non trivial and challenging task because of the specific characteristics of a MANET environment [1, 2]:

- Due to node movements, the network topology may change randomly and rapidly at unpredicted times.
- The available bandwidth is limited and can vary due to fading, noise, interference.
- Most mobile devices are battery powered; therefore energy consumption plays an important role.

In *ad–hoc* networks nodes geographically close to each other are grouped into non overlapping sub networks, clusters. Each cluster has a leading node called the clusterhead and a number of cluster members. When a

cluster member wants to communicate with another node, a route is provided by its clusterhead. A crucial question is which node will become a clusterhead. Typically a clusterhead is more burdened than its members and could easily become a bottleneck of the system if not chosen appropriately [1,2].

Hence solutions to this problem are based on heuristics approaches. A good clustering scheme should preserve its structure as much as possible, when nodes are moving and/or the topology is slowly changing. Otherwise, recompilation of clusterheads and frequent information exchange among the participating nodes will result in high computation overhead. Any node can become a clusterhead if it has the necessary functionality, such as processing and transmission power. Nodes register with the nearest clusterhead and become members of that cluster. Clusters may change dynamically, reflecting the mobility of the underlying network [1,2].

The rest of the paper is organized as follows: section II deals with routing in ad hoc networks. Section III introduces the genetic algorithm as an optimization technique. Section IV introduces Network Modeling and Constraints. The proposed technique is presented in section V. section VI analyzes simulation results. Conclusions are summarized in section VII.

II. ROUTING IN AD-HOC NETWORKS

a. Proactive/Reactive Ad Hoc Routing Protocols

The existing *ad hoc* routing protocols can be broadly classified into the following two categories [3,4]:

Proactive protocols (e.g. WRP or wireless routing protocol): by broadcasting control packets containing routing table information (e.g. distance vector), these protocols attempt to maintain at all time up-to-date routing information from each node to every node.

Reactive protocols (*e.g.* AODV or Ad hoc on-demand distance vector routing): only when a route to destination is required, a node initiates a route discovery process. Once a route has been established, it is maintained by a route maintenance procedure until the route is no longer desired.

Unfortunately, these protocols suffer from a number of shortcomings: scalability problems with growing network size and their performance is only optimal under certain

network conditions (mobility, network load, network topology).

b. Clusterhead Gateway Switch Routing Protocol

Clusterhead gateway switch routing protocol (CGSR) in Figure 1 is based on dynamic destination sequenced distance vector (DSDV) which belongs to proactive routing protocols [5]. CGSR uses a Least Cluster Chance (LCC) algorithm in which a clusterhead chance occurs only when two clusterheads come into one cluster or one of the nodes moves out of the range of all clusterheads. In this algorithm, each node maintains two tables, namely, a cluster member table which records the clusterhead for each destination node and routing table which contains the next hop to the destination. The cluster member table is broadcasted periodically. A node will update its cluster member table when it receives a new one from its neighbors using sequence numbers. To route a packet to a destination, the node first selects the shortest (minimal hop) clusterhead corresponding to the destination from the cluster member table and routing table and then transmits the packet to the next hop according to the routing table entry corresponding to that clusterheads, thus, the routing principle looks as follows: lookup of the clusterhead of the destination node, then lookup of the next hop, packet send to destination, at last destination clusterhead deliver packet [6].



Figure 1: Illustration of single level clustering hierarchy used in CGSR

This approach has numerous disadvantages: *first*, selection of clusterheads causes complexity and overhead, thus degrading performance. *Second*, there are traffic bottleneck and single point failures at the clusterheads and gateways. *Finally*, *CGSR* is hierarchical routing protocol that uses *DSDV* as its underlying routing algorithm but reduces the size of routing update packets in large networks by partitioning the whole network into multiple clusters [6,7].

III. GENETIC ALGORITHMS

The GA, which was introduced by John Holland, was adopted from natural evolution [8, 9]. Natural evolution has the following features:

- 1) The characteristics of an individual are encoded on a chromosome.
- 2) Each chromosome has a certain fitness according to the environment in which it exists.
- 3) Individuals judged stronger are able to survive and produce next generations of strong individuals.

