Applying Fuzzy Sliding Mode Control Based on Genetic Algorithms to Congestion Avoidance in Computer Network

Mehdi Galily, Farzad Habibipour Roudsari and Abdolmajid Riazi

Iran Telecommunication Research Center, Ministry of ICT, Tehran, Iran

roudsari@itrc.ac.ir

Abstract

In this paper, the design of Fuzzy Sliding Mode Control (FSMC) based on Genetic Algorithms (GAs) is studied. Using the proposed control strategy, a robust controller is designed for Active Queue Management (AQM) in computer networks. AQM takes a trade-off between link utilization and delay experienced by data packets. From control point of view, it is rational to regard AQM as a typical regulation system. We design a fuzzy sliding mode controller for the AQM system as an optimization problem and apply the optimal searching algorithms and GAs to find the optimal rules and membership functions of the FSMC. Some computer simulations are provided to show the effectiveness of the proposed control action. Simulation results show the FSMC based on GAs has better performance than that of SMC and FSMC in providing efficient queue management.

Keyword: Fuzzy control; sliding mode control, congestion control, computer networks

I. 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 limit 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 [1,2].

It should also be mentioned that, even for the present Internet architecture, network congestion control remains a critical and high priority issue, and is unlikely to disappear in the near future. Furthermore, if we consider the current utilization trends, congestion in the Internet maybe come unmanageable unless effective, robust, and efficient methods for congestion control are developed. For example, the existing congestion control solutions for TCP transported traffic [3] are increasingly becoming ineffective, and it is generally accepted that these solutions cannot easily scale up even with various proposed ‘‘fixes’’ [4,5], new approaches [6], and architectures [7]. The congestion control schemes employed by the TCP/IP protocol have been widely studied. The Internet protocol architecture is based on a connectionless, best-effort, end-to-end packet service using the IP protocol. TCP is an end-to-end transport protocol that provides reliable, in-order service over the IP packet service. Ever increasing demands on the Internet have led to a number of incremental changes over the last 10 years designed to improve TCP/IP performance:

- Improved round-trip time measurement algorithm (Karn's algorithm) [8];
- Slow-start and congestion avoidance [3];
- Fast retransmit, fast recovery algorithms [9];
- Improved operation over high speed, large delay networks [4];
- Improved congestion indication [5].

As demand for multimedia (streaming) applications increases, it becomes increasing important to ensure that these applications can co-exist with current TCP applications. It is becoming widely accepted that streaming media should be subjected to similar rate controls, as TCP traffic, and recently a number of researchers advocate that they should also exhibit TCP-friendly behavior [10,11]. This makes the control of the congestion in the Internet considerably more difficult. Furthermore, the congestion control problem in the Internet is exacerbated, as the Internet is increasingly transformed into a multi-services high-speed network. Asynchronous Transfer Mode (ATM) also witnessed a similar approach, with the performance of various congestion control schemes proposed for solution of Available Bit Rate (ABR) problem not proven analytically.
several well-known difficulties must be overcome in the Fuzzy Logic Control (FLC) design as follows: (1) Converting the experts' knowledge how into if–then rules is difficult and the results are often incomplete and unnecessary, since operators and control engineers are not capable of specific details or cannot express all their knowledge including intuition and inspiration. (2) Characteristics of fuzzy control systems cannot be pre-specified. (3) It is hard to search optimal parameters of controller to achieve maximum performance. To overcome (1) and (2), the fuzzy sliding mode control (FSMC) [12,13] schemes are proposed. Essentially, the active queue management using the FLC can be considered as an optimization problem for multi-parameters to ease difficulty (3) [14-16]. For improving performance of the sliding mode control (SMC), based fuzzy controller, we adopt optimal searching algorithms, that is, the genetic algorithms (GAs). The GAs have been demonstrated to be a powerful tool for automating the definition of the fuzzy control rule base and membership functions, because that adaptive control, learning, and self-organization can be considered in a lot of cases as optimization or searching processes. The advantages have extended the GA in development of a wide range of approaches for designing fuzzy sliding mode controllers over the last few years. Using the proposed control strategy, the AQM problem in a sample network will be studied. Simulation results show improvement of the performance using GAs.

