EAI Endorsed Transactions
on Context-Aware Systems and Applications
Research Article ICST.ORG

Improving Control Mechanism at Routers in TCP/IP Network

Nguyen Kim Quoc[1,]*, Vo Thanh Tu[2] and Nguyen Thuc Hai[3]

[1]Nguyen Tat Thanh University, Vietnam, email: quocnknet@yahoo.com,
[2]College of Sciences, Hue University, Vietnam, email: vothanhtu_hue@yahoo.com
[3]Ha Noi University of Science and Technology, Vietnam, email: haint.fit@gmail.com

Abstract

The existing control mechanisms at the network nodes have a good active and very effective at each local router, but they do not still strong enough to control nonlinear and dynamical behaviour of the network. Therefore, the control system requirements must be designed to be flexible to fully grasp the important status information of the variation and intelligent control methods to control network congestion in nonlinear network. To solve this problem, we propose a solution combined fuzzy reasoning with neural network control put on active queue management mechanisms at the network nodes

Received on 27 June 2014; accepted on 27 June 2014; published on 05 September 2014

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

doi: 10.4108/casa.1.1.e6

1. Introduction

Active queue management (AQM) is a queue control mechanisms and transduction activity in the network nodes, control the number of packets in router queues by scheduling and active packet to remove a blockage or notification to regulate traffic on the network [2][3]. In recent years, researchers have proposed a number of queue management mechanisms in the network nodes based on the size of the queue elements such as RED, FRED [23][28] and transduced as BLUE, SFB [15][20]. The mechanism for active queue management based on transduction works well in environments transduction network in the router changes, but not effective in the network environment in the queue size of different routing [13][22][23]. Mechanism of active queue management REM [1][32] overcomes the drawbacks of the mechanism on the way manage queues based on queue size and transduction in the router. REM was evaluated based on the specific material [1][11][14][15]. However, this mechanism is controls drawback and static in nature. In some cases, the information is not updated in accordance with the status of the network changes. Because of the above reasons, we chose to do queue management mechanism REM to improve positive, supplemented by fuzzy neural network system on it.

Although the study of nonlinear control has made important progress, however, the problem becomes more complex nonlinear systems containing unknown ingredients destabilize the system. The characteristics of this unknown may stem from various factors at the input, such as fluctuations in traffic load, the diversity of application types, change the connection network model. For stability control system, the design method using fuzzy neural control (also known as fuzzy neural network used) to approximate the unknown component, which seeks to reduce the impact of excluding components to achieve the best quality control. The application of fuzzy neural control also allows the development of adaptive control, so the parameters can be adjusted online during system operation. Artificial neural networks are often used to adjust the membership functions of the fuzzy system in the control device. Although fuzzy logic can encode expert knowledge directly using rules with linguistic labels of fuzzy logic but requires a lot of experience and does not allow automatic adjustment of the membership functions to the best of quantitative the input linguistic variables. Neural network training techniques to automate this process and dramatically reduce time and development costs while improving processing speed.

For neural networks, knowledge can be obtained automatically by the regression algorithm but the training process is relatively slow and analysis the trained network is difficult. Also we do not have the ability to draw structured knowledge (rules) from the neural network was trained, and cannot put more information into the neural network known to simplify the process of training Network. The fuzzy system is better in sense of their activities, can be explained based on the fuzzy rules and thus the performance of the system can be changed by adjusting the rules. However, the common knowledge is obtained quite difficult and the input variable to split into several regions, the application of fuzzy systems are limited in areas where expert knowledge to have, as well as in fact most are applicable only to the

*N. K. Quoc: quocnknet@yahoo.com

number of small input variables. Combines the advantages of fuzzy systems and neural networks led to hybrid systems, structures are widely used as hybrid fuzzy neural network. In modern control theory, fuzzy systems, neural networks and a combination of fuzzy systems with neural networks are considered versatile tool to solve nonlinear problems and uncertainties in general nonlinear control systems.

In this paper, we have developed a mechanism for active queue management FNNREM (REM Fuzzy Neural Network) based on the use of fuzzy neural network controller, which is integrated into the mechanism of active queue management REM, in order to proactively detect and control congestion. The results of the analysis and evaluation of simulation experiments based on NS2 software was installed, showed that the mechanism active queue management FNNREM works well at each router, reducing packet drop, reducing the latency and increase throughput flows. Therefore, queue management mechanism FNNREM has new advanced network performance, respond quickly to the changes of network traffic packets on the transmission line should have been improving the quality of network services. The contents of this article are presented consists of 5 parts: The first part of the article pointed out the necessity of queue management and idea management mechanism proposed new queue FNNREM. The second part of the article is to present the mathematical basis of the mechanism of active queue management and operation of queue management mechanism in the network environment TCP/IP. The third part, the paper presents the mathematical basis of fuzzy neural networks in active queue management at the network nodes. The fourth part, presented proposed new queue management mechanism FNNREM of authors. Finally, based on our theoretical analysis and implementation results, to draw conclusions and future research directions of the authors.

2. The control system TCP/AQM

In this section, we analyse the dynamical models of TCP, the stability of the control system TCP/AMQ and the operation mechanism of the active queue management REM in routers.

2.1. Dynamical model of TCP

We can realize that the entire TCP/IP is a feedback system linked as described in Figure 1.

