A Novel Routing Algorithm for Highly-Mobile Ad-Hoc Networks

R.Chandrasekar and Sudip Misra

***Abstract*-In this paper a Neural Gas Algorithm with Competitive Hebbian Learning is proposed for routing in highly-mobile ad-hoc networks. The inherent features of Neural Gas with Competitive Hebbian Learning (CHL) are utilized such as an implicit neighborhood ranking scheme to allow for only the local-best possible nodes to participate in the routing. Independence from topological arrangements of the network is achieved as is the case with highly-mobile ad-hoc networks and near optimal distribution of the information around the node's location is obtained based on current changes in the neighborhood of the node. A modified 2-hop acknowledgement method is used along with beacon timestamp values in order to select the routes according to the freshness of the individual responses. A Trust and Reputation Mechanism based on a Ranked Neighborhood and Location Information is additionally used to verify the veracity of the information obtained from the nodes.**

***Index Terms*-Ad-Hoc Networks, Neural Gas Algorithms, Routing**

I. INTRODUCTION

A mobile ad hoc network (MANET) [5] is a collection of mobile computers or devices that cooperatively communicate with each other without any pre-established infrastructures such as a centralized access point. Computing nodes in an ad hoc network act as routers to deliver messages between nodes that are not within their wireless communication range [5,11]. Based on the kinds of topological arrangements and movements they can be classified as minimally mobile or highly mobile.Here the focus is on highly-mobile ad-hoc networks the typical kinds of which can be found commonly in day to day life. An example would be of an ad-hoc network consisting of moving vehicles commonly termed as *VANETS* [14]. This paper is concerned with a more generalized version of the above example.

An Artificial Neural Network (ANN or ANS) information processing paradigm [4,8,9,10] that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. The single cell neuron consists of the cell body, or soma, the dendrites, and the axon. The dendrites receive signals from the axons of other neurons. The small space between the axon of one neuron and the dendrite of another is the synapse[4]. The dendrites conduct impulses toward the soma and the axon conducts impulses away from the soma. The function of the neuron is to integrate the input it receives through its synapses on its dendrites and either generate an action potential or not. In [2], a routing algorithm for ad-hoc networks called *NEURAL* was proposed which was based on the properties of the aforementioned neurons. It took into account the learning and self-organizing abilities of the brain. More precisely, it was inspired by the synapses process between neurons, when a signal is propagated. Basically, the most significant characteristic of NEURAL is the uniform distribution of the information around the node's location based on the current changes in its neighborhood. Using a 2-hop acknowledgment mechanism, local information is monitored in order to be used for route selection method, classification procedures and learning algorithms. This paper has adapted the above features to account for highly mobile ad-hoc networks. More precisely, the basic features of self-organizing maps[8,10] like the unsatisfactory learning rate for highly-mobile ad-hoc networks, near topological rigidity and absence of quick adaptation of routing paths entails a new routing paradigm which takes into account the above deficiencies. Also to optimize for low energy consumption and overhead, efficient broadcasting mechanisms are utilized by incorporating timestamp and hop count values for performing the pre-processing calculations. To reduce the pre-processing interval the proposed approach performs the broadcasting after a certain timestamp which depends on the beacon-reply interval from the neighboring nodes to effectively have a low overhead for network. The Neural gas[9,13] with Competitive Hebbian Learning(CHL), the algorithm which forms the base of this paper converges quickly to low distortion errors[1].To also improve the delivery ratio as appropriately mentioned in [2], only the local-best possible nodes are chosen as the network keeps migrating and randomly moving. In a neural gas algorithm with *CHL*, at each adaptation step of the algorithm a connection between the winner and the second-nearest unit is created [9]. A local edge ageing mechanism is used to remove edges which are not valid anymore. The synaptic weights are also adapted without any regard to the topological characteristics of the system. To determine the local-best possible nodes, we use an implicit ranking scheme to order the elements which has proven to be faster and requires less adaptation steps. The proposed approach also uses a geo/situation oriented trust and reputation mechanism [11,12]. Rest of the paper is divided as follows; Section 2 describes the architecture, Section 3 provides some experimental results while Section 4 concludes the work.