II. TCP Flow Control Model

In [17,18], a nonlinear dynamic model for TCP flow control has been developed based on fluid-flow theory. This model can be stated as follows

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

The above nonlinear and time-varying system was approximated as a linear constant system by small-signal linearization about an operating point [20] (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

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

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. 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. The variable structure sliding mode control action is one of the best that can help us.

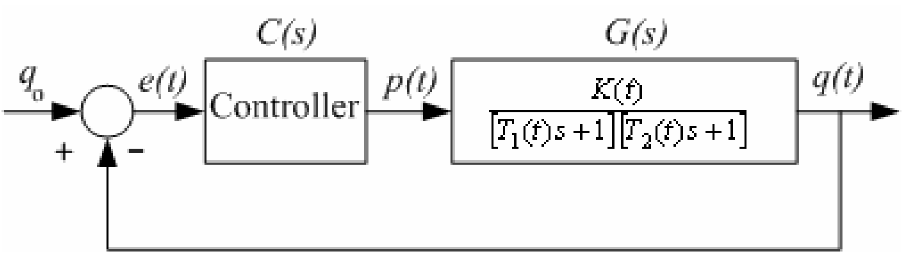

Fig. 1. Block diagram of AQM control system

III. Fuzzy Sliding Mode Control Based on Genetic Algorithms

A Sliding Mode Controller is a Variable Structure Controller (VSC). Basically, a VSC includes several different continuous functions that map plant state to a control surface, and the switching among different functions is determined by plant state that is represented by a switching function [19,20].

Consider the design of a sliding mode controller for the following system

$$\dot{z}(t)=A(z(t)-z_d)+Bu(t)+f(z,u,t) \qquad (3)$$

where z_d is reference trajectory and $u(t)$ is the input to the system. We choose *m* switching functions as follows

$$s_i(z)=c_i z=c_{i1}z_1+c_{i2}z_2+\ldots+c_{i2n}z_{2n} \qquad (4)$$

where $c_i=[c_{i1},c_{i2},\ldots,c_{i2n}]$, c_i is a sliding vector. We rewrite Equation (4) in the form

$$s(z) = cz \tag{5}$$

where $c = [c_1, \ldots, c_m]^T$.
The following is a possible choice of the structure of a sliding mode controller [19]

$$u = u_h + u_{eq} \tag{6}$$

where

$$\begin{aligned} u_{eq} &= -(cB)^{-1} cAz \\ u_h &= -(cB)^{-1}(\gamma + \sigma)\frac{s}{\|s\|} \end{aligned} \tag{7}$$

The control strategy adopted here will guarantee a system trajectory move toward and stay on the sliding surface $s = 0$ from any initial condition if the following condition meets

$$s\dot{s} \le -\sigma|s| \tag{8}$$

where η is a positive constant that guarantees the system trajectories hit the sliding surface in finite time [19].

It is proven that if k is large enough, the sliding model controllers of (4) are guaranteed to be asymptotically stable [19].

The fuzzy control rule is the spirit of fuzzy control design. However, when the fuzzy variables are more than two, establishing a complete fuzzy rule set becomes difficult. The SMC guarantees the stability and robustness of the resulting control system, which can be systematically achieved but at the cost of chattering effect. The FSMC is a hybrid controller, which combines the advantages of the fuzzy controller and the sliding mode controller. The combination becomes a feasible approach to rectify the shortcomings and preserve the advantages of these two approaches. The structure of fuzzy sliding mode controller is described as follows:

According to the control law (6) with one switching function, we have the fuzzy control rule j [21] as
R^j: If s is A^j, then u is u_j.
where $j=-q,-q+1,\ldots,q$, s is obtained from Equation (5) for one switching function, and A^j is a linguistic value with respect to s of rule j. The definition of membership function is

$$\mu_{A^j}(s) = \begin{cases} \dfrac{s - \sigma_{j-1}}{\sigma_j - \sigma_{j-1}}, \sigma_{j-1} < s < \sigma_j \\ \dfrac{\sigma_{j+1} - s}{\sigma_{j+1} - \sigma_j}, \sigma_j < s < \sigma_{j+1} \end{cases} \tag{9}$$

where σ_j is the centre of j^{th} membership function. The triangle membership function is determined by three parameters σ_{j-1}, σ_j and σ_{j+1}. The definition of membership functions is symmetrical, that is, $\sigma_o = 0$, $\sigma_{-1} = -\sigma_1, \ldots,$ $\sigma_{-q} = -\sigma_q$. The control law u_j is

