Journal of Southeast University (English Edition) Vol. 20 No. 3 Sept. 2004 ISSN 1003—7985

Fuzzy variable structure delay control algorithms and their applications to AQM

Wang Mingzheng Li Jianhua Zhang Wenjun Zhu Hongwen

(Department of Electronic Engineering, Shanghai Jiaotong University, Shanghai 200030, China)

Abstract: This paper analyzes the fuzzy variable structure control algorithms for delay systems and describes the compensation mechanism of the integral factor to the effect of the delay. Based on the linearized model of the congestion-avoidance flow-control mode of transmission control protocol (TCP), we present delay control algorithms for active queue management (AQM) and discuss the parameter tuning of the algorithms. The NS (network simulator) simulation results show that the proposed control scheme for the nonlinear TCP/AQM model has good performance and robustness with respect to the uncertainties of the round-trip time (RTT) and the number of active TCP sessions. Compared to other similar schemes, our algorithms perform better in terms of packet loss ratio, throughput and butter fluctuation.

Key words: time delay compensation; fuzzy variable structure control; active queue management; robustness

Unresponsive flows are flows that do not use end-to-end congestion control and, in particular, that do not reduce their load on the network when subjected to packet drops. This unresponsive behavior can result in both unfairness and congestion collapse for the Internet. The end-to-end congestion control mechanisms of transmission control protocol (TCP) have been a critical factor in the robustness of the Internet.

Improving the congestion control and queue management algorithms in the Internet has been one of the most active areas of research in the past few years. The basic idea behind an active queue management (AQM) algorithm, such as RED (random early detection)[1] and BLUE[2], is to convey congestion notification early enough to the senders, so that the senders are able to reduce the transmission rates before the queue overflows and any sustained packet loss occurs. However, due to the challenging nature of non-linearity in the TCP dynamics, most of the current results on the AQM analysis and design are based on linearized models, which are valid only in the neighborhood of the equilibrium points. To further complicate the situation, the TCP/AQM dynamics have time varying round-trip times (RTT) and uncertainties with respect to the number of active TCP sessions through the congested AQM router, which require more robustness for the designed schemes. The TCP connections through the congested routers can be modeled as a feedback dynamic system, where control theory can be used to analyze the network behavior, tune AQM's parameter settings, and design new AQM schemes. In Ref.[3], a control theoretic analysis was given for RED, which provided a more systematic and in-depth study on RED parameter tuning.

In this paper, we analyze the fuzzy variable structure control algorithms for delay systems and describe the compensation mechanism of the integral factor to the effect of the delay. Then we present delay control algorithms for AQM and discuss the parameter tuning of the algorithms. By analyzing the robustness of the proposed AQM scheme, we show that the proposed design has good stability robustness with respect to RTT and the number of active TCP sessions through the router. We also discuss the implementation issues and compare our design against related works via NS simulations.

1 Linearized Dynamic Model of TCP's Congestion-Avoidance Flow-Control Mode

The dynamic model of TCP behavior was developed using fluid-flow and stochastic differential equation analysis. Simulation results demonstrated that this model accurately captured the dynamics of TCP. The schematic of TCP model is shown in Fig.1.

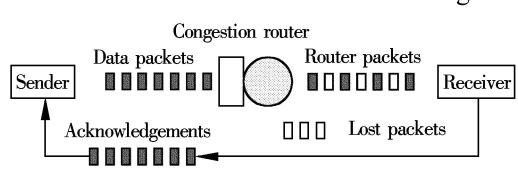

Fig.1 Schematic of TCP model

Received 2003-11-07.

Foundation item: The National High Technology Research and Development Program of China (863 Program) (No. 2001AA121042).

Biographies: Wang Mingzheng (1974—), male, graduate; Li Jianhua (corresponding author), male, doctor, professor, lijh888@sjtu.edu.cn.