R.Chandrasekar is a senior undergraduate with the Department of Information Technology, Sri Venkateswara College of Engineering, Sriperumbudur, India 602105 (Email: chandra85@gmail.com)

Dr.Sudip Misra is a Visiting Scientist at the School of Electrical and Computer Engineering, Cornell University, Ithaca, NY 14853 USA (Email: sudip_misra@yahoo.com)

II. ARCHITECTURE

The proposed approach consists of two modules; the preprocessing and route discovery module as shown in Figure 1.The word *neuron* may be substituted for a *node* and vice-versa. The effective utilization of the two modules provides an improvement over NEURAL in that it can be applied to highly-mobile ad-hoc networks. The Preprocessing phase consists of the K-NNR Rule [4] but with additional information about the timestamp of the beacon reply packet sent back to the originating node. The Route Discovery module consists of Broadcasting of Inputs, Ordering of Elements, Modified Adaptation Procedure, Connection and Age Incrementing module and finally the Mobile Trust and Reputation Mechanism.

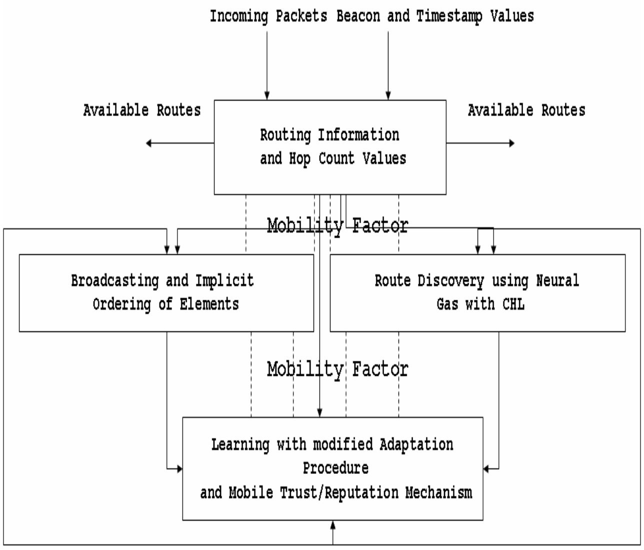


Figure1. Neural Gas with CHL architecture for highly mobile ad-hoc networks. Includes a mobility factor as shown.

A. Using Timestamps with Pattern Classification

In this approach we have adopted a similar pattern classification scheme called the K-Nearest Neighbors Rule or K-NNR [4] with additional information suitable to the needs of a highly mobile network. The K-NNR is a very intuitive method that classifies unlabeled examples based on their similarity with examples in the training set .For a given unlabeled example, find the k “closest” labeled examples in the training data set and assign to the class that appears most frequently within the k-subset. Based on sending hello request and reply packets during an interval of time it senses the continuous topology changes of the network. Besides an activation stage, a beacon reply timestamp called as *BEACON_TIMESTAMP* is associated with each responding node. This is important as nodes previously classified as inhibitory or excitatory may not remain the same after some time as the nodes constantly move in a highly-mobile ad-hoc network. A Maximum timestamp called *MAX_TIMESTAMP* is defined beyond which the reply beacon packets are ignored or discarded. K-NNR disseminates environmental changes using the activation stage (*A*s) configured using the *activation threshold* (*K* value from K-NNR) and represents an estimation of the local density based on the 1-hop information.

B. Route Discovery

A two hop acknowledgement mechanism is used from [2]. Additionally to maintain the integrity of the nodes and their routing capabilities, we assign hop count values to each node. Successive nodes when they receive the route discovery requests update the hop count values of the 2 hop neighboring nodes in the routing table. The additional overhead involved is compensated for the fact that the current node’s neighbors may not be present after some time due to the high mobility of the nodes and thus the weight values may not be pertinent at that point of time. The explanation of this concept is illustrated by the example hop count values given in Table 1.

