A New Self-tuning Active Queue Management Algorithm Based on Adaptive Control

Zhang Heying[1], Liu Baohong[2], Xiao Liquan[1], Dou Wenhua[1]

[1] School of Computer, National University of Defense Technology, 410073,
Changsha, Hunan, China
hey_zhang@hotmail.com, douwh@vip.sina.com
[2] Institute of Automation, National University of Defense Technology, 410073,
Changsha, Hunan, China
liu_baohong@hotmail.com

Abstract. Most Active Queue Management (AQM) algorithms based on control theory have difficulty in obtaining desirable performance once the network conditions or the traffic patterns change out of the presumed ones they are designed for. To address these problems, a new self-tuning AQM algorithm called STR is proposed in this paper. STR has the ability of keeping minimum variance between the instantaneous queue length of the router and the reference value by estimating the parameters of the model of controlled object online and adjusting the packet drop probability accordingly. The performance of STR is evaluated through extensive simulations. The results show that STR is robust against the great changes of the network parameters and the traffic load.

1   Introduction

Congestion control is very important to the stability and scalability of Internet. To improve the performance of traditional end-to-end congestion control, IETF (Internet Engineering Task Force) strongly suggests using active queue management (AQM) in routers [1]. RED (Random Early Detection) is one of the well-known AQM algorithms [2]. Many simulations and tests show that the performance of RED is very sensitive to its parameter settings [3]. This is considered partly due to lack of systematic analysis during the design of RED. In view of this, many AQM algorithms based on control theory have been proposed in recent years [4-11]. These algorithms are

usually characterized by simple implementation and easy configuration. However, they have some disadvantages. For example, these algorithms are usually designed based on the linearized model of the controlled object, which is made up of TCP (Transmission Control Protocol) and the queue dynamics. While the accurate model of the controlled object is nonlinear and time-varying in real networks. The inaccuracy of the model will lead to dissatisfied performance for AQM algorithms especially in terms of robustness and stability. Moreover, the parameters of most AQM algorithms are determined according to some specific network conditions. When the real network condition is far from that presumed, the performance of the algorithm will become unpredictable. In view of the fact that AQM algorithms with fixed parameters are not very suitable for the highly variable network conditions, some adaptive AQM algorithms have been proposed recently [12]. Unfortunately, they cannot solve the problems thoroughly.

The rest of this paper is organized as follows. In section 2, we describe the design of STR in detail. In section 3, the performance of STR is evaluated through simulations. Finally, we give the conclusions in section 4.

2   The Self-tuning Active Queue Management

Through detailed analyses, we find that it seems impossible to establish accurate model of TCP and the queue dynamics with fixed parameters. To efficiently control congestions even if the network conditions change widely, the AQM algorithm should identify the parameters of the controlled object timely and adjust the control laws, i.e. the packet drop probability, accordingly. So we use the self-tuning regulator to design AQM algorithm, which combines the recursive least square parameter identification and the minimum variance control as illustrated in Fig.1. In TCP/AQM system, the reference input r of the system is the reference queue length. The controlled object is made up of TCP window adjustment scheme and the queue dynamics. And the input u and output y of the controlled object is the packet drop probability and the queue length respectively. The estimator identifies the parameters of the model of controlled object. The regulator adjusts the packet drop probability according to the identified parameters. The estimator and regulator compose the self-tuning AQM algorithm.
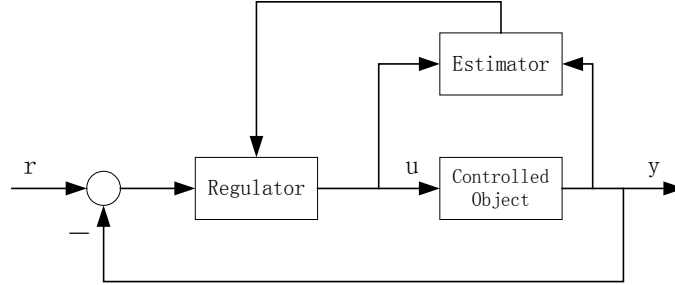
Fig. 1. Structure of self-tuning regulator

Now we will give the design of STR in detail. First of all, two key parameters of the system should be determined. One is the order n of the controlled object. The other is the delay d from the input of the controlled object to its output. If the parameters n and d of a system are unknown, it is feasible to select large values for them [13]. In [6], the differential equation model of TCP and queue dynamics is linearized about the operating point and a second-order system model is gained. So n can be selected as 2. Considering the stochastic disturbance frequently occurred in the network, we choose $n=3$ to guarantee the convergence of the parameter identification. In TCP/AQM system, the packet drop rate will delay one RTT (Round Trip Time) before affecting the queue length. To decrease the number of parameters which require identifying, we choose the sample cycle T=RTT. Then the delay between the packet drop and the change of the queue length is one sample cycle, namely $d=1$.