This model relates the average value of key network variables and is described by the following coupled nonlinear differential equations:

$$\dot{W}(t)=\frac{1}{R(t)}-\frac{W(t)W(t-R(t))}{2\quad R(t-R(t))}\cdot p(t-R(t))$$

$$\dot{q}=\begin{cases}-C(t)+\dfrac{N(t)}{R(t)}W(t) & q>0\\ \max\left\{0,\ -C(t)+\dfrac{N(t)}{R(t)}W(t)\right\} & q=0\end{cases}$$

where $\dot{x}$ denotes the time-derivative, W is the average TCP window size (packets), q is the average queue length (packets), $R(t)$ is the round-trip time (s), $C(t)$ is the link capacity (packets/s), $N(t)$ is the load factor (number of TCP sessions), and p is the probability of packet mark.

To linearize the above nonlinear differential equations, we first assume that the number of TCP sessions and link capacity are constant, i.e., $N(t)=N$ and $C(t)=C$. Taking (W, q) as the state and p as input, the operating point (W_0, q_0, p_0) is then defined by $\dot{W}=0$ and $\dot{q}=0$, so that

$$\dot{W}=0\Rightarrow W_0^2 p_0=2$$

$$\dot{q}=0\Rightarrow W_0=\frac{R_0 C}{N}$$

$$R_0=\frac{q_0}{C}+T_p$$

where T_p is the propagation delay (s).

We ignore the dependence of the time-delay argument on queue-length, and assume it fixed to $(t-R_0)$ and define $\delta W \equiv W-W_0$, $\delta q=q-q_0$, $\delta p=p-p_0$, then obtain the dynamics around the operating point (W_0, q_0, p_0) (see Fig.2).

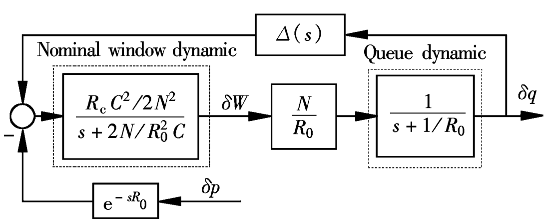

Fig.2 Linearized dynamic model of TCP's congestion-avoidance flow-control mode

2 Time Delay Problem and Existing Solution

Currently, the classical proportional, integral and derivative (PID) control is widely used with its gains manually tuned based on congestion control. Networks with different scenes requires different PID gains. In addition, network operation over wide ranges of point-set, for example, requires different gains at the lower and higher end to avoid overshoots and oscillation. This is necessary since even brief drop probability overshoots, for example, can initiate nuisance alarms and costly shut down to the process being controlled. Generally, tuning the proportional, integral, and derivative constants for a large network control process is costly and time consuming. The task is further complicated when incorrect PID constants are sometimes entered due to the lack of understanding of the network control process.

A number of time delay compensation and prediction schemes have been developed and/or improved with modifications. The performance of Smith predictor control (SPC)[4] was studied experimentally. It shows that the system performs well if the process model is accurate, but that performance degrades rapidly with inaccuracy in the process parameters and time delay. Clearly for an unknown or variable time delay, Smith predictive compensation is no longer a viable technique. Several control design methods for systems with varying time delays have appeared in recent literature including an estimation and self-tuning method, a variable structure controller, and a model reference adaptive approach. For systems with large time delays, most design approaches use a prediction mechanism as a part of the controller to simulate the process for given system parameters and time delay. In the well-known Smith predictor, the controller output is fed through models of the process with delay, and the process without delay, respectively. The difference of the output signals is added to the actual plant output and then fed back to the controller, thus allowing the controller to act on the prediction of the plant output. Using this well-known time delay compensation technique on a simple first order plant in an industry standard PID controller such as Bailey's Infi-90 single loop controller is still not an easy task. The predictor parameters including the plant gain, time constant, and time delay, in addition to the three PID parameters must be determined. These six parameters used in a predictive compensator increase tuning and operational complexity on even the simplest plants. The additional complexity of the Smith predictor is the main reason that industry still uses non-predictive PI or PID control for time delay using tuning methods such as Ziegler-Nichol's method.