TABLE.1. HOP COUNT VALUES FOR EACH OF THE NEIGHBORING NODES FROM FIGURE 2

Node	Neighbors	Hop Counts
A	B C D	1 2 2
B	A C D	1 1 1
C	A B D	2 1 1
D	A B C	2 1 1

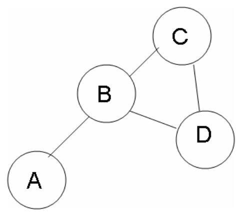


Figure2. An Example Ad-Hoc Network with the connections showing the direct possible links between the nodes.

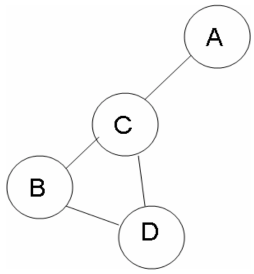


Figure3. An Example Ad-Hoc Network with the connections showing the direct possible links between the nodes and Node A moving from its original place in the network.

Figure 2 and 3 show some sample nodes in a highly-mobile ad-hoc network. From this figure node A has moved out of its position after some time (*but within a short period of time*) and thus the routes earlier discovered by B may not be valid as the Round trip time and/or the Number of Neighbors for B would have changed at that point of time. By maintaining the hop count values the node B can determine from its neighbors C or D that the hop count for A has changed and it updates the new hop counts in its routing table.

C. Ordering of Elements

To reduce the time taken for calculating the best possible an implicit ordering scheme is used taking into account a mobility factor of each of the originating nodes' neighbors as follows,

$$m_{Mi} \frac{l_i - l_{\min}}{l_{\max} - l_{\min}} \quad (1)$$

Where l_{max} and l_{min} are the minimum and maximum possible distances in the network and MF is a mobility factor denoting how fast the nodes are moving. If the Implicit Rank is more but the Mobility Factor is high then the ranking is changed as shown in Table 2.

TABLE II. HOPCOUNT VALUES AND MOBILITY FACTOR IN DECIDING THE RANKS OF NEIGHBORING NODES

Node	Neighbors	Hop Count	Mobility Factor	Rank
A	B	2	1	1
	C	1	2	2
	D	2	3	3
B	A	2	1	2
	C	1	1	1
	D	2	2	3
C	A	1	1	1
	B	1	3	3
	D	1	2	2
D	A	2	2	3
	B	1	1	1
	C	1	2	2

The values of weighted inputs Net_i are arranged in decreasing order. The top few elements are chosen with a minimum of 2 and a maximum $\Delta \max_i$ depending on the neighborhood of the originator given by,

$$\Delta \max_i = \frac{Net_{\max i} - Net_{\min i}}{2} \quad (2)$$

Where Net_{maxI} and Net_{minI} are the minimum and maximum of the weighted inputs. From Table1, by analyzing Node C's Hop count values to its neighbors we find that d_{max} and d_{min} are 2 and 1 respectively. Therefore the m_i or the rank index for Node C would have a *Rank 1* for node A and *Rank 2* for nodes B and D. Here multiple values of Rank 2 can be useful in situations when either of the nodes have failed. But from Table 2, after the nodes have moved and by applying an additional Mobility Factor to resolve which nodes would be the most suitable, we can see that Node D now gets Rank2 with respect to Node C and Node B is assigned as Rank3.

D. *Adaptation Procedure*

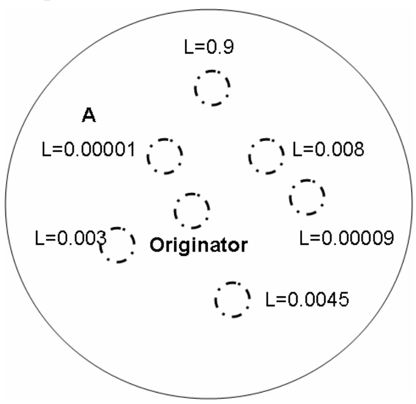


Figure 4. Learning Rates represented at each of the nodes. A has the least learning rate and may not be considered for further adaptation by the other nodes as well as the originator.