The GA is based on the above features in the following manner: the solution of the problem is encoded on a string comparable with the chromosome of the biological system. The GA keeps a population of randomly selected chromosomes and allows filter chromosomes to combine and produce offspring with new characteristics, which may replace low fitness old chromosomes. This is repeated until we find a chromosome with best characteristics, which represents the optimal solution of the problem. There are two mechanisms that link a genetic algorithm to the problem it is solving. These two mechanisms are:

- 1) Encoding solutions to the problem on chromosomes.
- Evaluation function that returns a measurement of the worth of a chromosome in the context of the problem. This is what we call the fitness of a chromosome. The evaluation function plays the same role in the genetic algorithm that the environment plays in natural evolution.

In order to use GA's for network topological design, the chromosome is chosen to contain the network parameters. A possible chromosome would be a string containing the weights of all nodes of the network. The evaluation function which assigns fitness to each chromosome is chosen according to the objective of the design problem. If the objective is to minimize the route between source and destination, then the evaluation function will compute the all distances of all possible paths between source and destination and give the dynamic optimal path with time change.

IV. NETWORK MODELING AND CONSTRAINTS

The network can formed by node and the links can be represented by an undirected graph G(V,E), where V represents the set of nodes v_i and E represents the set of links e_i . Where the cardinality of V remains the same but the cardinality of E always changes with certain and deletion of links. Clustering can be thought as a graph partitioning problem with some added constraints. As the underlying graph does not show any regular structure, partitions) with respect to certain parameters becomes an NP-hard problem [10, 11, 12]. More formally, we look for the set of vertices $S \subseteq V(G)$, such that:

$$\dot{U}_{v \in S} N [v] = V(G).$$

Here, N [v] is the neighborhood of node v, defined as:-

N [v]=
$$\dot{U}_{v' \in V, v' \neq v} v \{ \text{dist}(v, v') < tx_{\text{range}} \},\$$

Where, tx_{range} is the transmission range of v. This neighborhood of a clusterhead is the set of nodes which lie within its transmission range. The set S is called a dominating set such as that every vertex of G belongs to S or has a neighbor in S. The dominating set of the graph is the set of clusterheads. It might be possible that a node is physically nearer to a clusterhead but belongs to another clusterhead because of the other considerations such as:

- A clusterhead may not able to handle a large number of nodes due to resource limitations even if these nodes are its immediate neighbors and lie well within its transmission range.
- The area with the minimum number of clusterheads will put more burdens on the clusterheads. On the other hand, a large number of clusterheads will lead to a computationally expensive system. Although this may result in a good throughput, the data packets have to go through multiple hops thus implying high latency.
- When the node decides to become a clusterhead or stay as an ordinary node depending on the weights of its one hop neighbors. The node should be wait for all responses from its neighbors to make its own decision to be a clusterhead or ordinary node, so must take in account the time that a node may need to wait to receive responses from its neighbors.

For instance, a node might be physically closer to a clusterhead that is over loaded. In that case it will attach itself to a clusterhead which is far off due to mobility; the nodes may go outside the transmission range of their clusterhead thus changing its neighborhood. However, this does not result in a change of the dominate set. It might so happen that the detached node is not able to attach itself to any of the nodes in the dominant set. This implies that the existing dominant set can no longer cover the entire graph, hence the clustering algorithm overcome this problem and find a new dominant set [10,11,12].

V. THE PROPOSED TECHNIQUE

a. Clusterhead Election Basis

To decide how well suited a node is for being a clusterhead; some considerations must take into account such as degree, transmission power, mobility and battery power. The following features are considered in cluster algorithm [13,14]:

- The clusterhead election procedure is not periodic and is invoked as rarely as possible. This reduces system updates and hence computation and communication costs. The clustering algorithm is not invoked if the relative distances between the nodes and their clusterheads do not change.
- Each clusterhead can ideally support only δ (a predefined threshold) nodes to ensure efficient medium access control (*MAC*) functioning. If the clusterhead

tries to serve more nodes than it is capable of, the system efficiency suffers in the sense that the nodes will incur more delay because they have to wait longer for their run to get their share resource. A high system throughput can be achieved by limiting or optimizing degree of each clusterhead.

