Better Quality of Service Management With Fuzzy Logic In Mobile Adhoc Network

Onifade O.F.W.
Department of Computer Science
University of Ibadan
Ibadan, Nigeria

Ojesanmi O.A.
Department of Computer Science
Federal University of Agriculture
Abeokuta, Nigeria.

Oyebisi T.O.
African Institute of Science Policy and Innovation
Obafemi Awolowo University
Ile-Ife, Nigeria.
dejioje@yahoo.com

ABSTRACT

Quality of service (QoS)is a great concept in mobile Adhoc network (MANETs). It is of a great importance that we are very conscious of how packet is routed to maximize efficiency and minimize delay. In this paper, an efficient algorithm for transmitting packet for better quality of service in adhoc mobile network was proposed. Fuzzy Self Organizing Map (FSOM) provide very efficient algorithmic tools for transmitting packet in an efficient manner by taking the most efficient route and also the bandwidth, latency and range are considered to determine how good is the data delivered. The results shown that fuzzy logic can guarantee QoS of every packets inthe network.

Keywords – QoS, Adhoc network, Packet, Fuzzy logic.

1. INTRODUCTION

The earliest mobile ad-hoc networks were called "packet radio" networks, and were sponsored by DARPA in the early 1970s [3]. Then, the advantages such as flexibility, mobility, resilience and independence of fixed infrastructure, elicited immediate interest among military, police and rescue agencies in the use of such networks under disorganized or hostile environments. For a long time, ad hoc network research stayed in the realm of the military, and only in the middle of 1990, with the advent of commercial radio technologies, did the wireless research community became aware of the great potential and advantages of mobile adhoc networks outside the military domain, witnessed by the creation of the Mobile Adhoc Networking working group within the IETF [18].

In the past few years, we have seen a rapid expansion in the field of mobile computing due to the proliferation of inexpensive, widely available wireless devices. However, current devices, applications and protocols are solely focused on cellular or wireless local area networks (WLANs), not taking into account the great potential offered by mobile ad hoc networking. A mobile ad hoc network is an autonomous collection of mobile devices (laptops, smart phones, sensors, etc.) that communicate with each other over wireless links and cooperate in a distributed manner in order to provide the necessary network functionality in the absence of a fixed infrastructure, with no central server, self organizing networks [9].

The routers are free to move randomly and organize themselves arbitrarily, and thus, the network's wireless topology may change rapidly and unpredictable. Manets has the chance of potential ease of deployment, decreased dependency on infrastructure [11][14]. The majority of applications of MANets are in areas where rapid deployment and dynamic reconfiguration are necessary and wired network is not available. They are applicable in universities, rescue sites, classrooms and conventions where freedom of location, mobility may be of paramount importance and where participants share information dynamically using their mobile devices [9].

Applications of mobile ad hoc networks have increased requirements in order to ensure high quality of service for the provided services and all these identified areas of application need quality of service for their connection lifetime [8][26]. Manets are facing constraints such as: limited bandwidth, power constraints, mobility, dynamic topology, high link error rate, Ease of snooping on wireless transmissions (security hazard), multi-hop routing and network scalability makes quality of service in Manets a challenging task. In Mobile Adhoc Networks (MANETs), determining the physical location of nodes (localization) is very important for many network services and protocols by [10][22].

The QoS satisfaction problem in wireless ad hoc network has been studied by many researchers. Recently, there has been research in the area of supporting QoS in MANETs. [1] did a study on qos provision for IP-based radio access networks.[2] described a model towards achieving QoS guarantees in Mobile Ad hoc Networks. [24], [13] and [19] proposed methods of supporting quality of service in mobile ad hoc networks. [17] describes a semi-stateless approach based on a fuzzy logic system for wireless mobile ad hoc networks. The works that exist tend to be based on distributed scheduling algorithms that address QoS routing issue, QoS-based medium access controllers, rescheduling when the network topology changes, and fairness issues.

The works in [4][5][6][12][15][16][18][21][23][25] have studied the QoS routing issue. Most of the existing distributed algorithms (e.g., [23]) require the maintaining of a global network state at every node, which may cause the scalability problem. On the other hand, the source routing schemes such as [5] suffer from problems of scalability and frequent updates of the state of the network. Hence the need to have a scheme that guarantees good QoS for the packets.

The paper is organized as follows. Next section discusses our architecture for quality of service management adhoc network using fuzzy logic. Section III presents the simulation study and performance evaluation, followed by conclusion in section IV.