The difficulty in dealing with such problems is compounded with variable time delays existed in many such systems. Variations in network scene, new product development and physical constraints are placed at different locations, inducing variable time delays (dead time) in the system. It is also well known that PID controllers exhibit poor performance when applied to the systems containing unknown nonlinearity such as dead zones saturation. It is further understood that many network control processes are nonlinear. Equal increments, for example, do not necessarily produce equal increments in drop probability rise in many processes, a typical phenomenon of nonlinear systems. The complexity of these problems and the difficulties in implementing conventional controllers to eliminate variations in PID tuning motivate us to investigate intelligent control techniques, such as fuzzy logic as a solution to controlling systems, in which time delays, nonlinearities, and manual tuning procedures need to be addressed.

Engineers have known the principles of fuzzy logic for more than 35 years. Fuzzy logic acquired its name from the Fuzzy Set Theory developed by Zadeh at the University of California — Berkeley in 1965. Practical applications appear later, when the world market became flooded with inexpensive micro-controllers. Fuzzy control (fuzzy logic in the role of a control system)[5] becomes attractive, especially for the smallest micro-controllers, because it requires less computational power and demands less operational memory than conventional PID compensation. However, nothing can prevent you from applying fuzzy control to network congestion control. Fig. 3 shows the principle of fuzzy controller.

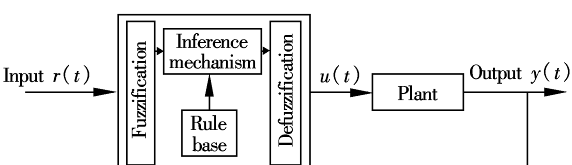

Fig.3 Fuzzy controller

3 Fuzzy Variable Structure Control with Time Delay Compensation (FVS-T)

The fuzzy two-term control system (see Fig. 4) employs a 2-D rule base. Since only two state variables e and $\dot{e}$ of the process are used, it is a low-order controller. As the difficulty of design mainly comes from the coupling of parameters in the knowledge base, the fuzzy logic control (FLC) design can follow the hybrid design methodology integrating both qualitative and quantitative approaches. Two separate stages are suggested for design and tuning. The nominal design is a top-down approach starting from the qualitative level (higher level) to the quantitative level (lower level). It intends to find out the nominal model of FLC, i.e., the initial parameters of FLC. If the nominal model is not satisfactory, the fine-tuning can be used to explore the fine parameters of FLC, through a bottom-up approach that continues learning from the nominal model. The nominal model of FLC includes the nominal rule base, which should be designed qualitatively, the nominal membership functions (MFs), and the nominal scaling gains, which should be designed quantitatively.

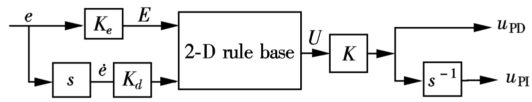

Fig.4 Fuzzy two-term control system

The qualitatively designed nominal rule base is often a linear 2-D rule base as shown in Fig.5 where two inputs are error position and its change rate. For simplicity, the nominal MFs can be chosen as triangular in shape. After determining the nominal rule base and MFs, all the design loads are shifted to the scaling gains that can be handled with the use of various quantitative methods. Properly designed scaling gains are very critical to the nominal performance of FLC. The linear rule base is divided into many inference cells (ICs), as shown in Fig.5.

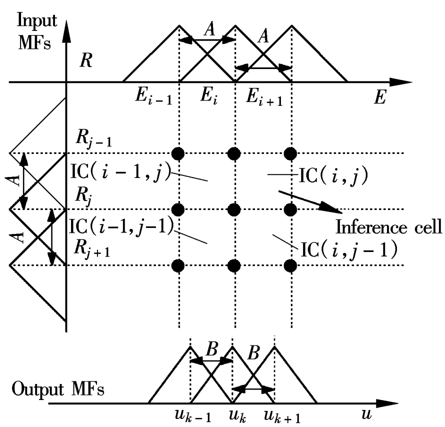

Fig.5 ICs divided by the linear rule

The mathematical model of a 2-D rule base can be derived from one of ICs:

1) The rule base output:

$$U = kB(1-\gamma) + \frac{B}{A}\gamma S$$

2) The fuzzy PD control output:

$$u_{PD} = KU$$

3) The fuzzy PI control output:

$$u_{PI} = KU/s$$

where the inference method is based on the max-min method, the membership functions of triangular shape are equally spread, and defuzzification is based on the center-of-gravity method. $S = K_d(\lambda e + \dot{e})$ is the first-order switching function with the same sign as U, i.e., $SU > 0$. K_e and K_d are the scaling gains for input e and $\dot{e}$, and $\lambda = K_e / K_d$ is the input gain ratio. K is the output scaling gain. $1/s$ is the integral operator. γ is a nonlinear time-varying parameter ($2/3 \leqslant \gamma \leqslant 1$). A and B are half of the spread of each input and output triangular MF, respectively. k is an index number depending on the inference cell under use; the larger the k value, the farther away the state $(e, \dot{e})$ is from the switching surface $S = 0$ in the phase plane (see Fig.6).

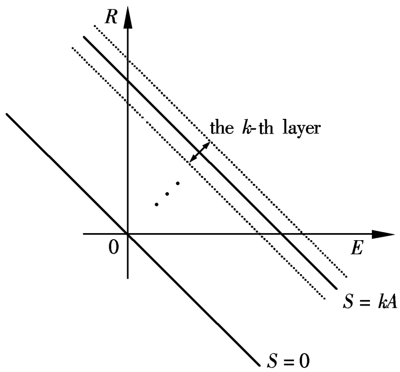

Fig.6 Sketch map for the slide-mode plane of fuzzy variable structure control

After modeling, various quantitative approaches can be applied to designing the nominal scaling gains. However, there is still no systematic way for fine-tuning of the scaling gains.

For the time-delay process, the observed information comes later than desired for taking the control action. The delayed information from the process when used directly gives the wrong information to the rule base, and hence the wrong control to the process. One way to compensate for the time-delay effect is to add the delay compensation factor in the control loop. The delay compensation factor can be expressed as

$$e^{\tau s} \approx 1 + \tau s + \frac{(\tau s)^2}{2!} + \cdots + \frac{(\tau s)^n}{n!}$$

When the delay is reasonably small, the delay compensation factor can be approximated as a first-order polynomial for easier implementation. When the fuzzy PI controller employs a first-order delay compensation $1 + \tau' s$, it becomes a fuzzy PID controller in structure. Practically, the exact delay information is difficult to obtain for a full compensation. Assuming that the delay time is τ and the compensation time is τ', the overall control system gives an equivalent PID effect. When the system is over-compensated, the equivalent PID effect will improve performance. On the other hand, under-compensation will leave some delay uncompensated, which will definitely deteriorate performance. In conclusion, over-compensation is better than under-compensation in terms of the equivalent PID effect on the overall control system.

We ignore the dependence of the time-delay argument on queue-length, and assume it fixed to $(t - R_0)$ and define $\delta W \equiv W - W_0$, $\delta q = q - q_0$, $\delta p = p - p_0$, then obtain the dynamics around the operating point (W_0, q_0, p_0) (see Fig. 2). We can obtain the following equation:

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

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

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

The sum of $P(s)$ and $\Delta(s)$ can be regarded as a second level system, therefore we utilize a fuzzy two-term control system to control its dynamic. The nonlinear TCP dynamics and FVS-T based AQM can be modeled as a feedback control system shown in Fig. 7, where FVS-T controller uses the queue and traffic incoming rate information to generate marking rate as congestion indication.

