Computer Network Routing Using Fuzzy Neural Networks

Khloud Ahmad Nasar ***Turki Y. Abdalla*** ***Abdulkareem Y. Abdalla***
Computer Science Dept. ***Computer Engineering Dept.*** ***Computer Science Dept.***
University of Basrah, Basrah, Iraq

Abstract

Fuzzy neural networks can be used efficiently as a solution technique for routing problem in computer networks, to improve the performance. In this paper, two methods are proposed to solve this problem. The fuzzy neural networks are used to make routing decision, and to mange the congestion with solving of the routing problem. The proposed methods are applied for typical examples of computer networks. The fuzzy neural networks are trained. Results of the testing assert their high achievement.

1. Introduction

In a communication network information is transferred from one node to another as data packets. Packet routing is a process of sending a packet from its source node (s) to its destination node (d). On its way, the packet spends some time waiting in the queues of intermediate nodes while they are busy processing the packets that came earlier. Thus the delivery time of the packet, defined as the time it takes for the packet to reach its destination, depends mainly on the total time it has to spend in the queues of the intermediate nodes. Normally, there are multiple routes that packet could take, which means that the choice of the route is crucial to the delivery time of the packet for any (source, destination) pair [1].

Successful operation of data communication network is critically dependent on the provision of an adequate routing algorithm. Routing algorithms are methods for finding the best way from a node s to another node d. This may be via a large number of other nodes or it may be in the next subnetwork. On a small, simple network the problem is almost trivial, statically allocating routes and defining them by hand, but when dealing with a huge internetwork such as the Internet this is not possible. It heavily interconnected network has many routes from one node to another, and these routes span many different types of link with different bandwidth and latency characteristics. Calculating the best route through such a complex system is computationally intractable and impossible to do by hand [2 , 3, 4].

If part of network becomes over_ filled with packets it can become impossible for packets to move. The queues into which they should be accepted are always full. This is called congestion. Routing algorithms strongly interact with congestion [2, 5].

Considerable research has been devoted toward solving the routing problem in the computer networks. Paper of J. Zuo, S. Ng and L. Hanzo [6] propose fuzzy logic based method to specific route having the highest route stability is finally selected for data transmission. L. Lertsuwanakul [7] compare the shortest path routing and probability functions using deterministic or adaptive approach. M. Dehyadegari and S. Mohammadi [8] propose routing method using two multimedia applications and a random traffic profile. A. Haboush [9] investigates three metrics, the mean link bandwidth, queue utilization and the mean link delay to solve routing problem. A. Dana and N. Salehi [10] propose routing algorithm is choosing channel that has more free-slots input buffer beyond adjacent routers and the less number of active requester for a given output port. M. Chelliah and B. Sivaselvan [11] provide an efficient routing method is a fuzzy multi-constraint AODV routing for wireless mesh networks.

2. Routing Using Neural Networks and Fuzzy Logic

In computer networks, adaptive routing methods are recommended in order to efficiently select the transmission path of the packets without causing network congestion. Routing algorithms must be adapted to the traffic and topology changes. The information regarding these changes have inherent uncertainty because it is at least as old as the propagation delay between nodes [12, 13].

The traditional network routing algorithms have not attempted to deal with this uncertainty in the information available at a node. Consequently, these algorithms often make poorer decisions when trying to route to distant nodes. Also, they have incapability to take into account possible node or link failures. Furthermore, their implementation requires complex and lengthy calculations [4, 13].

The parallel, distributed processing structure of the neural networks and their ability to learn are justified to use it as available structure for solving the routing problem [4].

Uncertain notions are described and implemented by fuzzy set theory. Therefore, in fuzzy logic an element can reside in more than one set to different degrees of similarity, and a formalism for implementing expert or heuristic rules is provided as fuzzy systems which are transparent. Then, the fuzzy systems can be integrated in complete routing systems [13].

The main idea of fuzzy neural networks is to combine the advantages of fuzzy systems which are interpretability and use of prior knowledge for initialization with the learning capabilities of neural networks [14]. Therefore, the fuzzy neural networks provide a high level of the performance to make routing decision.

3. Proposed Methods

This paper presents two proposed methods to use fuzzy neural networks for solving the routing problem in the computer networks. Fuzzy neural network is located at each node of the computer network to make a local routing decision. In other method, the congestion control is combined with solving of the routing problem.