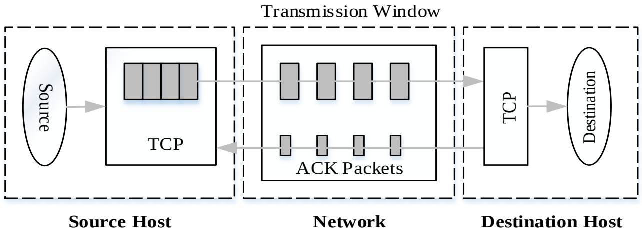

Figure 1. Models of TCP flow control

Therefore, feedback control theory is an appropriate tool for analysing and design of TCP/AQM algorithm. To use control theory, the paper presents a mathematical model of TCP behaviour was developed in [4] and [33]. This model is based on two assumptions: TCP traffic is seen as fluid flows and the packet loss is described obey Poisson distribution.

The algorithm avoids congestion on the network TCP/IP was developed to adjust the speed of the traffic flow so close to the limit as possible to convey maximum information transmission and avoid network congestion. The basic premise of this algorithm is considered a packet loss is equivalent to signal congestion. The principle of the algorithm is used to control the speed of traffic sources according to state this principle is quite simple. When starting operation, each source flow rate increases. This increase will take place continuously until packet loss occurs. This means that congestion is detected somewhere in the network. Therefore, traffic sources must slow down enough to get rid of congestion. This is the basic point of the congestion control algorithms, Additive Increase Decrease multiplicative (AIMD) on network TCP/IP [16].

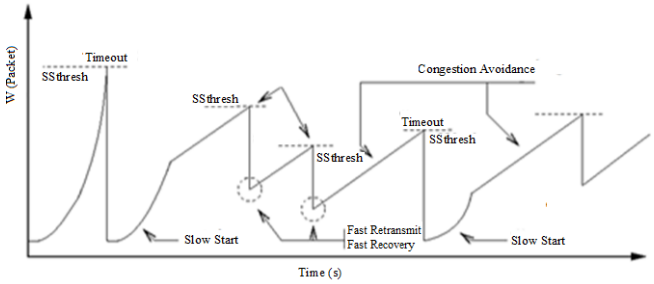

Figure 2. The control mechanism of TCP

Slow start algorithm and congestion [31]: Includes two-phase slow start and congestion avoidance. Initially, active TCP slow start phase. The goal is to get a congestion threshold prediction. Start slow start phase, TCP settings *cwnd =1*. Each time an ACK is received, *cwnd* will increase a unit (but not beyond that window to receive the message receiver). So, *cwnd* will increase exponentially until it reaches the slow start threshold (Ssthresh), then we move to avoid congestion. In the congestion avoidance phase, TCP will adjust *cwnd=cwnd +1/cwnd* each time an ACK is received until the packet loss occurs. Upon detection f packet loss, will put the source *sthresh = cwnd/2*, retransmit the lost packets and return to slow start phase by resetting *cwnd* = 1.

Fast retransmit algorithm [31]: The goal is to restore the operation of more efficient TCP from packet loss. When you get two identical copies acknowledgment, TCP as this is the case repeat the acknowledgment and three transmission lines were congested. Rather than waste time waiting for timeout, the source will decrease speed and retransmit lost packets.

Fast recovery algorithms [31]: When the packet loss is detected by the phenomenon repeated acknowledgment, returning TCP slow start phase by placing s*thresh = cwnd/2* and *cwnd=1*. If the window size is large and the error rate is small, then instead of continuing slow start algorithm, TCP will move fast recovery algorithms. At this time, congestion window *cwnd = Ssthresh/2 +3* phase and go straight to avoid congestion.

That means that when the packet loss occurs, all of the TCP connections react to the loss. The packet loss on networks not only occur in a single source that actually take place in a variety of sources. TCP is designed to ensure fairness in the connection to each source window size changes simultaneously. The first assumption implies that the congestion window will increase continuously instead of jumping level. Window size will increase after each RTT and expressed as dt/RTT. Second assumption process modelling packet loss. Assuming the packet loss is random and obey Poisson distribution. Based on the above two assumptions, the variation of congestion window size W can be described as follows:

$$dW(t) = \frac{dt}{RTT} - \frac{\mathrm{W}}{2} dN(t) \tag{1}$$

in that, dN(t) is defined as: dN(t)=1 if packet loss occurs, dN(t)=0 if no packet loss occurs. Equation (1) aspect reflects "Additive Increase Decrease multiplicative" of TCP. The first component corresponding to the Additive Increase, the window size will increase by one after each RTT. The second component corresponding to the Decrease multiplicative, halving the window size for each packet loss.

2.2. The control system TCP/AQM

Using stochastic analysis and difference equation to equation (1), considered on the basis of assumptions, documents [4] provide dynamic models of the behavior of TCP. This model consider a system in which there is a single congested router with transmission capacity is C. Combining the router is an AQM algorithm is characterized by a probability function drop package p, to stabilize the average queue length at the router and average congestion window size. The models proposed from [7] then takes the form:

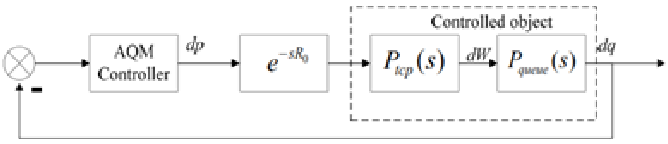

Figure 3. Flow chart of the control system TCP/AQM

$$\dot{w}(t) = \frac{1}{R(t)} - \frac{w(t)}{2}\frac{w(t-R(t))}{R(t-R(t))} p(t-R(t)) \tag{2}$$

$$\bar{q}(t) = -C(t) + \frac{N(t)}{R(t)} w(t) \tag{3}$$

$$R(t) = \frac{q(t)}{C(t)} + T_p \tag{4}$$

$$P_{queue}(s) = \frac{q(s)}{W(s)} = \frac{\frac{N}{R_0}}{s + \frac{1}{R_0}} \tag{5}$$

$$P(s) = P_{TPC}(s)\mathrm{P}_{queue}(s)\mathrm{e}^{-sR_0} \tag{6}$$

$$P(s) = \frac{\delta q(s)}{\delta p(s)} = \frac{\frac{C^2}{2N} e^{-sR_0}}{\left(s + \frac{2N}{R_0^2 C}\right)\left(s + \frac{1}{R_0}\right)} \tag{7}$$

In particular: operator $\dot{x}$ denotes the derivative of x over time, W: the average window size TCP, $\overline{\mathrm{q}}(t)$: the average queue length, R(t) : cycle time for a packet, C: bandwidth, T_p : transmission delay, N(t): load factor (sessions number of TCP) and p: probabilistic marking or removing packages.

The dynamical behaviour of TCP is described in the operation of a nonlinear system over time delay should be difficult in the analysis control theory perspective. Therefore, we are only interested in the design of AQM algorithms operate around a linear balance point (w_0,q_0,p_0). For the linearized model in Figure 3, it is assumed that the number of TCP sessions and routing capacity is constant. So N(t)≡N and C(t)≡C. Follow [7] linear model works around balance point is determined as follows:

$$\delta\dot{\mathrm{W}}(t) = -\frac{N}{R_0^2 C}(\delta\mathrm{W}(t) + \delta W(t-h)) - \frac{1}{R_0^2 C}(\delta q(t) + \delta q(\mathrm{t}-\mathrm{h})) - \frac{R_0 C_2}{2N^2}(\delta p(\mathrm{t}-\mathrm{h})) \tag{8}$$

$$\dot{q}(t) = -\frac{N}{R_0^2}(\delta W(\mathrm{t})) - \frac{1}{R_0^2}\delta q(t) \tag{9}$$

In particular: $\delta\mathrm{W} = \mathrm{W} - \mathrm{W}_0$, $\delta q = q - q_0$ is the state variable, $\delta p = p - p_0$ is the input.

Equation (8) and (9) also written as:

$$\begin{bmatrix} \delta\dot{\mathrm{W}} \\ \delta\dot{q} \end{bmatrix} = \begin{bmatrix} -\frac{N}{R_0^2 C} & -\frac{1}{R_0^2 C} \\ \frac{N}{R_0} & -\frac{1}{R_0} \end{bmatrix} \begin{bmatrix} \delta W \\ \delta q \end{bmatrix} + \begin{bmatrix} -\frac{N}{R_0^2 C} & -\frac{1}{R_0^2 C} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta W(t-h) \\ \delta q(t-h) \end{bmatrix} + \begin{bmatrix} \frac{R_0 C^2}{2N^2} \\ 0 \end{bmatrix} \delta p(t-h) \tag{10}$$

2.3. Analysis of the stability of AQM algorithms

The AQM algorithms need to maintain the operation of the closed-loop stability despite changing conditions on network activity. The changes include the number of TCP sessions, the change of the transmission delay T and routing capacity C. In this method, we view the delay as the noise impacts affecting the kinetic characteristics common. The objective of the algorithm is to achieve stability in relation to late. To achieve this, the system is divided into two parts as above. In particular, Δ disturbances and Σ is the usual kinetic.

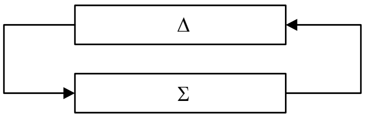

Figure 4. Linear analysis model for AQM

From the model, according to equation (10), we analyzed the kinetics of the usual two and interference:

$$\begin{bmatrix}\delta\dot{W}\\ \delta\dot{q}\end{bmatrix}=\begin{bmatrix}-\dfrac{N}{R_0^2C} & -\dfrac{1}{R_0^2C}\\ \dfrac{N}{R_0} & -\dfrac{1}{R_0}\end{bmatrix}\begin{bmatrix}\delta W\\ \delta q\end{bmatrix}+$$

$$\begin{bmatrix}-\dfrac{N}{R_0^2C} & -\dfrac{1}{R_0^2C}\\ 0 & 0\end{bmatrix}\begin{bmatrix}\delta W(t-h)\\ \delta q(t-h)\end{bmatrix}+ \tag{11}$$

$$\begin{bmatrix}\dfrac{R_0C^2}{2N^2}\\ 0\end{bmatrix}\delta p(t-h)$$

use $\dot{x}(t)=\begin{bmatrix}\delta W\\ \delta q\end{bmatrix}$, $D_h v(t)=v(t-h)$ is the delay operator, the above equation can be rewritten as:

$$\dot{x}(t)=Ax(t)+B\omega_1(t)+G\omega_2(t) \tag{12}$$

with $\omega_1(t)=D_h\delta p(t)$, $\omega_2(t)=(1-D_h)\,x(t)$

$$A=\begin{bmatrix}-\dfrac{2N}{R_0^2} & 0\\ \dfrac{N}{R_0} & -\dfrac{1}{R_0}\end{bmatrix},\ B=\begin{bmatrix}-\dfrac{R_0C^2}{2N^2}\\ 0\end{bmatrix},\ G=\begin{bmatrix}-\dfrac{N}{R_0^2C} & -\dfrac{1}{R_0^2C}\\ 0 & 0\end{bmatrix}$$

