International Journal of Soft Computing and Engineering (IJSCE)
ISSN: 2231-2307, Volume-1, Issue-3, July 2011

Routing in Ad Hoc Wireless Networks using Soft Computing techniques and performance evaluation using Hypernet simulator

Siddesh.G.K, K.N.Muralidhara, Manjula.N.Harihar

Abstract— **An ad-hoc network is a collection of wireless mobile hosts forming a temporary network without the aid of any established infrastructure or centralized administration. In such an environment, it may be necessary for one mobile host to enlist the aid of other hosts in forwarding a packet to its destination, due to the limited range of each mobile host's wireless transmissions. Much effort has gone into mobile ad-hoc network (MANET) research over the past decade. Yet, even to-day, mobile ad-hoc networking is seen as a relatively new area of research. The reason for this can be traced to the fact that the maturity in truly understanding these networks is still alarmingly low and actual deployment of these networks rare. This paper presents a protocol for routing in ad hoc networks that uses soft computing techniques like neural networks, fuzzy logic and genetic algorithm. The simulation has been performed using hyper net simulator for various existing protocols like proactive routing , reactive routing, power aware routing protocol, hybrid routing . Our protocol uses soft computing techniques protocol for establishing the link between the nodes in minimum time. The results of our experimentation have been very satisfactory and we have achieved the goal of optimal route finding to a large extent. The simulation results are obtained using hypernet simulator and are also compared with the results obtained using NS-2 .**

Index Terms— **Routing Protocol, MANET, Neural Nets, Fuzzy Logic, Genetic Algorithm, Soft Computing, hyper net.**

I. INTRODUCTION

An ad-hoc or short-live network is the network of two or more mobile devices connected to each other without the help of established infrastructure [1]. In contrast to a fixed wireless network, an ad-hoc network can be deployed in remote geographical locations and requires minimum setup time and administration costs. Moreover, the integration of an ad-hoc network with a bigger network- such as the Internet-or a wireless infrastructure network increases the coverage area and application domain of the ad-hoc network.

Mobile ad-hoc sensor networks are very beneficial in different scenarios. These networks advance operational efficiency of certain civilian applications. For example, in a military operation, it can be used to gather information about enemy location, movement, etc. As a mobile traffic sensor networks, it can be used to monitor vehicle traffic on motorways and as a mobile surveillance sensor network, it can be used for providing security in various places such as shopping malls, hotels and in other similar facilities. Mobile ad-hoc sensor networks can also be use to locate free and occupied spots in a parking area and to monitor environmental changes in places like forests, oceans, etc. Ad-hoc network uses multiple hops[8] for communication between source and destination as shown in Fig.1.

Wireless Ad Hoc Networks (MANETs for short) [6] are characterized by their mobility, ease of deployment, self-configuration without a centralized administration and ability of nodes to communicate with each other even in out-of-range conditions with intermediate nodes performing the routing functions. MANETs are also flexible enough to get connected to cellular as well as wired networks. The features that delineate them from traditional networks are the mobility of the nodes, the absence of need for an infrastructure/ centralized administration [1,2] and the ability to configure on the fly as the situation demands. These unique features impose additional overheads in protocol implementations. Compared to Cellular Networks. MANETs are adaptable to changing traffic demands and other physical conditions.

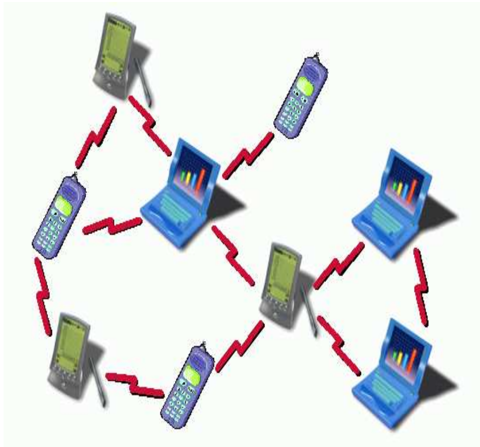


Fig. 1. Multi hop communication between source and destination

Since the attenuation characteristics of wireless media are non-linear, energy efficiency will be superior and increased spatial reuse will guarantee superior capacity and spectral efficiency.