The computer networks which are considered in this paper modeled as graphs. Two examples, the first is a 9- node mesh computer network (CN1) shown in Figure (1). While the second is a random computer network (CN2) shown in Figure (2). The imposing values of cost of the links (packet delay) and queue length of the nodes for these two computer networks are shown in Tables (1) and (2), respectively. The values of the queue lengths in these Tables are located under changing continuously, when the nodes send or receive the packets.

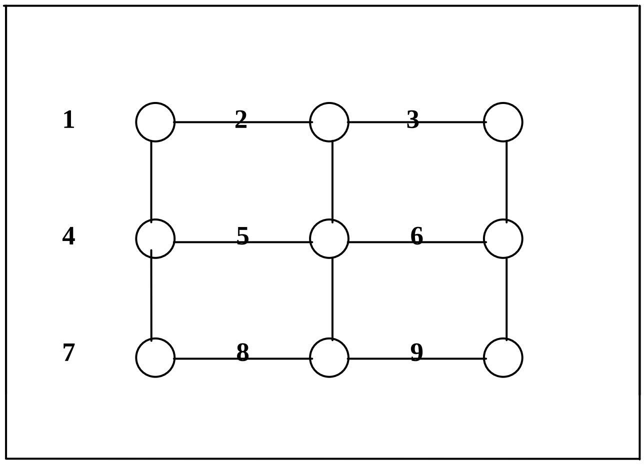

Figure (1) computer network (CN1)

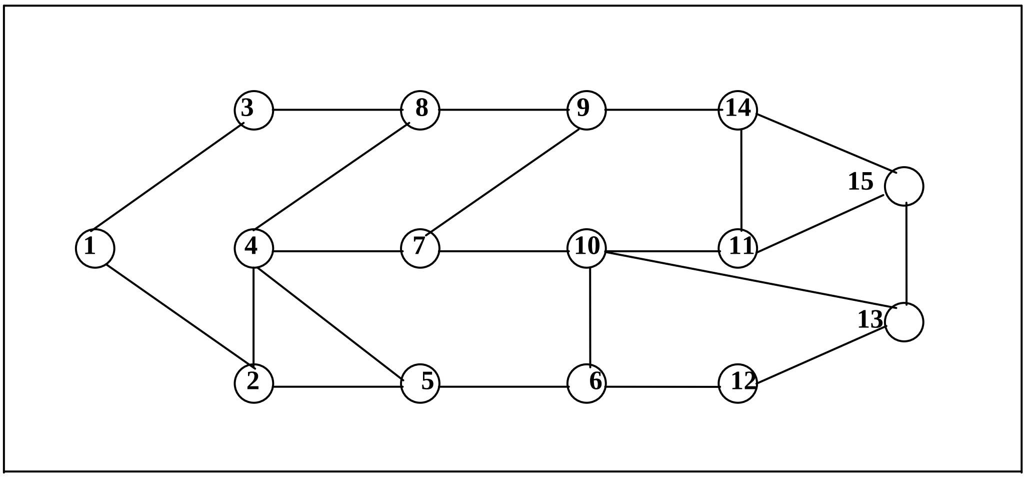

Figure (2) computer network (CN2)

Table (1) link costs and queue lengths of the computer network (CN1)

Links		Nodes	
link	**cost (packet delay) (in second)**	**node**	**queue length (in packet)**
1─2	10	1	6
1─4	2	2	5
2─3	3	3	8
2─ 5	15	4	4
3─ 6	8	5	7
4─ 5	2	6	7
4─7	1.6	7	4
5─ 6	1.2	8	6
5─ 8	2.1	9	6
6─ 9	1.5		
7─ 8	3		
8─ 9	1		

Table (2) link costs and queue lengths of the computer network (CN2)

Links		Nodes	
link	**cost (packet delay) (in second)**	**node**	**queue length (in packet)**
1—2	6.312	1	5
1—3	6.312	2	5
2—4	1.544	3	7
2—5	6.312	4	7
3—8	6.312	5	4
4—5	3.352	6	8
4—7	6.312	7	8
4—8	3.352	8	5
5—6	12.624	9	6
6—10	3.352	10	5
6—12	6.312	11	4
7—9	3.152	12	6
7—10	3.152	13	3
8—9	6.312	14	4
9—14	12.624	15	6
10—11	3.152		
10—13	3.152		
11—14	3.152		
11—15	3.152		
12—13	6.312		
13—15	6.312		
14—15	6.312		

