**EAI Endorsed Transactions**
**on Context-Aware Systems and Applications**

Research Article ICST.ORG

# Model Integrating Fuzzy Argument with Neural Network Enhancing the Performance of Active Queue Management

Nguyen Kim Quoc[1,*], Vo Thanh Tu[1] and Nguyen Thuc Hai[2]

[1]College of Sciences, Hue University, Vietnam
[2]Ha Noi University of Science and Technology, Vietnam

## Abstract

The bottleneck control by active queue management mechanisms at network nodes is essential. In recent years, some researchers have used fuzzy argument to improve the active queue management mechanisms to enhance the network performance. However, the projects using the fuzzy controller depend heavily on professionals and their parameters cannot be updated according to changes in the network, so the effectiveness of this mechanism is not high. Therefore, we propose a model combining the fuzzy controller with neural network (FNN) to overcome the limitations above. Results of the training of the neural networks will find the optimal parameters for the adaptive fuzzy controller well to changes of the network. This improves the operational efficiency of the active queue management mechanisms at network nodes.

**Keywords:** Congestion Control, Active Queue Management, Fuzzy Logic, Neural Network

Received on 20 April 2015; accepted on 21 April 2015, published on 04 July 2015

doi: 10.4108/eai.4-8-2015.150042

## 1. Introduction

Along with the advantages that the fuzzy control system has brought to improvements on the active queue management mechanism, there still exists some shortcomings such as the design and optimization of the fuzzy system require some experiences in controlling subjects. On the other hand, for the fuzzy controller to work well, it should have the parameters (the number of fuzzy sets in each language variable , the shape of fuzzy sets, the number of laws and the weight of each law) to be optimal [13][28][29]. To solve this problem, we offer the model integrating fuzzy control with neural network in order to improve the performance of active queue management mechanisms.

Therefore, the structure of this paper is organized as follows: the first presents the mathematical basis of neural network. Since then, building the model combining fuzzy control with the neural network to improve the active queue management mechanism. During network training, we propose a number of back-propagation algorithm improvements so that the learning process can converge faster and parameters of the fuzzy controller is good. Based on the theoretical model of the combination in the fuzzy neural network, we build FNN model for AQM improvement mechanism. To inspect the effectiveness of the application of FNN into improving AQM, we build and install simulation for FNNRED and FNNREM mechanisms. The next of the paper is the assessment on the simulation of the proposed FNNRED and FNNREM mechanism. The last of the paper confirms the role of neural network to adjust the parameters in the fuzzy controller, and the benefits of the combination of soft computing tools, namely that combines fuzzy logic and neural networks for improving active queue management mechanisms.

## 2. Related Works

### 2.1. Operation of FLRED mechanism

The FLRED's goal is to use the fuzzy controller to adjust the queue of network nodes in a level to be determined, by achieving the target queue length (TQL), in order to maintain a high level of user queue medium and low latency. FLRED use Sugeno fuzzy systems with input variables are deviations instantaneous queue length than the target queue length at two consecutive time and the output is the probability of packet marking.

FLRED mechanism have fuzzy input values as a function of bell-shaped with seven value domains. The value of the selected language to denote the case of two input *e(t)* and *e(t-T)* corresponding to two consecutive times and fuzzy membership functions by the bell-shaped with seven domain values is: {NB, NM, NS, ZE, PS, PM, PB}. The function of the input linguistic variables *e(t)* and *e(t-T)* are show

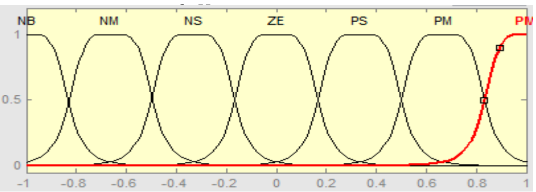


*Figure 1. Membership functions for input of the FLRED*

[*]Corresponding Author. E-mail: quocnknet@yahoo.com

Meaning of dismissal following values: Negative Big (NB), Negative Medium (NM), Negative Small (NS), Zero (ZE), Small Positive (PS), Positive Medium (PM, Positive Big (PB).

Similarly, we define the language value for the output variable is the probability of marking / remove packages as follows *p(t)*: Zero (Z), Tiny (T), Very Small (VS), Small (S), Big (B), Very Big (VB), Huge (H). The value *p(t)* is fuzzy singleton with values representing the 7 levels in the interval [0,1] as follows: $p(t) = \{\mathrm{H, VB, B, S, VS, T, Z}\} = \{0.000, 0.167, 0.333, 0.500, 0.667, 0.834, 1.000\}$

Because each input variable has seven value domains, so the system is built FLRED rule include 49 rules in Table 1.

*Table 1. Fuzzy rule system of FLRED*

| $p(t)$ | | $K_2.e(t-T)$ | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | NB | NM | NS | ZE | PS | PM | PB |
| $K_1.e(t)$ | PB | VS | T | Z | Z | Z | Z | Z |
| | PM | S | VS | T | Z | Z | Z | Z |
| | PS | B | S | VS | T | Z | Z | Z |
| | ZE | VB | B | S | VS | T | Z | Z |
| | NS | H | VB | B | S | VS | T | Z |
| | NM | H | H | VB | B | S | VS | T |
| | NB | H | H | H | VB | B | S | VS |

## 2.2. Operation of FLREM mechanism

AFC for FLREM also based on variable rates (*Pr*) measure of congestion as REM algorithm. Therefore, we use two inputs, one for the sample at the current time *Pr (t)* and a time to form in the previous cycle *Pr (t -T)*. Based on these two input values, the fuzzy controller determines the value of the packet marking probability represents the output of the fuzzy system.

To improve the accuracy we increase the number of membership functions for input and output of the fuzzy system is 9, this will make some inference rules become 81 rules. The downside of the calculation of the price mechanism is the magnitude of the price REM always positive, this is not the actual queue length and flow rate to be much smaller than the reference queue length and bandwidth transmission.

The value of the selected input language to denote the case corresponding to two consecutive times and fuzzy membership functions by a bell-shaped domain nine values: $P_r(t) = \{NH, NB, NM, NS, ZE, PS, PM, PB, PH\}$ and $P_r(t-T) = \{NH, NB, NM, NS, ZE, PS, PM, PB, PH\}$. In particular: Negative Huge (NH), Negative Big (NB), Negative Medium (NM), Negative Small (NS), Zero (ZE), Positive Small (PS), Positive Medium (PM), Positive Big (PB), Positive Huge (PH). The function of the $P_r(t)$, $P_r(t-T)$ input linguistic variables are shown in Figure 2. as follows:

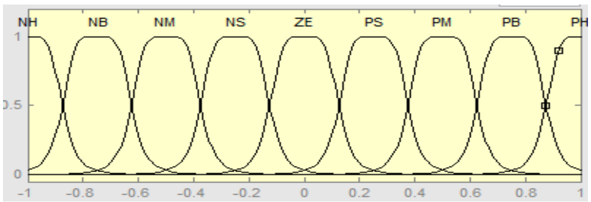


*Figure 2. Membership functions of the FLREM input*

The value of language in the output variable is the singleton value represents the probability nine ascending remove packages from the interval [0,1], is defined as follows:

$$p(t) = \{Z, T, VS, MS, S, B, MB, VB, H\}$$

$$p(t) = \{0, 0.125, 0.25, 0.375, 0.5, 0.635, 0.75, 0.875, 1\}$$

In particular, Zero (Z), Tiny (T), Very Small (VS), Medium Small (MS), Small (S), Big (B), Medium Big (MB), Very Big (VB), and Huge (H).