Manuscript received June 17, 2011.
Siddesh. G. K, Department of Electronics and Communication, Sri Bhagawan Mahaveer Jain College of Engineering, Bangalore, India, 080-26715595, (e-mail: professorsiddeshgk@gmail.com)..
K. N. Muralidhara ,Department of Electronics and Communication, P.E.S College of Engineering, Mandya, India, 08232-225010, (e-mail: knm08@rediffmail.com).
Manjula. N. Harihar, Department of Electronics and Communication, Sri Bhagawan Mahaveer Jain College of Engineering, Bangalore, India., (e-mail: manjulaharihar@gmail.com).

Since MANETs are generally deployed in disaster management and critical situations, there is a substantial amount of real-time content in their operation. Time plays a crucial role in the communication activities, be it a protocol transfer session or a plain routing operation. In view of these facts, efficient protocol implementation assumes the highest level of importance in practical implementations. It is therefore no surprise that a huge amount of time and effort has gone into inventing various kinds protocols to suit different needs and varying conditions. The efficiency of a routing protocol [5] (at the outermost level) is directly related to numerous factors such as node mobility, dynamic topology, the communication capabilities of the nodes, power consumption issues, bandwidth constraints, traffic congestion, security and a host of other related parameters, all of which have to be well orchestrated to achieve an optimal performance that is adequate at the minimum level.

When we take all these factors into consideration, evolution of an optimal routing strategy is an indomitable task. These factors are mutually exclusive and there is no explicit relationship of these factors amongst themselves and more importantly we do not see how these are related to an optimal routing strategy. Herein lies a highly complex non-linear problem to solve, a problem that is not amenable to any classical solution. Artificial Neural Network (ANN for short) steps in at this juncture as our savior. An ANN is akin to a biological network, capable of thinking, reasoning, decision making and a high degree of parallelism. It draws inferences from a vast storehouse of knowledge and experience gained over a period of time in solving problems. It can work with imprecise and ill-defined parameters in arriving at solutions. Fuzzy Logic and Genetic Algorithms are additional ingredients that can make an ANN more powerful and aggressive in solving unsolvable problems by analytical methods. Fuzzy Logic helps us to work with ill-defined parameters and Genetic Algorithms represent a powerful paradigm in searching for optimal solutions in a solution space. A judicious mixture of ANN with Fuzzy Logic and Genetic Algorithms personifies a powerful mechanism in protocol development and routing strategies in Ad Hoc Networks.

A. *Artificial Neural Networks*

According to a simplified account, the human brain consists of about ten billion neurons and a neuron is, on average, connected to several thousand other neurons. By way of these connections, neurons both send and receive varying quantities of energy. One very important feature of neurons is that they don't react immediately to the reception of energy. Instead, they sum their received energies, and they send their own quantities of energy to other neurons only when this sum has reached a certain critical threshold. The brain learns by adjusting the number and strength of these connections. Even though this picture is a simplification of the biological facts, it is sufficiently powerful to serve as a model for the neural net. The first step toward understanding neural nets is to abstract from the biological neuron, and to focus on its character as a threshold logic unit (TLU). A TLU is an object that inputs an array of weighted quantities, sums them, and if this sum meets or surpasses some threshold, outputs a quantity. Let's label these features. First, there are the inputs and their respective weights: $X_1, X_2, \ldots, X_n$ and $W_1, W_2, \ldots, W_n$. Then, there are the $X_i * W_i$ that are summed, which yields the activation level a, in other words

$$\hat{A}\ a = \sum_{k-1}^{n} X_k W x_k$$

The threshold is called theta. Lastly, there is the output: y.

When $a >= theta$, $y = 1$, else $y = 0$. Notice that the output doesn't need to be discontinuous, since it could also be determined by a squashing function, s (or sigma), whose argument is a, and whose value is between 0 and 1. Then, $y = s(a)$.

The threshold logic unit is as shown in Fig. 2

B. *Fuzzy Logic (FL)*

FL is a simple but powerful methodology in logic building. It was originally conceived in the context of building control systems based on micro-controllers. FL incorporates a simple, rule-based *IF X AND Y THEN Z* approach to solve a control problem.

