Adaptive Fuzzy Active Queue Management

M. Jalili-Kharaajoo, F. Habibipour Roudsari,
A. Dehestani, H. Hashemi Fesharaki
Young Researchers Club
Islamic Azad University
P.O. Box: 14395/1355, Tehran, Iran
mahdijalili@ece.ut.ac.ir

Abstract

Recently many Active queue management (AQM) algorithms have been proposed to address performance degradations of end-to-end congestion control. However, these AQM algorithms show weaknesses to detect and control congestion under dynamically changing network situations. In this paper, an adaptive fuzzy AQM is designed to congestion avoidance in TCP/AQM networks. This kind of control action has robust performance, which is suitable for time varying and complex systems such as computer and communication networks. A candidate Lyapunov function is employed in the adaptive law synthesis to ensure convergence.

Keywords: Active queue management, adaptive control, fuzzy control, traffic management, congestion control.

1 Introduction

TCP congestion control mechanism, while necessary and powerful, are not sufficient to provide good service in all circumstances, specially with the rapid growth in size and the strong requirements to Quality of Service (QoS) support, because there is a limitation to how much control can be accomplished at end system. It is needed to implement some measures in the intermediate nodes to complement the end system congestion avoidance mechanisms. Active Queue Management (AQM), as one class of packet dropping/marking mechanism in the router queue, has been recently proposed to support the end-to-end congestion control in the Internet [1-5]. It has been a very active research area in the Internet community. The goals of AQM are (1) reduce the average length of queue in routers and thereby decrease the end-to-end delay experimented by packets, and (2) ensure the network resources to be used efficiently by reducing the packet loss that occurs when queues overflow. AQM highlights the tradeoff between delay and throughput. By keeping the average queue size small, AQM will have the ability to provide greater capacity to accommodate nature-occurring burst without dropping packets, at the same time, reduce the delays seen by flow, this is very particularly important for real-time interactive applications. RED [6,7] was originally proposed to achieve fairness among sources with different burst attributes and to control queue length, which just meets the requirements of AQM. However, many subsequent studies verified that RED is unstable and too sensitive to parameter configuration, and tuning of RED has been proved to be a difficult job [8-10].

Fuzzy logic controllers have been developed and applied to nonlinear system for the last two decades [11]. The most attractive feature of fuzzy logic control is that the expert knowledge can be easily incorporated into the control laws [12-14]. In [15,16] an adaptive fuzzy controllers has been proposed for robust control performance, which will be used in our controller design.

The intuition and heuristic design is not always scientific and reasonable under any conditions. Of course, since Internet is a rather complex huge system, it is very difficult to have a full-scale and systematic comprehension, but importance has been considerably noted. The mathematical modeling of the Internet is the first step to have an in-depth understanding, and the algorithms designed based on the rational model should be more reliable than one original from intuition. In some of the references, the nonlinear dynamic model for TCP flow control has been utilized and some controllers like PI and Adaptive Virtual Queue Algorithm have been designed for that [17-21].

Although PI controller successfully related some limitations of RED, for instance, the queue length and dropping/marking probability are decoupled, whenever the queue length can be easily controlled to the desired value; the system has relatively high stability margin. The shortcomings of PI controller are also obvious. The modification of probability excessively depends on buffer size. As a result, for small buffer the system exhibits sluggishness. Secondly, for small reference queue length, the system tends to performance poorly, which is unfavorable to achieve the goal of AQM because small queue length implies small queue waiting delay. Thirdly, the status of actual network is rapidly changeable, so we believe that it is problematic and unrealistic, at least inaccurate, to take the network as a linear and constant system just like the designing of PI controller. Affirmatively, the algorithm based on this assumption should have limited validity, such as inability against disturbance or noise. We need more robust controller to adapt complex and mutable network environment, which will be our motivation and aim in this study.

In the paper, we will apply an adaptive fuzzy controller to design the AQM system for congestion avoidance. First, a fuzzy logic based controller is designed and them the adaptive law will be applied to the designed controller. A candidate Lyapunov function is employed in the adaptive law synthesis to ensure convergence. The performance of the proposed fuzzy adaptive controller is compared with that of classic PI controller. The simulation results show the superior performance of the proposed controller in comparison with classic PI controller.