3.1. Routing Decision

A fuzzy neural network (FN1) is designed as a part of system which also consists of central monitor, to solve the routing problem locally. It has three inputs, they are link time cost, queue time cost and neighbor_ destination path length, and two outputs, they are neighbor time cost and destination path level. Where link time cost is the packet delay on link to the one neighbor nodes, queue time cost is the waiting time of packet in queue of the one neighbor nodes, neighbor_ destination path length is the distance (number of the nodes) between the neighbor node of the router and the destination, neighbor time cost is the cost of sending packet by one neighbor node and destination path level is level of evaluating the suitability of the path from the router to the destination by the neighbor node. With h units of product and normalization layers where h = 48 is the number of rules, as shown in Figure (3).

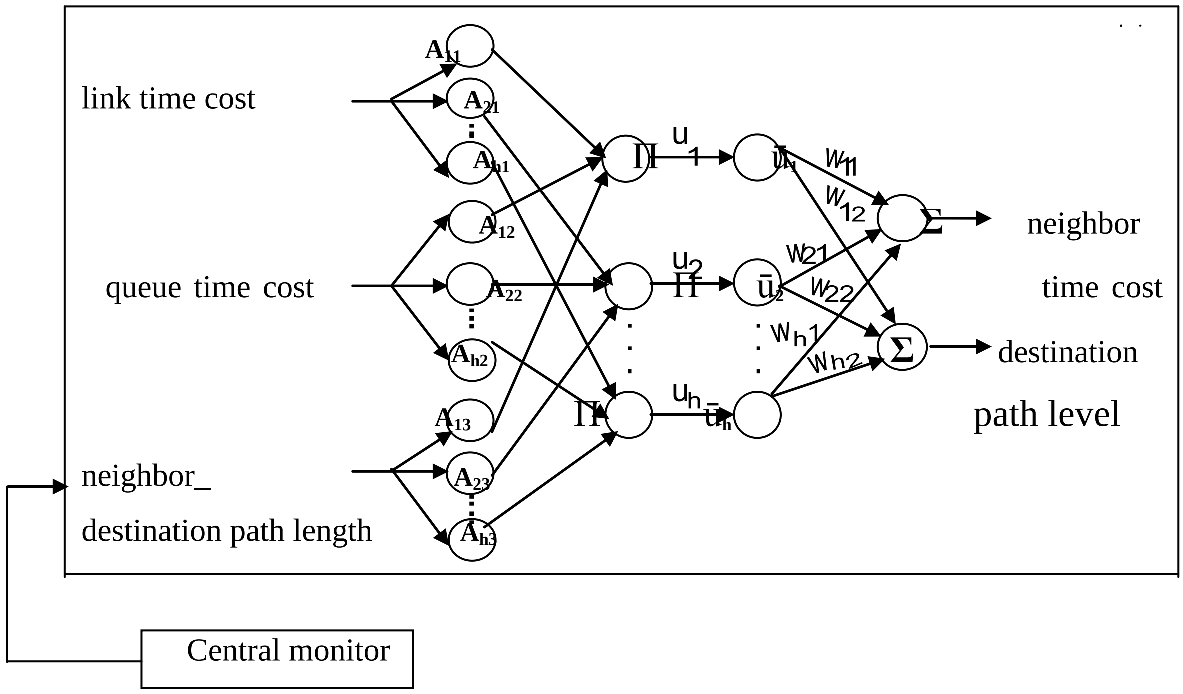

Figure (3) the fuzzy neural network (FN1)

The computer network contains the fuzzy neural network (FN1) at each node. When, node (j) has packet to send to a given destination node, the fuzzy neural network receives time cost of link and time cost of queue of the one neighbor nodes, and also receives length of the path from this neighbor node to the destination node from the central monitor. It decides the time cost of this neighbor node for sending packet by it and level of the path to the destination node. In r_j_ steps, the node (j) determines the best neighbor node, where r_j is a number that represents the number of neighbor nodes to node (j). For example, in the computer network (CN1), if node 7 has packet to send to node 5. The fuzzy neural network in node 7 receives time cost of link, time cost of queue of the one neighbor nodes (4, 8) and path length from this neighbor node to destination 5, and determines the time cost of this neighbor node and level of the path to destination 5 by this neighbor node. Then, the routing system of node 7 determines its best neighbor node to reach destination 5 quickly. The same process is repeated at each node successively till the destination node.