In the context of modeling protocol for an Ad Hoc Network, most of the parameters are imprecise or not so-well defined. For example, mobility can be expressed in vague terms by means of a motion vector (precise values will never be known and not essential either). Similarly, distance limitations, power available at the nodes, traffic density etc. is parameters where determination of precise values are not practical and not important either. A fuzzy model helps us to work with imprecise values in a very predictable way.

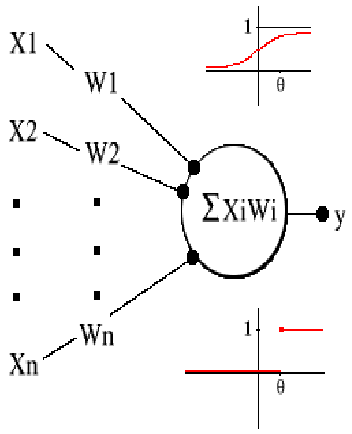


Fig. 2. Threshold logic unit, with sigma function (top) and cutoff function (bottom)

C. *Genetic Algorithm (GA)*

The basic purpose of genetic algorithms (GAs) is optimization. Since optimization problems arise frequently, this makes GAs quite useful for a great variety of tasks. As in all optimization problems, we are faced with the problem of maximizing/minimizing an objective function $f(x)$ over a given space X of arbitrary dimension. A brute force which would consist in examining every possible x in X in order to determine the element for which f is optimal is clearly infeasible. GAs give a heuristic way of searching the input space for optimal x that approximates brute force without enumerating all the elements and therefore bypasses performance issues specific to exhaustive search.

We will first select a certain number of inputs, say, $x_1, x_2 \dots x_n$ belonging to the input space X. In the GA terminology, each input is called an organism or chromosome. The set of chromosomes is designated as a colony or population. Computation is done over epochs. In each epoch the colony will grow and evolve according to specific rules reminiscent of biological evolution.

To each chromosome x_i, we assign a fitness value which is nothing but $f(x_i)$. A stronger individual that is those chromosomes with fitness values closer to the colony optimal will have greater chance to survive across epochs and to reproduce than weaker individuals which will tend to perish. In other words, the algorithm will tend to keep inputs that are close to the optimal in the set of inputs being considered (the colony) and discard those that under-perform the rest.

The crucial step in the algorithm is reproduction or breeding that occurs once per epoch. The content of the two chromosomes participating in reproduction are literally merged together to form a new chromosome that we call a child. This heuristic allows us to possibly combine the best of both individuals to yield a better one (evolution).

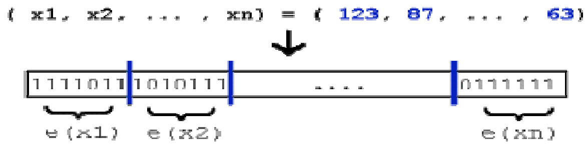
Moreover during each epoch, a given fraction of the organisms is allowed to mutate. This provides a degree of randomness which allows us to span the whole input space by generating individuals with partly random genes.

Each epoch ends with the deaths of inapt organisms. We eliminate inputs exhibiting bad performance compared to the overall group. This is based on the assumption that they're less inclined to give birth to strong individuals since they have poor quality genes and that therefore we can safely disregard them (selection).

II. The Algorithm

Let's examine in detail how this whole process is accomplished and how the algorithm works in practice. Let's take the example of optimizing a function f over a space X contained in Nd.

Every input x in X is an integer vector $x = (x_1, x_2, \dots, x_n)$. For the sake of simplicity, assume $0 <= x_i <= k \ for \ i = 1 \dots n$. In order to implement our genetic algorithm for optimizing f, we first need to encode each input into a chromosome. We can do it by having $log(k)$ bits per component and directly encoding the value x_i. Each bit will be termed gene. Of course, we may choose any other encoding based on our requirements and the problem at hand.



At epoch 0, we generate (possibly randomly) an initial set of inputs in X. Then at each epoch i, we perform fitness evaluation, reproduction, mutation and selection. The algorithm stops when a specified criterion providing an estimate of convergence is reached.