A modified adaptation procedure [1] is used to enable the weight update rule to modify those weights with non-zero values of Δw_i to eliminate the explicit ranking mechanism completely due to the large overhead involved. To allow the nodes with only non-negligible learning rates to participate in the adaptation process, preference can be given for nodes closer to the training or originator node to enable robust and efficient routing. This is represented by Figure 4.

E. *Incrementing Ages and Creating Connections*

Every node has a *LINK_TIMESTAMP* field in its routing table to check for the validity of the link after the connection has been established. It is incremented for every rank 1 node obtained after training. A connection is created between the top ranked nodes as defined by $\Delta \max_i$. If the *LINK_TIMESTAMP* value exceeds a certain threshold then those links are broken. This facilitates deleting links between nodes which move farther away from each other's range. To compensate for the loss of connections, the modified training and learning procedure enables faster acquisition of links between the originator and its neighboring nodes.

F. *Verifying Veracity of Routing Information*

We account for the geographical positioning of nodes since the network keeps moving frequently. Our approach consists of two modules; An Opinion Generation which consists of the direct and indirect opinions, and a threshold function for reputation which is used to test the confidence level of the estimates. This is loosely based on [11,12]. The Opinion Generation Module could be a combination of various factors such as experience, indirect or direct trust. This is necessary because opinions from far away nodes could also count for while determining the veracity of information from a node providing routing information. A Trust and Neighborhood Mechanism based on the Ranked Neighborhood values as well as location information from the nodes. The term *direct trust* is used for reputation information that can be identified based on past interaction with those particular nodes.

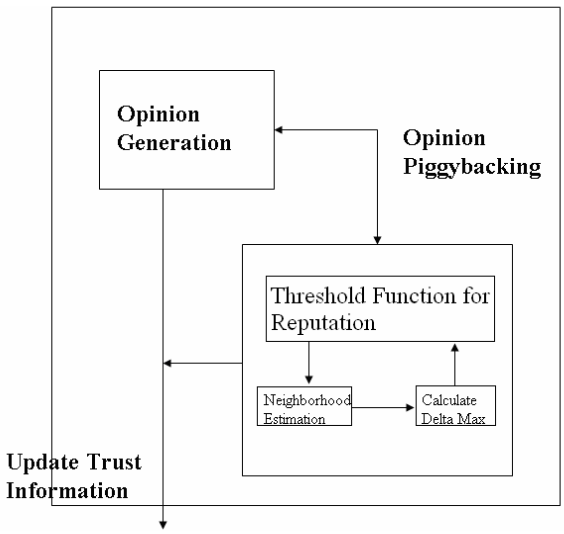


Figure 5. Architecture of the Trust and Reputation Mechanism based on Location and Ranked Neighborhood Information.

Indirect trust is reputation[15] provided by nodes of which reputation information is already known and appended to intermediate nodes to reach the requesting node. On arrival of an event message every forwarding node generates an opinion on the trustworthiness of this message. Due to the short amount of time available to nodes seeking to verify routing information obtained, $\Delta \max_i$ is a good measure of the extent to which partial opinion should be collected for opinion generation. Depending on the available set of messages T, a function f is calculated to determine the threshold of trustworthiness of the opinions. It is adapted here for a high mobility network from [11].