- The battery power can be efficiently used within certain transmission range, which means, it will take less power for a node to communicate with other nodes. A clusterhead consumes more battery power than an ordinary node since a clusterhead has extra responsibilities to carry out for its members.
- Mobility is an important factor in deciding the clusterheads. In order to avoid frequent clusterhead changes, it is desirable to elect a clusterhead that does not move very quickly. When the clusterhead moves fast, the nodes may be detached from the clusterhead and as a result, a reaffiliation occurs. Reaffiliation takes place when one of the ordinary nodes moves out of a cluster and joins another existing cluster. In this case, the amount of information exchange between the node and the corresponding clusterhead is local and relatively small. The information update in the event of a change in the dominant set is much more than a reaffiliation.
- A clusterhead is able to communicate better with its neighbors having closer distances from it within the transmission range. As the nodes move away from the clusterhead, the communication may become difficult due mainly to signal attenuation with increasing distance.

b. Clusterhead Election Algorithm

In this section we will discuss the steps of electing clusterheads procedures based on the weight of each node [11,13,14]. Table (1) list the used notations

Step 1: Find the neighbors of each node v which defines its degree dv as:

 $dv = |N(v)| = \sum_{v' \in V \ v' \neq v} \{dist(v, v') < t_range\}.$

Symbol	Definitions
V	Node in clusterhead.
Wv	Weight of each node.
Δv	Degree difference.
Δu	Sum of distances of the members of the
	clusterheads.
Mv	The average speed of the nodes.
Pv	The accumulative of a node being a
	clusterhead.
	Table (1): Notation List.

Step 2: For every node v, compute the degree-difference, $\Delta_v = |\mathbf{d}_v - \delta|$.

Step 3: For every node v, compute the sum of distances, $D_v = \sum_{v' \in N(v)} \{ dist(v, v') \}.$ **Step 4:** Compute the running average of the speed for every node until current time *T* (measure of mobility, Mv), $Mv = 1/T \sum_{t=1..T} [(X_t - X_{t-1})^2 + (Y_t - Y_{t-1})^2]^{1/2}$ where, $(X_t - X_{t-1})$ and $(Y_t - Y_{t-1})$ are the coordinates of the node *v* at time *t* and (t-1), respectively.

Step 5: Compute the cumulative time, Pv, during which a node v, acts as a clusterhead. Pv implies how much battery power has been consumed.

Step 6: Calculate the combined weight Wv for each node v, Where:- $Wv = W_1\Delta_v + W_2\mathbf{D}_v + W_3M_v + W_4P_v$

Being W_1, W_2, W_3 and W_4 the weighing factors for the corresponding system parameters.

Step 7: Choose the node with the smallest W_V as the clusterhead. All the neighbors of that clusterhead can't participate in election procedure.

step 8: Repeat steps 2-7 for the remaining nodes not yet selected as clusterhead or assigned to a cluster.

The first component, $W_I\Delta_v$, contributing towards the combined metric Wv helps in efficient MAC functioning because it is always desirable for a clusterhead to handle up to a certain number of nodes in its cluster.

In the second component, W_2D_{ν} , The motivation of D_{ν} is mainly related to energy consumption. It is known that more power is required to communicate to larger distance. As a result, one might think that it would be more appropriate to use the sum of the squares of the distances, because the power required to support a link increases considerably faster than linearly with distance, (at least in the far-field region). The usual attenuation in the signal strength is inversely proportional to some exponent of the distance, which is usually approximated to "4" in cellular networks where the distance between mobiles and base stations is of the order of "2-3 miles". But in ad hoc networks, the distance involved are rather small (approximately hundreds of meters". In this range, the attenuation can be assumed to be linear.

The third component, $W_3 M_{\nu}$, (M_{ν}) , is due to mobility of the nodes, a node with less mobility is always a better choice for a clusterhead.

The last component, $W4 P_{v}$, $(P_{v)}$, is measured as total cumulative time a node acts as a clusterhead. Once a node becomes a clusterhead, neither that node nor its members can participate further in the cluster election algorithm. The algorithm terminates once all the nodes either become a clusterhead or a member of clusterhead. All the clusterheads are aware of their one-hop neighbors as well as the ordinary (non-clusterhead) nodes know their clusterheads.

c. Routing optimization using a Genetic Algorithm

We discuss the steps of routing optimization. The goal consists of allocate near optimal path from source to

destination based on time, giving priority to clusterheads to maximize utilization and minimum delay [9,15,16].