Build system for FLREM rules as Table 2. System rules for the fuzzy controller is made up of the control rule of the form:

*Table 2. Fuzzy rule system of FLREM*

| p(t) | | e(t-T) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | NH | NB | NM | NS | ZE | PS | PM | PB | PH |
| e(t) | PH | S | MS | VS | T | Z | Z | Z | Z | Z |
| | PB | B | S | MS | VS | T | Z | Z | Z | Z |
| | PM | MB | B | S | MS | VS | T | Z | Z | Z |
| | PS | VB | MB | B | S | MS | VS | T | Z | Z |
| | ZE | H | VB | MB | B | S | MS | VS | T | Z |
| | NS | H | H | VB | MB | B | S | MS | VS | T |
| | NM | H | H | H | VB | MB | B | S | MS | VS |
| | NB | H | H | H | H | VB | MB | B | S | MS |
| | NH | H | H | H | H | H | VB | MB | B | S |

# 3. Mathematical basis of fuzzy neural network

## 3.1. Motivation using fuzzy neural network

One of the problems of alternative methods of estimating the state function from measured data and understanding of the system to determine the function of replacing conservative estimates so wrong of approximately $\dot{x}_n$ always response design requirements. Assuming that can find continuous functions, bounded $\tilde{f}$ and $\tilde{g} > 0$ so that the error of approximation $\dot{x}_n$ satisfying $|\Delta_{dxn}| \leq |\Delta_{dxn}| \leq \mathrm{W}$; $\forall x \in \Omega_x$ ; $\forall x \in \Omega_x$. However, if the conditions unsatisfied approximately, or in other words if do not find the Estimators satisfactory alternative is clearly not applicable method.

Also another issue to consider is determining control parameter (if exists) to static stability controller meets the conditions binding on the state trajectory and input during activities, such as x and u should keep always in the corresponding domain valid $\Omega_{\mathbf{x}}$ and $\Omega_u$. This problem is significant considering the application perspective to answer the only question is if the function is defined to replace approximately satisfy the conditions in a particular region does not cover the valid domain can design static stability controller satisfies design requirements or not.

As mentioned, due to the advantages of fuzzy systems and neural networks in identifying non-linear characteristics of the object that the method is built towards fuzzy neural application to build compensate nonlinear components in the control rule static feedback. This section of the paper will show that this problem is solved by the universal approximation in function approximation theory.

## 3.2. The ministry universal approximation

Symbol $F(x,\theta)$ is the approximation with $\theta \in \Re^{p}$ is the parameter vector to be modified and $\Omega_F \subset \Re^{p}$ is the set of all the valid parameter value of approximately. If the call $Z = \left\{ \mathrm{F}(\mathrm{x},\theta) : \theta \in \Omega_{\mathrm{F}} \subset \Re^{\mathrm{p}}, \mathrm{p} \geq 0 \right\}$ is class contains all functions of the form $F(\mathrm{x},\theta)$, $\theta \subset \Omega_F$; with p≥0, whereas concepts are regularity approximation or uniformly approximation is defined as follows:

**Definition 1**: function $f : D \rightarrow \Re$ possible are regularity approximation or uniformly approximation on $D \subseteq \Re^{n}$ by functions of class Z if the ε>0 always exists $F(\mathbf{x}) \in \mathrm{Z}$ to $\sup_{x \in D} \left| F(x) - f(x) \right| < \varepsilon$.

Note that the definition chosen $F(\mathbf{x})$ can depend on ε and value $p \geq 0$ is the number of parameters required of $F(\mathbf{x})$ to ensure error of approximation blocked or $\sup_{x \in D} \left| F(x) - f(x) \right| < \varepsilon$.

In addition, signs $F(\mathbf{x})$ shortened to just parameter vector $\boldsymbol{\theta}$ identify objects and is not of concern. In function approximation theory, the universal approximation has an important role because they can indicate approximate certain function layers with any accuracy. The definition of universal approximation is statement based on the concept of are approximately as follows:

**Definition 2** : Class function structure $Z_1$ is called the universal approximation of functions of functions class $Z_2$ if every function $f \in Z_2$ then have to be approximated by $Z_1$.

If sign $Z_c(n,D)$ is the set of all continuous scalar function defined on a closed domain $D \in \Re^{n}$, While it can use one of the common universal approximation follows ([4], [8], [30]) to approximate the continuous function:

– The class definition content jump function; 2-layer neural network with activation function buttons hidden under the threshold or sigmoid activation function $\psi : \Re \rightarrow [0,1]$ and output nodes are linear approximations of the universal constant scalar functions $f \in Z_C(1,D)$, $D = [a,b] \subset \Re$.

– The polynomial defined function class $Z_p = \left\{ g(x) = \sum_{i=0}^{p} a_i x^{i} : \mathrm{a}_0, \mathrm{a}_1, \ldots \mathrm{a}_p \in \Re, p \geq 0 \right\}$ with g:D→ℜ, D⊆ℜ (The theorem Weiertrass); functions defined function class $Z_a = \left\{ g(x) = \sum_{i=1}^{m} a_i \cos(\mathrm{b}_i^{T} x + c_i) : \mathrm{a}_i, \mathrm{c}_i \in \Re, \mathrm{b}_i \in \Re^{n} \right\}$; fuzzy systems with Gaussian membership functions, defuzzification COG focal point methods; 2-layer neural network with hidden nodes sigmoid activation function and output nodes are linear approximations of the universal constant scalar functions $f \in Z_C(n,D)$, $D \subset \Re$.

Proof of a function class structure is the universal approximately function for another function class normally by definition. However, there is a proven effective tool is to use the results of Stone Weierstrass theorem can refer to in the literature [35], [20].

## 3.3. Mathematical approximation of the fuzzy neural network system

As presented, the universal approximation parameters selected quantities large enough to be approximated any continuous function with arbitrary accuracy on compact sets (compact set), so in addition to the ability to adjust Online (online) but they can also be used in the adaptive controller. The study makes use of approximately adaptive controller used primarily oriented fuzzy systems ([12], [30]), using neural networks ([14], [32]) or a combination of fuzzy systems and network use neurons ([8], [15]).

Although theoretically can use any approximations that satisfy the requirements of such methods as the fuzzy approximation with Gauss membership functions, defuzzification methods COG; 2-layer neural network with hidden nodes sigmoid activation function and a linear output node or network adaptive ANFIS. However, not only the structure easily approximate the optimal in each case so dependent on measurement data available, non-linear characteristics of the object and the boundary conditions.

Often the design of universal approximate size of the smallest structure satisfies the approximation error and need more time to try and test the structure due to the approximations given only a multimeter to ensure that error is approximately bounded by $W_F > 0$ was not possible to determine the value $W_F$ small as long as. However, there are certain things that need to increase the size of the structure and selected approximately parameters adjusted accordingly to achieve approximation error arbitrarily small.

Also research the issues choose between the linear approximation and nonlinear parameters for the structure of the same size (or number of parameters), the problem of parameter tuning method in the approximation or nonlinear structure determination of the best approximations are the subject being studied ([35]). But due to the advantages of fuzzy systems and neural networks in nonlinear processing, and adjust parameters on articles online, but only consider the approximation is based on fuzzy systems and neural networks. The following section will show that the neuron can approximate fuzzy controller used in both static stability and dynamic stability control (adaptation), and as a basis for problem solving stability control in our method production. The following presents a mathematical representation of the number and structure of fuzzy neural network system is used as the universal approximation, as well as a number of issues on the use and optimization of the fuzzy neural approximation.

## 3.4. Mathematical representation of the fuzzy approximation

At the MISO fuzzy systems are nonlinear mapping from input vectors $x = \left[x_1,...,x_n\right]^T \in \Re^n$ to output y$\in\Re$ [30].