3.2. Routing and Congestion Control

For solving the routing problem with congestion control, the system is designed as a combination of two fuzzy neural networks. The first, fuzzy neural network (FN2) has two inputs, packet average level and packet variance level, and one output, the congestion prediction level. Where packet average level is level of estimating the average number of packets of a node, packet variance level is level of estimating the variance of packets of a node and congestion prediction level is level of determining the degree of congestion of a node, as shown in Figure (4) (where h = 16).

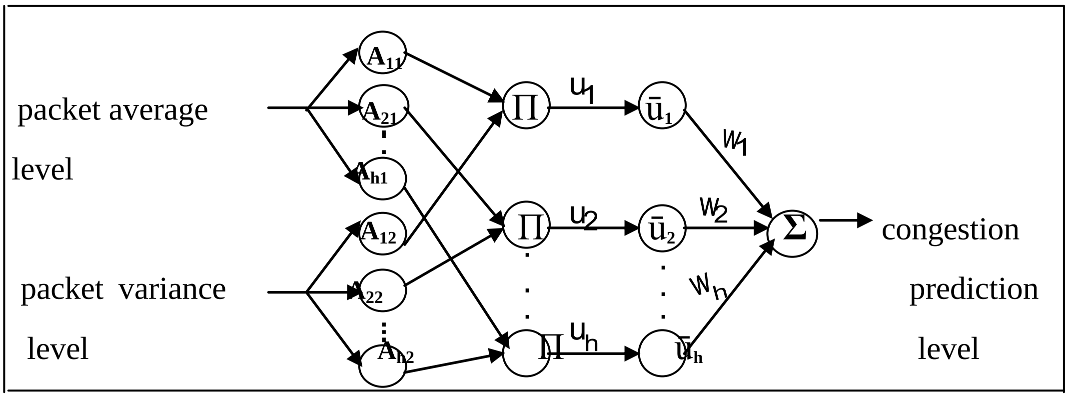

Figure (4) the fuzzy neural network (FN2)

The second, fuzzy neural network (FN3) has two inputs, link time cost and congestion prediction level which is the output of the first part (FN2), and one output, the neighbor selection level. Where neighbor selection level is level of evaluating the suitability of the selected neighbor node, as shown in Figure (5) (where h = 16).

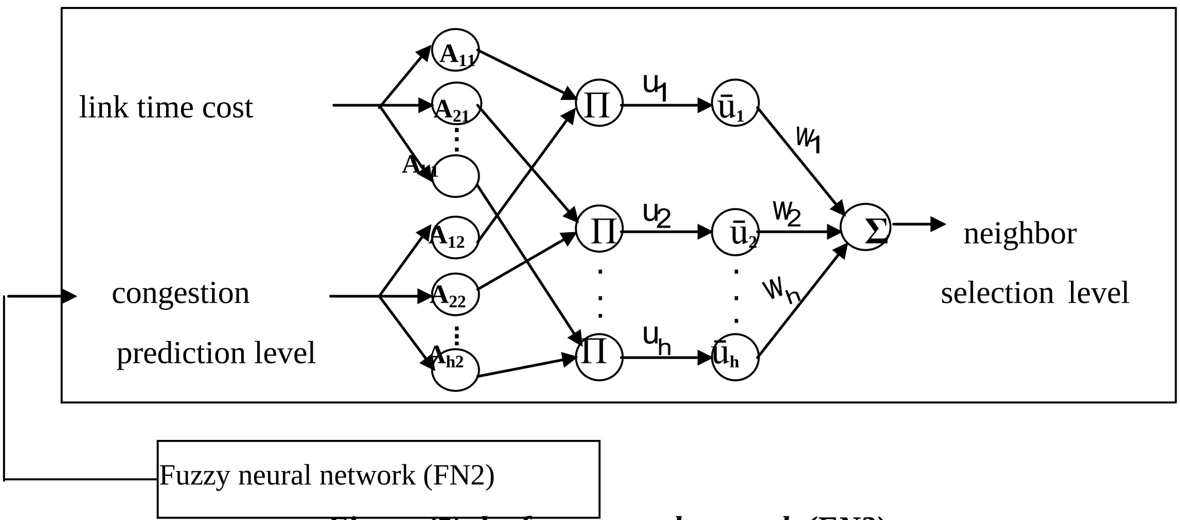

Figure (5) the fuzzy neural network (FN3)