2. SYSTEM ARCHITECTURE
2.1 Overview of the architecture
A system is a collection of various component which interact with one another in a manner that satisfy objective and goals according to a set of functional and performance specifications. There are two stages of design which is the system design and program design. The system design is the stage of system development which determine what the proposed system will do and how it would be done. While the program design deals with the mechanisms that best implements the solution. System design can be described as either a product or as a process. As a product, it is viewed as a product resulting in the transformation of the problem into solution. On the other hand, system design is two step process: the data model part and the process model part.
Process modeling: it shows how data flows among different processes and how they transform.

2.2 Network Learning Algorithm
In neural network structure, each output neuron directly corresponds to a class of trajectories. The number of output neurons used to describe the activity patterns is essentially arbitrary. The more neurons used the greater the accuracy of the model. The number of output neurons needed for good accuracy depends on the complexity of a scene. The more complex the scene is, the more output neurons are required. The number of output neurons is manually selected. The weights (W) connect the input vector components and the output neurons.

The weight vectors are of the same dimensions as the sample vectors. The weight components are initialized randomly and adjusted gradually using a self-organizing learning algorithm, and ultimately a mapping, from input to output, that keeps the distribution features of trajectories formed [15].

$$\begin{cases} (x_n + (x_n - x_{n-1}) * i, \, y_n + (y_n - y_{n-1}) * i, \, x_n - x_{n-1}, \, y_n - y_{n-1}), & i = 1, \ldots, c \\ (x_n + (x_n - x_{n-1}) * c, \, y_n + (y_n - y_{n-1}) * c, \, 0, 0), & i = c+1, \ldots, g-n, \end{cases}$$

Let M denote the number of input samples, N the number of input vector components and K the number of output neurons. The learning algorithm consists of the following steps.

Step 1
Randomize the initial values of the components of the weight vectors.

Step2
Input all samples

$$X_l \quad = \quad [X_{l,1}, X_{l,2}, \ldots, X_{l,N}], _{l\,=\,1,2;\,\ldots}$$

;M.

Step 3
Calculate the Euclidean distances from each sample X_l to all output neurons

$$d_{lj}(t) = \sqrt{\sum_{i=1}^{N}(X_{li} - W_{ij}(t))^2}$$
$$l = 1, 2, \ldots, M,$$
$$j = 1, 2, \ldots, K.$$

Step 4
Compute the memberships of each sample to all neurons

$$R_{lj}(t) = \frac{\frac{1}{d_{lj}^2(t)}}{\sum_{m=1}^{K}\left(\frac{1}{d_{lm}^2(t)}\right)}$$
$$l = 1, 2, \ldots, M,$$
$$j = 1, 2, \ldots, K. \qquad (3)$$

Step 5
Adjust the weights of each neuron according to the computed memberships

$$W_{ij}(t+1) = W_{ij}(t) + \frac{\sum_{l=1}^{M} R_{lj}(t) \cdot (X_{li} - W_{ij}(t))}{\sum_{l=1}^{M} R_{lj}(t)}. \qquad (4)$$

Step 6) Determine the stability condition of the network

$$\max_{\substack{1 \le i \le N \\ 1 \le j \le K}} \left\{ |W_{ij}(t+1) - W_{ij}(t)| \right\} < \varepsilon. \qquad (5)$$

If the stability condition is satisfied or the predefined number of iterations is achieved, then the learning process terminates; otherwise go to Step 2 for another loop of learning. From the above learning procedure, we can see that the fuzzy SOM eases the difficulty of selecting network parameters. The weights are adjusted only once in each learning loop and the features of all input samples are taken into consideration once the weights are adjusted, so the learning speed and estimation accuracy are both greatly improved. In fact, different kinds of fuzzy membership functions can be used in the above learning algorithm.

The generation of fuzzy membership function via SOFM has, so far been a two-step procedure. The first step generates the proper clusters. Then, the fuzzy membership function is generated according to the clusters in the first step. However, it is possible to integrate the two-step procedure and generate the fuzzy membership function directly during the learning phase. The main idea is to augment the input feature vector with the class labeling information.
The variables associated are semantic with the objective to cluster and visualize the data distribution. The focus was on how SOFM could be used to handle fuzzy information. Therefore, the information being associated are all fuzzy variables [7].