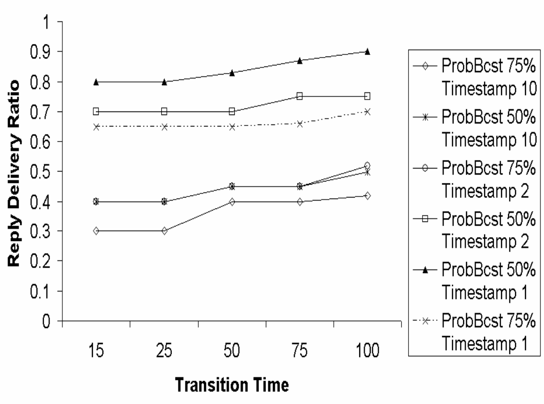
$$f = |T| - \left(\frac{r_{\max} - r_{fin}}{a_v . \Delta \max_{m,i}}\right) \tag{3}$$

Depending on the network size and $\Delta \max_i$, f has to be above a certain threshold. This is to announce the importance of *node i* to other nodes in the neighborhood. Other nodes update their trust information about node i if the value of f is satisfied. r_{max} and r_{fin} are reputation levels from [11] and a_v is a constant value depending on the neighborhood of the network. $\Delta \max_i$ is also used for normalizing the final values.

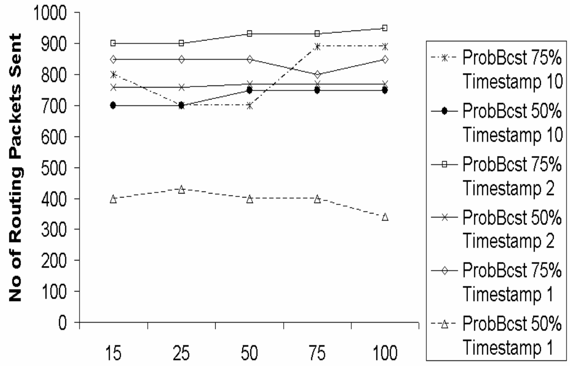
III. EXPERIMENTAL RESULTS

We experimentally try to evaluate our routing algorithm by using *ns-2* [3]. First, each mobile host has an omni directional antenna having unity gain with a nominal radio range of 250 m. The random waypoint model is selected as mobility model in a rectangular field with nodes' speed uniformly between cero and a maximum value of 20 $m.s{-}1$. Number of sources and the sending rate are varied to study different loads for each configuration. For each configuration, reported measurements are the mean of 10 runs with different random seeds. We use a similar *Post-Synapse* [2,6] algorithm for flooding the network during pre-processing but with *reduced pause times* to account for highly mobile nodes. To evaluate the performance of the Post-synapse algorithm, 20 nodes are configured using 2 to 5 seconds of pause time and a pre-processing interval depending on the *BEACON_TIMESTAMP* values from the nodes. Simulation was run during 150 seconds taking into account 5 different transition times: 15, 25, 50, 75 and 100 seconds. The performance of the *Post-Synapse* algorithm is evaluated by measuring the hello beacon packet reply delivery ratio for different percentages of ProbBcst[2] and *BEACON_TIMESTAMP* values and transition times in Figure 6a. We find that with increasing ProbBcst and Timestamp values, both the number of collisions increase as well as the routes become stale. This means that nodes which hitherto were neighbors may have changed their positions hence the beacon reply packets are not delivered back to the originator nodes. In Figure 6b we evaluate the number of routing packets sent for different transition times in the network along with different values of ProbBcst and BEACON_TIMESTAMP values in the network. We find that with increasing values of ProbBcst and increasing BEACON_TIMESTAMP values the number of routing packets sent increases and vice-versa. This is because of more number of collisions. Also the packets sent may not reach the destination nodes with higher timestamps as they might have moved from their original positions in a highly-mobile ad-hoc network.