Step1: Encoding and Initial Population

All nodes in the search space should be present and have a representation. If there is a one-to-one unique correspondence between the search space and string representation, the design of the genetic operator would be considerably less complex. These unique id's are used to encode the chromosome using integer permutation Encoding the individual chromosomes is an essential part of the mapping process; each chromosome contains information about the clusterheads and the members thereof, as obtained from the original clustering algorithm. Each chromosome is represented by a link weight vector W $= \langle w1....w(n) \rangle$ where (n) is the total number of links in the network. The value of each weight is within the range from 1 to MAX WEIGHT. We define the value of MAX WEIGHT to be 64 for reducing the search space. The population size is set to 100, with the initial values inside each chromosome randomly varying from 1 to MAX WEIGHT.

Step2: Fitness Evaluation

Chromosomes are selected according to their fitness. The bandwidth constraint is embedded into the fitness function as a penalty factor, such that the search space is explored with potential feasible solution. The fitness of each chromosome can be defined to be a two-dimensional function as shown in Equation 1. The overall network load (L1) and excessive bandwidth allocated to overloaded links (L2).

Fitness = $f(L1,L2) = \mu / (\alpha x L1 + \beta xL2)$ (1) Where α , β and μ are manually configured coefficients.

L1 and L2 are expressed as shown in equations 2, 3, 4.

$$I = \sum_{g=1}^{G} D_g \tag{2}$$

$$L2 = \sum_{(ij)} W_{ij} x \left(\sum_{g=1}^{G} Dg - C_{ij} \right)$$
(3)

Where

L

$$W_{ij} = \{ \begin{array}{ll} 0 & if \quad \sum^G D_g \\ \text{or} \quad \{ 1 \quad \text{otherwise} \end{array} \right\}$$
(4)

Where:

Dg : Bandwidth demand for *cluster g* on each link;

Cij : Bandwidth capacity of link *(i,j)*;

G: total number of active clusters.

So the objective function is two fields: first chromosomes of the new generations. And second, solutions obtained from the offspring should be feasible in that the total bandwidth allocated flows traveling through each link should not exceed its capacity. The tuning of $\dot{\alpha}$ and β can be regarded as a tradeoff between overall bandwidth conservation and load balancing. For example we let $\beta = 0$ then the objective is to conserve bandwidth resources only, while setting $\dot{\alpha} = 0$ infers to minimize link overloading within the network. The function of fitness calculation is shown in Figure 2.

Procedure computing_fitness(chromosome i)	
Begin	
set the weight of each link according to the g	gene values in
chromosome i;	
<i>for</i> each link in the cluster g	
Compute the shortest path Tg rooted to all	nodes in the
cluster g;	
for each link (u,v) in Tg	
update link load Luv according to bandwidt	th demand Dg
of cluster g;	_
end for	
Load1 = 0; $load 2 = 0$;	
<i>for</i> each link (u, v) in the network	
Load1 = load1 + Luv;	
If $Luv > Cuv$	
Load2 = load2 + (Luv - Cuv);	
end for	
return fitness = f(load1,load2);	
end	

Figure 2: Fitness Calculation Procedures

Step3 crossover and mutation

According to the basic principle of Genetic algorithms, chromosomes with better fitness value have higher probability of being inherited into the next generation. To achieve this, first we rank all the chromosomes in descending order to their fitness, so the chromosomes with high fitness (lower overall load) are placed on the top of the ranking list. Then we partition this list into two disjointed sets, with the top 50 chromosomes belonging to the upper class (UC) and the bottom 50 chromosomes to the lower class(LC). During the crossover procedure, we select one parent chromosome C_{u}^{i} from UC and other parent C_{l}^{i} from LC in generation "*i*" for creating the child C^{i+1} in generation *i*+1. We use a crossover probability threshold $K_C \varepsilon$ (0.05) to decide the genes of which parent to be inherited into the child chromosome in the next generation. We introduce a mutation probability threshold K_M to randomly replace some old genes with new ones. The function of crossover and mutation are shown in Figure 3.

Procedure crossover $(C_u^i C_l^i)$
Begin
for all genes $j = 1 \dots (E)$
Generate $r = random (MAX_WEIGHT);$
$if r > K_C$
$C^{i+1}(j) = C^{i}_{\ u}(j);$
else if $r > K_M$
$C^{i+1}(j) = C^{i}_{1}(j);$
else
$C^{i+1} = random [1, MAX_WEIGHT];$
end for
return C^{i+1} ;
end

Figure 3: Crossover and Mutation Procedures.