3. THE PROPOSED SYSTEM

The fuzzy SOM introduces the concept of membership function in the theory of fuzzy sets to the learning process in the batch manner. The membership of each sample to each neuron is calculated, and then the weight vector of each neuron is adjusted according to all the memberships of all samples to the neuron. In the fuzzy SOM, some network parameters related to the neighborhood in the SOFM are replaced with the membership function. So the burden of choosing network parameters is eased. Integration of the SOFM and other fuzzy set based algorithms can produce other variants of fuzzy SOM.

3.1 Fuzzy Inference Structure Model
The FIS Editor, figure 1, handles the high level issues for the system: How many input and output variables? What are their names? The Fuzzy Logic Toolbox doesn't limit the number of inputs. However, the number of inputs may be limited by the available memory of your machine. If the number of inputs is too large, or the number of membership functions is too big, then it may also be difficult to analyze the FIS using the other GUI tools. The Rule Viewer and the Surface Viewer are used for looking at, as opposed to editing, the FIS. They are strictly read-only tools.

The Rule Viewer is a MATLAB-based display of the fuzzy inference diagram shown at the end of the last section. Used as a diagnostic, it can show (for example) which rules are active, or how individual membership function shapes are influencing the results. The Surface Viewer is used to display the dependency of one of the outputs on any one or two of the inputs—that is, it generates and plots an output surface map for the system. The Membership Function Editor is used to define the shapes of all the membership functions associated with each variable.

Fig 1: Fuzzy Inference System Model

The steps involved in the design are:

Figure 2: Stages in Fuzzy inference system

3.2.1 Fuzzification
This is the process of generating membership values for a fuzzy variable using membership functions. The first step is to take the crisp input variables and determine the degree to which these inputs belong to each appropriate fuzzy set. This crisp input is always a numeric value limited to the universe of discourse. Once the crisp inputs are obtained, they are fuzzified against appropriate linguistic fuzzy sets.

3.2 Fuzzy Logic System Analysis
Fuzzy inference is the process of formulating the mapping from a given input to an output using fuzzy logic. The mapping then provides a basis from which decisions can be made, or patterns discerned. A fuzzy inference system is created based on the known sensitivity algorithm parameters to bandwidth, latency, range and this utilizes the Mamdani Fuzzy logic System. The singleton fuzzifier, the product operation fuzzy implication for fuzzy inference, and the center average defuzzifier will be used.

The parameter depending on their availability are fed into a fuzzifier in which are converted into fuzzy sets. A fuzzy set contains varying degree of membership in a set. The membership values retrieved for a particular variable into a membership function (see figure 2).
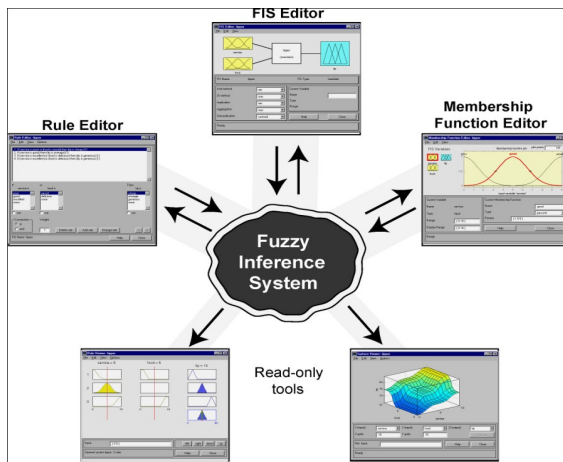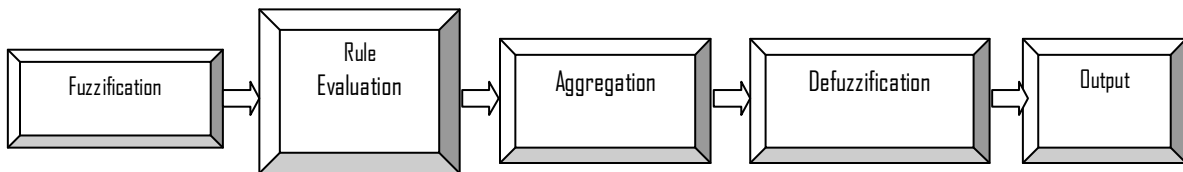
Membership function is designed for each quality of service condition which is a curve that defines how each point in the input space is mapped to a membership value (or degree of membership).