Figure 5 shows the AQM control system has been linearized. We determine the transfer function by the Laplace transform, equation (12):

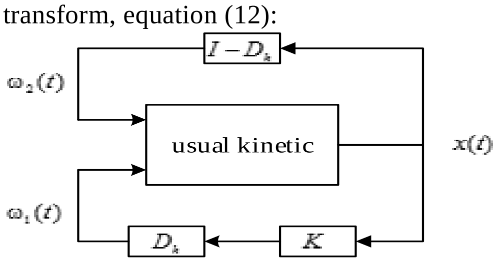

Figure 5. AQM linear system models

$$sX(s)=Ax(s)+BW_1(s)+G\Delta_h(s)X(s); \tag{13}$$

$$\Delta_h(s)=1-e^{-sR_0}$$

$$\delta Q(s)=[01]X(s)=CX(s)$$

$$\frac{\delta Q(s)}{W_1(s)}=C[sI-A-G\Delta_h(s)]^{-1}B \tag{14}$$

Calculate:

$$\frac{\delta Q(s)}{W_1(s)}=\frac{\dfrac{N}{R_0}\cdot\dfrac{R_0C^2}{2N^2}}{\left(s+\dfrac{2N}{R_0^2C}\right)\left(s+\dfrac{1}{R_0}\right)-\dfrac{N}{R_0^2C}s\Delta_h} \tag{15}$$

$$=\frac{P(s)}{1-P(s)\Delta(s)}$$

Here, the transfer function P(s) in equation (16) shows the kinematics influence of probability mark or remove packages on the queue length. Transfer function Δ (s) representing the noise impact:.

$$P(s)=\frac{\dfrac{C^2}{2N}}{\left(s+\dfrac{2N}{R_0^2C}\right)\left(s+\dfrac{1}{R_0}\right)} \tag{16}$$

$$\Delta(s)=\frac{2N^2s}{R_0^2C^3}(1-e^{-sR_0}) \tag{17}$$

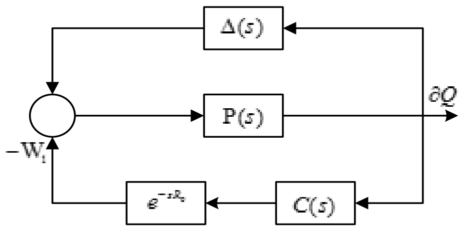

Figure 6. The control system feedback AQM

2.4. Stabilizing AQM control rule

Based on Figure 6, we see C (s) is the control rule. The stability of the closed loop is basic factors to meet operational goals. References [7] provide conditions for C (s) stable objects usually late $P(s)e^{-sR_0}$ and stable confounding factors impact $\Delta(s)$. In the following statement, we assume that there exists a transfer function:

$$V(s)=\frac{P(s)}{1+P(s)C(s)e^{-sR_0}} \tag{18}$$

Statement:

With the feasible network parameters $\eta=(N,C,Tp)$ and operational point linear control AQM described in (12) is stable if C(s) stabilizing kinetic object usually late $P(s)e^{-sR_0}$ and noise $\Delta(s)$ amplification coefficient stability, ie: $|\Delta(j\omega).V(j\omega)|<1,\forall\omega>0$

Proof:

If C (s) stabilizing kinetic object usually late $P(s)e^{-sR_0}$ then V (s) stable. Because Δ(s).V(s) stability (gain ratio amplifies small $|\Delta(j\omega).V(j\omega)|<1)$ along with the conditions of the Nyquist stability leads to a stable closed-loop.

2.5. Operation of REM mechanism

The first idea of REM is stable input rate and link capacity of the queue, regardless of the number of users sharing links [4][23]. Each output queue of router installed REM mechanism and maintains a variable called 'price'. Price as a factor in evaluating congestion. Price is updated periodically or asynchronously, based on the asymmetry of the asymmetric load and the asymmetry queue size. The load is unworthy of the difference between the speed of the flow of data into and existing capacity of the link. Unworthiness of the queue size is the difference between the required target size queue with queue size of the router.

This price increase if the total weight of unworthiness is positive, and decreases in the opposite case. Total weight is positive when one input exceeds the binding capacity or there are too many packages in the queue backlog should be cleared, and negative in the opposite case. As the number of users increases, the load is not uniform among the queue size increases, boosting the price and therefore the probability of congestion marked. This will send a stronger signal congestion to the source, then the source of reduced growth. When downloading the source is too small, will not sync negative, and marked probability reduce and enhance the of load power. Until unworthiness is towards to zero, high efficiency and negligible decrease, so the delay in equilibrium. The queue is deleted in equilibrium if the queue set to 0.

REM clear control through updates its price, this is the first character of REM. Exactly, queue size l, price $p_l(t)$ in stages t is updated by the following formula [1][26]:

$$p_l(t+1)=\left[p_l(t)+\gamma(\alpha_l(b_l(t)-b_l^*)+x_l(t)-c_l(t))\right]^+ \qquad (19)$$

Where, $\gamma>0$ and $\alpha_l>0$ constants are small and $[z]^+=\max\{z,0\}$. Here, $b_l(t)$ the average queue size of the queue l in time t and $b_t^*\geq 0$ the target queue length, x(t) average download speeds of queue l at the time of t, and $c_l(t)$ allowed bandwidth of queue l at the time of t. The difference in load is $x_l(t)-c_l(t)$ and the queue size $b_l(t)-b_l^*$. The constant α_l can be set up by each individual queue and are updated according to the performance and latency in each queue. The constant γ REM control response changes depending on network conditions. Parameters price will increase if the speed deviation of the load and queue size is positive and the price will decrease in the opposite case. At equilibrium, stability and p is calculated using the formula:

$$\alpha_l(b_l(t)-b_l^*)+x_l(t)-c_l(t)=0 \qquad (20)$$

This can be kept when loading the input speed equal to the speed of processing in routers ($x_l=c_l$) and the number of packets in the queue with the desired queue size $b_l=b_l^*$