$$u_j = k_j \operatorname{sgn}(s_j) + u_{eqj} \tag{10}$$

where

$$u_{eqj} = G_j z \tag{11}$$

in which $G_j = \lfloor g_{j,1}, g_{j,2},..., g_{j2,n} \rfloor$, and $j=-q,-q+1,\ldots,q$. With respect to the SMC, the parameters are assumed as follows:

$$\begin{aligned} &q = 1 \qquad \sigma_o = 0 \qquad \sigma_{-1} = \sigma_1 = \varepsilon \qquad with \quad \varepsilon \to 0 \\ &G_{-1} = Go = G_1 = -(cB)^{-1} cA \\ &k_{-1} = (cB)^{-1}(\gamma + \sigma) \qquad k_o = 0 \qquad k_1 = -(cB)^{-1}(\gamma + \sigma) \end{aligned} \tag{12}$$

From the control point of view, the parameters of structures should be modified automatically by evaluating the results of fuzzy control. In this section, we will introduce the GAs [22] to the problem of determining and optimizing the FSMC for the system (2). The key to put a genetic search for the FSMC into practice is that all design variables to be optimized are encoded as a finite length string. Each design is represented by a binary string, which consists various smaller strings that can be decoded to the value for each design variable. According to the structure and parameters of the FSMC in section 4, individual multivariable binary coding is arranged in the following form

c_1	C_2	….	c_{2n}	σ_1	σ_2	….	σ_q	k_{-q}	k_{-q+1}	….	k_q	G_{-q}	G_{-q+1}	…	G_q

where $\sigma_j, j = 1,2,...q$ are parameters of membership functions in antecedent fuzzy sets as shown in Equation (9), k_j and G_j are parameters of the consequent part as shown in Equations (10), (11). The binary string of this FSMC has $1 + 4n+3q +4nq$ variables.

Fitness as a qualitative attribute measures the reproductive efficiency of living creatures according to the principle of survival of the fittest. In the FSMC design, the parameters of controller are determined and optimized through assessing the individual fitness. In order to employ the GAs to optimize the FSMC for the system, we establish the fitness function according to the objective of active vibration control. Thus the FSMC design based on the GAs can be considered as an optimization search procedure over a large parameter space. For the active vibration control, we define the performance index [23] as

$$J = \frac{1}{M} \sum_{k=1}^{M} |s_k| \tag{13}$$

where s_k is the value of switching function s at the k^{th} time step, $M = \text{int}(t_{\max} / \Delta t)$ denotes the number of computing steps, $t_{\max}$ is the running time, and Δt is the sampling period. The fitness function can then be defined as

$$F = \frac{1}{J + \tau} \tag{14}$$

where τ is a small positive constant used to avoid the numerical error of dividing by zero.
The GAs control parameters play an important role in the procedure of optimizing the parameters of the fuzzy logic controller. Some worthwhile discussions of the GAs parameters are made as follows:

- *Encoding form:* The linear encoding form is used. The length of binary coding string for each variable is important for the GAs. There is always a compromise between complexity and accuracy in the choice of string length. Here, a 16-bit binary coding is used for each parameter.
- *Crossover and mutation rates:* Crossover and mutation rates are not fixed during evolution period. At the beginning, crossover and mutation rates are, respectively, fixed to 0.9 and 0.1, then decrease 10 percent in each generation until crossover rate is 0.5 and mutation rate is 0.01.

- *Population size:* The population size has to be an even number and is kept fixed throughout. Generally, the bigger the population size, the more design features are included. The population size should not be too small, but the procedure of optimizing will be slow when the population size is big.

IV. Simulation Results

The network topology used for simulation, is depicted in Fig. 2. The only bottleneck link lies between node A and node B. the buffer size of node A is 300 packets, and default size of the packet is 500 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 time 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, τ_1). The pair (L_2, τ_2) and (L_3, τ_3) define the parameter of links AB and BC, respectively.