Input Variables with their value range
Bandwidth [0,10]
Latency [0,10]
Range [0,10]

Output Variables with their value range
Data delivery   [0,30]

3.2.2   Rule Evaluation

This is the second step where the fuzzified inputs are applied to the antecedents of the fuzzy rules. Since the fuzzy rule has multiple antecedents, fuzzy operator (AND or OR) is used to obtain a single member that represents the result of the antecedent evaluation. We apply the AND fuzzy operation (intersection) to evaluate the conjunction of the rule antecedents. Rules added to this system are derived by mapping the three inputs to one output by using conjunction (AND). Examples of some of the rules are:

(1) If (bandwidth is low) and (latency is low) then (data delivery is poor) (1)
(2) If (bandwidth is high) and (latency is low) then (data delivery is excellent) (1)
(3) If (bandwidth is high) and (latency is low) and (range is very close) then (data delivery is excellent) (1)
(4) If (bandwidth is low) and (latency is high) and (range is far) then (data delivery is poor)

3.2.3   Aggregation Rule Output

This is the process of unification of the outputs of all rules. In other words, we take the membership functions of all the rules consequent previously scaled and combine them into single fuzzy sets (output). Thus, input of the aggregation process is the list of scaled consequent membership functions and the output is one fuzzy set for each output variable (Name='data delivery' Type='mamdani', Numinputs=3, NumOutputs=1, NumRules=' ')

3.2.4   Defuzzification

This is the last step in the fuzzy inference process, which is the process of transforming a fuzzy output of a fuzzy inference system into a crisp output. Fuzziness helps to evaluate the rules, but the final output this system has to be a crisp number. The input for the defuzzification process is the aggregate output fuzzy set and the output is a number. This step was done using centroid technique because it is most commonly used method of Defuzzification (DefuzzMethod='centroid').

The fuzzy logic-based resource management modeling architecture is given in figure 3.
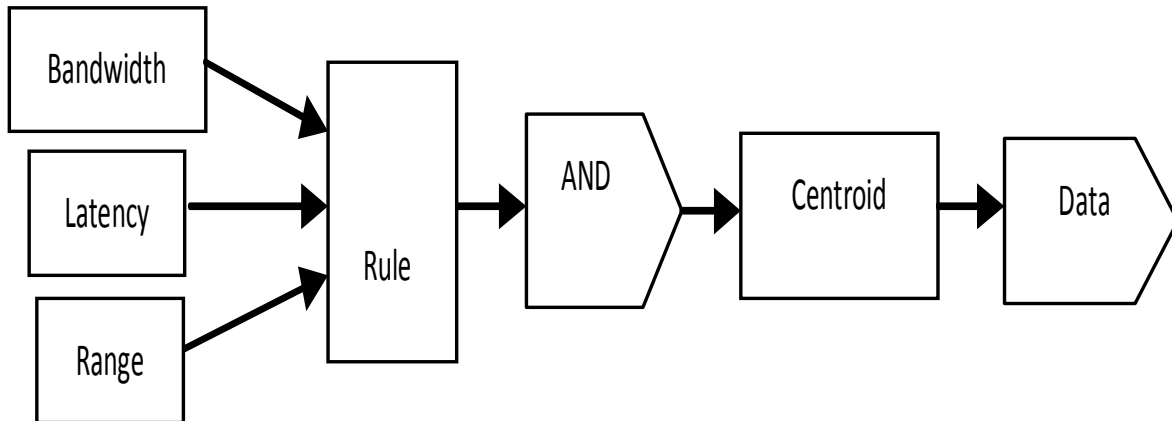

Figure 3: Architecture for fuzzy logic based Qos modeling.