The rest of the paper is organized as follows: Section 2 presents the nonlinear dynamic model for TCP flow control and the state space description of this model. In Section 3, the basic principles of adaptive fuzzy controller are presented. Some simulations are provided using MATLAB package in Section 4 and the performance of the various controllers are compared. Finally, the paper is concluded in section 5.

2 TCP flow control model

In [17], a nonlinear dynamic model for TCP flow control has been developed based on fluid-flow theory. This model can be stated as follows

Top

$$\begin{cases} \dfrac{dW(t)}{dt} = \dfrac{1}{R(t)} - \dfrac{W(t)W(t-R(t))}{2R(t)} p(t-R(t)) \\ \dfrac{dq(t)}{dt} = \dfrac{N(t)}{R(t)} W(t) - C(t) \end{cases} \tag{1}$$

The above nonlinear and time-varying system was approximated as a linear constant system by small-signal linearization about an operating point [20,21] (Fig. 1). In the block diagram, *C(s)* and *G(s)* are the controller and the plant, respectively. The meaning of parameters presented in Fig. 1 are as following

$$K(t) = \frac{[R(t)C(t)]^3}{[2N(t)]^2}, \quad T_1(t) = R(t), \quad T_2(t) = \frac{R^2(t)C(t)}{2N(t)} \tag{2}$$

where

C(t) : Link capacity (packets/sec)
q_o : Queue reference value
N(t) : Load factor, i.e., number of active sessions
R(t) : Round-trip time (RTT), $R(t) = 2(q(t)/C(t) + T_p)$, T_p is the fixed propagation delay
p(t) : Dropping/marking probability
q(t) : Instantaneous queue

We believe that the AQM controller designed with the simplified and inaccurate linear constant model should not be optimal, because the actual network is very changeful; the state parameters are hardly kept at a constant value for a long time [2,5]. Moreover, the equations (1) only take consideration into the fast retransmission and fast recovery, but ignore the timeout mechanism caused by lacking of enough duplicated ACK, which is very usual in burst and short-lived services. In addition to, there are many non-respective UDP flows besides TCP connections in networks; they are also not included in equations (1). These mismatches in model will have negative impact on the performance of controller designed with the approach depending with the accurate model. For the changeable network, the robust control should be an appropriate choice to design controller for AQM.

To describe the system in state space form, suppose that $x_1 = e$, $x_2 = de/dt$, so the plant depicted in Fig. 1 is described by a second order system as

$$\begin{cases} \dfrac{dx_1}{dt} = x_2 \\ \dfrac{dx_2}{dt} = -a_1(t)x_1 - a_2(t)x_2 - b(t) + F(t) \end{cases} \tag{3}$$

$$a_{1\min} \le a_1 \le a_{1\max}, a_{2\min} \le a_2 \le a_{2\max}, 0 < b_{\min} \le b \le b_{\max} \tag{4}$$

where

$$\begin{aligned} & a_1(t) = \frac{1}{T_1(t)T_2(t)}, a_2(t) = \frac{T_1(t)+T_2(t)}{T_1(t)T_2(t)}, b(t) = \frac{K(t)}{T_1(t)T_2(t)} \\ & F(t) = \frac{d^2}{dt^2} q_o + \frac{T_1(t)+T_2(t)}{T_1(t)T_2(t)} \frac{d}{dt} q_o + \frac{1}{T_1(t)T_2(t)} q_o \end{aligned} \tag{5}$$

F(t) is regarded as the system disturbance.

Top

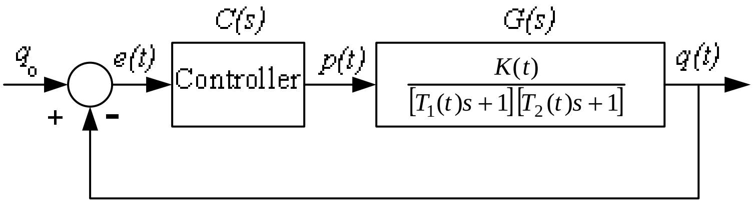


Figure 1: Block diagram of AQM control system

3 Fuzzy adaptive controller design

3.1 Fuzzy control design

Fuzzy logic control (FLC) has been demonstrated to solve some practical problems that have been beyond the reach of conventional control techniques. Fuzzy logic control is a knowledge-based control that uses fuzzy set theory, fuzzy reasoning and fuzzy logic for knowledge representation and inference [11,12,16]. The apparent success of FLC can be attributed to its ability to incorporate expert information and generate control surfaces whose shape can be individually manipulated for different regions of the state space with virtually no effects on neighboring regions.