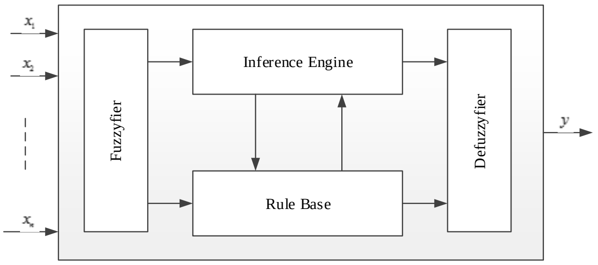


*Figure 3. MISO fuzzy control system*

In the theory of fuzzy sets and fuzzy logic ([4], [9]), allows dimming chemical inputs using fuzzy operators chemical transfer function data clearly the basis of fuzzy sets and fuzzy rules with the assumption including: $p$ fuzzy rules are represented as a set of fuzzy descriptive (fuzzy Implications) after:

$$\Re_i : \left(A^1_{J_{1i}} \cap A^2_{J_{2i}} \cap ...A^n_{J_{ni}}\right) \Rightarrow B_{ki} \qquad (1)$$

with i=1..p, notation $A \Rightarrow B$ or just fuzzy describe for conditions statements IF A THEN B and $A^i_j$, $B_k$ is the fuzzy set is defined as follows:

$$\begin{aligned} A^i_j &= \left\{\left(x_i, \mu_{A^i_j}\left(x_i\right)\right) : \mathrm{x}_i \in \Re\right\} \\ B_k &= \left\{\left(y, \mu_{B_k}\left(y\right)\right) : y \in \Re\right\} \end{aligned} \qquad (2)$$

with $\mu_{A^i_j}, \mu_{B_k} \in [0,1]$ respectively, the membership functions of j and k inputs $x_i$ and output $y$.

The basic problem of fuzzy systems in fuzzy inference mechanism (fuzzy inference) and defuzzification methods (defuzzification) used to calculate the output of fuzzy system clearly specify when the input given on the basis of fuzzy rules have known. This fuzzy inference mechanism is built on the success of inference rules. To calculate the the first clause in the formula (1) we can use any t-norm does [30] as the smallest t-norm $T_{MIN}(a,b) = \min\{a,b\}$, algebraic integrated $T_{PAND}(a,b) = ab$, Łukasiewicz function $T_{LAND}(a,b)$ $= \max\{a+b-1,0\}$, In case of using smallest t-norm $T_{MIN}$ then fuzzy describe (23) can be written as Decac integrated $\left(A^1_{J_{1i}} \times A^2_{J_{2i}} \times ... \times A^n_{J_{ni}}\right) \Rightarrow B_{ki}$ the first clause in the formula (3) calculated as follows:

$$\mu_{A^1_{J_{1i}} \times A^2_{J_{2i}} \times ... \times A^n_{J_{ni}}}\left(x\right) = \min\left\{\mu_{A^1_{J_{1i}}}\left(x_1\right), \mu_{A^2_{J_{2i}}}\left(x_2\right), ..., \mu_{A^n_{J_{ni}}}\left(x_n\right)\right\} \qquad (3)$$

To calculate the fuzzy description of each rule or the output of each rule can be used to describe the fuzzy operator (t-norm or t-conorm) [30]. Some fuzzy operator describe common include: Zadeh $x \Rightarrow y = max\{1-x, \ min\{x, y\}\}$, Lukasiewicz $x \Rightarrow y = min\{1, \ 1-x+y\}$, Mamdani $x \Rightarrow y = min\{x, y\}$ và Larsen $x \Rightarrow y = xy$ ([28], [25], [30]). Where the operator using fuzzy Mamdani describes the output of the $i^{th}$ rule ($R_i$), denoted as $C_i = \left\{\left(y, \mu_{C_i}\left(x, y\right)\right) : x \in \Re^n, \mathrm{y} \in \Re\right\}$ calculated as follows:

$$\begin{aligned} \mu_{C_i}\left(x, y\right) &= \mu_{A^1_{J_{1i}} \times A^2_{J_{2i}} \times ... \times A^n_{J_{ni}}}\left(x, y\right) \\ &= \min\left\{\mu_{A^1_{J_{1i}} \times A^2_{J_{2i}} \times ... \times A^n_{J_{ni}}}\left(x\right), \mu_{B_{ki}}\left(y\right)\right\} \end{aligned} \qquad (4)$$

Finally, to calculate the output of fuzzy system can be used as a defuzzification method of defuzzification focal point COG (Center Of Gravity) after:

$$y = F(x, \theta) = \frac{\sum_{i=1}^{p} C_i \int_z \mu_{C_i}(x, z)dz}{\sum_{i=1}^{p} \int_z \mu_{C_i}(x, z)dz} \qquad (5)$$

with $c_i$ is the center of $\mu_{B_{ki}}\left(y\right)$ for to i^th rule. Normally we choose $\mu_{B_{ki}}\left(y\right)$ symmetry across a vertical axis through the peak to $c_i$ is the midpoint of $\mu_{B_{ki}}\left(y\right)$

The equation (5) is a model of Mamdani fuzzy system with COG defuzzification method and is used as the fuzzy approximation $F(\mathbf{x}, \boldsymbol{\theta})$ with $\theta = \left[c_1, ..., c_p\right]^T$.

## 3.5. Mathematical representation of the neural network approximation

At the feedforward artificial neural network $p \geq 1$ output layer with linear activation function, in which $0,1..p-1, p$ respectively order index from the input layer, the hidden layer (when $p \geq 2$) to the output layer. If sign:

$x_i$, $y_j$ : respectively the input and output of the network with $i = 1..n$, $j = 1..m$;

$L_i$ : the neural number of hidden layer i^th (when $p \geq 2$) with $i = 1..p-1$;

$\mathrm{w}^i_{jk}$ is the weight from node k in layer i-1 to node j in layer i with i = 1.. p,

$$j = \begin{cases} \{1..\mathrm{L}_i, 1 \leq i < p \\ 1..m, i = p \end{cases}; \quad k = \begin{cases} \{1..\mathrm{L}_i, 1 < i \leq p \\ 1..\mathrm{n}, i = 1 \end{cases}$$

$\theta^i_j$ is the threshold value of node j in layer i with *i = 1.. p*,

$$j = \begin{cases} 1..\mathrm{L}_i, 1 \le i < p \\ 1..m, \mathrm{i} = \mathrm{p} \end{cases}$$

$\sigma_j^i$ function is activate network node $j^{th}$ in the hidden layer $i$ with $i^{th}$ *=1..p-1, j =1..* $L_i$ the output of neuron $j^{th}$ in the hidden layer $i^{th}$ (when $p \ge 2$), denoted as $O_{i,j}$ with $i$ =1..$p-1$, $j$ =1..$L$ and the output of the network $y_r$ with $r$ =1..$m$:

$$o_j^i = \begin{cases} \sigma_j^1\left(\sum_{k=1}^{n} \mathrm{w}_{jk}^1 x_k + \theta_j^1\right), i = 1 \\ \sigma_j^i\left(\sum_{k=1}^{L_{i-1}} \mathrm{w}_{jk}^i o_k^{i-1} + \theta_j^i\right), 1 < i \le p-1 \end{cases} \tag{6}$$

$$y_r = \begin{cases} \sum_{s=1}^{n} \mathrm{w}_{rs}^1 x_s + \theta_r^1, p = 1 \\ \sum_{s=2}^{L_{p-1}} \mathrm{w}_{rs}^p o_s^{p-1} + \theta_r^p, p \ge 2 \end{cases}$$

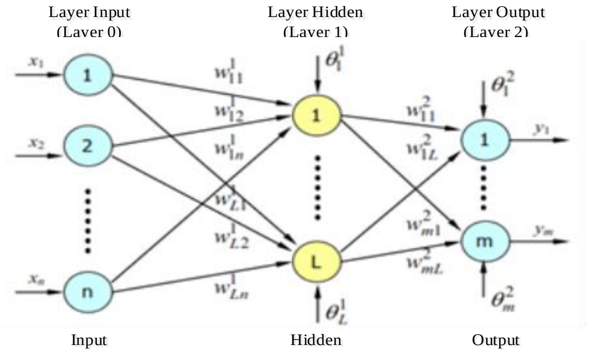


*Figure 4. System 2-layer feed forward neural networks*

Figure 4 performances 2 layer feed forward neural networks (p = 2) has n input, m output linear and L buttons hidden. Network model is written as follows:

$$y_i = \sum_{j=1}^{L} w_{ij}^2 \sigma_j \left( \sum_{k=1}^{n} w_{jk}^1 x_k + \theta_j^1 \right) + \theta_i^2 \tag{7}$$

with $i$ =1..$m$. Where there network is only one output (m = 1) with sigmoid activation function $sig(x) = \frac{1}{1+e^{-2x}}$ in the hidden layer, the network can use the universal approximation of scalar continuous functions $f(x) = \Re^n \to \Re$ either:

$$F(x,\theta) = y_1 = \sum_{j=1}^{L} \mathrm{w}_{1j}^2 sig\left( \sum_{k=1}^{n} \mathrm{w}_{jk}^1 x_k + \theta_j^1 \right) + \theta_1^2 \tag{8}$$