4. EVALUATION

We can infer from figures 4 and 5 that the time of delivery varies based on the route taken to transmit packet. We can conclude from figure 6 that the time and bandwidth for figure 4 is better than that of figure 5.
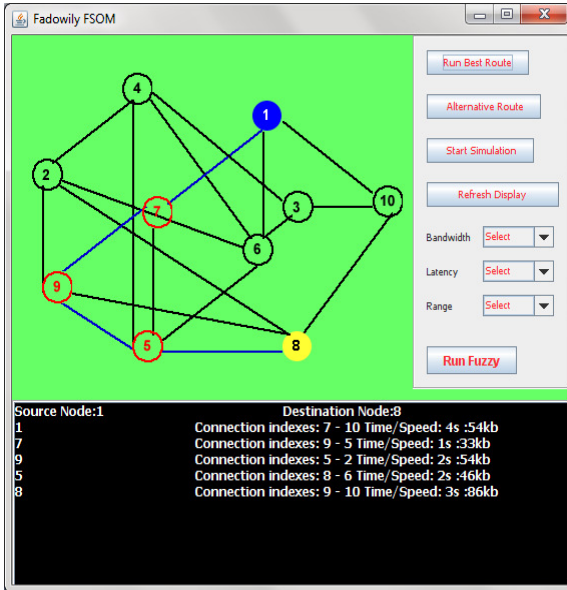
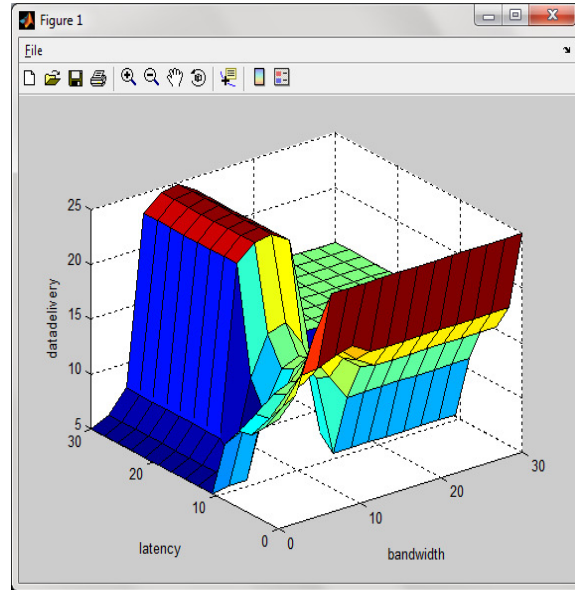**Figure 4: Interface for the simulation
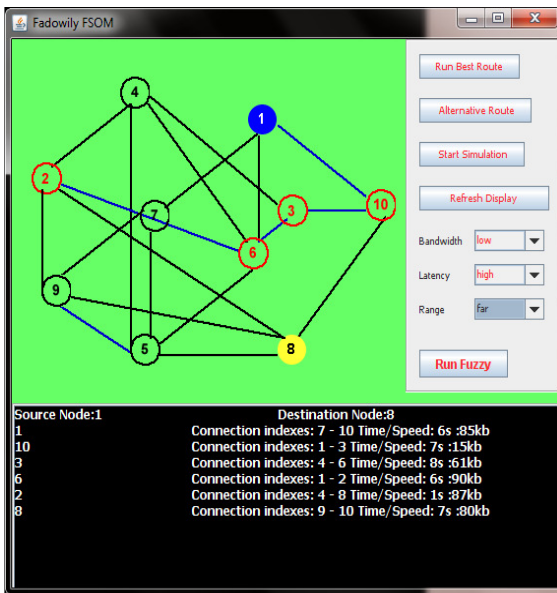of the best route.**

**Figure 5: Interface for the simulation of the
alternative route.**

**Figure 6: Surface viewer from the
simulation interface**

FIS Editor: Figure 7 is the window through which a new FIS type with any particular model can be selected, variable can be added, and input or output variable names can be changed. In this case, the chosen model is **Mamdani**. The pop-up menus in front of (And method to Defuzzification) are used to adjust the fuzzy inference functions, such as the defuzzification method. Double-click on an input or output variable icon to open the membership function editor, or on the system diagram to open the Rule Editor.
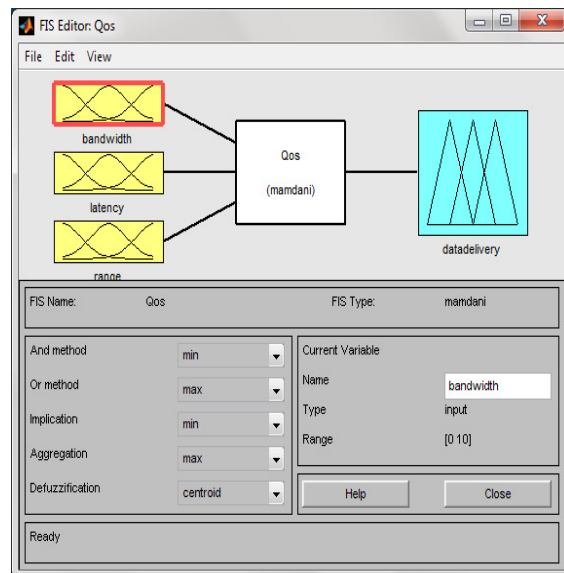