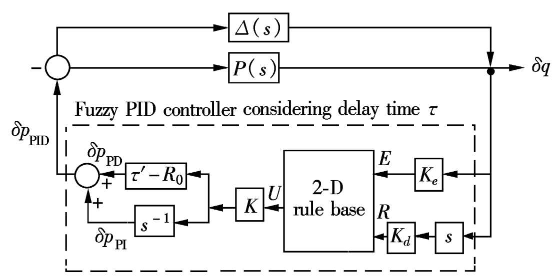

Fig.7 FVS-T based linearized TCP's congestion-avoidance flow-control model

A numerical illustration considers the case when $q_0 = 175$ packets, $T_p = 0.2$ s and $C = 3\ 750$ packets/s. Then, for a load of $N = 60$ TCP sessions, we have $W_0 = 15$ packets, $p_0 = 8 \times 10^{-3}$ and $R_0 = 0.246$. According to the linearized dynamics, we can obtain the practical model around the operating point. For AQM, performance objectives include efficient queue utilization, regulated queueing delay, and robustness.

4 Parameter Tuning for FVS-T Algorithms

As the difficulty of design mainly comes from the coupling of parameters in the knowledge base, therefore the FLC design can follow the hybrid design methodology integrating both qualitative and quantitative approaches, as shown in Fig. 8. Two separate stages are suggested for design and tuning. The nominal design is a top-down approach starting from the qualitative level (higher level) to the quantitative level (lower level). It intends to find out the nominal model of FLC, i.e., the initial parameters of FLC. If the nominal model is not satisfactory, the fine-tuning can be used to explore the fine parameters of FLC, through a bottom-up approach that continues learning from the nominal model. The nominal model of FLC includes the nominal rule base, which should be designed qualitatively, the nominal MFs, and the nominal scaling gains, which should be designed quantitatively.

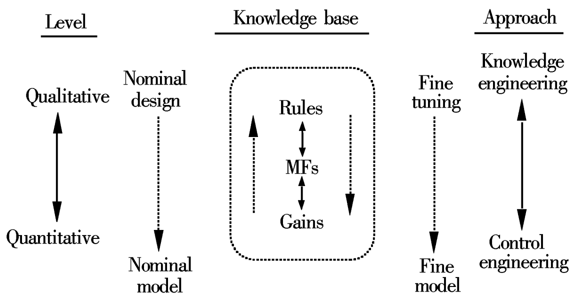

Fig.8 Design process for fuzzy variable structure algorithm

The FVS-T's parameters γ, A, B and k are determined by the structure of the MFs. In our example, we assume that $A = B = 1/3$, and $\gamma = 3/4$. Therefore, only three I/O scaling gains K_d, λ, τ' and K can be tuned by the user.

1) According to the variable structure control (VSC) theory, the ratio λ is usually chosen for the satisfactory dynamics of the switching surface. Good dynamics of the switching surface are essential to the stability and robustness of the entire system, generally $\lambda = 1$ to 10.

2) From the stability design, we can see that the gain K_d has little influence on the stability. However, too much input saturation may cause instability. Then K_d should be chosen to have a reasonable K_e so that both inputs have the best resolution and less saturation, generally $K_d = 0.2$ to 4.

3) The output gain K significantly affects the stability. Generally speaking, smaller K produces a smaller control signal which will reduce chattering (one feature of VSC) greatly. As there is the accumulation in PI-type ($K = 0.2$ to 5) control, the output gain K may be much smaller than that in PD-type ($K = 2$ to 20) control in order to keep the stability criterion valid. Therefore the chattering in PI-type control is usually much smaller than PD-type. Too large K may also cause instability, in this sense, K behaves similarly to the proportional gain of the conventional two-term control.

4) RTT is time varying obeying $R_0 \leqslant \tau(t) \leqslant \bar{\tau} + 0.05\sin\frac{2\pi}{100}t$, generally $\tau = 0.1$ to 0.4. In order to over-compensate the effect of R_0, we can set $\tau' = 1.2R_0$. Fig.9 shows the effect of delay compensation.

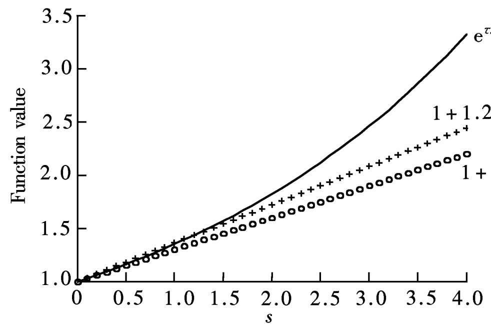

Fig.9 Effect of delay compensation factor ($\tau = 0.3$)

5 Related Work

In this section, we give a brief overview of the other two active queue management algorithms, RED and BLUE. These will be used for performance comparison.