A. Neuro-Fuzzy-Genetic Based Network

Based on the previous discussion of the three essential ingredients, our ANN acts like a powerful inference engine, drawing all the inference rules from an extensive knowledge base. Our hybrid Neural Network functions with the cooperation of Fuzzy Logic, operating on inputs (which are fuzzy in nature) and generating a set of solutions in the solution space with minimal searching using Genetic algorithms. A representative schema of the proposed network is as shown in Fig.3

III. Implementation of the Neuro-Fuzzy-Genetic Network

Some amount of introduction on the design of our routing protocol will be in order at this point. The aim of the protocol is to establish the best possible route within the minimum time possible and the appropriate approach to this problem is the use of soft-computing technologies such as Neural Nets to reduce the dimensionality of the problem, Fuzzy Logic to deal with imprecise inputs and Genetic Algorithms to find an optimum solution in the solution space by search and related heuristic techniques.

The problem therefore reduces to finding an acceptable solution in an optimal sense. Linear Programming is not appropriate when the inputs to the system are not crisp. A judicious combination of Neural Network with Fuzzy Logic and Genetic Algorithms appears to be the ideal solution.

By examining the input layers of the Neural Network (there are many of them), it is obvious that the proposed NFG Network has many input layers as opposed to conventional NeuralNetworks.

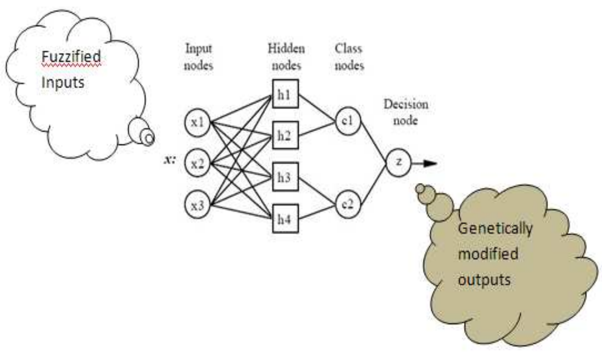


Fig. 3. Neuro – Fuzzy– Genetic Based Network

The input to output relationship depends on various parameters enumerated in the above table and this fact makes the network much more complex than the traditional ones. Associated with each input parameter, there is a set of connection weights. In our case there are 11 input parameters $(I_{x,} \ with \ x = 0 \ thru \ 10)$. The associated connection weights are W_{xk} with x = 0 thru 11 and k = 0 thru n-1. W_x is the set of weights associated with layer I_x and W_{xk} is the k^{th} weight in the set W_x. The weighted sum of the layer I_x is: $\sum_{x=0}^{n-1} I_{xn} * W_{xn}$. We propose a Neural Back Propagation Network with several input layers instead of just one. The input layers to the system are enumerated in the table mentioned above.

Corresponding to each input layer with a set of connection weights, the weighted sum is computed: $\sum_n I_k W_k$. This weighted sum, we will designate as: Q_k, where k is the index corresponding to the set of input layers. Assuming a set of N input layers, we have a set of N weighted sums: Q_k with $k \in 1 \ldots N$. Normal squashing functions used in Neural Networks have either sharp cutoff or exponential cutoff boundaries. Extensive experimentation and empirical studies have shown that the best form of squashing is achieved through the function:

$$Output - sqrt(e^{(x^2+y^2)})$$

Where the x and y values are chosen initially as 0.1 and updated through the training phase. Ultimately we have a set of N squashed values that are further processed in the route finding mechanism.

A Bayesian estimator is pressed into service at this stage to find the optimal solution. This estimator is an essential ingredient of the Fuzzy-Genetic component of the system. The Neural Network gives a number of feasible solutions, the Bayesian Estimator picks up the best possible solution out of the solution space.

The results have led us to conclude that substantial redesigning of protocols will be essential to make Ad Hoc Networks really useful. There are more of documentation bugs in NS2 and simulation on NS2 is somewhat unpredictable under certain conditions.

In consideration of the above limitations, we have used HyperNet. This simulator has better functionalities than NS2. Simulation of various existing and user-defined Protocol Stacks has been performed

A detailed report of the simulator results for both NS2 and HyperNet has been presented in for various types of standard protocols as well as our own protocol. The results show a remarkable improvement in performance of hypernet simulator over NS2

IV. CHARACTERISTICS OF SIMULATOR

A. *Characteristics of Network Simulator NS2*

While evolving a network topology and designing new protocols, it is customary to use a simulator which accepts the input parameters to the network structure, implements the protocol and using generates outputs relating to the performance of the protocol. There are a number of simulators available in the public domain and NS2 Simulator is quite popular. Most of the researchers in networking domain use NS2 for simulation and protocol evaluation purposes.