In this paper a fuzzy system consisting of a fuzzifier, a knowledge base (rule base), a fuzzy inference engine and defuzzier will be considered. The knowledge base of the fuzzy system is a collection of fuzzy IF-THEN rules. Fuzzy logic control is ideal for the AQM problem, since there is no complete mathematical model. However, human experience and experimental results, can be used in the control system, design.

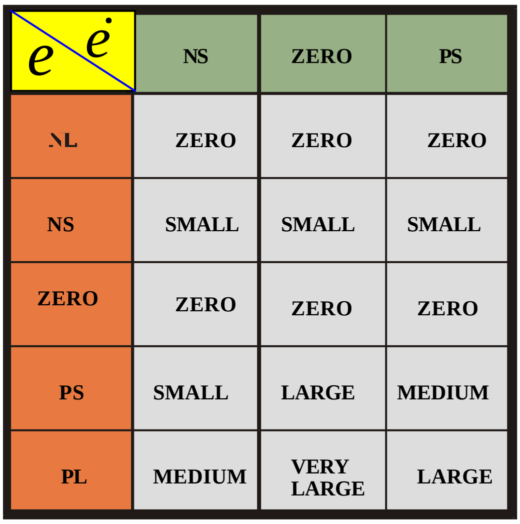
The controller has two inputs, the error (e) and its derivative $(\dot{e})$ and the control input (p). Five triangular membership functions are defined for error (Fig. 2), namely, Negative Large (NL), Negative Small (NS), Zero, Positive Small (PS), and Positive Large (PL). Similarly three triangular membership functions are defined for derivative of the error (Fig. 3) and there are as follows, Negative Small (NS), Zero, and Positive Small (PS). Also five triangular membership functions are defined for the control input (Fig. 4) and there are Zero, Small, Medium, Large and Very Large. The complete fuzzy rules are shown in Table 1: The first rule is outlined below,

Rule 1:
If (e) is **PL** *AND* $(\dot{e})$ is **Zero** *THEN* (p) is **Large**.

The rest of the rules are derived similarly. The label names used here give an intuitive sense of how the rules apply. Through experimentation and tuning of the membership functions it was determined that the number of rules was sufficient to encompass all realistic combinations of inputs and outputs. This fuzzy logic controller is implemented using product inference and a center-average defuzzifier.

Table 1: Fuzzy Rules

Top



e \ $\dot{e}$	NS	ZERO	PS
NL	ZERO	ZERO	ZERO
NS	SMALL	SMALL	SMALL
ZERO	ZERO	ZERO	ZERO
PS	SMALL	LARGE	MEDIUM
PL	MEDIUM	VERY LARGE	LARGE