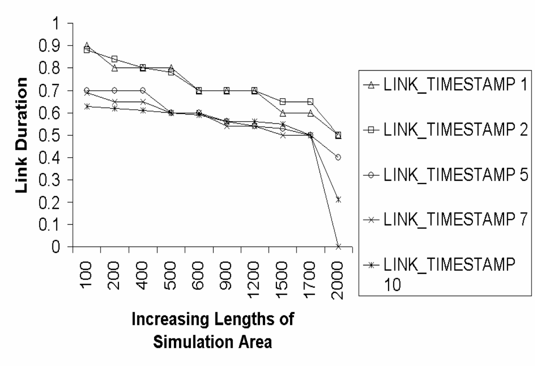
We avoid the use of Random Query Processing Delay (*RQPD*) [6] as we find that these two parameters are sufficient to test the number of packets sent in the network. Figure 6c shows the normalized link duration for increasing LINK_TIMESTAMP values and lengths of the simulation area in the network. We find that the link duration decreases with increasing LINK_TIMESTAMP values as the nodes are continuously moving. Figure 6d shows the packet delivery ratio for increasing lengths of the simulation area and increasing hop count values. We find that with increasing hop count the delivery ratio decreases drastically. This is because of the fact that as the nodes continuously move in the network, the packets have to traverse a large number of nodes for higher hop counts. These packets may not reach their intended destination as the nodes would have moved from their original positions. This factor is again compounded by the fact that with increasingly sparse networks there are lesser number of nodes available for routing in the network. Figures 6e and 6f shows the corresponding performance comparisons against the Post synapse(PS) algorithm for NEURAL and Neural Gas(NG) for different Timestamps(TS). From Figures 6e and 6f we can find out that generally the Post Synapse Algorithm with NEURAL performs worse than most instances of Neural Gas with CHL for different Timestamp values. Occasionally NEURAL has a better delivery ratio than Neural Gas with CHL, but this is in the initial stages of the transition times and as the simulation progress the ratio spirals downwards.



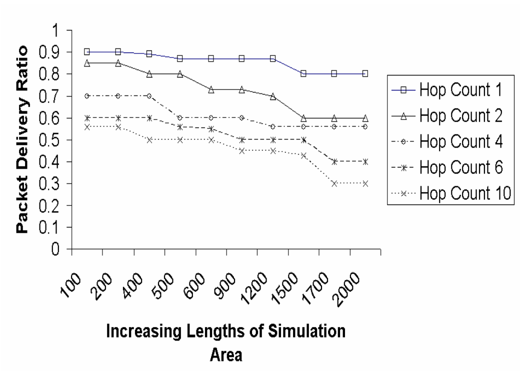
6(a)



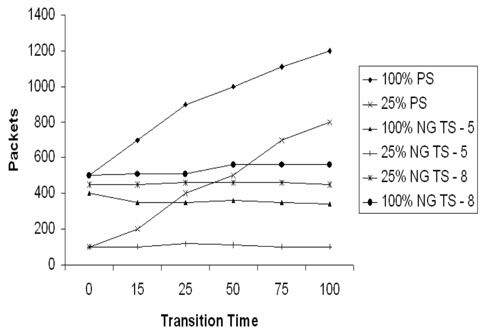
6(b)



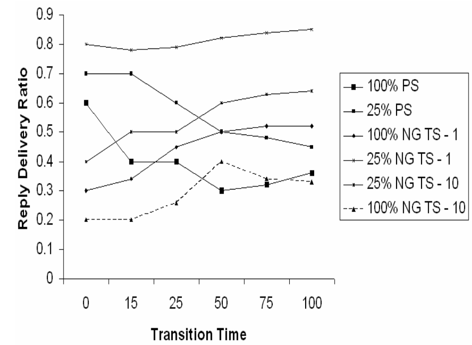
6(c)



6(d)



6(e)



6(f)

Fig. 6a) Beacon reply delivery ratio for ProbBcst and BEACON_TIMESTAMP values and different transition times
b) No of routing packets sent for ProbBcst and BEACON_TIMESTAMP values and different transition times
c) Link Duration over the average link duration for simulation area (in metres) and increasing LINK_TIMESTAMP values
d) Packet delivery ratio for simulation area (in metres) and hop count values in the highly mobile ad-hoc network
e) No. of Routing Packets sent for Post Synapse (PS) algorithm in NEURAL and Neural Gas (NG) with different Timestamps (TS)
f) Reply Delivery Ratio for Post Synapse (PS) algorithm in NEURAL and Neural Gas (NG) with different Timestamps (TS)