After determining n and d, the controlled object can be represented by:

$$q(k) + a_1 q(k-1) + a_2 q(k-2) + a_3 q(k-3) = b_0 p(k-1) + \\ b_1 p(k-2) + b_2 p(k-3) + b_3 p(k-4) \quad . \tag{1}$$

where $q(k)$ is the queue length and $p(k)$ is the packet drop probability.

Equation (1) can be rewritten as:

$$q(k+1) = b_0 p(k) + \varphi^T(k)\theta \quad . \tag{2}$$

where

$$\theta = \left[-a_1, -a_2, -a_3, b_1, b_2, b_3 \right]^T \\ \varphi(k) = \left[q(k), q(k-1), q(k-2), p(k-1), p(k-2), p(k-3) \right]^T \quad . \tag{3}$$

Calculating the variance between $q(k+1)$ and the reference value q_0, we yield:

$$E\left[q(k+1) - q_0 \right]^2 = E\left[b_0 p(k) + \varphi^T(k)\theta - q_0 \right]^2 \quad . \tag{4}$$

We have the following condition when the minimum variance is gained:

$$b_0 p(k) + \varphi^T(k)\theta - q_0 = 0 \quad . \tag{5}$$

Then, we get the packet drop probability of STR:

$$p(k) = \frac{1}{\hat{b}_0} \left[q_0 - \varphi^T(k)\hat{\theta} \right] \quad . \tag{6}$$

where $\hat{b}_0$ is the estimated value of b_0 and $\hat{\theta}$ is the estimated value of θ. To simplify the parameter identification and guarantee its convergence, we will determine the value of b_0 by experiments and use the basic least squares method to estimate θ. Since θ is variable rather than constant in real networks, we will use the recursive parameter identification algorithm with forgetting property which is suitable for the slowly variable parameter estimation. The estimation method is represented by:

$$\begin{cases} \hat{\theta}(k) = \hat{\theta}(k-1) + K(k)\left[q(k) - \hat{b}_0 p(k-1) - \varphi^T(k-1)\hat{\theta}(k-1)\right] \\ K(k) = H(k-1)\varphi(k-1)\left[\lambda(k) + \varphi^T(k-1)H(k-1)\varphi(k-1)\right]^{-1} \quad . \\ H(k) = \frac{1}{\lambda(k)}\left[I - K(k)\varphi^T(k-1)\right]H(k-1) \end{cases} \tag{7}$$

where λ is the "forgetting factor" ranging between 0.95 and 1. Generally, let $\hat{\theta}(0) = 0$, $H(0) = \alpha I$, where α is positive and large enough and I is a unit matrix. In TCP/AQM system, we will change the value of λ as follows [13]:

$$\lambda(k+1) = \begin{cases} \lambda(k)(1-\lambda_0) + \lambda_0 & |\varepsilon(k)| \le E \\ \lambda_0 & |\varepsilon(k)| > E \end{cases} \quad . \tag{8}$$

where, λ_0 is the lower bound of λ and set to 0.95. E is a constant. $\varepsilon(k)$ is the prediction error and defined as follows:

$$\varepsilon(k) = q(k) - \hat{b}_0 p(k-1) - \varphi^T(k-1)\hat{\theta}(k-1) \quad . \tag{9}$$