Figure 7: Fuzzy Inference System Editor

Membership function editor: Figure 8 is the window through which the input or the output of the membership function can be changed and membership function can be added or removed. The graph field displays all the membership functions of the current variable. Click on a line in the graph and it is possible to change any of its attributes, including name, type and numerical parameters. The pop-menu right in front of type lets you change the type of the current membership function. The status line describes the most recent operation.
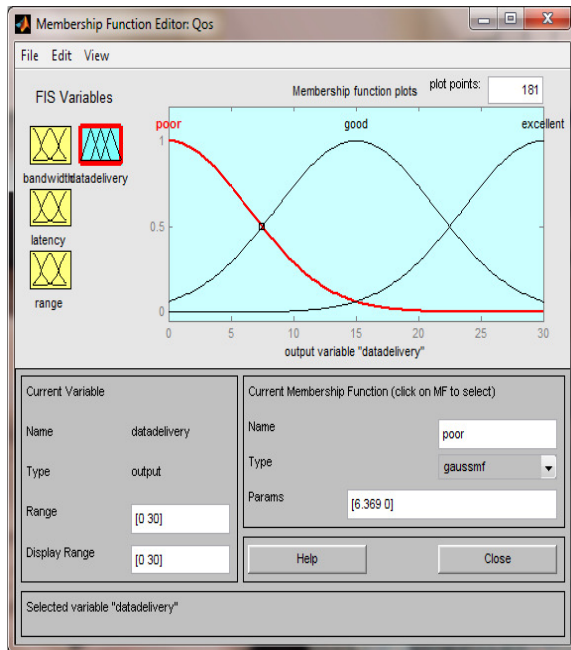

Figure 8: Membership function specification for Qos

Rule editor: Figure 9 is used to add, change or delete rules, as the name implies. The rules are entered automatically using the GUI tools. It provides opportunity to change the connections and weight applied to the rules (the default is always 1). Connection, link input statements in rules. The (Delete, Add, Change rule) Create or edit rules with the GUI buttons and choices from the input or output selection menus and the not negate input or output statements in rules.
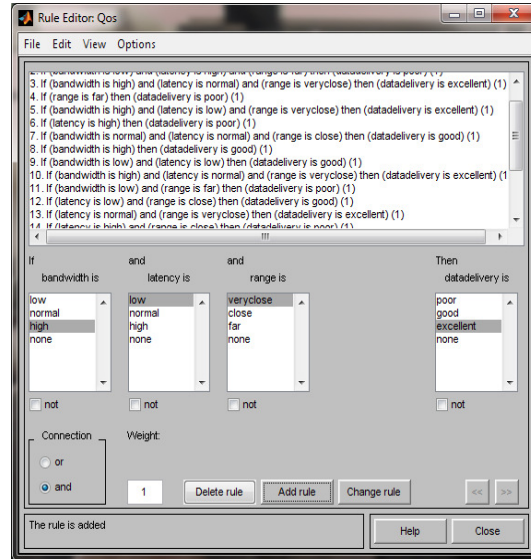

Figure 9: Rule Editor

Rule Viewer: The rule viewer in figure 10 displays a roadmap of the whole fuzzy inference process. It shows a graphical representation of each of the variable through all the rules, a representation of the combination of the rules, and a representation of the output from the defuzzification. It also shows the crisp value of the system. Data are entered for analysis through the rule viewer at the input text field.

Each column of plots (yellow) shows how the input variable is used in the rules. The input values are shown at the top, and the column of the plots (blue) shows how the output variable is used in the rules. Sliding the red line changes your input values, and generate a new output response although the edit field allows you to set the input explicitly. The last output plot, a blue triangle having a red line in between, the red line provides a defuzzified value, while the plot shows how the output of each rule is combined to make an aggregate output and then defuzzified.
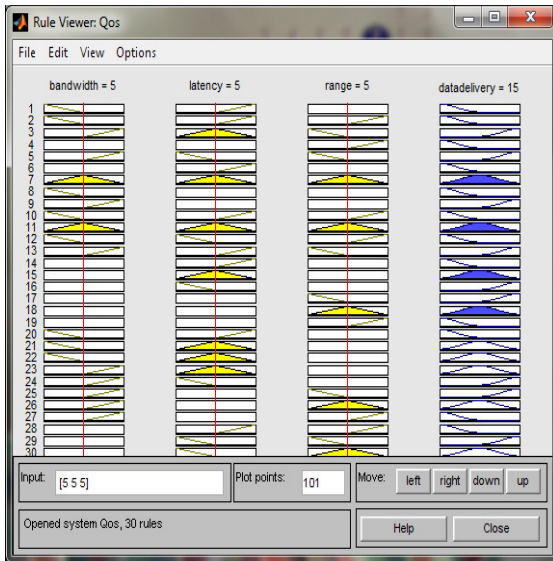
Figure 10: Rule Viewer for Qos