REM's second idea is to use a total of price links along the path of the packet, this total reflects signs of congestion on the path, the total price to be embedded into the terminal to mark probability. Forms exponentially increase the probability of marking is very important in a large network. At last, the probability of marking a packet passes through multiple congested link from source to destination depends on the probabilities associated marks at all links in the path. When and only when the associated component marking probability is exponential in its price link. This led to the arrival terminal will be marked probability increases exponentially with the total price of all links in congested link in its path. This total is a true measure of congestion in the path of the packets. Since it is embedded in the terminal to mark probability, it can be easily estimated from the sources of its own packets are marked, and is used to design adaptation loads of it.

Suppose that a packet passes through the links l=1,2, …, L have price $p_l(t)$ at time t, the marking probability $m_l(t)$ queue l at time t is:

$$m_l(t)=1-\phi^{-p_l(t)} \qquad (21)$$

Where, $\phi>1$ is constant and marking probability at the end of the packet is [24][25][28]:

$$1-\prod_{l=1}^{L}(1-m_l(t))=1-\phi^{-\sum_l p_l(t)} \qquad (22)$$

Probability high marks when large sum $\sum_l p_l(t)$. When the marking probability at the link $m_l(t)$ small, should $p_l(t)$ small. So, the probability is marked at the end by the above formula is proportional to the sum of the $p_l(t)$ along the way and mark probability at end approximately: $(\log_e\phi)\sum_l p_l(t)$.

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 f and $\tilde{g}>0$ so that the error of approximation $\dot{x}_n$ satisfying $|\Delta_{dxn}|\leq$ w với $\forall x\in\Omega_x, \forall x\in\Omega_x$. 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 Ω_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 $Z = \{F(x,\theta) : \theta \in \Omega_F \subset \Re^p, p \geq 0\}$ is class contains all functions of the form $F(x,\theta)$, $\theta \subset \Omega_F$ 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 $F(\mathbf{x}) \in Z$ to $\sup_{x \in D} |F(x) - f(x)| < \varepsilon$.

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 $\sup_{x \in D} |F(x) - f(x)| < \varepsilon$.

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], [35]) 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 $Z_p = \left\{ g(x) = \sum_{i=0}^{p} a_i x^i : a_0, a_1, \ldots a_p \in \Re, p \geq 0 \right\}$ with g:D→ℜ, D⊆ℜ (The theorem Weiertrass); functions defined function class $Z_a = \left\{ g(x) = \sum_{i=1}^{m} a_i \cos(b_i^T x + c_i) : a_i, c_i \in \Re, b_i \in \Re^n \right\}$; 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], [35]), using neural networks ([14], [35]) 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 $y \in \Re$ (Figure 7). 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:

$$\Re_i : \left(A^1_{J_{1i}} \cap A^2_{J_{2i}} \cap ... A^n_{J_{ni}}\right) \Rightarrow B_{ki} \quad (23)$$

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:

$$A^i_j = \left\{\left(x_i, \mu_{A^i_j}(x_i)\right) : x_i \in \Re\right\}$$
$$B_k = \left\{\left(y, \mu_{B_k}(y)\right) : y \in \Re\right\} \quad (24)$$

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 (23) 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 (23) calculated as follows:

$$\mu_{A^1_{J_{1i}} \times A^2_{J_{2i}} \times ... \times A^n_{J_{ni}}}(x) = \min\left\{\mu_{A^1_{J_{1i}}}(x_1), \mu_{A^2_{J_{2i}}}(x_2), ..., \mu_{A^n_{J_{ni}}}(x_n)\right\} \quad (25)$$

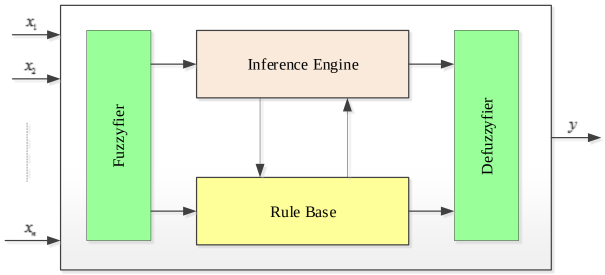

Figure 7. MISO fuzzy control system

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 $C_i = \left\{\left(y, \mu_{C_i}(x,y)\right) : x \in \Re^n, y \in \Re\right\}$ calculated as follows:

$$\mu_{C_i}(x,y) = \mu_{A^1_{J_{1i}} \times A^2_{J_{2i}} \times ... \times A^n_{J_{ni}}}(x,y)$$
$$= \min\left\{\mu_{A^1_{J_{1i}} \times A^2_{J_{2i}} \times ... \times A^n_{J_{ni}}}(x), \mu_{B_{ki}}(y)\right\} \quad (26)$$

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

$$y = F(x,\theta) = \frac{\sum_{i=1}^{p} C_i \int_z \mu_{C_i}(x,z)dz}{\sum_{i=1}^{p} \int_z \mu_{C_i}(x,z)dz} \quad (27)$$