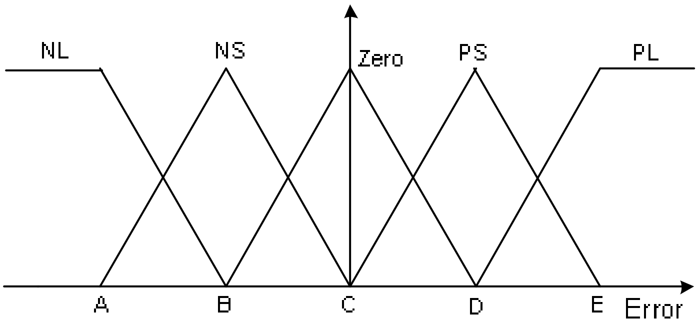


Figure 2: Error membership function

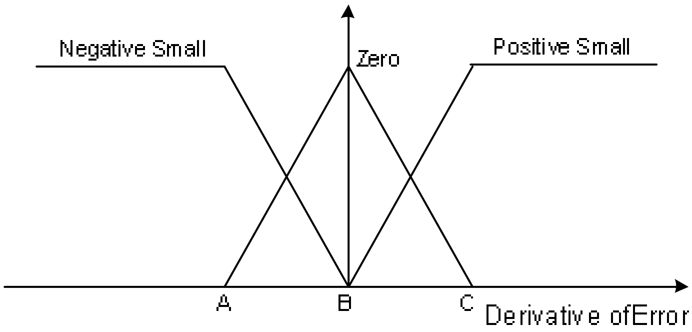
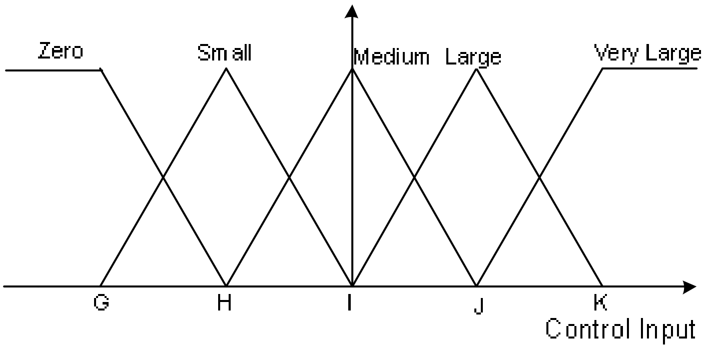


Figure 3: Membership function for the derivative of Error



Top

Figure 4: Control input membership function

3.2 Adaptive Fuzzy Control

Assume that the rule base consists of multiple-input single-output (MISO) rules of the form

$$\mathrm{R}^{(j)}:\mathrm{IF}\ x_1\ \text{is}\ A_1^j\ \text{and}\ldots\text{and}\ x_n\ \text{is}\ A_n^j\ \mathrm{THEN}\ y_j\ \text{is}\ C^j \tag{6}$$

where $\underline{x}=(x_1\ldots x_n)\in N$, y denotes the linguistic variables associated with inputs and outputs of the fuzzy system. A_i^j and C^j are linguistic values of linguistic variables x and y in the universes of discourse N and S respectively, $j=1,2..Q_R$ (number of rules). A fuzzy system consisting of a singleton fuzzifier, product inference, center-average defuzzifier and triangular membership functions can be written as [14]

$$f(x)=\frac{\sum_{j=1}^{Q_R}\bar{y}^j\left(\prod_{i=1}^{n}m_{A_i^j}(x_i)\right)}{\sum_{j=1}^{Q_R}\left(\prod m_{A_i^j}(x_i)\right)} \tag{7}$$

where $f:N\subset\Re^n\rightarrow\Re,\underline{x=(x_1\ldots x_n)^T}\in N$ and $m_{A_i^j}(x_i)$ is a triangular membership function and $\bar{y}^j$ is the point in S were m_{C^j} is maximum or equal to 1.If the $m_{A_i^j}(x_i)$'s and $\bar{y}^j$'s are free (adjustable) parameters, then (7) can be written as

$$f(\underline{x})=\underline{J}^T\,\underline{\Psi}(\underline{x}) \tag{8}$$

where $\underline{J}=(\bar{y}^1,\ldots\bar{y}^{Q_R})$ is a parameter vector and $\underline{\Psi}(\underline{x})=(y^1(x),\ldots y^{Q_R}(x))^T$ is a regression vector with the regressor given by

$$y_i(x)=\frac{\prod_i^n m_{A_i^j}(x_i)}{\sum_{j=1}^{Q_R}\left(\prod m_{A_i^j}(x_i)\right)} \tag{9}$$

Equation (8) is referred to as adaptive fuzzy systems [14-16].There are two main reasons for using adaptive fuzzy systems as building blocks for adaptive fuzzy controllers. Firstly, it has been proved in [14] that they are universal function approximators. Secondly, all the parameters in $\Psi(x)$ can be fixed at the beginning of adaptive fuzzy systems expansion design procedure, so that the only free design parameters are $\underline{J}$. In this case $f(x)$ is linear in the parameters. This approach will be adopted in synthesizing the adaptive control law in this paper. The advantage of this approach is that very simple linear parameter estimation methods can be used to analyze and synthesize the performance and robustness of adaptive fuzzy systems. If no linguistic rules are available, the adaptive fuzzy system reduces to a standard nonlinear adaptive controller.