Strictly speaking, NS2 is designed to work for wired networks where the topology is well established in advance and the protocols are defined in advance. NS2 does not address some of the most important issues of Ad Hoc Networks – issues such as mobility of the nodes, the limitations in communication, power saving aspects and many others. Since these issues are extremely important in the evaluation of protocols for Ad Hoc Networks, NS2 is not strictly suitable as an evaluation tool. Nonetheless, we are using NS2 as a measure of comparison against our simulator HyperNet, taking into account its limitations.

B. *HyperNet Simulator*

HyperNet Simulator has been designed specifically for protocol evaluation of Ad Hoc Networks. Based on the study of various simulators available in the public domain, we designed the features of the simulator that would most aptly reflect the inner workings of an Ad Hoc Network. The following capabilities were incorporated into the simulator:

- Real Time Node Mobility evaluation
- Communication capabilities of the nodes
- Monitoring of Inactivity of the nodes and intermittent connections
- Power Saving issues during inactive periods
- Route optimization based on heuristic and Neuro-Fuzzy Algorithms
- And many other issues specific to Ad Hoc Networks

For protocol evaluation issues, Back Propagation Artificial Neural Networks with Fuzzy Logic & Genetic Algorithms have been used. These Soft-Computing ingredients make the simulator highly suitable in conducting realistic evaluation of existing and newly-designed protocols.

Training sets required to train the ANN were prepared on a rough estimate and these were later refined by generation of intermediate results. These refined training sets were used to train the network towards fine-tuning of the performance.

The following parameters were used for evaluation purposes:

- Maximum number of nodes limited to 200
- Timeout for on-demand connections programmed for 1200 milliseconds
- Mobility of the nodes restricted to a radius of 750 meters from initial position
- Power status monitoring wherever the nodes are capable of emitting this information
- Real Time monitoring of nodes' idling states
- Route activity information maintained in a knowledge base
- Inference Engine based on Back-Propagation ANN with Fuzzy Logic and Genetic Algorithms
- High amount of parallelism within the network activities
- Based on the above parameters, the results of protocol evaluation are being presented here below in the form of graph outputs. The graphs are depicted with the number of nodes in the network along the x axis with the routing time between 2 selected nodes represented along the y axis in milliseconds.

These graph outputs were obtained after 100 iterations through the Neural Networks, reflecting the averaged results in statistical sense. This approach affords maximum reliability in performance.

The results can be summarized in terms of connection time for a maximum of 200 nodes as shown in Table I

A few observations on the results can be made:

- The performance of routing is approximately logarithmic in nature
- Connection time increases with increase in the number of nodes
- The worst case performance is encountered in Power-Aware and Hybrid Protocols [10].
- The best performance is to be found in Neuro-Fuzzy and Neuro-Fuzzy-Genetic cases

The superior performance of Neuro- Fuzzy- Genetic Routing Algorithms comes with a price: a substantial overhead in preparation of training sets and training the simulator to achieve an equilibrium point. But this is really not an issue as the training phase is a one time effort.

Table I. Link establishment time for a maximum of 200 nodes using various protocols Using Hyper net simulator

Protocol	Maximum Link Establishment Time in milliseconds
Pro Active (Table Driven) Routing	60
Reactive (On Demand) Routing	35
Flow Oriented Routing	35
Power Aware Routing	70
Hybrid (Pro Active & Reactive	70
Neuro Fuzzy Optimal Routing	4
Neuro Fuzzy Genetic Routing	2.5

In contrast, the results generated from NS2 Simulator are summarized and it is as shown in Table II

Table II. Link establishment time for a maximum of 200 nodes using various protocols Using NS2 simulator

Protocol	Maximum Link Establishment Time in milliseconds
Pro Active (Table Driven) Routing	87
Reactive (On Demand) Routing	61
Flow Oriented Routing	62
Power Aware Routing	110
Hybrid (Pro Active & Reactive)	113
Neuro Fuzzy Optimal Routing	Not Available
Neuro Fuzzy Genetic Routing	Not Available

In conclusion, it appears reasonable to assume that the essential ingredients of ANN with Fuzzy Logic and Genetic Algorithms go a long way in improving the performance of protocol in very dramatic terms.