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

The equation (27) 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 $i = 1..p-1$;

w_{jk}^{i} is the weight from node k in layer i-1 to node j in layer i with i = 1.. p,

$$j=\begin{cases}\{1..L_i, 1\le i<p\\ 1..m, i=p\end{cases};\quad k=\begin{cases}\{1..L_i, 1<i\le p\\ 1..n, i=1\end{cases}$$

θ_j^i is the threshold value of node j in layer i with i = 1.. p, $j=\begin{cases}1..L_i, 1\le i<p\\ 1..m, i=p\end{cases}$

σ_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:

$$o_j^i=\begin{cases}\sigma_j^1\left(\sum_{k=1}^{n} w_{jk}^1 x_k+\theta_j^1\right), i=1\\ \sigma_j^i\left(\sum_{k=1}^{L_{i-1}} w_{jk}^i o_k^{i-1}+\theta_j^i\right), 1<i\le p-1\end{cases}$$
$$y_r=\begin{cases}\sum_{s=1}^{n} w_{rs}^1 x_s+\theta_r^1,\ p=1\\ \sum_{s=2}^{L_{p-1}} w_{rs}^p o_s^{p-1}+\theta_r^p,\ p\ge 2\end{cases}\qquad (28)$$

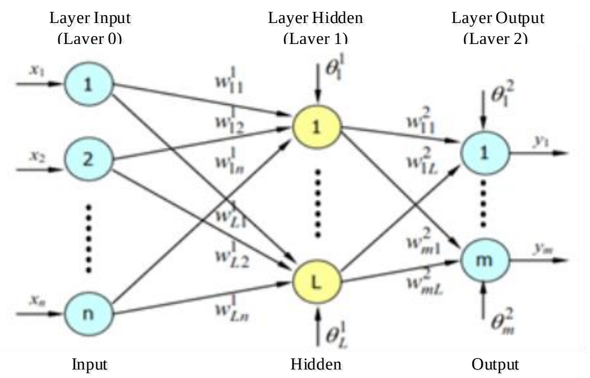

Figure 8. System 2-layer feed forward neural networks

Figure 8 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:

$$y_i=\sum_{j=1}^{L} w_{ij}^2\sigma_j\left(\sum_{k=1}^{n} w_{jk}^1 x_k+\theta_j^1\right)+\theta_i^2 \qquad (29)$$

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:

$$F(x,\theta)=y_1=\sum_{j=1}^{L} w_{1j}^2 sig\left(\sum_{k=1}^{n} w_{jk}^1 x_k+\theta_j^1\right)+\theta_1^2 \qquad (30)$$

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:

$$R_i:\left(A_{J1i}^1\times A_{j2i}^2\times\ldots\times A_{Jni}^n\right)\Rightarrow g_i(x) \qquad (31)$$

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:

$$F(x,\theta)=\theta^T\varphi(x) \text{ hay } \frac{\partial F(x,\theta)}{\partial\theta}=\varphi^T(x) \qquad (32)$$

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 (27) and RBN neural networks (NN Radial Basis).

In case the approximation using fuzzy systems under (27) if $\theta=\left[c_1,\ldots,c_p\right]^T$ then :

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

$$\varphi_i(x)=\frac{\int_z \mu_{C_i}(x,z)dz}{\sum_{i=1}^{p}\int_z \mu_{C_i}(x,z)dz}$$ 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 $\boldsymbol{\varphi}(\mathbf{x},\boldsymbol{\theta})$ containing parameters vector $\boldsymbol{\theta}$ as in the case of approximation based on multi-layer neural network represented in (30), the approximation is called nonlinear for parameter, because $F(\mathbf{x},\boldsymbol{\theta})$ is a nonlinear function of the parameter $\boldsymbol{\theta}$:

$$F(x,\theta)=\theta^T\varphi(x,\theta) \qquad (33)$$

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 [35]. 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:

$$\Delta_F(x,\theta) = F(x,\theta^*) - F(x,\theta) = -\frac{\partial F(x,\theta)}{\partial\theta}\Delta_\theta + \delta(x,\theta,\theta^*) \tag{34}$$

Where $\boldsymbol{\theta}$ 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, $\Delta_\theta = \theta - \theta^*$, $\delta(x,\theta,\theta^*) = -\sigma(|\Delta_\theta|)$ with $\lim_{|\Delta_\theta|\to 0}\frac{\sigma(\Delta_\theta)}{|\Delta_\theta|} = 0$.

Addition $\delta(x,\theta,\theta^*)$ blocked by $|\delta(x,\theta,\theta^*)| \le L|\Delta_\theta|^2$ with L is Lipschitz constant. Therefore, if you find the governing rule θ to reduce $|\Delta_\theta|^2$ then θ tend toward θ^* and $F(x,\theta)$ will forward to $F(x,\theta^*)$. 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 Ω_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 [35], [20], [4], [30].