The routing system, that consists of two fuzzy neural networks is included at each node of the computer network. In order for, node (j) to send packet to a given destination node, the congestion predictor (FN2) receives statistics data (average number of packets and variance of packets) of the one neighbor nodes and gives its output about the congestion state of this neighbor node to the other part (FN3) to determine the selection level of this neighbor node. In r_j_ steps, the system determines the best neighbor node to node (j). For example, in the computer network (CN2), if node 7 has packet to send to node 15, its congestion predictor (FN2) reports the congestion state of the one neighbor nodes (4, 9, 10) to the fuzzy neural network (FN3) which decides the selection level of this neighbor node. So, the selection level of each neighbor node can be determined. Then, the routing system at node 7 selects best neighbor node. Which has less value of congestion prediction with less cost of link. The same process is repeated at each node till the destination node.

3.3. Simulation Result

The simulation has been realized using C++ programming language, in order to test the proposed methods, which use fuzzy neural networks to solve the routing problem. Through that, they are applied for the two computer networks (CN1, CN2).

The proposed fuzzy neural networks have numbers of inputs, outputs, and units of product and normalization layers as described in Table (3). For setting the parameters (weight, center and width) of these fuzzy neural networks the following method of initialization is used [15].

Step1_ For k = 1, ..., h-2, the training sets $(x_1(k), ..., x_m(k), y_1(k), ..., y_p(k))$ which are inputs and desired outputs are presented, where h is the number of rules, m is the number of inputs, p is the number of outputs.

Step2_ For k = 1, ..., h-2, i = 1, ..., m, j = 1, ..., p, the center $c_{ki} = x_i(k)$ and the weight $w_{kj} = y_j(k)$. For k = h-1, the center $c_{ki} = a_i$ and the weight $w_{kj} = 1$, and for k = h, the center $c_{ki} = b_i$ and the weight $w_{kj} = 1$, where a_i is the lower bound and b_i is the upper bound of the universe of discourse $[a_i, b_i]$ of the input x_i.

Step3_ For k = 1, ..., h, i = 1, ..., m, the width $\sigma_{ki} = \max(|c_{ki} - c_{ri}|, |c_{ki} - c_{li}|) / \sqrt{|\ln \lambda_i|}$, where c_{ri} is the center of membership function at right side for input variable x_i, c_{li} is the center of membership function at left side for input variable x_i, and λ_i is a factor, $0 < \lambda_i < 1$.

The membership function used to perform a fuzzification operation of the proposed fuzzy neural networks is Gaussian function as in equation:

$$A_{ij} = e^{-\frac{1}{2}((x_j - c_{ij})/\sigma_{ij})} \quad (1)$$

where x_j is the input variable, for (j = 1, ..., m) inputs, c_{ij} and σ_{ij} denote the mean (center) and variance (width) with respect to A_{ij}.

These fuzzy neural networks are trained by backpropagation algorithm, when the parameters (weight, center and width) are adjusted to minimize the error function. For every one of the computer networks (CN1, CN2), 30 training sets are taken. The training operation is continue, until the minimum value of mean squared error is obtained. For that, some values of learning rates (η_w, η_c, η_σ) and momentum rates (α_w, α_c, α_σ) are employed to get best convergence which are selected by trial and error, they are exposed in Table (4). Results obtained at this stage proof decreasing of the mean squared error with increasing of number of epochs, as shown in Figures (6)_ (11).

The testing is performed on the trained sets, and on other test sets for each of the computer networks (CN1, CN2). Some of the results of testing are exposed in Tables (5)_ (9). Results are summarized in Tables (10) and (11) which show the efficiency of the performance of the fuzzy neural networks through the testing. These Tables describe the success rates to test trained sets and test sets.

From the simulation results, the following points are noticed.

The structure of each proposed fuzzy neural network is not related to the size of the computer network. The inputs and the outputs of each proposed fuzzy neural network are information about the one neighbor nodes in the computer network.