V. RESULTS

It is important to observe that the objective function of any protocol is to establish a link as quickly as possible, taking into account the various input constraints. Our goal has been to solve the objective function and establish a route within the shortest possible time. Our implementation surpasses traditional routing algorithms by implicitly taking into account all the network input parameters at the same time in reaching an optimal solution. These include: Varying number of nodes in the network, The mobility of the nodes across a geographical region Limitations in the communication capabilities of the nodes, Congested and blocked routes,Nodes that are currently active, Link failure history.

The results generated from the network are independent of the protocol philosophy that one chooses to follow.The chief merit in our implementation is the independency of our simulator to traditional protocol paradigms such 'pro-active', 'reactive', 'power-aware' and similar such principles.

A detailed report of the simulator results for HyperNet has been presented in for various types of standard protocols as well as our own protocol.

HyperNet Simulator has been designed specifically for protocol evaluation of Ad Hoc Networks. Based on the study of various simulators available in the public domain, we designed the features of the simulator that would most aptly reflect the inner workings of an Ad Hoc Network. The following capabilities were incorporated into the simulator: Real Time Node Mobility evaluation Communication capabilities of the nodes,Monitoring of Inactivity of the nodes and intermittent connections , Power Saving issues during inactive periods, Route optimization based on heuristic and Neuro-Fuzzy Algorithms.

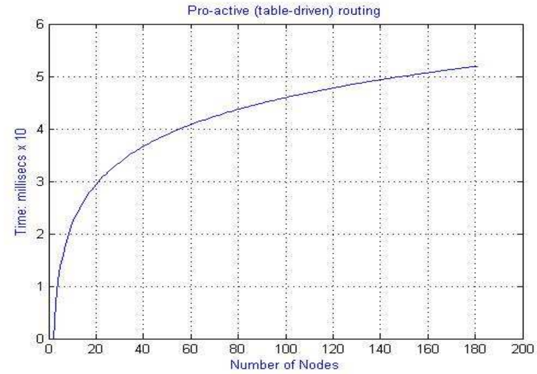
For protocol evaluation issues, Back Propagation Artificial Neural Networks with Fuzzy Logic & Genetic Algorithms have been used. These Soft-Computing ingredients make the simulator highly suitable in conducting realistic evaluation of existing and newly-designed protocols.

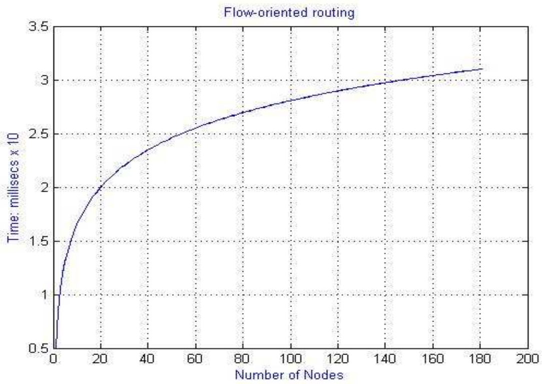
Training sets required to train the ANN were prepared on a rough estimate and these were later refined by generation of intermediate results. These refined training sets were used to train the network towards fine-tuning of the performance.

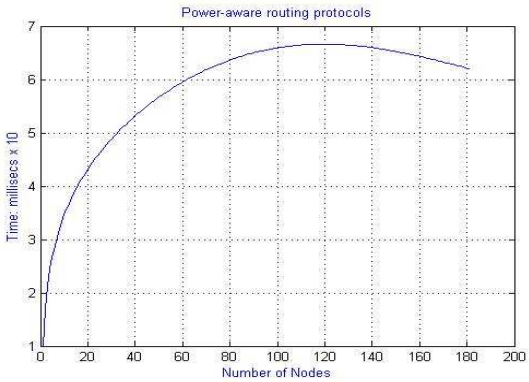
The following parameters were used for evaluation purposes: Maximum number of nodes limited to 200,Timeout for on-demand connections programmed for 1200 milliseconds Mobility of the nodes restricted to a radius of 750 meters from initial position, Power status monitoring wherever the nodes are capable of emitting this information, Real Time monitoring of nodes idling states, Route activity information maintained in a knowledge base, Inference Engine based on Back-Propagation ANN with Fuzzy Logic and Genetic Algorithms, High amount of parallelism within the network activities