3.3 Adaptive Law Synthesis

The mathematical model given by equation (3) can be expressed as

$$\dot{\mathrm{z}}=\mathrm{Az}+\mathrm{Bu}+\mathrm{E(z)} \tag{10}$$

Top

where A is Hurwitz. Therefore there exists a unique positive definite matrix P that satisfies the Lyapunov equation.

$$A^T P + PA = -Q \tag{11}$$

If the control input, u, is expressed as an adaptive fuzzy system then (10) becomes,

$$\dot{z} = Az + BJ^T y(z) + E(z) \tag{12}$$

Let [15,16],

$$\dot{\hat{z}} = A\hat{z} + BJ^{*T} y(\hat{z}) \tag{13}$$

be the ideal systm model with no uncertainty (identification model) with $e = z - \hat{z}$, where J^* denotes the optimal J defined as,

$$J^* \equiv \arg\min_{|u| \le M}\left[\sup_{z \in \Omega}\left|u(z|J^*) - u(z|J)\right|\right] \tag{14}$$

Therefore,

$$\dot{e} = Ae + Bf^T y(e) + \hat{E} \tag{15}$$

where $f = J - J^*$. To derive a control law that ensures that $e \to 0$ as $t \to \infty$ a candidate Lyapunov function is defined as [14,16];

$$V = \frac{1}{2}\left(e^T P e + \frac{f^T f}{g\|\hat{E}\|}\right) \tag{16}$$

where $g > 0$ is a design parameter. The time derivative of V is

$$\dot{V} = -e^T Q e + e^T PB(\hat{E} + f^T y(e)) + \frac{f^T \dot{f}}{g\|\hat{E}\|} \tag{17}$$

Rearranging equation (17) yields

$$\dot{V} = -e^T Q e + e^T PB\hat{E} + f^T\left(g\|\hat{E}\|\|e^T PB\| y(e) + \dot{f}\right) \tag{18}$$

Now choosing the adaptive law (recalling that $\dot{f} = \dot{J}$)

$$\dot{J} = -g\|\hat{E}\|\|e^T PB\| y(e) \tag{19}$$

The equation (18) reduces to

$$\dot{V} = -e^T Q e + e^T PB\hat{E} \tag{20}$$

Top

The equation (21) can be recast using vector norms;

$$\dot{\mathrm{V}} = -l_{\min}(Q)\|e\|^2 + \|e^{\mathrm{T}}\mathrm{PB}\|\|\hat{\mathrm{E}}\| \tag{21}$$

Let $\|\hat{\mathrm{E}}\|$ be selected such that

$$\|\hat{\mathrm{E}}\| \geq \frac{l_{\min}(Q)\|e\|^2 - a\|e\|}{\|e\mathrm{PB}\|} \tag{22}$$

where $a > 0$, substituting for $\hat{\mathrm{E}}$ in equation (22) gives

$$\dot{\mathrm{V}} \leq -a\|e\| \tag{23}$$

Therefore the control law of equation (19) will ensure that the state e converges.

4 Simulation results

The network topology used for simulation, is depicted in Fig. 5 [2,5]. The only bottleneck link lies between node A and node B the buffer size of node A is 200 packets, and default size of the packet is 350 bytes. All sources are classed into three groups. The first one includes N_1 greedy sustained FTP application sources, the second one is composed of N_2 burst HTTP connections, each connection has 10 sessions, and the number of pages per session is 3. The thirds one has N_3 UDP sources, which follow the exponential service model, the idle and burst times are 10000msec and 1000msec, respectively, and the sending rate during "on" duration is 40kbps. We introduced short-lived HTTP flows and non-responsive UDP services into the router in order to generate a more realistic scenario, because it is very important for a perfect AQM scheme to achieve full bandwidth utilization in the presence of noise and disturbance introduced by these flows. The links between node A and all sources have the same capacity and propagation delay pair (L_1, t_1). The pair (L_2, t_2) and (L_3, t_3) define the parameter of links AB and BC, respectively.