IV. CONCLUSION

In this paper we have proposed a neural gas algorithm with competitive hebbian learning for routing purposes in highly-mobile ad-hoc networks. To impart a certain degree of speed and efficiency while performing training and learning we have utilized an implicit ranking scheme for identifying and adapting a select few neighboring nodes of the originator node in the network. We have also utilized a *BEACON_TIMESTAMP* value to determine the freshness of the links during the pre-processing phase of the etwork. To maintain the integrity and the quality of the routes obtained, hop count values and *LINK_TIMESTAMP* values are assigned to each and every node in the network. From the simulation results we have proved that with lesser percentage of ProbBcst values along with lesser hop counts and timestamp values accurate and efficient routing can be obtained. Thus an independence from topological arrangements in the network as is the case with highly mobile ad-hoc networks and selection of the local-best possible nodes for routing is achieved.

REFERENCES

[1] Ajantha S. Atukorale, P.N. Suganthan, "Hierarchical overlapped neural gas network with application to pattern classification," In *Neurocomputing* 35 (2000) Apr 2000.

[2] V. E. Mujica-V., D. Sisalem, and R. Zeletin-Popescu, "Neural: A self organizing routing algorithm for ad hoc networks," in *Proceeding of 3rd International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, Trentino, Italy, April 2005.

[3] U. Berkeley and U. ISI. The network simulator ns-2, 1998. Part of the VINT project. http://www.isi.edu/nsnam/ns.

[4] C. Bishop. *Neural Network for Pattern Recognition*. Oxford University Press, Great Britain, 2000.

[5] J. Broch, D. Maltz, D. Johnson, Y. Hu, and J. Jetcheva. "A performance comparison of multi-hop wireless ad hoc networks routing protocols," In *Fourth annual MOBICOM*, pages 85–97. 1998.

[6]Z. Hass and M. Pearlman. "The performance of query control schemes for the zone routing protocol," In *IEEE/ACM Transactions on networking*, volume 9 N4, Aug. 2001.

[7] S. Haykin. *Adaptive Filter Theory*. Pretice-Hall, Englewood Cliffs, NJ, 1991.

[8]R. Helge, T. Martinetz, and K. Schulten. *Neural Computation and Self-Organizing Maps- An Introduction*. Addisonn-Wesley, NY, 1992.

[9]T. Martinetz, S.G. Berkovich, K. Schulten, "Neural gas network for vector quantization and its application to times-series prediction," IEEE Trans. Neural Networks (4) (1993).

[10]T. Kohonen. "Self-organized formation of topologically correct feature maps," *Biological Cybernetics*, 43:59–69, 1982.

[11]Robert Morris, John Jannotti, Frans Kaashoek, Jinyang Li, Douglas Decouto, "*CarNet: A Scalable Ad Hoc Wireless Network System*", *proceedings of the 9th ACM SIGOPS European Workshop*.

[12]Pietro Michiardi and Refik Molva. "Core: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks," In *Proceedings of Communication and Multimedia Security 2002 Conference*, September 2002.

[13]T.M.Martinez and K.J.Schulten, "Topology representing networks," *Neural Networks*, 7(3):507-522, 1994.

[14] Christian Lochert, Hannes Hartenstein, Jing Tian, Holger Fübler, Dagmar Hermann, Martin Mauve , "A Routing Strategy for Vehicular Ad Hoc Networks in City Environments," *In Proc of the IEEE Intelligent Vehicles Symposium* 2003.

[15] Partha Dasgupta, Prashant Dewan and Amiya Bhattacharya. "On using reputations in ad hoc networks to counter malicious nodes," In *Proceedings of QoS and Dynamic Systems (in conjunction with IEEE ICPADS)*, 2004.