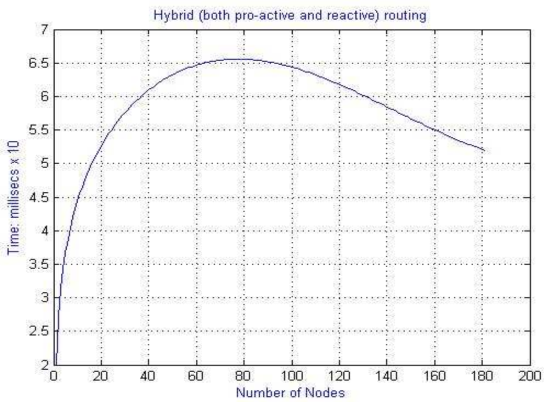
Based on the above parameters, the results of protocol evaluation are being presented here below in the form of graph outputs. The graphs are depicted with the number of nodes in the network along the x axis with the routing time between 2 selected nodes represented along the y axis in milliseconds.

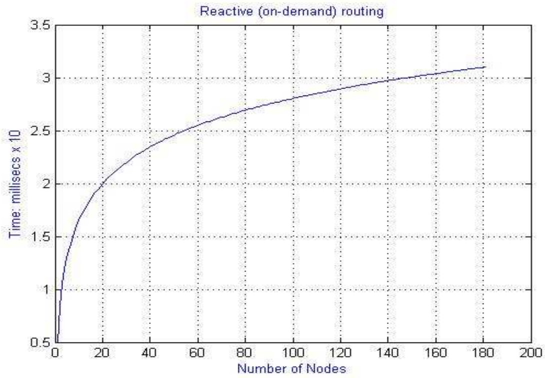
These graph outputs were obtained after 100 iterations through the Neural Networks, reflecting the averaged results in statistical sense. This approach affords maximum reliability in performance. A plot of Link Establishment time and Number of nodes using various protocols are as shown in Fig. 4

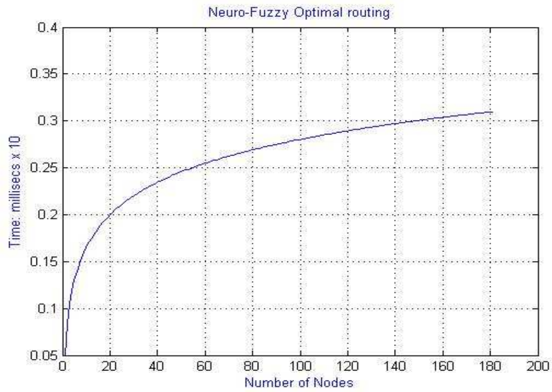












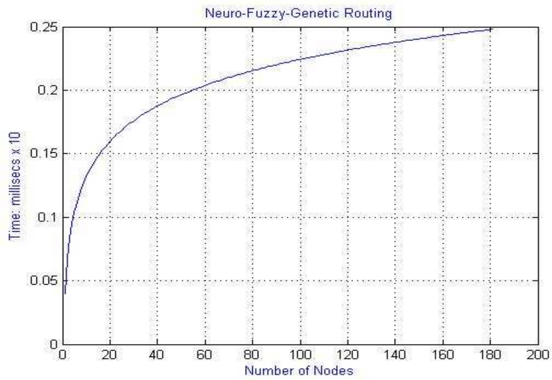


Fig. 4: A plot of Link Establishment time and Number of nodes using various protocols

VI. CONCLUSIONS

A few observations on the results can be made:

- The performance of routing is approximately logarithmic in nature and Connection time increases with increase in the number of nodes
- The best performance is to be found in Neuro-Fuzzy and Neuro-Fuzzy-Genetic cases

The superior performance of Neuro- Fuzzy- Genetic Routing Algorithms comes with a price : a substantial overhead in preparation of training sets and training the simulator to achieve an equilibrium point. But this is really not an issue as the training phase is a one time effort.

In conclusion, it appears reasonable to assume that the essential ingredients of ANN with Fuzzy Logic and Genetic Algorithms go a long way in improving the performance of protocol in very dramatic terms.