## 3.6. Approximate fuzzy neural network

There are many research results in order to combine the advantages of fuzzy systems and neural networks in the building structure is approximately [30]. One of the results of the study are positive fuzzy inference system based on adaptive network ANFIS (Adaptive Network based Fuzzy Inference System) proposed by Jang [18], [19], [20], [30]. This is a hybrid neural network structure model based on fuzzy rules fuzzy systems Takagi Sugeno have been described as fuzzy follows:

$$R_i : \left( A_{J1i}^1 \times A_{j2i}^2 \times ... \times A_{Jni}^n \right) \Rightarrow g_i(x) \tag{9}$$

ANFIS networks using linear functions $g_i(x) = \sum_{j=1}^{n} a_{ij} x_j$ input membership functions and bell has proven to be a universal approximation of nonlinear function.

## 3.7. Mathematical representation of the linear and nonlinear approximation for parameters

The approximate (fuzzy neurons) can be represented as linear either nonlinear for parameters. Ministry of approximation is called linear if for parameters representation in the form:

$$F(x,\theta) = \theta^T \varphi(x) \text{ hay } \frac{\partial F(x,\theta)}{\partial \theta} = \varphi^T(x) \tag{10}$$

where $\boldsymbol{\varphi}(\mathbf{x})$ is a vector function of x and $\boldsymbol{\theta}$ is the vector of input parameters in the linear behavior of the function approximation. For example, the fuzzy neural linear approximation for fuzzy system parameters such as formula (10) and RBN neural networks (NN Radial Basis).

In case the approximation using fuzzy systems under (10) if $\theta = \left[c_1, ..., c_p\right]^T$ then:

$$\frac{\partial F(x,\theta)}{\partial \theta} = \varphi^T(x) = \left[\varphi_1(x), ..., \varphi_p(x)\right] \text{ with}$$

$$\varphi_i(x) = \frac{\int_z \mu_{C_i}(x,z)dz}{\sum_{i=1}^{p} \int_z \mu_{C_i}(x,z)dz}$$ should be able to represented:

$F(x,\theta) = \theta^T \varphi(x)$ linear form for the parameter.

Case $\frac{\partial F(x,\theta)}{\partial \theta} = \varphi^T(x)$ or $\varphi(x,\theta)$ vector $\theta$ containing parametersas in the case of approximation based on multi-layer neural network represented in (30), the approximation is called nonlinear for parameter, because $F(x,\theta)$ is a nonlinear function of the $\theta$ parameter:

$$F(x,\theta) = \theta^T \varphi(x,\theta) \tag{11}$$

## 3.8. The linearized approximation

The non-linear approximation to the parameters usually more simple (in size and number of parameters) than the linear approximation to achieve accuracy approximately equivalent. For static feedback control rule in the alternative state estimation function, the use of linear approximations to the nonlinear parameter is not set up by simply ensuring the approximation error should set of valid domain. To adjust the parameters of the approximation in the feedback control rule, the use of the linear approximation for nonlinear parameter significance. Although the findings of the paper in the next chapter of applying one of two types of approximation, however, can also linearized approximation of nonlinear parameters according to each application.

The problem of the linearized approximation is presented in [17]. These results indicate if the approximation is Lipschitz continuous adjustments to the parameters (excluding the performers are in the form of linear or nonlinear), it can be written as follows:

$$\begin{aligned}\Delta_F(x,\theta) &= F(x,\theta^*) - F(x,\theta)\\ &= -\frac{\partial F(x,\theta)}{\partial\theta}\Delta_\theta + \delta(x,\theta,\theta^*)\end{aligned} \qquad (12)$$

Where **θ** is the current parameters, $\theta^* = \arg\min_{\theta\in\Omega_\theta}\left(\sup_{x\in\Omega_x}|F(x,\theta) - f(x)|\right)$ the optimal parameters, $\Delta_\theta = \theta - \theta^*$, $\delta(\mathrm{x},\theta,\theta^*) = -\sigma(|\Delta_\theta|)$ with $\lim_{|\Delta_\theta|\to 0}\frac{\sigma(\Delta_\theta)}{|\Delta_\theta|} = 0$.

Addition $\delta(\mathrm{x},\theta,\theta^*)$ blocked by $|\delta(\mathrm{x},\theta,\theta^*)| \le L|\Delta_\theta|^2$ with L is Lipschitz constant. Therefore, if you find the governing rule θ to reduce $|\Delta_\theta|^2$ then θ tend toward $\theta^*$ and F(x,θ) will forward to $F\left(x,\theta^*\right)$. Thus, if $|\Delta_\theta|^2$ bounded then approximation error is also blocked.

### 3.9. Optimization of fuzzy neural approximation

The optimization problem of fuzzy neural approximation generally seek to minimize the value function (cost function) $J(\theta) = \sup_{x\in\Omega_x}|f(x) - F(x,\theta)|^2$ with $\theta \in \Omega_\theta \subseteq \Re^p$ p is the vector of parameters to be adjusted by fuzzy systems or neural networks or need to find the optimum tuning parameters $\theta^* \in \Omega_\theta$ from the measured data to $\theta^* = \arg\min_{\theta\in\Omega_\theta} J(\theta)$

Thus for small approximation error upon request measured data must be large enough and cover all valid domain $\Omega_x$ . But in fact most can not choose how to measure distributed data in and can not change the measurement data to improve the precision that can only directly use finite amount of data has been measured. This is really a complex issue and in many cases optimization methods do not guarantee to meet the requirements of error of approximation. Normally vector to find the optimum tuning parameters $\theta^* \in \Omega_\theta$ from the measured data can be applied least squares algorithm (Least Squares) linear (batch, recursive) or nonlinear (gradient, conjugate gradient, line search, Levenberg Marquardt) are presented in the document [4] [20] [30] [35].

## 4. Proposed FNN fuzzy neural network model

To improve the performance of AQM mechanisms at the network nodes, we have a solution using soft computing tools to improve this mechanism. In [31] has presented the building adaptive fuzzy controller AFC for improved AQM mechanisms were effective and better for the mechanism. However, to the fuzzy controller works well in different network environments, there should be the appropriate parameters for its operation. Therefore, we propose fuzzy neural network model, temporarily called FNN to solve this problem.

The composition and operation of the FNN model is presented as follows:

### 4.1. FNN model for AQM

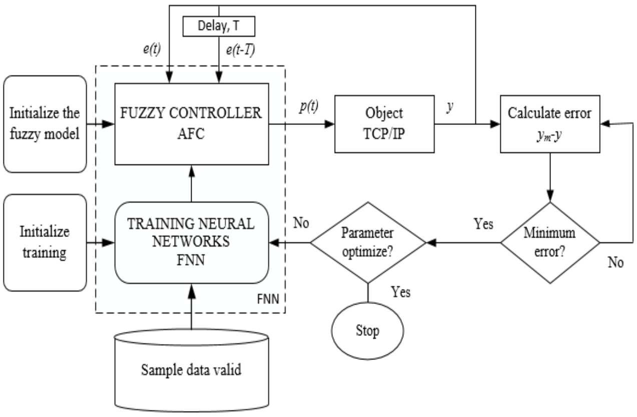


*Figure 5. The model proposed FNN for AQM*

– AFC is an adaptive fuzzy controller is constructed in [31] has components: fuzzy controller, adaptive structure, $G_m$ models, the coefficient input $K_i$ and output $K$.