5.1 RED[6]

A network with RED algorithm detects congestion and measures the traffic load in the network by using the average queue size s_{avg}. It is calculated using an exponentially weighted moving average filter and expressed as

$$s_{avg} = (1 - w_q)s_{avg} + w_q q$$

where w_q is the queue weight.

When the average queue size is smaller than the minimum threshold t_{min}, no packets are dropped. Once the average queue size exceeds the minimum threshold, the router considers the network in congestion and randomly drops the arriving packets with a given drop probability. The probability that a packet arriving at the RED queue is dropped depends on the average queue length, the time elapsed since the last packet was dropped, and the maximum drop probability parameter P_{max}. The drop probability P_a is computed as

$$P_a = \frac{P_b}{1 - mP_b}$$

where $P_b = P_{max}\frac{s_{avg} - t_{min}}{t_{max} - t_{min}}$, P_{max} is the maximum value for P_b, and m is a variable that keeps track of the number of packets that have been forwarded since the last dropped. If the average queue size is larger than the maximum threshold t_{max}, all the arriving packets will be dropped.

The average queue length s_{avg} is related to the number of active connection N in the system as follows:

$$s_{avg} = 0.91 N^{\frac{2}{3}} \left(\frac{t_{max}}{P_{max}} \right)^{\frac{1}{3}}$$

For the RED algorithm, this equation implies that s_{avg} increases with N until t_{max} is reached, and there is always an N where t_{max} will be exceeded. Since most existing routers operate with a limited amount of buffer, t_{max} is usually small and can easily be exceeded even with small N. Dropping all incoming packets may result in global synchronization, which is usually followed by a sustained period of low-link utilization.

5.2 BLUE[2]

Instead of calculating the average queue size, BLUE uses buffer overflow and link-idle events to manage congestion. If the queue size keeps on exceeding a certain value after a given period of time, called the freeze time, the drop probability will be increased by a constant value d_1. On the other hand, when the link remains idle for the freeze time, the drop probability will be decreased by a constant value d_2. BLUE algorithm can be approximated by a closed-loop negative feedback system, in which the control variable is increased when the processor output is smaller than a threshold value, and the control variable is decreased when the processor output is larger than the threshold value. Let e be defined as an error signal, which is the difference between the control target and the processor output (actual queue size). Then the following control law can be defined in discrete time.

$$p(n) = p(n-1) + df(e(n))$$

where the drop probability updates are limited to $\pm d$ which can be implemented with a relay. For the drop probability of BLUE, the previous expression can be expressed as

$$p(n) = p(n-1) + \alpha^{\mathrm{sgn}\left[\left\lfloor \frac{q(n)-T}{T} \right\rfloor - 1\right]} \cdot d\,\mathrm{sgn}\left[\frac{q(n)-T}{T}\left(\left\lfloor \left| \frac{q(n)-T}{T} \right| \right\rfloor\right)\right]$$

where T is the control target, and $\alpha = d_1 / d_2$. If d is the drop probability adjustment parameter d_1, and the control target is set to $B/2$, the drop probability of BLUE can be expressed as the above equation. This type of control leads to a system where the process output oscillates. Thus, the queue size in a router can theoretically be unstable with the BLUE algorithm.

6 Simulation Results

In this section, we evaluate the performance of FVS-T, and compare it with those of RED and BLUE. The TCP sources are based on a TCP-Reno implementation. The Reno version uses the fast-retransmit and fast-recovery mechanisms. The maximum segment size (MSS) for TCP is set to 536 bytes. The receiver's advertised window size is set sufficiently large so that TCP connections are not constrained at the destination. The ACK-every-segment strategy is used at the TCP destinations. The TCP timer granularity "tick" is set to 500 ms and the minimum retransmission timeout is set to two ticks. Because of the different design approaches in each algorithm, it is difficult to fairly compare them. In order to be fair, we set most of the parameter values in each algorithm using the recommended values from the original papers.