4. Proposed control mechanisms FNNREM

4.1. The algorithm of FNNREM

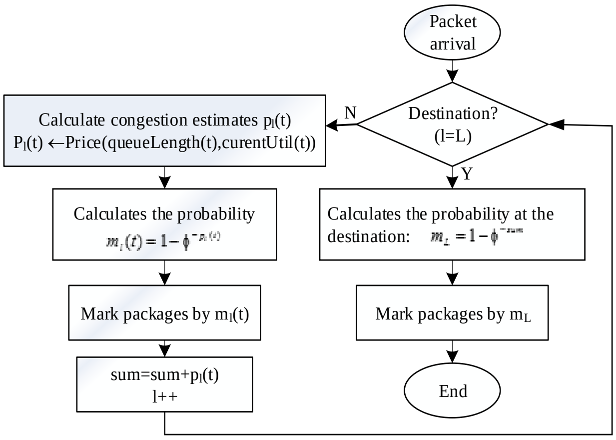

Figure 9. Algorithm control mechanism FNNREM

4.2. Structure FNNREM

FNNREM structure is shown in Figure 10 consists of five layers, which is:

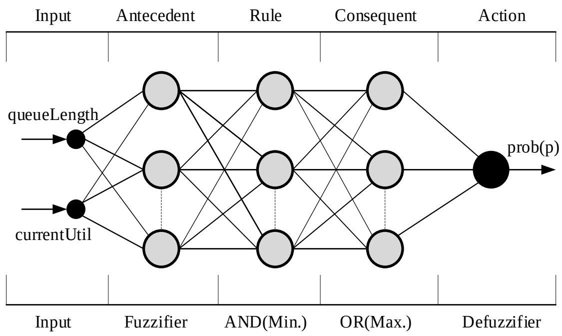

Figure 10. Architecture of FNNREM

(i) Layer of inputs (Input): language input variable is the variable that represents the main factors affecting the operation mechanism of REM. Here we choose to use level factors queue (queueLength), use line level (currentUtil) as input linguistic variables FNNREM system.

(ii) Layer antecedent (Antecedent): This layer performs the function of tissues in the value of the variable input, will pass each of the input values to set the corresponding language. Because chemical methods under fuzzy triangular/trapezoidal simple and efficient, so we chose this method to construct the membership function for the input linguistic variables.

(iii) Layer rules (Rule): This layer contains the basis for inference rules. Code base is a set of fuzzy rules of the form IF-THEN, for n variables in x_1.. x_n, the y variable, fuzzy rules of the form R: IF (x_1 is A_1)∧.. ∧ (x_i is A_i) ∧ … ∧(x_n is A_n) THEN y is B. Where A and B are fuzzy sets of linguistic variables x_1, x_2,.., x_n and the outcome variable y.

(iv) Layer consequences (consequent): This layer performs the function of the total results of the nodes in the rule layer to send, through the permit OR (Max.).

(v) Layer Action (Action) functional class implements defuzzification to obtain results, the results are calculated probability of packet marking optimization under the current state of the network.

4.3. Simulation Settings

We performed simulations based on NS2 software [23]. During the experiment, the network model is described in the model after:

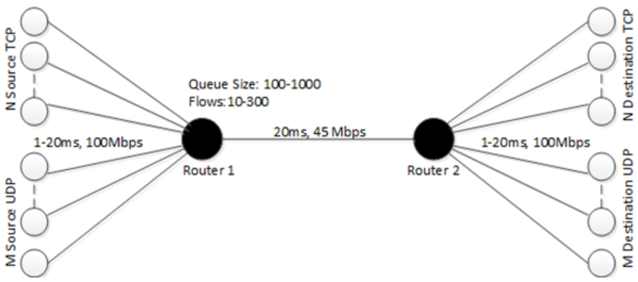

Figure 11. Network simulation model

In simulation, we use N flows TCP and M unresponsive flows UDP responses flows. The transmission lines from source TCP and UDP to bottleneck and from bottleneck to destinations has a 100Mbps bandwidth, latency is changed from 1 to 20ms. Transmission line in the script is the link between two routers. We put the transmission bandwidth is 45Mbps and the latency is 20ms. Router bottlenecks in the algorithm settings REM, FNNREM to evaluate the queue size at a bottleneck circumstances change and flow into the M and N vary from 10 to 300

In addition, parameters such as packet size of all TCP and UDP flows are set to 1000 bytes, TCP window size is 2000 packets, transmission speed of UDP flows changes in the simulation as a evaluation basis. Selected simulation time is 60 seconds.

4.4. The evaluation criteria

The performance evaluation of congestion control mechanisms is usually through criteria such as packet loss probability at place where congestion occurs, achieved network throughput, transmission line utilization level, the level of fairness of transmission line when the together connection to the transmission bottleneck and the queue utilization level at bottleneck. The criteria for this evaluation represents quality network service (QoS) and is defined by [3][24]:

Packet loss rate: The ratio of the number of packets lost to the total development package. For network stability, the lower the rate, whereas this ratio is very high. Packet loss rate is determined by the formula:

$$packet\ loss\ rate = \frac{\sum_{i=1}^{N} packet\ loss}{\sum_{j=1}^{M} packet\ sent} \quad (35)$$

Transmission line utilization level: As the ability to take advantage of network traffic that said the index's ability to communicate through the network connection is strong or weak and is calculated by the following formula:

$$utilization = \frac{byte\_departures_t}{bandwidth \times t} \quad (36)$$

Where utilization is the level of using transmission lines, byte_departurest is the number of bytes transmitted in t seconds, the bandwidth is the bandwidth of the transmission line and t is time of transmission.

Fairness level: is level of flows in network with ensuring fairness of connections when network has many other throughput types. Level of fairness is 1 when throughput of flows is equal, unless when throughput of flows is unequal, this value is less than 1. This value demonstrates greater, assurance of the congestion control algorithms is well. Fairness level is calculated as following formula:

$$fairness = \frac{\left(\sum_{i=1}^{N} x_i\right)^2}{N \times \sum_{i=1}^{N} x_i^2} \quad (37)$$

where, fairness is fair level of flows, fairness ∊ [0, 1], x_i: is the throughput of flow i and N is the number of flows.