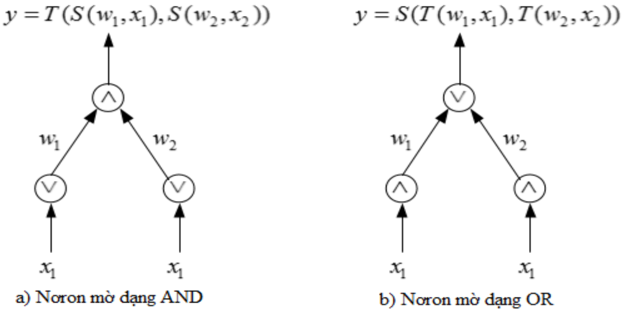
– Ministry to train the network using errors back-propagation algorithm.

– Sample data is valid sample of the form $\{e(t), e(t-T), y\}$ is updated periodically T operation of the fuzzy controller. For example, with T = 0.1s to 10s for the 100 samples for training.

– Minimum error is the smallest error of the fuzzy controller (corresponding to the parameter). If the error is less than or equal sub-optimal error, the optimal system and end, opposite the FNN trained by IBP to find the best parameters for the fuzzy controller.

### 4.2. Construction of a fuzzy controller FNN

### 4.2.1. Fuzzy neuron in FNN


$y = T(S(w_1,x_1), S(w_2,x_2))$
∧
$w_1$
$w_2$
∨
∨
$x_1$
$x_1$
$y = S(T(w_1,x_1), T(w_2,x_2))$
∨
$w_1$
$w_2$
∧
∧
$x_1$
$x_1$

a) Nơron mờ dạng AND

b) Nơron mờ dạng OR

*Figure 6. The fuzzy neuron types*

Fuzzy neural is Neural combined signals $x_i$ and weighted

$w_i$ calculations used t-norm, t-conorm. So two types of fuzzy neurons, AND neural fuzzy and OR neural fuzzy forming. Figure 6a) shows the structure neural fuzzy AND and Figure 6b) shows the structure fuzzy neurons OR.

FNN to train the fuzzy controller to find the best parameters for the fuzzy controller. FNN network structure consisting of five layers is shown in Figure 7:

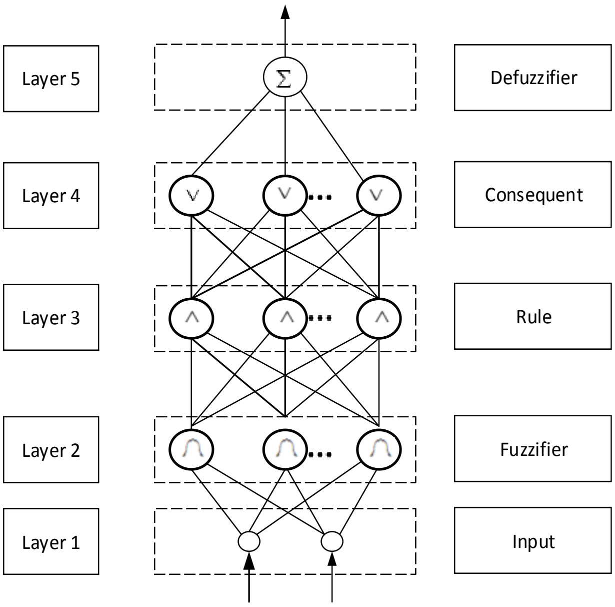


*Figure 7. FNN structures*

— Input layer: input variables of FNN the deviation of the immediate output value ($y$) of the adaptive fuzzy controller in 2.3.3 compared with the reference output value ($y_m$) at two consecutive time $t$ and $t$-$T$. The expression $e(t) = y_m - y(t)$ determines the current time errors and $e(t\text{-}T) = y_m - y(t\text{-}T)$ was determined time error after a delay time T (in cycles taken previous sample).

— Fuzzifier Layer: this layer performs the function of fuzzy values of input variables, will transmit each input value to set the corresponding language. For the control of the controller is sleek and smooth, FNN defuzzification method according bell shape. FNN divide domain of the value *e(t)* and *e(t-T)* were normalized value in the values domain [-1,1] .

— Rule layer: This layer contains the rules basis for inference. FNN Use the rules of the AFC is presented in Section 2.1

— Consequent layer: this layer performs the function of the total results in the class node inference rules sent by the union to have been presented in Section 3.4.

— Defuzzification layer: layer performs the function of the fuzzy grade results, the results are calculated probability of packet marking optimum current state of the network, presented in Section 3.4. Fuzzy controller determines the value of the package removal rate represents the output of the fuzzy system.

### 4.2.2. Training FNN

The goal of the FNN is trained to select the best parameters for the fuzzy controller fuzzy AQM mechanisms. Training process based on the data obtained from the operation of AQM mechanisms in TCP/IP networks. Sample data for calibrating process at any one time is the actual input and actual output of the system TCP/AQM. Depending on the required accuracy and the computation time of the system that we shall select the number of samples as well as the time interval between the time of sampling.

Based on simulation results of mechanisms, most of AQM mechanisms are stable after 5-8 seconds. Therefore, in the proposed FNN, we selected sampling in early 10s, including the transition period and the period setting and some sample data to learn the 100. The 100 model is selected because the sample period will be 0,1s, consistent comparisons with other AQM mechanisms. The samples taken much more will make the optimization problem more precisely, but brought about the complexity and computation time when it will be bigger.

### 4.2.3. Adjust the parameters of the membership functions

The problem is to find the jurisprudence updates adjust the parameters of the second layer FNN (performing fuzzy input values), when used in the membership functions of the second layer in the form of bell-shaped:

$$\mu_{A_i^j}(x_i) = \frac{1}{1 + \left[\dfrac{x_i - c_i^j}{a_i^j}\right]^{2b_i^j}} \qquad (13)$$

Updating the adjustable parameters in the learning process is given by the formula:

$$a_i^j(t+1) = a_i^j(t) - \eta_a \frac{\partial E}{\partial a_i^j(x_i)} \qquad (14)$$

$$b_i^j(t+1) = b_i^j(t) - \eta_b \frac{\partial E}{\partial b_i^j(x_i)} \qquad (15)$$

$$c_i^j(t+1) = c_i^j(t) - \eta_c \frac{\partial E}{\partial c_i^j(x_i)} \qquad (16)$$

$$p_i^j(t+1) = p_i^j(t) - \eta_p \frac{\partial E}{\partial p_i^j(x_i)} \qquad (17)$$

with $\eta_a, \eta_b, \eta_c, \eta_p$ is the training coefficient. Parameters a, b, c of the membership function is calculated:

$$\frac{\partial E}{\partial a_i^j(x_i)} = \frac{2\mu_j}{\sum_{j=1}^{M} \mu_j}(y - y^m)(f_j - y)(1 - \mu_{A_i^j}(x_i)) \frac{b_i^j(x_i)}{a_i^j(x_i)} \qquad (18)$$

$$\frac{\partial E}{\partial b_i^j(x_i)} = \frac{\mu_j}{\sum_{j=1}^{M} \mu_j}(y - y^m)(f_j - y)(1 - \mu_{A_i^j}(x_i)). \qquad (19)$$

$$\left[\ln\left(a_i^j(x_i)\right) - \ln\left(x_i - c_i^j(x_i)\right)\right]$$

$$\frac{\partial E}{\partial c_i^j(x_i)} = \frac{2\mu_j}{\sum_{j=1}^{M}\mu_j}(y-y^m)(f_j-y)(1-\mu_{A_i^j}(x_i))\frac{b_i^j(x_i)}{1-c_i^j(x_i)} \quad (20)$$

$$\frac{\partial E}{\partial p_i^j(x_i)} = \frac{\mu_j}{\sum_{j=1}^{M}\mu_j}(y-y^m)x_i \quad (21)$$

## 5. Construction FNNRED mechanism

The goal of building FNNRED mechanism is find the value of the parameter optimization for FLRED mechanisms have been developed in Section 2.1, using BP for training.