We describe the network configuration and the network topology. Fig.10 shows the network topology that represents a simple bottleneck network configuration with two routers and the number of subnet nodes. Each subnet has a number of TCP sources (e.g. 100 TCP sources per subnet). Therefore, given that $(L_1, \tau_1) = (10$ Mbit/s, 20 ms), $(L_2, \tau_2) = (25$ Mbit/s, 20 ms) and $(L_3, \tau_3) = (55$ Mbit/s, 20 ms). After a very short regulating process, the queue settles down to its stable operating point.

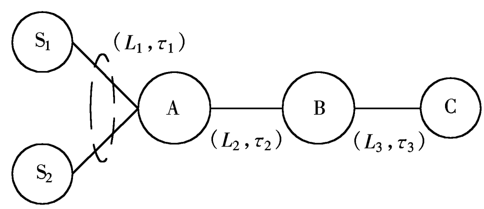

Fig.10 Network topology

Fig.11 shows the queue size of each algorithm with 800 TCP connections respectively. It can be seen that the queue sizes are unstable in both RED and BLUE algorithms. This is due to the design principles of both algorithms. The queue size of FVS-T is stable around the target buffer occupancy and load-independent.

Fig. 12 shows the drop probability of each algorithm with 800 TCP connections respectively. The drop probability of RED is the highest for all three algorithms. The drop probability of FVS-T seems to adapt faster compared with BLUE. BLUE's drop probability does not react fast enough, thus leading to the periods of buffer overflow and underflow.