Average Queue Size: The index indicates directly the level of resource use at router. This index is defined as the ratio of the average queue size to the actual size of the queue. Mechanism with this small ratio will have small latency at the queue and risk of overflew queue is low. In contrast, the mechanism will make large latency and high risk of overflew queue. We use the quadratic average of control deviation to be index of queue utilization level and it is defined as:

$$S_e = \sqrt{\frac{1}{M+1}\sum_{i=0}^{M} e_i^2} = \sqrt{\frac{1}{M+1}\sum_{i=0}^{M}\left(Q_i - Q_{ref}\right)^2} \quad (38)$$

where, Q_{ref} the queue size, Q_i the queue size at the ith sampling time and M is the number of sampling.

4.5. Evaluation of packet loss rate

From the graph Figure 12, we see that the queue size in the router increases, the packet loss rate of mechanisms reduce and the increased number of connections to the router is the increased packet loss rate. In all cases, REM always has the highest packet loss rate and the FNNREM always have the lowest packet loss, when the queue size of 400 or more and the number of connections is less than 100, the packet loss rate of FNNREM less than 1%.

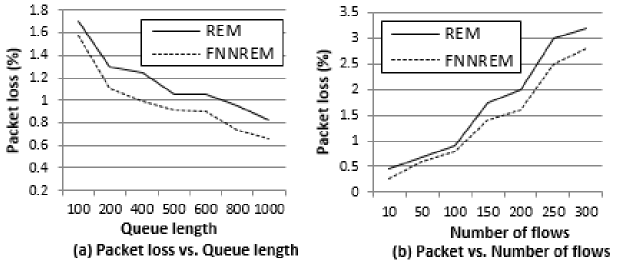

Figure 12. Packet loss rate of the active queue management mechanisms

4.6. Evaluation of Link Utilization level

The graph in Figure 13 shows the level of transmission line utilization of the mechanisms. The ability to take advantage of the transmission line utilization of the mechanisms increases, when the queue size and loading (number of connection flows) increases. When the queue size from 400 and over or the number of connections into router from 50 and over, mechanism FNNREM uses better of transmission line, transmission rate used is over 80%, and is always higher than the mechanism REM.

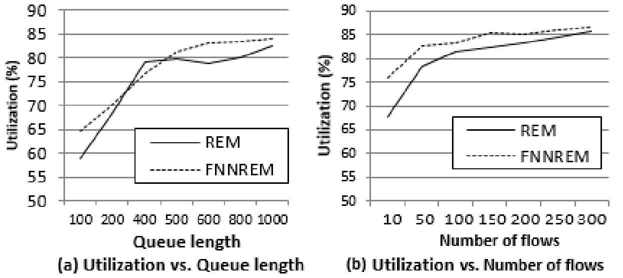

Figure 13. Utilization of the transmission of active queue management mechanism

4.7. Evaluation of Fairness

Based on the graph of Figure 14 shows the fairness of the algorithm, we found that the fairness level of the algorithm by REM and FNNREM is very high at over 75% for all cases. Particularly, mechanism FNNREM always balance over 80% in the cases which there are the changed number of connection flows

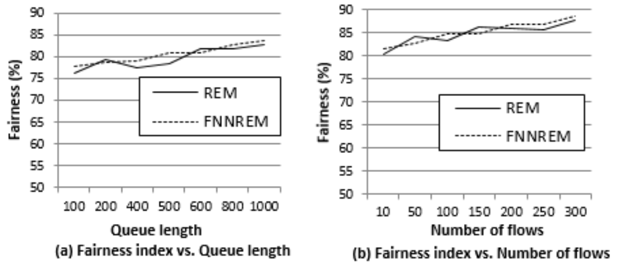

Figure 14. The fairness of active queue management mechanisms

4.8. Evaluate the average queue size

Based on the simulation results and graph demonstrating usage rate of the queue size of algorithm in Figure 15, we find that FNNREM usage level is always lower than REM, in cases of the changing queue size, this figure is less than 60%, and less than 80% for all cases having changed flows. This matter makes the latency and the ability to overflow queue at routers of low mechanism FNNREM

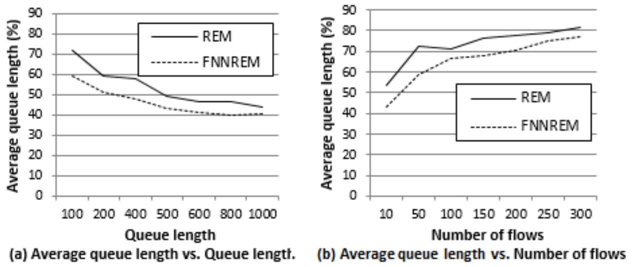

Figure 15. Utilization of the queue of active queue management mechanism

5. Conclusion

The congestion control mechanism in the active queue management in routers is essential. However, these factors need to be taken computational intelligence, fuzzy control mechanism in the active queue management, to these mechanisms operate more efficiently, in order to improve the quality of service and network performance. In this paper, we have changed the mechanism of REM queue management by introducing fuzzy neural network controller involved in the process of calculating the probability of packet marking based on the level of use and queue level used in the routing path. Results theoretical analysis and simulation experiments on NS2 software mechanisms for REM and REM traditional fuzzy controller (FNNREM) in the same network model, showed FNNREM packet loss rates low, utilization levels and high transmission latency at small router queue using low. So FNNREM control and congestion control better than REM.

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

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

[32] 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

[33] 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

[34] 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

[35] Yeffrey 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.