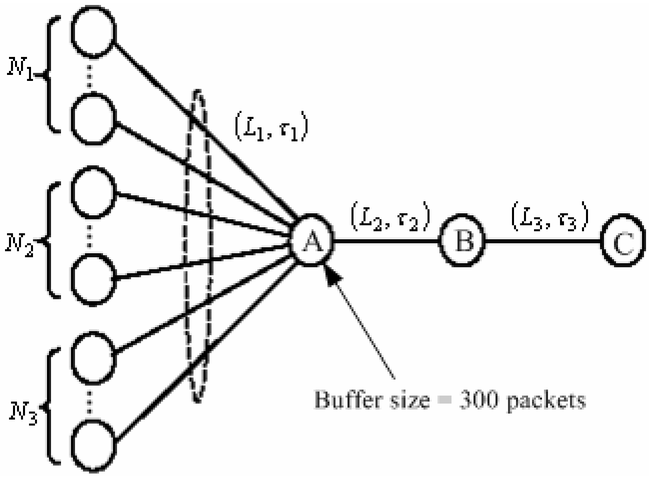

Fig. 2. The simulation network topology

In the first study, we will use the most general network configuration to testify whether the proposed controller can reach the goals of AQM, and freely control the queue length to stabilize at the arbitrary expected value. Therefore, given that $(L_1, \tau_1) = (10Mbps, 15ms)$, $(L_2, \tau_2) = (15Mbps, 15ms)$, $(L_3, \tau_3) = (45Mbps, 15ms)$. $N_1 = 270$, $N_2 = N_3 = 0$. Let the expected queue length equal to 75 packets.
To design the controller at first stage, we choose one switching function as

$$s = cz = c_1 z_1 + c_2 z_2$$

in which

$$c = [1 \quad 1].$$

For the FSMC, we adopt the following three control rules

R^{-1}: If s is *NB*, then u is $G_{-1}z + k_{-1}$.

R^{o}: If s is *NB*, then u is $G_o z + k_o$.

R^{1}: If s is *NB*, then u is $G_1 z + k_1$.

where the membership functions with respect to fuzzy sets NB, ZO, and PB are shown in Fig. 3. In this paper, the design parameters of the FSMC are selected as follows:

$$\sigma_1 = 0.12 \quad k_{-1} = 2500 \quad k_o = 0 \quad k_1 = -2500 \quad G_{-1} = G_o = G_1 = [23.95 \quad -84.55]$$

The design parameters of the FSMC based on the GAs associated with the above three control rules are specified as follows: sampling time interval = 0.02sec, population size = 50, initial crossover probability = 0.9, initial mutation probability = 0.1, bit length for parameter =16, generations = 60, σ, k, c and G are [0,1], [2500,2500], [0,15000] and [0,15000], respectively. The optimal parameters of the FSMC are generated after 60 generations, namely,

$c = [3381 \quad 4423]$ $\quad G_{-1} = [-12.75 \quad 93.05]$ $\quad G_o = [32.08 \quad -14.71]$ $\quad G_1 = [54.72 \quad 23.61]$

$\sigma_1 = 0.0316$ $\quad k_{-1} = 2460$ $\quad k_o = 93.7$ $\quad k_1 = -2274$

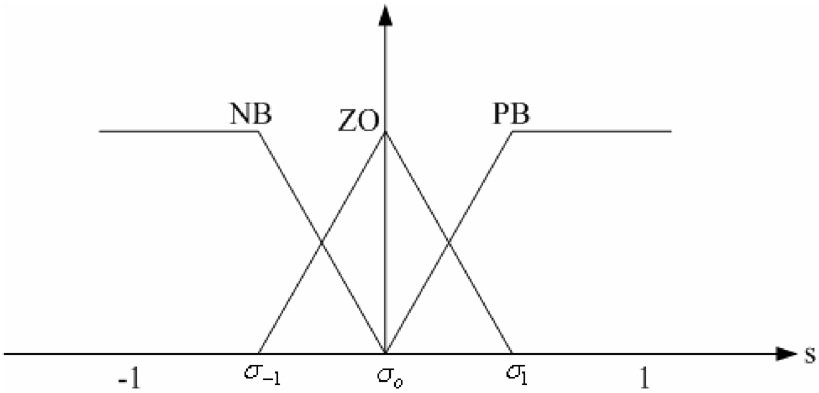

Fig. 3: The membership function of fuzzy sets

The instantaneous queue length using the proposed FSMC based on GAs is depicted in Fig. 4. 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. 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. Although the FSMC 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. The queue evaluation of router A, controlled by FSMC (without using GAs) (q_o=75packets), is plotted in Fig. 5. Evidently, FSMC takes the longer time to settle down the reference point.