### 5.1. Network training for FNNRED

Training process for FNNRED with the goal of finding the optimal parameters for the fuzzy controller be done in layers structure in Section 4.2 by back-propagation algorithm enhancements are wrong, the input data is deviation queue length at two consecutive time *Qe(t)* and *Qe(t-T)* and output data is instantaneous queue length *q(t)*. Figure 8. describes the process of training for FNNRED. Fuzzy control of FNNRED is installed in NS2 [20] software according to theoretical models in Section 7.1 was tested on Matlab software is structured as following:

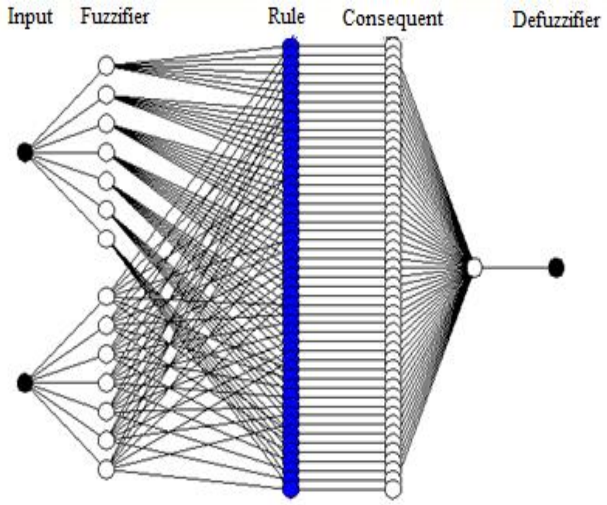


*Figure 8. Training model for FNNRED*

Number of sample data for the correction process is K = 100 samples, were taken in the first 10 seconds of simulation activities. Because the two inputs of the fuzzy controller is the deviation of the queue in the current cycle *Qe(t)*, deviation of the queue in the previous cycle *Qe(t - T)* and an output queue length *q(t)*, so the sample data for training is a collection of samples of the form $\{Qe_i(t), Qe_i(t-T), q_i(t)\}, i = 1 \div K$.

### 5.2. Results Training for FNNRED

After the training is finished, the results obtained are the optimal parameters for the membership functions of the FNNRED. The result of training is tested on the following Matlab software:

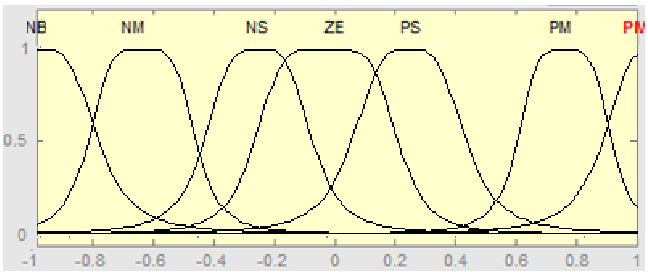


*Figure 9. The membership functions of* Qe(t) *after training*

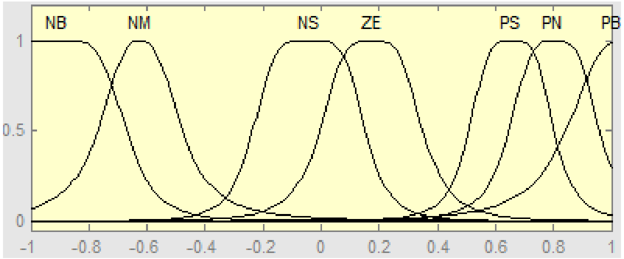


*Figure 10. Membership functions of Qe(t-T) after training*

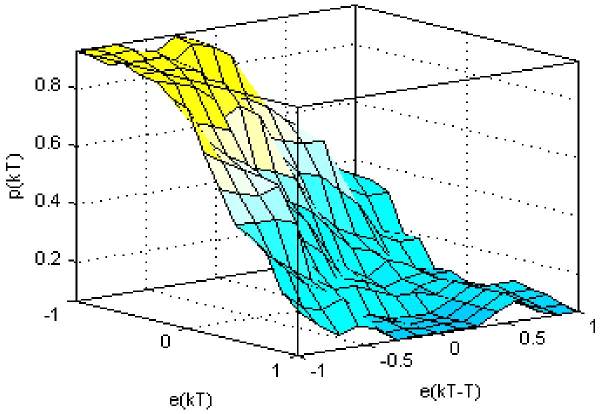


*Figure 11. Inference surfaces of FNNRED after training*

## 6. Construction FNNREM mechanism

The goal of building FNNREM mechanism is find the value of the parameter optimization for FLREM mechanisms have been developed in Section 2, using IBP for training.

### 6.1. Network training for FNNREM

Training process for FNNREM with the goal of finding the optimal parameters for the fuzzy controller FLREM be done in 5 layers structure in Section 4.2.2, the input data is deviations resource nodes (queue length and bandwidth) with the use of (price variable) at two consecutive time *Pr(t)* và *Pr(t-T)* and output data is instantaneous queue length *q(t)*. Figure 12 describes the process of training for FNNREM. Fuzzy control of FNNREM is installed in NS2 software according to theoretical models in Section 7.1 was tested on Matlab software is structured as following :

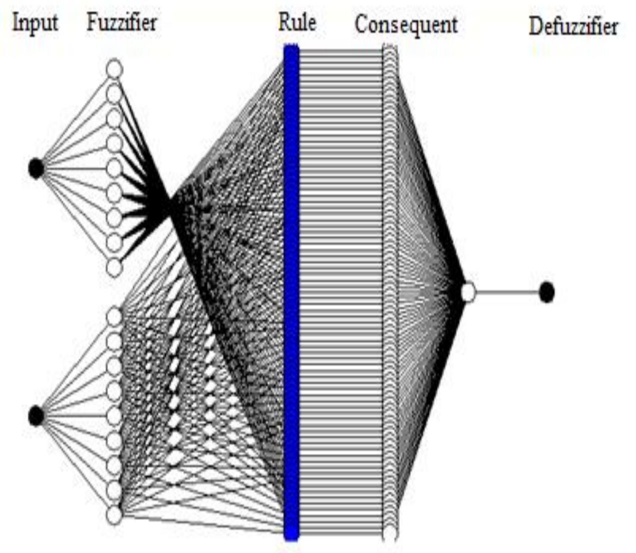


*Figure 12. Training model for FNNREM*

Number of sample data for the correction process is K = 100 samples, were taken in the first 10 seconds of simulation activities. Because the two inputs of the fuzzy controller is the deviations resource nodes (queue length and bandwidth) with the use of (price variable) at two consecutive time *Pr(t)* and *Pr(t-T)* and output data is instantaneous queue length *q(t)*, so the sample data for training is a collection of samples of the form $\{Pr_i(t), Pr_i(t-T), q_i(t)\}, i = 1 \div K$.

### 6.2. Results Training for FNNREM

After the training is finished, the results obtained are the optimal parameters for the membership functions of the FNNREM. The result of training is tested on the following Matlab software:

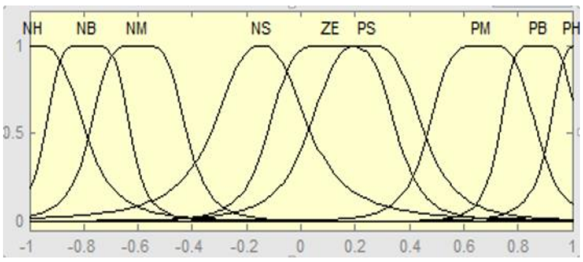


*Figure 13. The membership functions of Pr(t) after training*

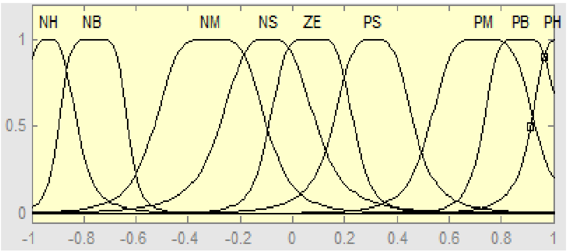


*Figure 14. Membership functions of Pr(t-T) after training*

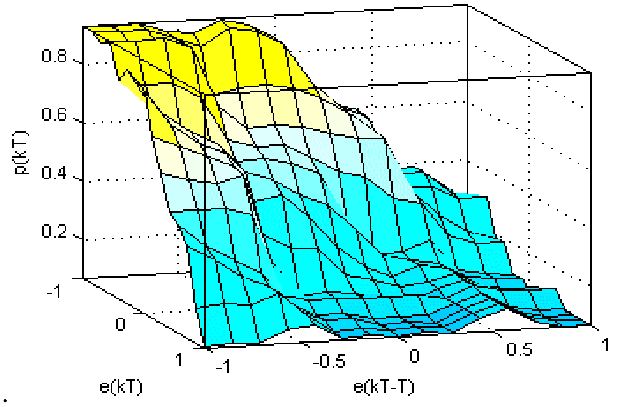


*Figure 15. Inference surfaces of FNNREM after training*

## 7. Simulation and results

### 7.1. Simulation Settings

To evaluate the effectiveness of the AQM mechanism using fuzzy controllers FNN, we use common network model and simulation results of the mechanism was announced. At the same time, we combine the extensive network model in Figure 3:20 to extend the simulations for the case of multi-flow send and receive, and multi-queue.

In bottlenecks, respectively installing traditional AQM mechanisms represent the group queue management mechanism (such as RED, REM), the corresponding AQM mechanism using fuzzy control to improve primitive mechanisms (such as FEM, FUZREM), the AQM mechanism using adaptive fuzzy controller AFC (as FLRED, FLREM) and AQM mechanisms used self-learning fuzzy controller FNN (as FNNRED, FNNREM).

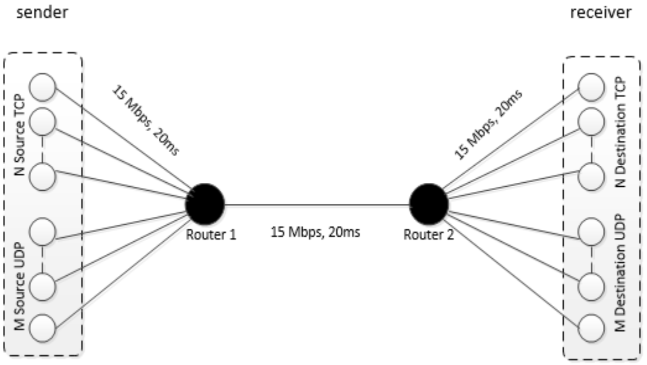


*Figure 16. Network simulation model*

In the simulation, using N flows changed from 10 to 300, the queue length at the bottleneck change from 100 to 1000. The path from the source to the bottleneck and the bottleneck to the destination are bandwidth of 15 Mbps and 20 ms latency change over the specific model. Transmission bottleneck in the script in this transmission bandwidth is 15Mbps and latency of 20 ms. Bottlenecks in the network nodes are installed to evaluate the algorithm. Buffer size at the bottleneck Q vary from case to evaluate.

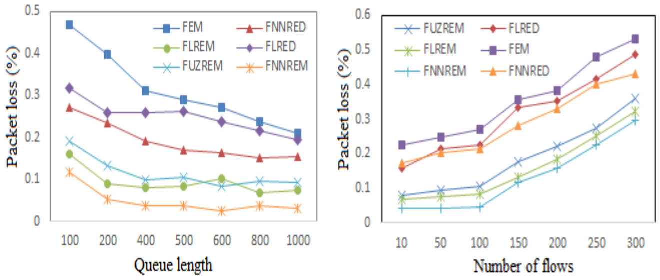
Establish communication protocols TCP/NewReno with congestion window is 240 packets, each packet size of 1000 bytes. Buffer size of all queues is 500 packets. Reference queue length for the mechanism to be set to 200 packets (40% buffer size). Sampling distance is 0.1 seconds, the execution time for the simulation program for mechanisms is 100 seconds.

To evaluate the mechanism when the volatility of the network, we built two main scenarios: the queue length at the bottleneck change from 100 to 100 while the number of connections does not change (60 connections) and the packet flow to change (change of flow) from 10 to 300 in the queue length at the bottleneck constant (500 packets).

## 7.2. Evaluation FNNRED and FNNREM

### 7.2.1. Evaluation of packet loss rate

From the graph Figure 3.21, shows that when the queue size at the router increases then the packet loss rate of mechanisms are reduced and when the increased number of connections to the router then the packet loss rate increases.



(a) Packet loss vs. Queue length

(b) Packet vs. Number of flows

*Figure 17.. Packet loss rate of AQM mechanisms*

All these mechanisms have low packet loss ratio, even in the case of maximum load (the number of flow is 300), this figure is less than 0.6%. This is due to the mechanism using fuzzy control to control the queue to queue instantaneous value around the reference queue, should remain stable queue length, leading to the variable natural small latency low packet loss ratio.

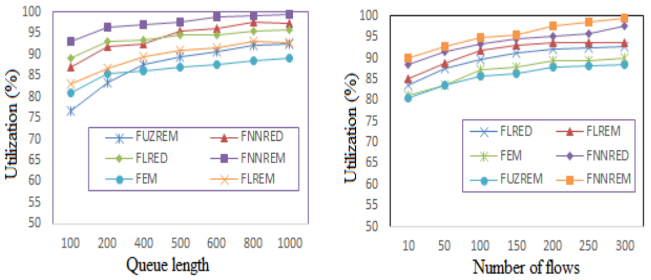
In all cases FEM always high packet loss rate and FNNREM packet loss rate lowest. This is due to the fuzzy controller of FEM using Mamdani fuzzy with triangular membership function and the parameters of the fuzzy system it is not optimal. Conversely, due FNNREM use Sugeno fuzzy systems with bell-shaped function of parameter values 9 domains and the optimal fuzzy FNNREM should have the lowest rate of loss.

The mechanism AQM improve REM (FUZE, FLREM, NREM) control queues based on the queue length and packet flow to, packet loss rate should be lower than the AQM mechanism improved RED (FEM , FLRED, FNNRED) based on factors queue length.

The control queue of AQM mechanisms depends heavily on the fuzzy controller. From the graph in Figure 17, shows that the improvement with a traditional mechanisms (RED, REM), the mechanism using the adaptive fuzzy AFC will packet loss rate be lower than the mechanisms used the traditional fuzzy but the packet loss rate is higher than the mechanisms used fuzzy controller FNN.

### 7.2.2. Evaluation of Link Utilization level

The graph in Figure 18, indicates the level of use link utilization level of the mechanisms. The ability to make use transmission of the mechanisms increased when the queue size and the number of connections increases. In all cases, almost all of the mechanisms used are level on 80% bandwith, FUZREM mechanism always use the lowest transmission, FNNREM mechanism always use the highest transmission. This suggests that the efficiency of the mechanism when using the fuzzy controller FNN, when FUZREM and FNNREM improved REM mechanism, but using different fuzzy systems, FUZREM fuzzy systems used traditionally fuzzy system, FNNREM fuzzy systems used FNN fuzzy system.



(a) Utilization vs. Queue length

(b) Utitlization vs. Number of flows

*Figure 18. Link Utilization level of AQM mechanisms*

Based on the graph, find the partition mechanism on the level of traffic. In both graphs of Figure 18, the level of use of transmission is increased from group mechanisms using traditional fuzzy controller (such as FEM, FUZREM), followed by a group of mechanisms used controller AFC adaptive fuzzy (as FLRED, FLREM) to group the mechanism used fuzzy controller FNN (as FNNRED, FNNREM). This is consistent with the results of theoretical analysis, the AFC using Sugeno fuzzy systems have mechanisms to adjust output parameters K and the method of determining the form $G_m$ for the target value, and FNN is built from AFC by train to get the value for the parameter optimization.