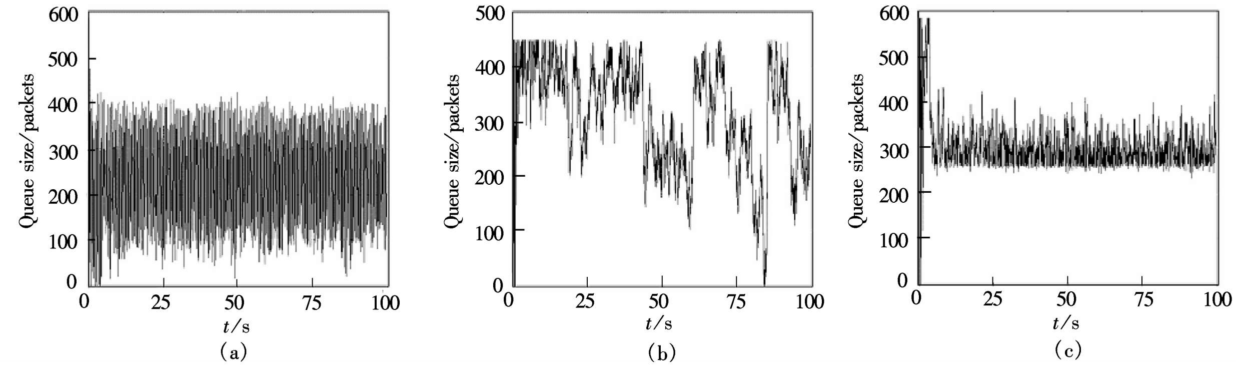

Fig.11 Queue sizes for 800 TCP connections. (a) RED; (b) BLUE; (c) FVS-T

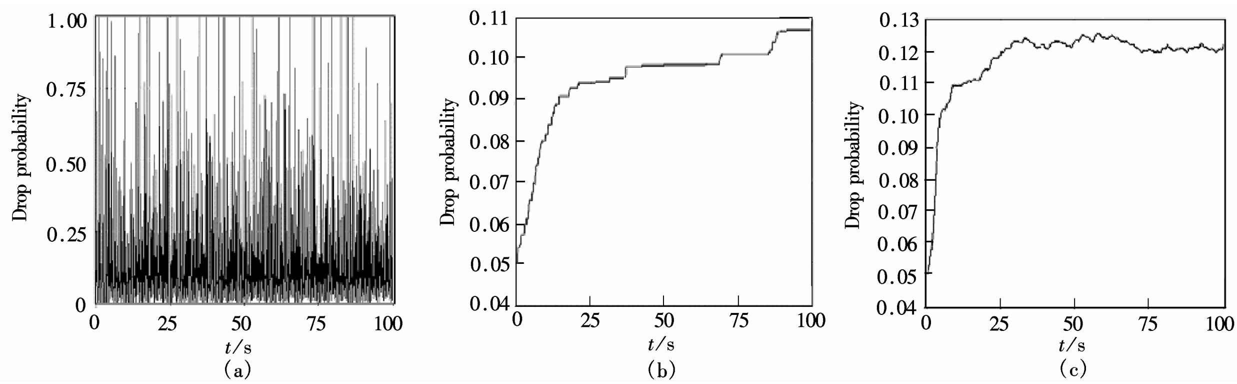

Fig.12 Drop probabilities for 800 TCP connections. (a) RED; (b) BLUE; (c) FVS-T

7 Conclusion

In this paper, we analyze the fuzzy variable structure delay control algorithms and apply them to active queue management. Simulation results show that our method is effective. RED algorithm is difficult to stabilize the queue size as well as to parameterize a queue to give good performance under different network scenarios and over a wide range of load levels. Like RED, BLUE has a hard time stabilizing the queue size. It took BLUE some fine-tuning before getting those results. Even with fine-tuning, BLUE does not seem to have good responsiveness when the traffic load changes. FVS-T stabilizes the queue size very well, thus resulting in a more predictable packet delay inside the network. FVS-T is much simpler to implement than RED, since it does not require any per-flow accounting mechanism.

There still remain many simulation scenarios under which all three algorithms can be evaluated and compared, e. g., different network configurations, short-lived and long-lived flows and other variations found in the Internet. We are currently investigating them according to this direction.

References

[1] Floyd S, Jacobson V. Random early detection gateways for congestion avoidance [J]. *IEEE/ACM Transactions on Networking*, **1993, 1**(4): 397 – 413.

[2] Feng W, Shin K G, Kandlur D D, et al. The BLUE active queue management algorithms [J]. *IEEE/ACM Transactions on Networking*, **2002, 10**(4): 513 – 528.

[3] Hollot C, Misra V, Towsley D, et al. Analysis and design of controller for AQM routers supporting TCP flows [J]. *IEEE Trans Auto Control*, **2002, 47**(6): 945 – 959.

[4] Benouarets M, Atherton D P. Autotuning design methods for a smith predictor control scheme [A]. In: *International Conference on Control* [C]. Warwick, UK, 1994. 795 – 800.

[5] Li H X, Gatland H B, Green A W. Fuzzy variable structure control [J]. *IEEE Transactions on Systems, Man and Cybernrtics, Part B*, **1997, 27**(2): 306 – 312.

[6] Hollot C, Misra V, Towsley D, et al. A control theoretic analysis of RED [A]. In: *IEEE INFOCOM* 2001 [C]. Anchorege, Alaska, 2001. 1510 – 1519.

模糊变结构时延控制算法及在 AQM 中的应用

王明政　李建华　张文军　诸鸿文

(上海交通大学电子工程系, 上海 200030)

摘要: 研究了带有延迟时间补偿的模糊变结构控制算法, 分析了模糊变结构控制中积分因子对于时延因子的补偿机理, 并基于网络拥塞路由器中 TCP 连接的线性化模型设计了适用于主动队列管理的时延控制算法. 最后讨论了该控制算法的实现问题和参数整定, 进行了仿真实验. 对于 TCP 主动队列管理的性能分析和比较试验表明, 本算法在 RTT 和 TCP 会话数量变动时保持了良好性能和鲁棒性, 在丢包率、吞吐量和缓冲波动方面相比其他类似方案有明显的性能优势.

关键词: 时延补偿; 模糊变结构控制; 主动队列管理; 鲁棒性

中图分类号: TP393; TP273