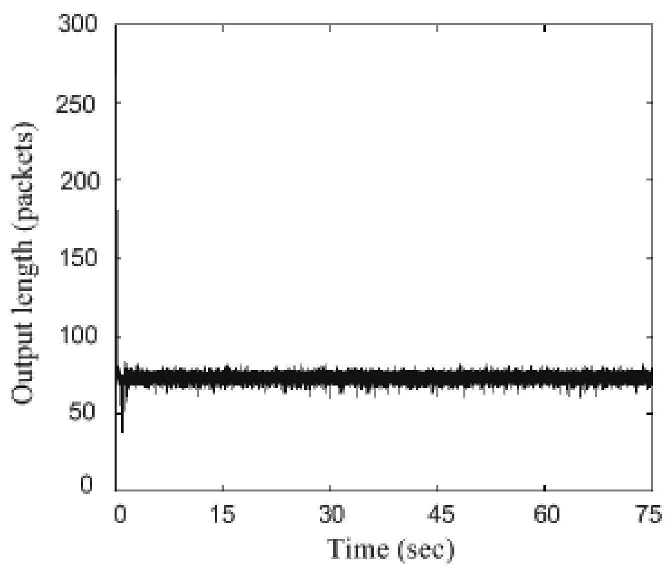

Fig. 4. Queue evaluation using FSMC based on GAs

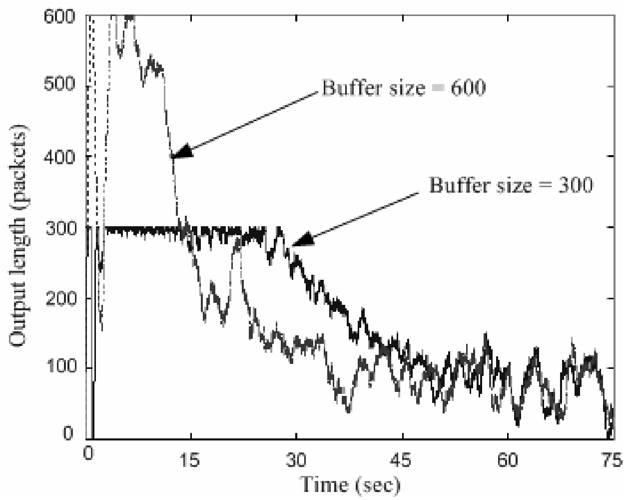

Fig. 5. Queue evaluation using FSMC (without GAs)

With the higher sampling frequency, the computation will be significantly exhausted. The only feasible way is to add the buffer size. In order to illustrate this ability, we redo the above simulation with 600 packets buffer size, which the results are also plotted in Fig. 5. Indeed, the large buffer is able to enhance the responsibility of FSMC but this ability is limited, moreover it seems to be wasteful. Conversely, the FSMC based on GAs has the ideal performance without any additional regulation mechanism. In order to evaluate the performance in steady state, we calculate the average and the standard deviation of the queue length in steady state. For the convenience of comparison, choose the queue length between 60 and 75 seconds as sample data. In this case, the standard deviation of FSMC (34.2) is much larger than that of FSMC based on GAs (3.1). Fig. 6 presents the case of small reference queue length. Except $q_o = 15$, the other parameters are unchangeable.

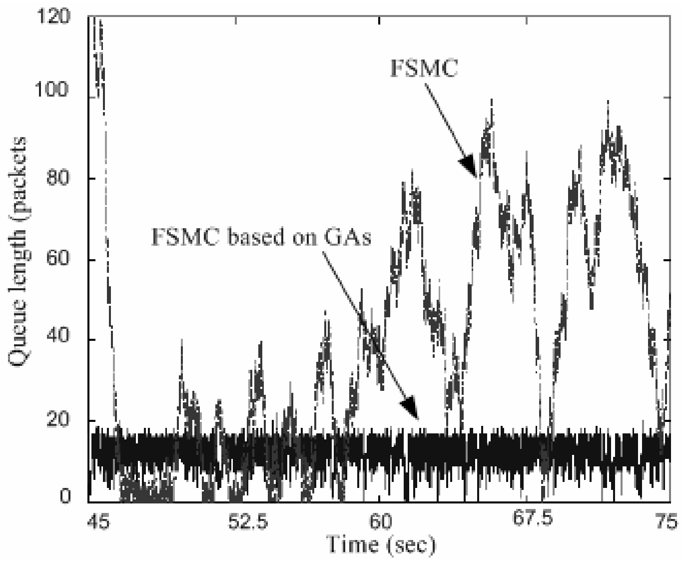

Fig. 6. Small expected queue (q_o=*15*)