3   Simulations

We implement STR algorithm in NS simulator and compare its performance with PI and ARED. The network topology is shown in Fig.2. The propagation delays of connections range between 40 ms and 240 ms. The link between routers r1 and r2 is the only congested link with the capacity being 100 Mbps and the delay being 10 ms. The

buffer size of the routers is 800 packets, with the average packet size being 500 bytes. The flows from s_i to d_i are forward, while those from d_i to s_i are backward. The reference queue length of STR and PI is 100 packets. For STR, $E=20$, $\hat{b}_0 = -10^4$. The parameters of PI are selected according to [7]. The minimum and maximum queue thresholds of ARED are 50 packets and 150 packets respectively.
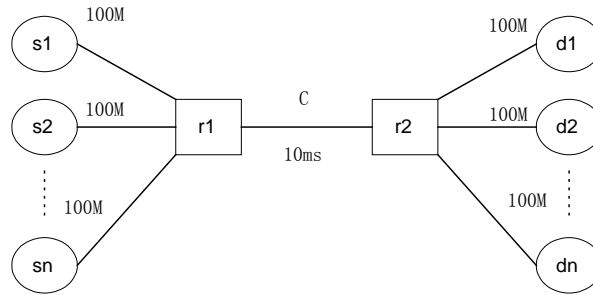

Fig. 2. Network topology used in the simulations

In the first experiment, we evaluate the responsiveness and robustness of STR when the traffic load changes. At the beginning of the simulation, there are 500 forward TCP flows. At $t=100$s, 450 TCP flows stop. And at $t=200$s, the stopped 450 TCP flows started again. The queue dynamics of STR, PI and ARED are depicted in Fig.3. It's easy to find that STR can keep small queue length and oscillations in either heavy or light load condition. For PI, the buffer is almost full when the traffic load is heavy. It however becomes empty frequently when the traffic load is light. For ARED, the queue oscillations are very large in heavy load condition.

The second experiment evaluates the performance of STR when the unresponsive flows and short TCP flows exist. At the beginning of the simulation, there are 100 FTP flows. At $t=100$s, 50 ON/OFF flows based on UDP protocol and 300 web flows based on HTTP protocol started simultaneously. The queue dynamics of STR, PI and ARED are shown in Fig.4. We can find that the queue length of PI increases abruptly and changes in a wide range when the ON/OFF flows and web flows exist. The queue oscillations of ARED become large. On the contrary, both the queue length and oscillations of STR remain small all the time.
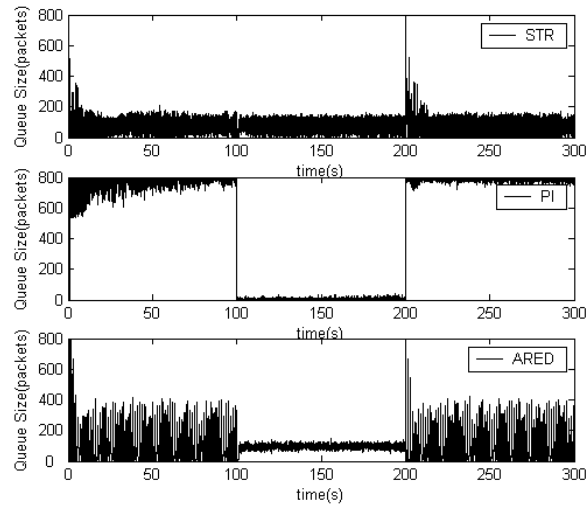
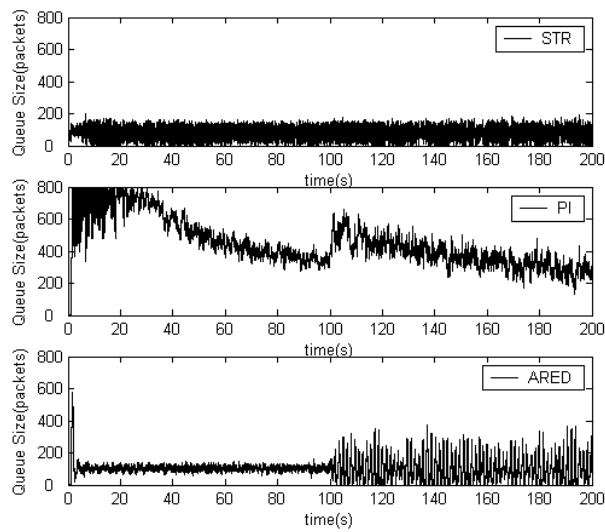**Fig. 3.** Queue dynamics when the traffic load changes

Fig. 4. Queue dynamics under mixed traffic

In the last experiment, all the routers in Fig.2 use AQM algorithm with the same configuration. At the beginning of the simulation, there are 100 forward TCP flows. At t=100s, 100 backward TCP flows joined. The queue dynamics of STR, PI and