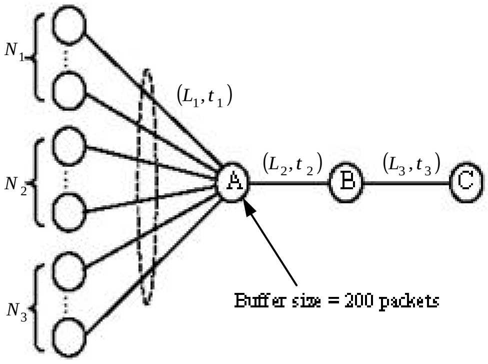


Figure 5: The simulation network topology

In the first study, we will use the most general network configuration to testify whether the proposed Adaptive Fuzzy Logic Controller (AFLC) can reach the goals of AQM, and freely control the queue length to stabilize at the arbitrary expected value. Therefore, given that

Top

$(L_1,t_1)=(10Mbps,15ms)$, $(L_2,t_2)=(15Mbps,15ms)$, $(L_3,t_3)=(45Mbps,15ms)$. $N_1=270$, $N_2=N_3=0$. Let the expected queue length equal to 75 packets. To implement the control law, the fuzzy rule Table 1 is used and the insight gained from the non-adaptive fuzzy logic control is used to select the $\underline{J}$ values to lie within the interval [1.0,2.0]. The remaining control parameters are set as: $\mathrm{Q}=\mathrm{diag}(3,3)$, $\hat{\mathrm{E}}=120$, $g=0.00025$. $\Psi(e)$ is formulated using the IF part of fuzzy rule Table 1.

The instantaneous queue length using the proposed AFLC is depicted in Fig. 6. After a very short regulating process, the queue settles down its stable operating point. RED algorithm is unable to accurately control the queue length to the desired value [7,9]. The queue length varies with network loads. The load is heavier the queue length is longer. Attempting to control queue length through decreasing the interval between high and law thresholds, then it is likely to lead queue oscillation.

To investigate the performance of the proposed AFLC, we will consider a classic PI controller as

$$p(k)=(a-b)(q(k)-q_o)+b(q(k)-q(k-1))+p(k-1) \tag{24}$$

The coefficients a and b are fixed at 1.822e^{-5} and 1.816e^{-5}, respectively, the sampling frequency is 500Hz, the control variable p is accumulative [5]. Because the parameter b is very small, and the sample interval is very short, the negative contribution to p made by the second item in the right can be omitted in initial process, then the positive contribution mainly come from the first item. The queue evaluation using PI controller is shown in Fig. 7. Although PI controller could regulate the queue to the fixed point, the integrated performance needs to be improved, such as the transient process is too long and the fluctuation in steady state is great, for small queue length, which lows the link utilization.

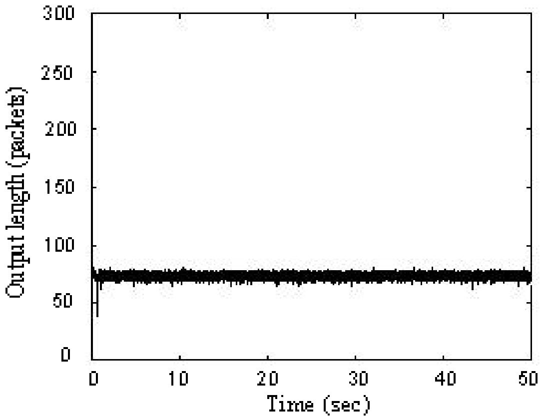


Figure 6: Queue evaluation (AFLC)

Top

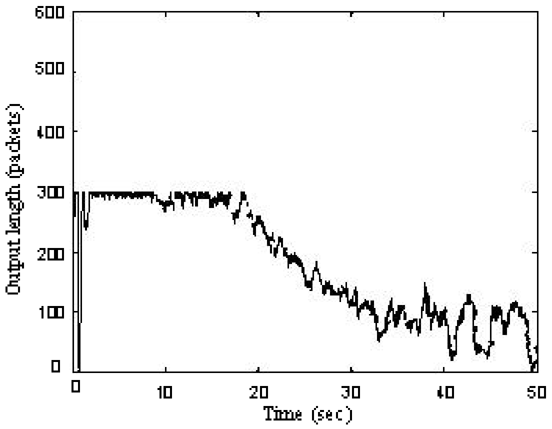


Figure 7: Queue evaluation (PI)

In this section, Firstly, let $N_1 = 270, N_2 = 400, N_3 = 0$, the evaluation of queue size is shown in Fig. 8. As it can be seen, the proposed AFLC has better performance than that of PI one. Next, given that $N_1 = 270, N_2 = 0, N_3 = 50$, we further investigate performance against the disturbance caused by the non-responsive UDP flows. Fig. 9 shows the results, obviously, PI is very sensitive to this disturbance, while AFLC operates in a relatively stable state. The queue fluctuation increases with introducing the UDP flows, but the variance is too much smaller comparing with PI controller.