In this section, Firstly, let $N_1 = 270, N_2 = 400, N_3 = 0,$ the evaluation of queue size is shown in Fig. 7. As it can be seen, the proposed FSMC based on GAs has better performance than that of FSMC. 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. 8 shows the results, obviously, FSMC is very sensitive to this disturbance, while FSMC based on GAs operates in a relatively stable state. The queue fluctuation increases with introducing the UDP flows, but the variance is too much smaller comparing with FSMC.

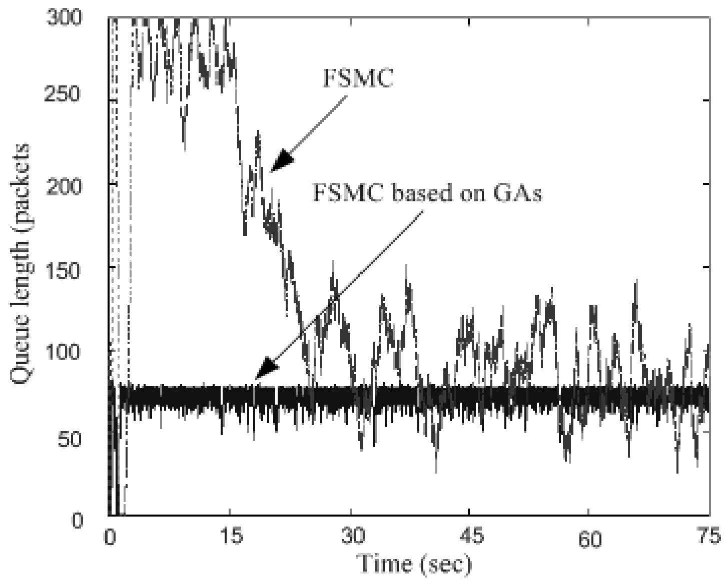

Fig. 7. Queue evaluation (FTP+HTTP)

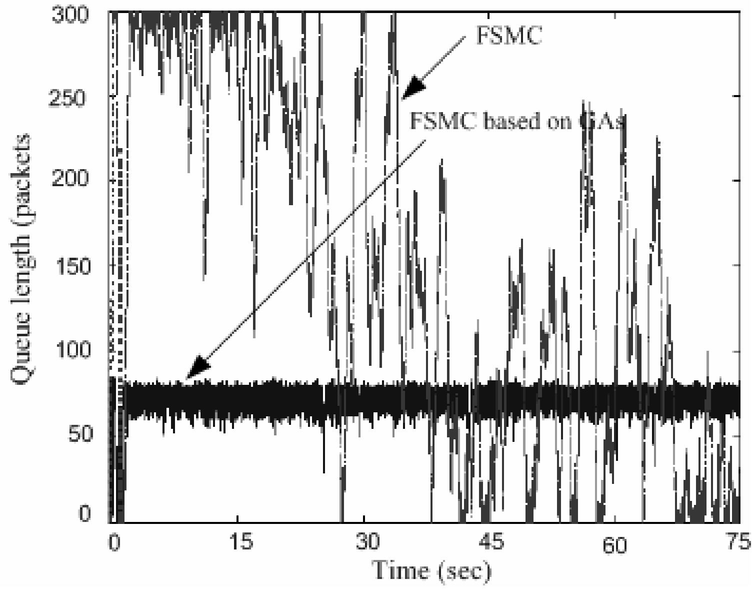

Fig. 8. Queue evaluation (FTP+UDP)

V. Conclusion

In this paper, a fuzzy sliding mode controller based on Genetic Algorithms was designed for the objective of active queue management. For this purpose, a linearized model of the TCP flow was considered. The proposed control strategy was insensitive to system dynamic parameters and was capable of being against disturbance and noise, which is very suitable for the mutable network environment. We took a complete comparison between performance of the proposed FSMC based on GAs and FSMC under various scenarios. The conclusion was that the integrated performance of FSMC based on GAs was superior to that of FSMC itself. The former was very responsive, stable and robust, especially for the small reference queue system, but its performance was inferior when active TCP sessions were relatively small. Thus, it will be very imperious to design the controller suitable for light load, and then integrate it with the proposed controller using classical adaptive control technology.