Table (3) the number of input, output and units of product and normalization layers of the proposed fuzzy neural networks

Fuzzy neural network	Number of inputs	Number of outputs	Number of units of product and normalization layers
FN1	3	2	48
FN2	2	1	16
FN3	2	1	16

Table (4) learning rates and momentum rates of the proposed fuzzy neural networks

Fuzzy neural network	Learning rate (η)			Momentum rate (α)		
	η_w	η_c	η_σ	α_w	α_c	α_σ
FN1	0.9	0.9	0.9	not used		
FN2	0.9	0.9	0.9	0.6	0.6	0.6
FN3	0.9	0.9	0.9	0.6	0.6	0.6

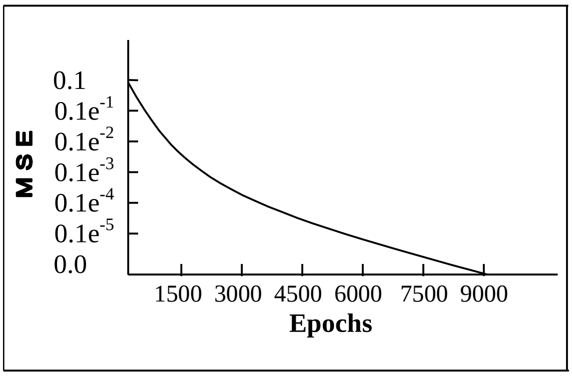

Figure (6) error versus number of epochs of the fuzzy neural network (FN1) for the computer network (CN1)

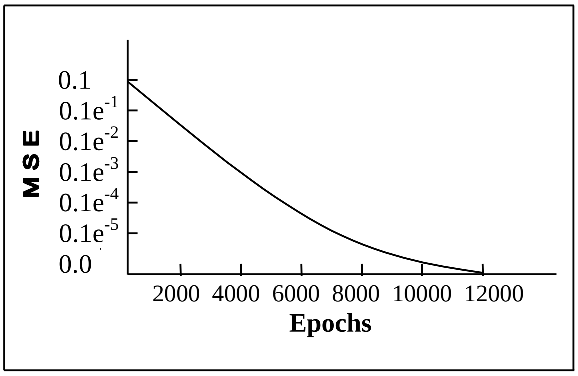

Figure (7) error versus number of epochs of the fuzzy neural network (FN1) for the computer network (CN2)

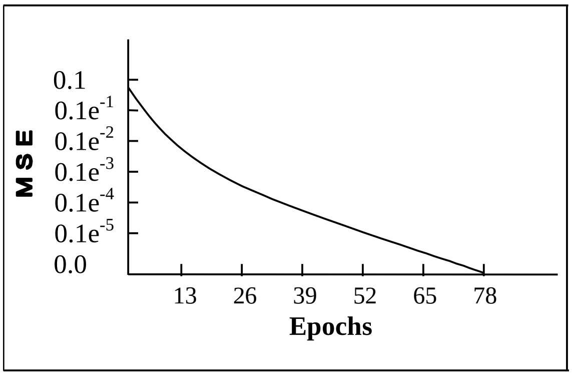

Figure (8) error versus number of epochs of the fuzzy neural network (FN2) for the computer network (CN1)

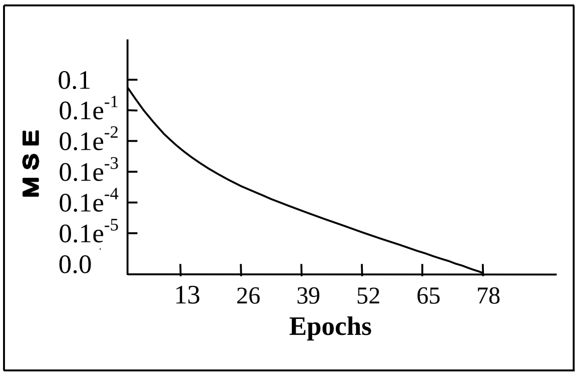

Figure (9) error versus number of epochs of the fuzzy neural network (FN2) for the computer network (CN2)

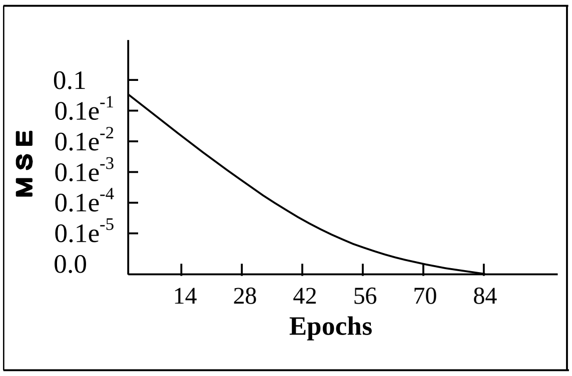

Figure (10) error versus number of epochs of the fuzzy neural network (FN3) for the computer network (CN1)