ARED are shown in Fig.5. When the backward traffic is active, the queue length of PI becomes unstable and the queue oscillations of ARED also become large. In contrast, both the queue length and the oscillations of STR have no evident changes.
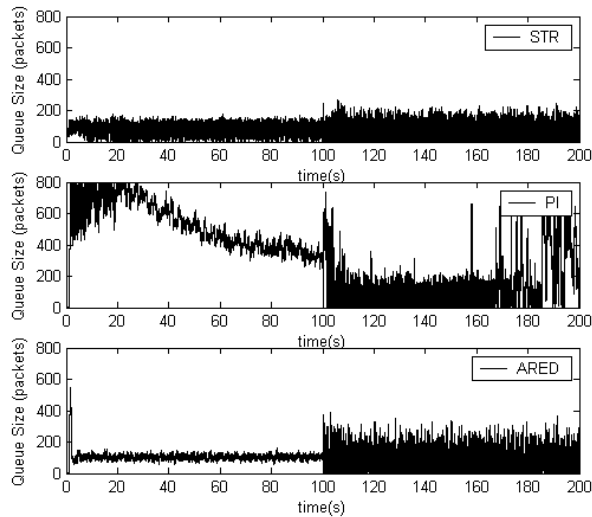

Fig. 5. Queue dynamics under bi-directional traffic

We also evaluate the performance of STR when the capacity of the bottleneck link is small, i.e. 15 Mbps. The results are similar to those described above. For space limitation, we don't show them here.

4   Conclusions

This paper proposes a new self-tuning active queue management algorithm called STR. Through estimating the parameters of the model of controlled object online and adjusting the packet drop probability accordingly, STR can minimize the variance between the transient queue length and the reference queue length, and eliminate the limitation of configuring AQM algorithms according to some specific network conditions. The results of the extensive simulations indicate that STR can keep the queue length at low level and achieve high link utilization simultaneously even when the network conditions change widely.

References

1. Braden, B., et al.: Recommendations on Queue Management and Congestion Avoidance in the Internet. RFC2309, (1998)
2. Floyd, S., Jacobson, V.: Random Early Detection Gateways for Congestion Avoidance. IEEE/ACM Transactions on Networking, Vol.1, (1993) 397-413
3. Firoiu, V., Borden, M.: A Study of Active Queue Management for Congestion Control. In Proceedings of IEEE INFOCOM 2000, Tel-Aviv, Israel (2000) 1435-1444
4. Misra, V., Gong, W.B., Towsley, D.: Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED. In Proceedings of ACM SIGCOMM 2000, Stockholm, Sweden (2000) 151-160
5. David, L., Steven, L.: Random Early Marking for Internet Congestion Control. In Proceedings of IEEE Globecom1999, Rio de Janeiro, Brazil (1999) 1747-1752
6. Hollot, C., Misra, V., Towsley, D., Gong, W.B.: A Control Theoretic Analysis of RED. In Proceedings of IEEE INFOCOM 2001, Anchorage, Alaska, USA (2001) 1510-1519
7. Hollot, C., Misra, V., Towsley, D., Gong, W.B.: On Designing Improved Controllers for AQM Routers Supporting TCP Flows. In Proceedings of IEEE INFOCOM2001, Anchorage, Alaska, USA (2001) 1726-1734
8. Ren, F.Y., Lin, C.: Speed up the Responsiveness of Active Queue Management System. IEICE Transactions on communication, Vol.2, (2003) 630-636
9. Zhang, H.Y., Liu, B.H., Dou, W.H.: Design of a Robust Active Queue Management Algorithm Based on Feedback Compensation. In Proceedings of ACM SIGCOMM2003, Karlsruhe, Germany (2003) 277-286
10. Ren, F.Y., Lin, C., et al.: A Robust Active Queue Management Algorithm Based on Sliding Mode Variable Structure Control. In Proceedings of INFOCOM2002, New York, USA (2002) 64-79
11. Ren, F.Y., Ren, Y., Shan, X.M.: Design of a Fuzzy Controller for Active Queue Management. Computer Communications, Vol.25, (2002) 874-883
12. Wu, W., Ren, Y., Shan, X.M.: A Self-configuring Proportional-Integral Controller for AQM Routers Supporting TCP-like Flows. In Proceedings of 7th Asia Pacific Conference on Communications. Tokyo, Japan (2001) 368-371
13. Wu, G.Y.: System Identification and Adaptive Control. Harbin Institute of Technology Press, Harbin, China (1987) (in Chinese)