Surface Viewer: Figure 11 is a three dimensional curve that represents the mapping of three input. This is a three-input one-output systems as they generate three-dimensional plots that MATLAB can adeptly manage. When we move beyond three dimensions overall, we start to encounter trouble displaying the results. Accordingly, the surface viewer is equipped with pop-up menus that let you select any two inputs and any one output for plotting. Just below the pop-up menus are two text input fields that let you determine how many x-axis and y-axis grid lines you want to include. This allows you to keep the calculation time reasonable for complex problems.

Clicking the evaluate button initiates the calculation and the plot comes up soon after the calculation is complete. To change the x-axis or y-axis grids or Y-grids, according to which text field you changed, to redraw the plot. The surface viewer has a special capability that is very helpful in cases with two (or more) inputs and one output: you can actually grab the axes and reposition them to get a different three-dimensional view on the data. The Ref. input field is used in situations when there are more inputs required by the system than the surface is mapping. Suppose you have a five-input one-output system and would like to see the output surface.

The surface viewer can generate a three-dimensional output surface where any two of the inputs vary, but two of the inputs must be held constant since computer monitors cannot display a five-dimensional shape. In such a case the input would be a four-dimensional vector with NaNs holding the place of the varying inputs while numerical values would indicate those values that remain fixed, A NaN is the IEEE symbol not number.
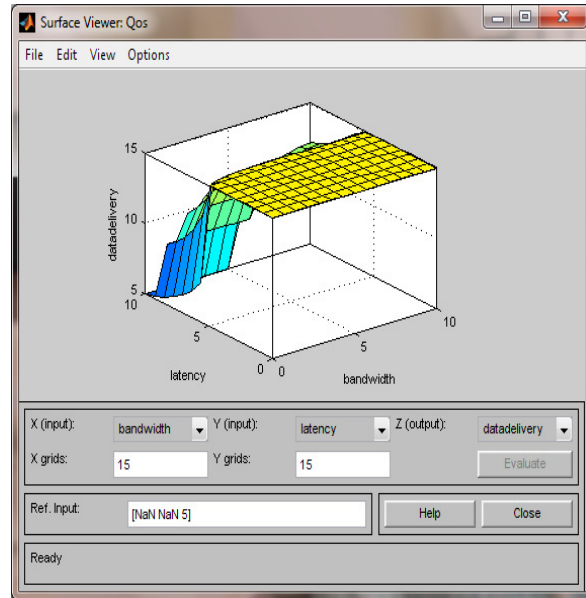

Figure 11: Surface Viewer

CONCLUSION
Fuzzy Self Organizing Map (FSOM) has been developed in this paper. Incorporation of fuzziness in the input and output of the proposed model was seen to result in better performance. It should be noted that input variables are only three properties i.e. low, normal, and high were used and for the output the variables are poor, good and excellent.

REFERENCES

[1] Alberto L., Jukka M., Andrej M., Hector V., Eleanor H., and Youssef K. (1999), "A study on qos provision for ip-based radio access networks", IST project IST-1999-10050 BRAIN.

[2] Arora, H. and Shetu, H. (2003). October "Towards achieving QoS guarantees in Mobile Ad hoc Networks", M.Sc Thesis, Faculty of Drexel University.

[3] Carlos D. and Agrawal D.P. (2006) "Adhoc & Sensor Networks: Theory and application', 2nd Edition, World Scientific Publishing Company, pp. 1-641.

[4] Chansu Y., (2007)." Routing in Mobile Adhoc Network", Cleveland State University.

[5] Chen S. and Nahrstedt K. (1998), "On finding multi-constrained paths", IEEE International Conference on Communication, pp. 874 -879.

[6] Chen S. and Nahrstedt K. (1999), "Distributed Quality-of-Service in Ad Hoc Networks", IEEE Journal on Selected Areas in Communications, vol.17, no. 8.

[7] Chih-Chung Y. and Bose, N.K. (2004), "Generating fuzzy membership function with self-organizing feature map" Department of Electrical Engineering, Spatial and Temporal Signal Processing Center, the Pennsylvania State University, USA.