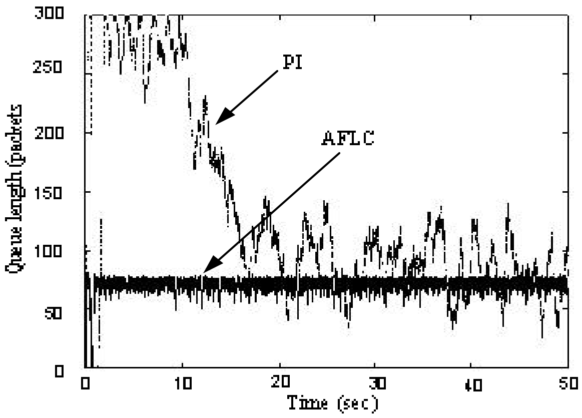


Figure 8: Queue evaluation (FTP+HTTP)

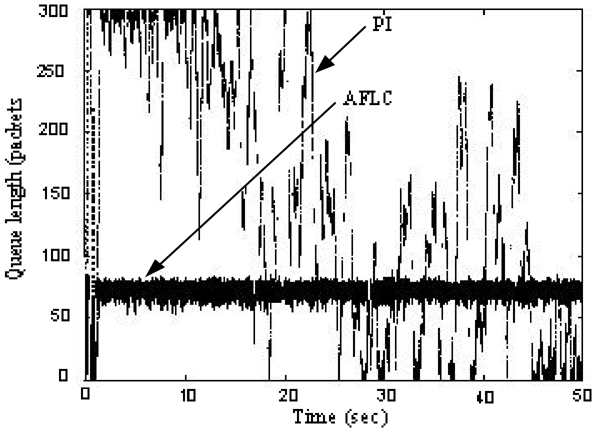


Figure 9: Queue evaluation (FTP+UDP)

Top

Finally, we evaluate the integrated performance of AFLC using one relatively real scenario, i.e., the number of active flows is changeable, which has 270 FTP flows, 400 HTTP connections and 30 UDP flows. Figs. 10 and 11 show the evaluation of queue controlled by AFLC and PI controllers, respectively. It is clear that the integrated performance of AFLC controller, namely transient and steady state responses is superior to that of PI controller. The AFLC controller is always keeping the queue length at the reference value, even if the network loads abruptly change, but PI controller has the inferior adaptability. In other words, the former is more powerful, robust and adaptive than the later one, which is in the favor of achievement to the objectives of the AQM policy.

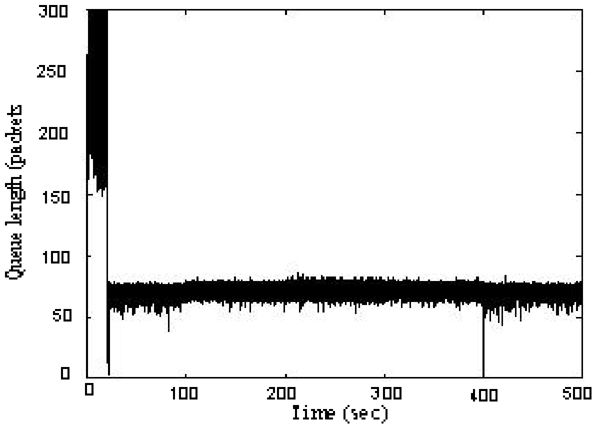


Figure 10: Queue evaluation (AFLC)

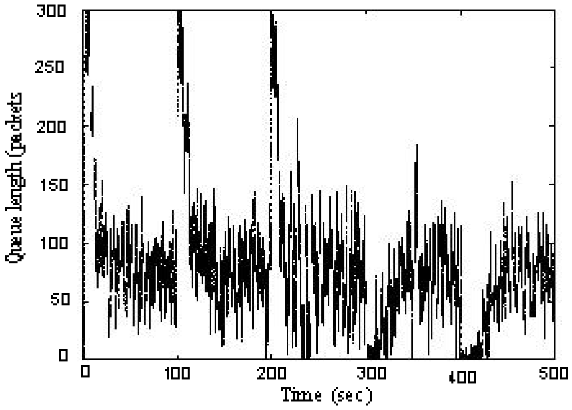


Figure 11: Queue evaluation (PI)

5 Conclusions

In this paper, an adaptive fuzzy logic based controller was applied to TCP/AQM networks for the objective of queue management and congestion avoidance. For this purpose, a linearized model of the TCP flow was considered. A candidate Lyapunov function is employed in the adaptive law synthesis to ensure convergence. We took a complete comparison between performance of the proposed AFLC and classical PI controller under various scenarios. The conclusion was that the integrated performance of AFLC was superior to that of PI one.