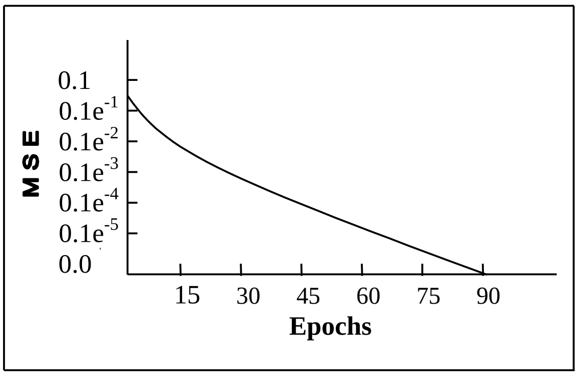

Figure (11) error versus number of epochs of the fuzzy neural network (FN3) for the computer network (CN2)

Table (5) some of the test results of the fuzzy neural network (FN1) for the computer network (CN1)

Inputs of FN1			Desired outputs		outputs of FN1	
Link time cost	**Queue time cost** (Tq=Lq / P, let P=2)	**Neighbor_ destination path length**	**Neighbor time cost**	**Destination path level**	**Neighbor time cost**	**Destination path level**
10	2.5	4	11.8825	18.3784	11.8821	18.3787
8	4	2	13	20	13.0035	19.9965
10	2.5	8	11.8825	23.3784	11.8821	23.3787
8	4	6	13	22.1441	13.0004	22.1444
1.6	4	2	5.6043	11.8626	5.6041	11.8629
15	5	5	23.9035	30.9035	23.8244	30.8244

Table (6) some of the test results of the fuzzy neural network (FN1) for the computer network (CN2)

Inputs of FN1			Desired outputs		outputs of FN1	
Link time cost	**Queue time cost** (Tq=Lq / P, let P=2)	**Neighbor_ destination path**	**Neighbor time cost**	**Destination path level**	**Neighbor time cost**	**Destination path level**
3.352	2.5	3	6.3277	12.4482	6.3277	12.4482
3.352	2.5	6	6.3545	14.6233	6.3512	14.6206
2.624	4	5	16.7822	23.7822	16.7864	23.7805
2.624	4	8	16.7822	33.7822	16.7864	33.7805
6.312	2.5	2	9.7045	15.6941	9.7042	15.6935
3.152	4	4	8.6055	14.533	8.3482	14.465

Table (7) some of the test results of the fuzzy neural network (FN2) for the computer networks (CN1, CN2)

Inputs of FN2		Desired output	Output of FN2
Packet average	**Packet variance level**	**Congestion prediction**	**Congestion prediction**
4	0.1	1.46316	1.46316
9	0.4	5.13696	5.13696
12	0.6	9.19229	9.1923
6	0.5	4.73698	4.73699
8.5	0.4	4.84137	4.84134
5.5	0.5	4.4763	4.7369

Table (8) some of the test results of the fuzzy neural network (FN3) for the computer network (CN1)

Inputs of FN3		Desired output	Output of FN3
Link time cost	**Congestion prediction**	**Neighbor selection level**	**Neighbor selection level**
2	1.46316	7.0755	7.0755
2.1	5.13696	18.3602	18.3602
2	9.19229	35.1393	35.1393
8	4.73698	28.8917	28.8917
2.1	1.46316	7.4201	7.4206
1.6	11.2359	41.8882	42.0732

Table (9) some of the test results of the fuzzy neural network (FN3) for the computer network (CN2)

Inputs of FN3			Desired output	Output of FN3
Link time cost	**Congestion prediction**		**Neighbor selection level**	**Neighbor selection level**
3.352	1.46316		9.0804	9.0804
6.312	5.13696		26.8333	26.8333
12.624	9.19229		65.8843	65.8844
3.352	4.73698		19.5252	19.5253
3.152	10.4559		38.6157	38.6152
6.312	8.367		34.5458	34.8242

Table (10) the success rates of testing the proposed fuzzy neural etworks for the computer network (CN1)

Fuzzy neural network	Success rate of test on trained sets	Success rate of test on other sets
FN1	100 %	92 %
FN2	100 %	94 %
FN3	100 %	95 %

Table (11) the success rates of testing the proposed fuzzy neural networks for the computer network (CN2)

Fuzzy neural network	Success rate of test on trained sets	Success rate of test on other sets
FN1	100 %	88 %
FN2	100 %	94 %
FN3	100 %	93 %

4. Conclusions

The proposed methods which used fuzzy neural networks to solve the routing problem of the computer networks are described in this paper. These fuzzy neural networks can be included as a part of a routing system that contains a monitor for providing them with necessary information. The congestion control with solving of routing problem is realized at one of these proposed methods as a combination of two fuzzy neural networks. For making the routing decision, system contains central monitor and the fuzzy neural network (FN1) at each node, to determine the best neighbor node depending on two primary criteria that are link time cost and queue time cost. The fuzzy neural networks (FN2, FN3) as congestion predictor and routing decision maker are inculded at each node to manage routing with congestion control. Results of the training and the testing of the fuzzy neural networks proof on their good performance for both computer networks (CN1, CN2).