## 8. Conclusions

This paper has proposed FNN model which is the combination of Sugeno fuzzy system with supervised learning techniques on the neural network according to back-propagation algorithm with error is the average of the squared error. This model has solved the problem posed in the introduction which is how to get the best parameters for the fuzzy system? The results of the combination is illustrated by the construction of the two FNNRED and FNNREM mechanisms. The simulation results show the performance of AQM mechanisms to increase when applying of the traditional fuzzy controllers, adaptive fuzzy controller - AFC and self-learning FNN fuzzy controller to improve AQM mechanisms.

## References

[1] Athuraliya Sanjeewa, Steven H. Low, Victor H. Li and Qinghe Yin, “REM Active Queue Management”, IEEE Network, 2001

[2] Arash Dana and Ahmad Malekloo, “Performance Comparison between Active and Passive Queue Management”, IJCSI

International Journal of Computer Science Issues, Vol. 7 Issue 3 No 5, 2010.

[3] Bartek Peter Wydrowski, Techniques in Internet Congestion Control, Electrical and Electronic Engineering Department The University of Melbourne, 2003.

[4] Bart Kosko (1997), Fuzzy Engineering, Prentice Hall, USA.

[5] Braden B., Clark D., Crowcroft J., Davie B., Deering S., Estrin D., Floyd S., Jacobson V., Minshall G., Partridge C., Peterson L., Ramakrishnan K. K., Shenker S., and Wrocruleski J., (1998), "Recommendations on queue management and congestion avoidance in the internet." Internet Draf

[6] C. Chrysostomou, A. Pitsillides, G. Hadjipollas, M. Polycarpou, A. Sekercioglu (2004), Fuzzy logic control for active queue management in TCP/IP Networks. 12th IEEE Mediterranean Conference on Control and Automation Kusadasi, Aydin, Turkey, (IEEE MED'04), pp. 2-8

[7] C. V. Hollot, V. Misra, D. Towsley, and W. Gong (2002). Analysis and design of controllers for AQM routers supporting TCP flows. IEEE Trans. on Automat. Control, 47:945–959, jun 2002

[8] F.L. Lewis, J. Camos, R. Selmic (2002), Neuro-Fuzzy Control of Industrial Systems with Actuator Nonlinearities, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, USA.

[9] George J. Klir, Ute St. Clair, Bo Yuan (1997), Fuzzy Set Theory: Foundations and Applications, Prentice Hall, New Jersey, USA.

[10] G.F.Ali Ahammed, Reshma Banu, "Analyzing the Performance of Active Queue Management Algorithms", International journal of Computer Networks & Communications (IJCNC), Vol.2 No.2, 2010

[11] G.Thiruchelvi and J.Raja, "A Survey On Active Queue Management Mechanisms", IJCSNS International Journal of Computer Science and Network Security, Vol.8 No.12, 2008

[12] Hugang Han, Chun-Yi Su, and Yury Stepanenko (2001), "Adaptive Control of a Class of Nonlinear Systems with Nonlinearly Parameterized Fuzzy Approximators", IEEE Transactions on Fuzzy Systems, Vol. 9, No. 2, pp.315– 323.

[13] I. K. Tabash, M. A. A. Mamun and A. Negi "A Fuzzy Logic Based Network Congestion Control Using Active Queue Management Techniques", J. Sci. Res. 2 (2), 273-284, 2010

[14] Jang-Hyun Park, Seong-Hwan Kim, Chae-Joo Moon (2006), "Adaptive Fuzzy Controller for the Nonlinear System with Unknown Sign of the Input Gain", International Journal of Control, Automation, and Systems, Vol. 4, No. 2, pp. 178–186.

[15] Jan Jantzen (2003), Neural and Neurofuzzy Control, Tech. report no 99-H 999, Technical University of Denmark.

[16] Jasem, H.N., Z.A. Zukarnain, M. Othman and S. Subramaniam (2011), Efficiency and fairness of new-additive increase multiplicative decrease congestion avoidance and control algorithm. J. Applied Sci., 11: 438-449

[17] Jeffrey T. Spooner, Mangredi Maggiore, Raúl Ordónez, Kelvin M. Passino (2002), Stable Adaptive Control and Estimation for Nonlinear Systems: Neural and Fuzzy Approximator Techniques, Wiley Interscience, USA.

[18] Jyh-Shing Roger Jang (1993), "Adaptive-Network-Based Fuzzy Inference System", IEEE Transactions on Systems, Man and Cybernetics, vol. 23(3), pp.665–685.

[19] Jyh-Shing Roger Jang, Chuen-Tsai Sun (1995), "Neuro-Fuzzy Modeling and Control", The Proceedings of the IEEE, vol. 83(3), pp.378–406.

[20] Jyh-Shing Roger Jang, Chuen-Tsai Sun, Eiji Mizutani (1996), Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence, Prentice Hall, USA.

[21] Karl J. Åström, Björn Wittenmark (1995), Adaptive Control, 2nd Edition, Addison-Wesley Publishing, USA.

[22] K.Chitra and G.Padamavathi, "Classification and Performance of AQM-Based Schemes for Congestion Avoidance", (IJCSIS) International Journal of Computer Science and Information Security, Vol.8 No.1, 2010

[23] Kevin Fall and Kannan, The ns Manual, A Collaboraten between researchers at UC Berkeley, LBL, USC/ISI, and Xerox PARC, 2010.

[24] Long Le, Kevin Jeffay, F. Donelson Smith, "A Loss and Queuing-Delay Controller for Router Buffer Management", Proceedings of the 26th IEEE International Conference on Distributed Computing Systems (ICDCS'06), 2006

[25] Michael A. Arbib (2003), The Handbook of Brain Theory and Neural Networks, The MIT Press, USA.

[26] Nguyen Kim Quoc, Vo Thanh Tu, "Improving service quality in Internet routers", Proceedings of the National Workshop on the 14th - Some selected issues of information technology and communications (@), Can Tho, pp. 322-342, 2011

[27] Nguyen Kim Quoc, Vo Thanh Tu, "Performance evaluation of a number of mechanisms active queue management based on queue size and transduced", Scientific Journal of Hue University, Volume 74A, Number 5, pp. 109-119, 2012

[28] Nguyen Kim Quoc, Vo Thanh Tu, Nguyen Thuc Hai, "Proposed mechanisms active queue management on high-speed network environment", Proceedings of the 6th National Conference - Basic research and application of information technology (FAIR), Thua Thien Hue, pp. 108-115, 2013

[29] Nguyen Kim Quoc, Vo Thanh Tu, Nguyen Thuc Hai, "Fuzzy Logic Control for SFB Active Queue Management Mechanism", ICCASA 2013, Springer, Vol. 128, No. 3, pp. 97-104, 2013

[30] Nguyen Kim Quoc, Vo Thanh Tu, Nguyen Thuc Hai (2014), "Improving control mechanism at routers in TCP/IP networks", Scientific Journal of EAI Transactions on Context-Aware Systems and Applications, Vol. 1, ISSN 2409-0026, pp. 52-66.

[31] Nguyen Kim Quoc, Vo Thanh Tu, Nguyen Thuc Hai (2015), " Some Improvements the Active Queue Management Mechanism Based on Adaptive Fuzzy Control", Scientific Journal of EAI Transactions on Context-Aware Systems and Applications, Vol. 5, ISSN 2409-0026, pp. 45-55

[32] Robert Fullér (1995), Neural Fuzzy Systems, Åbo Akademi University.

[33] S. H. Low, F. Paganini, and J. C. Doyle (2002). Internet congestion control. IEEE Control Systems Magazine, Feb 2002

[34] V. Santhi and A. M. Natarajan, "Active Queue Management Algorithm for TCP Networks Congestion Control", European Journal of Scientific Research ISSN 1450-216X, Vol.54 No.2, pp.245-257, 201

[35] V. Misra, W. Gong, and D. Towsley (1999), Stochastic differential equation modeling and analysis of TCP windowsize behavior. Technical report, University of Massachusetts, October 1999

[36] V. Misra, W. Gong, and D Towsley. Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED. In SIGCOMM, pages 151–160, August 2000