References

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

[2] Floyd, S. and Jacobson, V., Random early detection gateway for congestion avoidance, IEEE/ACM Trans. Networking, August 1993.
[3] Stevens, W. TCP slow start, congestion avoidance, fast retransmit, and fast recovery algorithms. Request for Comments RFC 2001, Internet Engineering Task Force, 1997.
[4] Jacobson, V., Braden, R., Borman, D. TCP extensions for high performance. Request for Comments RFC 1323, Internet Engineering Task Force, 1992.
[5] Brakmo, L., Peterson, L. TCP Vegas: End to end congestion avoidance on a global Internet. IEEE Journal of Selected Areas in Communications, 13(8), pp.1465–1480, 1995.
[6] Floyd, S., & Fall, K. Promoting the use of end-to-end congestion control in the Internet. IEEE/ACM Trans. Networking, 7 (4), pp.458-472, 1999.
[7] Chrysostomou, C., A. Pitsillides, L. Rossides, A. Sekercioglu, Fuzzy Logic Controlled RED: Congestion Control in TCP/IP Differentiated Services Networks, Soft Computing, 8 (2), pp.79-92, 2003.
[8] Karn, P., Partridge, C. Improving round-trip time estimates in reliable transport protocol. in Proc. ACM SIGCOMM, vol. 2, pp.1018-1020, 1987.
[9] Stevens, W. TCP/IP illustrated, vol. 1: the protocols. Reading, MA: Addison-Wesley, 1994.
[10] Madhavi, J., S. Floyd, TCP-Friendly unicast rate-based flow control. Note sent to end2end-interest mailing list, January, 1997.
[11] Padhye, J., Kurose, D. Townsley, R. Koodli, A Modeling TCP Reno performance: a simple model and its empirical validation, IEEE/ACM Trans. Networking, 8 (2), pp.133-145, 2000.
[12] Zhang, Y., Wang, F., Hesketh, T., et al., Fault accommodation for nonlinear systems using fuzzy adaptive sliding control, International Journal of System Science, 36(4), pp.215-220, 2005.05.19
[13] Lu, Y.S., Chen, J.S., A self-organizing fuzzy sliding mode controller design for a class of nonlinear servo systems. IEEE Transactions of Industrial Electronics, 41, pp.492– 502, 1994.
[14] Ren, F.Y., Ren, Y., Shan, X.M.,Design of a fuzzy controller for active queue management, Computer Communications, 25(9), pp.874-883, 2002.
[15] Di Fatta, G., Hoffmann, F., Lo Re G., et al.,A genetic algorithm for the design of a fuzzy controller for active queue management, IEEE Transactions on System, Man and Cybernetics, Part C-Applications and Review, 33(3), pp.313-324, 2003.
[16] Jalili-Kharaajoo, M., Fuzzy logic based congestion avoidance in TCP/AQM networks, Lecture Notes in Computer Science, 3261, pp.575-584, 2004.
[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] Jalili-Kharaajoo, M., Araabi, B.N., Application of predictive control algorithim to congestion control in differentiated service networks, Lecture Notes in Computer Science, 3124, pp.1157-1162, 2004.
[19] Edwards, C. & Spurgeon, S. K., Sliding mode control. Taylor & Francis Ltd, 1998.
[20] Young, D. K., Utkin,V. I. & Özgüner, A control engineer’s guide to sliding mode control. IEEE Transaction on Control System Technology 7(3), May, 1999.
[21] Chen JY. Expert SMC-based fuzzy control with genetic algorithms. Journal of the Franklin Institute, 6, pp.589–610, 1999.
[22] Falsafian, K., Jalili-Kharaajoo, M.,Genetic algorithm based fuzzy sliding mode with application to building structures, Lecture Notes in Artificial Intelligence, 3070, pp.960-965, 2004.
[23] Lin SC, Chen YY. Design of self-learning fuzzy sliding mode controllers based on genetic algorithms. Fuzzy Sets and Systems, 86, pp.139 –153, 1997.