[8] Demetris Z. (2000). "A Glance at QoS in Mobile Ad-Hoc Networks (MANETs)", Department of Computer Science, University of California, http://www.cs.ucr.edu/~csyiazti/cs260.html

[9] Hoebeke, J., Moerman, I, Dhoedt, B., and Demeester, P (2005), "Towards adaptive ad hoc network routing", *International Journal of Wireless and Mobile Computing*: Special Issue on 'Wireless AdHoc Networking' to be published.

[10] Jeroen H., Ingrid M., Bart D. and Piet D., "An Overview of Mobile Ad Hoc Networks: Applications and Challenges", Department of Information Technology (INTEC)", Ghent University – IMEC, Belgium.

[11] Jukka M., Alberto L., Andrej M., Hector V., Eleanor H., and Youssef K. (2002), "Evaluation of Mobility and Qos Interaction", Elsevier Journal of Computer Networks, 38, pp. 137–163.

[12] Khoukhi L., Cherkaoui S. (2004), "Flexible QoS Routing Protocol for Mobile Ad Hoc Networks", In Proc. of the 11th IEEE International Conference on Telecommunication (ICT2004), Brazil.

[13] Kui W. and Janelle H. "Qos support in mobile ad hoc networks", Department of Computing Science, University of Alberta,

[14] Kurt G. (2002)." Analysis of adaptation strategies for mobile qos-aware applications", M. Sc Thesis, Drexel University.

[15] Liao W.-H., Tseng Y.-C., Sheu J.-P., and Wang S.-L. (2001), "A Multi-Path QoS Routing Protocol in a Wireless Mobile AdHoc Network", In Proc. of IEEE ICN'01, International Conference on Networking, Part II, pp. 158–167.

[16] Lin C. R. and Liu J.-S.. (1999), "QoS Routing in Ad Hoc Wireless Networks", IEEE Journal on Selected Areas in Communication, Vol. 17, No. 8, 1426–1438.

[17] Lyes K. and Soumaya C. (2008), "Experimenting with Fuzzy Logic for QoS Management in Mobile Ad Hoc Networks", International Journal of Computer Science and Network Security (IJCSNS), Vol.8, No.8, pp. 372-386.

[18] Munaretto A., Badis H., Al Agha K. and Pujolle G. (2002), "A Linkstate QoS Routing Protocol for Ad Hoc Networks", In the proceedings of IEEE MWCN02, Stockholm, Sept.2002.

[19] Patrick S., (2003). "Quality of Service for Mobile Ad Hoc Networks", Diploma Thesis, Eidgenossiche Technische Hochschule, Swiss Federal Institute of Technology, Zurich.

[20] Perkins, C.E. and Royer, E.M. (2000). Quality of Service for Ad Hoc on Demand Distance Vector Evaluation of the insignia signaling system. For ad hoc wireless networks (AODV) Routing, IETF Internet Draft, draft-ietf-manet-aodvqos-00.txt.

[21]   Sivakumar R. et al. (1999), "CEDAR: a core extraction distributed ad hoc routing algorithm", IEEE Journal on Selected Areas in Communications, vol. 17, no. 8, pp. 1454-1465.

[22]   Tinh, P. D., Noguchi, T. and Kawai, M. (2009)." Distributed Range-Free Localization Algorithm Based on Self-Organizing Maps", Graduate School of Science and Engineering, Ritsumeikan University, Japan.

[23]   Wang Z. and Crowcroft J. (1996), "QoS routing for supporting resource reservation", IEEE Journal on Selected Areas in Communications, vol. 14, no. 7. http://www.meshnetworks.com.

[24]   Xiao H., Seah W. K.G., Lo A. and Chua K. C. (2000), "Flexible Quality Service Model for Ad-HOC Networks", In IEEE Vehicular Technology Conference, pp. 445–449, Tokyo, Japan. htpp://www.ece-icr.nus.edu.sg/journal1/ fqmmhandbook02.pdf.

[25]   Xue Q. and Ganz A. (2003), "Ad hoc QoS on-demand routing (AQOR) in mobile ad hoc networks", Journal of Parallel and Distributed Computing, Elsevier Science, USA.

[26]   Zeinalipour-Yazti D., (2001) "A Glance at Quality of Services in Mobile Ad-Hoc Networks," Seminar in Mobile Ad hoc Networks, Department of Computer Science, University of California-Riverside, USA.