References

[1] S. Kumar and R. Miikkulainen, " Confidence_ Based Q_ Routing: An On_ Line Adaptive Network Routing Algorithm ", Proceedings of the Artificial Neural Networks in Engineering Conference, Texas University, Austin, 1998.

[2] D. Davies, D. Barber, W. Price and C. Solomonides, " Computer Networks and Their Protocols ", J. Wiley and Sons Ltd, New York, 1979.

[3] J. Malrand, " Communication Networks: A First Course ", R. D. Irwin and Akson Associates, Inc, 1991.

[4] W. Newton, " A Neural Network Algorithm for Internetwork Routing ", Report in Software Engineering, for Degree of Bachelor, 2002.

[5] G. Caro and M. Dorigo, " AntNet: A Mobile Agents Approach to Adaptive Routing ", Technical Report, Iridia University, No. 12, 1997.

[6] J. Zue, S. Ng and L. Hanzo, " Fuzzy Logic Aided Dynamic Source Routing in Cross-Layer Operation Assisted Ad Hoc Networks ", Southampton University, United Kingdom, 2010.

[
7] L. Lersuwanakul, " Fuzzy Logic Based Routing in Grid Overlay Network ", Fern University, Hagen, 2010.

[8] M. Dehyadegari and S. Mohammadi, " An Adaptive Fuzzy Logic-based Routing Algorithm for Networks-on-Chip ", NASA/ESA Conference on Adaptive Hardware and Systems, 2011.

[9] A. Haboush, " Fuzzy Optimized Metric for Adaptive Network Routing ", International Journal of Computer Science and Security (IJCSS), Vol. 6, 2012.

[10] A.Dana and N. Salehi, " Congestion Aware Routing Algorithm for Mesh Network-on-Chip Platform ",
Indian Journal of Science and Technology, Vol. 5, No. 6, 2012.

[11] M. Chelliah and B. Sivaselvan, " Routing for Wireless Mesh Networks with Multiple Constraints Using Fuzzy Logic ", International Arab Journal of
Information Technology, Vol. 9, No. 1, 2012.

[12] A.Tanenbaum, " Computer Networks ", Fourth Edition, Pearson Education, Inc, 2003.

[13] S. Pithani and A. Sethi, " A Fuzzy Set Delay Representation for Computer Network Routing Algorithms ", Second International Symposium on Uncertainty Modeling and Analysis, College Park, MD, April, 1993.

[14] R. Habil, " Data Analysis with Neuro_ Fuzzy Methods ", Academic Report, Guericke University, Magdeburg, 25 Februar, 2000.

[15] Y. Chen and C. Teng, " Fuzzy Neural Network Systems in Model Reference Control Systems ", Fuzzy logic and Expert Systems Applications, Academic Press, 1998.

ايجـاد مسـار شبكة حاسبـات باستخـدام الشبكات العصبية المضببـة

خلود احمد نصار ***تركي يونس عبد الله*** ***عبد الكريم يونس عبد الله***

قسم علوم الحاسبات ***قسم هندسة الحاسبات*** ***قسم علوم الحاسبات***

جامعــة البصـرة، البصـرة ، العـراق

المستخلص

الشبكات العصبية المضببة ممكن ان تستخدم بكفاءة كتقنية حل لمسألة ايجاد مسار في شبكات الحاسبات وذلك لغرض تحسين الانجازية. في هذا البحث، اقترحت طريقتين لحل هذه المسألة. تم استخدام الشبكات العصبية المضببة لصنع قرار ايجاد مسار، ومعالجة مشكلة الاكتظاظ مع حل مسالة ايجاد مسار. الطرق المقترحة طبقت لأمثلة نموذجية لشبكات الحاسبات. الشبكات العصبية المضببة تم تدريبها، ونتائج الفحص اكدت على انجازيتها العالية.