VI. SIMULATION AND RESULTS

a. Simulation Parameters and Performance Metrics

We develop an ad hoc simulator using Visual Basic version 6 under windows XP platform. In our simulation, we generate a random graph of 100 nodes. The nodes could move in all possible directions with displacement varying uniformly between 0 to a maximum value (max_disp), per unit time. In our simulation experiments the transmission range was varied between 0 and 70. The nodes moved randomly in all possible directions with a maximum displacement of 10 along each of the coordinates. The weights used for our simulation were w_1 =0.7, w_2 =0.2, w_3 = 0.05, and w_4 =0.05. We have used LibGA [17] which is a library of routines for developing genetic algorithms. The GA parameters of the genetic algorithm are population size=100, maximum generation (M) = 500, maximum link weight (MAX WEIGHT) = 64, crossover threshold $(K_C) = 30$, mutation threshold (K_M) =0.01, $\mu = 10^7$, $\dot{\alpha} = 1.0$, and $\beta = 10$. We measure the performance technique with four performance metrics:

- The number of reaffiliations,
- The number of clusterheads,
- Load balancing factor (LBF).
- And path lifetime distribution.

The reaffiliation count is incremented when node gets dissociated from its clusterhead and becomes a member of another cluster within the current dominant set. The number of clusterheads in the network defines the dominant set. The first three parameters studied for varying number of nodes N (10, 30, 50, 70), transmission range (tx range=30m), and maximum displacement $(\max_disp=5m)$. To quantitatively measure how well balanced the clusterheads are we use Load balancing factor which defined in [13,14]. The load handled by a clusterhead is essentially the number of nodes supported by it. A clusterhead, apart from supporting its members with the radio resources, has to route messages for other nodes belonging to different clusters. It is difficult to maintain a perfectly load balanced system at all times due to frequent detachment and attachment of the nodes from and to the clusterheads. As the load of a clusterhead can be represented by the cardinality of its cluster size, the variance of the cardinality of its cluster size, the variance of the cardinalities will signify the load distribution. The LBF are defined as the inverse of the variance of cardinality of the clusters [13,14]. Thus, $LBF = n_c / \sum (x_i - \mu) 2$, Where n_c is the number of clusterhead, x_i is the cardinality of cluster *i*, and $\mu = (N - n_c) / n_c$ is the average number of neighbors of a clusterhead (N being the total number of nodes in the system). Clearly, a higher value of LBF signifies a better load distribution and it tends to infinity for a perfectly In the fourth parameter we study the balanced system. lifetime of a repairable unicast path between two nodes as the duration, in which there is one path between them, so

we use "unicast path" or "path" to represent "repairable unicast path". That is, during this period these two nodes can find a path to their messages. In our study, the path between two nodes at time T is chosen to be the "shortest" one between them at time T. The lifetime distribution is important as it gives us a sense of how long generally an establish unicast path can last on an ad hoc network. We prefer to see long lifetime rather than short lifetime for these unicast paths.

b. EXPERIMENTAL RESULTS

Figure 4 shows the reaffiliations per unit time with the transmission range before and after using optimization, respectively. For low transmission ranges, the nodes in a cluster are relatively close to the clusterhead. There is an optimal transmission range for which the reaffiliations are maximal. Further increase in transmission range decreases the reaffiliations since the nodes tend to stay inside the large area covered by the clusterhead irregardless of the movement of the nodes. For fewer numbers of nodes, the reaffiliation count is lower with genetic algorithm. Figure 5 shows the average number of clusterheads before and after using optimization, respectively. We observe that the genetic algorithm yields fewer numbers of clusters, as well as fewer numbers of clusterheads. Figure 6 shows how the load balancing factor (LBF) varies with time before and after using optimization technique respectively. We observe that after optimization there is a gradual increase in the LBF after every dominate set update. This is due to the diffusion of the nodes among clusters. We observe that the value of LBF with the genetic algorithm technique is more balanced. Figure 7 shows the cumulative distribution of the life time of paths in four different paths by using optimization. The Y axis shows the percentage of the paths whose lifetime is less than a particular value. The X axis shows the range between 1 and 50 seconds rather than the whole range between 1 and 300 seconds. This is because over 91% of all paths have a lifetime less than 50 seconds. So we focus only on this range. The distribution curves of these three different paths are almost the same. The information lets us know that when a source node wants to set up a path to destination, it need not care about whether that destination is moving in the same direction with itself. We observe that, 70% of all paths have a lifetime less than 10 seconds, 80% of all paths have a lifetime less than 20 seconds.



Figure 4: Reaffiliations per unit time before and after optimization, max_disp = 5 meter.