A careful study of the above graphs shows a superior performance of our protocol in establishing route over shorter periods of time and the results are encouraging.

REFERENCES

[1] F. Baker, "An outsider's view of MANET," Internet Engineering Task Force document ,17 March 2002.
[2] C. Barrett et al., "Characterizing the Interaction Between Routing and MAC Protocols in Ad-hoc Networks," Proc. MobiHoc 2002 , pp. 92-103
[3] J. Broch et al., "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols," Proc. Mobicom '98.
[4] D. Cavin et al., "On the accuracy of MANET simulators," Proc. ACM Workshop on Princ. Mobile Computing (POMC'02), Oct. 2002, pp. 38-43.
[5] K.W. Chin, et al., "Implementation Experience with MANET Routing Protocols," ACM SIGCOMM Computer Communications Review, Nov. 2002, pp. 49-59.
[6] M. S. Corson et al., "Internet-Based Mobile Ad Hoc Networking," IEEE Internet Computing, July-August 1999, pp. 63-70
[7] C. Elliott and B. Heile, "Self-Organizing, Self-Healing Wireless Networks," Proc. 2000 IEEE Int'l Conf. on Personal Wireless Comm., pp. 355-362.
[8] L. M. Feeney, "A Taxonomy for Routing Protocols in Mobile Ad Hoc Networks," Swedish Institute of Computer Science Technical Report T99/07, October 1999.
[9] M. Frodigh, et al, "Wireless Ad Hoc Networking: The Art of Networking without a Network," Ericsson Review, No. 4, 2000.
[10] David F. Bantz and Fr´ed´eric J. Bauchot. Wireless LAN design alternatives. IEEE Network, 8(2):43–53, March/April 1994.
[11] Vaduvur Bharghavan, Alan Demers, Scott Shenker, and Lixia Zhang. MACAW: A media access protocol for wireless LAN's. In Proceedings of the SIGCOMM '94 Conference on Communications Architectures, Protocols and Applications, pages 212–225, August 1994.
[12] Robert T. Braden, editor. Requirements for Internet hosts—communication layers. Internet Request For Comments RFC 1122, October 1989.

Siddesh.G.K. is an Assistant Professor in the Department of Electronics and Communication Engineering, Sri Bhagawan Mahaveer Jain College of Engineering, Jain University, Belgaum. He obtained his Bachelor degree in Electronics and Communication Engineering from Dayananda Sagar College of Engineering ,Bangalore University and Master degree in Digital Electronics and Advanced Communication from Manipal Institute of Technology, Manipal University formerly called as Manipal Academy of Higher Education. He is pursuing Ph.D. in Electronics and Communication Engineering, Visvesvaraya Technological University, Belgaum under the guidance of Dr.K.N.Muralidhara, Professor , Department of Electronics and Communication , P.E.S College of Engineering, Mandya. His research interest includes Computer Networks and Wireless Communication. He has altogether 3 intenational journals to his credit.

K N Muralidhara obtained his BE degree in Electronics and Communication from University of Mysore in 1981. He completed the ME and Ph.D. degrees in 1990 and 1998 respectively from Indian Institute of Technology, Roorkee (formerly known as University of Roorkee, Uttaranchal , India). At present, he is working as Professor and Head of the Dept. of Electronics and Communication Engineering, P.E.S College of Engineering, Mandya, Visvesvaraya Technological University, Karnataka, India,. He is guiding for research scholars for award of PhD degree from VTU, Karnataka.He has altogether 20 intenational/ national journals/ conferences to his credit. He is guiding 4 research scholars for PhD in the field of Wireless Communication and Low Power VLSI. His research interests includes in the areas of Wireless Communication, VLSI design, Solid State Electronics.

Manjula. N. Harihar is a lecturer in the Department of Electronics and Communication Engineering, Sri Bhagawan Mahaveer Jain College of Engineering, Jain University, Bangalore. She obtained her Bachelor degree in Electronics and Communication Engineering from S.T.J Institute of Technology, Ranebennur and Master degree in Communication Systems from P.D.A College of Engineering, Gulbarga, Karnataka, India. She is pursuing Ph.D in Electronics and Communication Engineering, Jain University, Bangalore. Her research interest includes Image Processing, VLSI, Neural Networks and Wireless Communication.