References

Top

[1] Barden, B. et al., Recommendation on queue management and congestion avoidance in the internet, REC2309, April 1998.

[2] Jalili-Kharaajoo, M., Application of robust fuzzy adaptive second-order sliding-mode control to active queue management, Lecture Notes in Computer Science, 2957, pp.109-119, 2004.

[3] S. Ryu, C. Rump and C. Qiao, A Predictive and Robust Active Queue Management for Internet Congestion Control, in Proc. Eighth IEEE International Symposium on Computers and Communication (ISCC'03), Antalya, Turkey, 2003.

[4] S. Ryu, C. Rump, and C. Qiao. Advances in Internet congestion control. IEEE Communication Survey and Tutorial, 2002.

[5] R. Fengyuan, L. Chuang, Y. Xunhe, S. Xiuming, and W. Fubao, A Robust Active Queue Management algorithm based on Sliding Mode Variable Structure Control. in Proc. INFOCOM'2002, 21, pp.13–20, 2002.

[6] Floyd, S. and Jacobson, V., Random early detection gateway for congestion avoidance, IEEE/ACM Trans. Networking, August 1993.

[7] C. V. Hollot, V. Misra, D. Towsley, and W. Gong. A control theoretic analysis of RED. In Proc. of INFOCOM'2001, pages 1510–1519, April 2001.

[8] Firoiu, V. and Borden, M., A study of active queue management for congestion control, in Proc. INFOCOM, March 2000.

[9] May, M., Bonald, T. and Bolot, T., Analytic evaluation of RED performance, in Proc. INFOCOM, March 2000.

[10] S. Floyd and V. Paxson. Difficulties in simulating the Internet, IEEE/ACM Transactions on Networking, 9(4), pp.392–403, 2001.

[11] Zadeh, L.A., Fuzzy sets, Inf. Control (1965), 338-353.

[12] Jalili-Kharaajoo, M., and Ebrahimirad, H., Improvement of second order sliding mode control applied to position control of induction motors using fuzzy logic, Lecture Notes in Artificial Intelligence, 2715, 2003.

[13] Trebi-Ollennu, A., Robust output tracking for MIMO nonlinear systems: An adaptive fuzzy systems approach. IEE Proceedings, Control Theory and Applications 144(6): 537-544, 1997.

[14] Li-Xin, W., Adaptive Fuzzy Systems and Control Design and Stability Analysis, Prentice Hall, 1994.

[15] Trebi-Ollennu, A. and J. M. Dolan, Adaptive Fuzzy Throttle Control for an All Terrain Vehicle, 2000.

Top

[16] Trebi-Ollennu, A., Robust nonlinear control designs using adaptive fuzzy systems, Ph.D. Thesis. Department of Aerospace & Guidance Systems, Royal Military College of Science, Shrivenham, Cranfield University, 1996.

[17] Misra, V., Gong, W.B. and Towsley, D., Fluid-based analysis of network of AQM routers supporting TCP flows with an application to RED, in Proc. ACM/SIGCOMM, 2000.

[18] Hollot, C., Misra, V., Towsley, D. and Gong, W.B., On designing improved controllers for AQM routers supporting TCP flows, in Proc. INFOCOM, 2001.

[19] Misra, V., Gong, W.B. and Towsley, D., Analysis and design an adaptive virtual queue (AVQ) algorithm for active queue management, in Proc. ACM/SIGCOMM, 2001.

[20] Kelly, F.P., Maulloo, A. and Tan, D., Rate control in communication networks, Journal of the Operation research Society, 49, pp.237-252, 1998.

[21] Athuraliya, S., Lapsley, D.E. and Low, S.H., Random early marking for internet congestion control, in Proc. Globecom, 1999.

Top