Figure 5: Average Number of Clusters before and after optimization, tx_range= 30 meter.



Figure 6: Load Distribution before and after optimization.



Figure 7: Cumulative Life Time for four paths.

VII. CONCLUSION

We presented a genetic algorithm as an optimization technique for routing in MANET. The results show that, with the genetic algorithmic technique each clusterhead handles the maximum possible number of mobile nodes in its cluster in order to facilitate the optimal operation of the medium access control (MAC) protocol, reduce the number of clusters and hence clusterheads, as well as, the loads among clusters are more evenly balanced by factor of ten. A genetic algorithm technique mapped the possible solutions given by a weight based distributed clustering algorithm in order to find the better solution from a pool of solutions. Each clusterhead handles the maximum possible number of nodes in its cluster. Also a fewer clusterheads are obtained by the genetic algorithm technique. With the genetic algorithm technique the cumulative distributions of the paths are almost the same. Generally, another criterion of research can concentrate to simplify parameters of GA's optimization to leave the bad one out and optimize the good parameters. The genetic algorithm can also be implemented to other criteria of research such as robotic systems for the purpose of achieving the best performance.

REFERENCES

- [1] Abolhasan, M., Wysocki, T., & Dutkiewicz, E. (2004). "A Review of Routing Protocols for Mobile Ad hoc Networks.", VOL 2, ISSUE 1, 1-22.
- [2] S.Narayanaswamy, V.Kawadia, R.S. Sreenivas, and P. R.Kumar "Ad-Hoc Networks: Theory, Architecture, Algorithm and implementation of the Compow protocols " Proc. of European wireless (2002). Next generation wireless networks: Technologies, Protocols, Service and Applications, PP. 156-162, Floence, Italy.
- [3] G. Pei et al.," *wireless Hierararchical Routing Protocol with Group Mobility*," proc. IEEE WCNC'99, New Orleans,LA, Sept.(1999).
- [4] C.E. Perkins and E.M.Rayer, "Ad-Hoc On Demand Distance Vector Routing," Proc. (IEEE) WMCSA '99 New Orleans, LA, Feb. (1999), pp. 90-100.

- [5] Clarles E. Per Kins and Pravin Bhagwat " *Highly Dynamic Destination Sequenced Distance Vector Routing (DSDV) for mobile computer* " SIGCOMM, ACM (1994).
- [6] Ching-Chuan Chiang and Hasiao Kuang Wu and Winston Liu and Mario Gerla" *Routing in Clustered Multihop, Mobile Wireless Networks with fading Channel*" IEEE Sigapore Internationl Conference on networks, SICON'97, April (1997).
- [7] David B. Johnson, "Routing in Ad Hoc Networks of Mobile Hosts," Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications, pages 158–163, December 1994.
- [8] L Davis, " *Handbook of Genetic Algorithms*". New York: Van Nostrand Reinhold, (1991).
- [9] A.Riedl,"A Genetic Algorithm for Network Planning" in Proc. of EUNICE'98, Munich sep,1998.
- [10] S.Bassagni, I. Chlamtac and A.Fargo, "A generalized Clustering Algorithm for peer to peer networks", Proceedings of workshop on algorithmic Aspects of communication, Bologn Italy, July (1997).
- [11] DiwekarU., "Introduction to Applied optimization", Kluwer Academic Publishers(2003).
- [12] C.R. Lin and M. Gerla, "Adaptive clustering for mobile wireless networks", IEEE J.on Selected Areas in Comm., 1997, pp1265-1275
- [13] M. Chatterjee, S.K.Das and D.Turgut "An on-demand weighted clustering Algorithm for ad hoc networks", November (2000), PP. 1697-1701.
- [14] L. Ramaswamy, B. Gedik, and L. Liu. "Connectivity Based Node Clustering in Wireless Networks" technical Report, College of Computing, Georgia Tech, 2003.
- [15] D. Awdcheetal," Overview and principles of Internet Traffic Engineering" RFC3272,(2002).
- [16] Dalma Turgut, Sajal K.Das. "Optimizing Clustering algorithm in Mobile Ad hoc Networks", Journal of Clustering Computing, Vol. 5, No. 2. Oct. 2002. pp.193-204.
- [17]A.L.Corcoran,http://www.cgi.cs.cmu.edu/asf/cs/project /ai- repository/ai/areas/genetic/ga/systems/libga/0.html