IEICE Electronics Express, Vol.6, No.22, 1548–1555

A Robust Discrete-Time Controller for delay sensitive applications

Mahdi Alinaghizadeh Ardestani[a] **and Mohammad T. H. Beheshti**
Electrical Faculty, Tarbiat Modares University,
Tehran, Iran
a) *ardestani@modares.ac.ir*

Abstract: Real time network applications require effective mechanisms for managing Internet traffic in order to avoid or at least limit the level of congestion. We propose, design and analyze an Active Queue Manager (AQM) for controlling jitter using robust control theory. Here, we use an output feedback controller and this controller is based on discrete-time model of TCP. The natural properties of robust controller help us decreasing effect of model uncertainties and undesirable flows on performance of traffic control. The proposed Robust Discrete-Time Controller (RDTC) ensures the maintenance of the router's queue length at the desired value. We used *ns2* for simulation of controller performance and the results obtained verify the validity of the algorithm.

Keywords: network congestion control, robust control, output feedback, disturbance attenuation, jitter
Classification: Electronic instrumentation and control

References

[1] S. Floyd and K. Fall, "Random early detection gateways for congestion avoidance," *IEEE/ACM Trans. Networking*, Aug. 1993.
[2] B. Braden, *et al.*, "Recommendations on queue management and congestion avoidance in the internet," *IETF RFC 2309*, April 1998.
[3] B. Wydrowski and M. Zukerman, "GREEN: an active queue management algorithm for a self managed Internet," *in Proc. IEEE International Conference on Communications ICC '2002*, vol. 4, pp. 2368–2372, April 2002.
[4] S. Athuraliya, S. H. Low, V. H. Li, and Y. Qinghe, "REM: active queue management," *IEEE Network Mag.*, vol. 15, no. 3, pp. 48–53, 2001.
[5] F. Wu-chang, K. G. Shin, D. D. Kandlur, and D. Saha, "The BLUE active queue management algorithms," *Proc. IEEE/ ACM Trans. Networking*, vol. 10, pp. 513–528, 2002.
[6] S. Jinsheng, K.-T. Ko, C. Guanrong, S. Chan, and M. Zukerman, "PD-RED: to improve the performance of RED," *IEEE Commun. Lett.*, vol. 7, pp. 406–408, 2003.
[7] C. V. Hollot, V. Misra, D. Towsley, and G. Weibo, "Analysis and design of controllers for AQM routers supporting TCP flows," *Proc. IEEE Trans. Autom. Control*, vol. 47, pp. 945–959, 2002.

DOI: 10.1587/elex.6.1548
Received August 26, 2009
Accepted October 01, 2009
Published November 25, 2009

IEICE Electronics Express, Vol.6, No.22, 1548–1555

[8] S. Jinsheng, C. Guanrong, K. King-Tim, S. Chan, and M. Zukerman, "PD-controller: a new active queue management scheme," *in Proc. IEEE GLOBECOM'03*, vol. 6, pp. 3103–3107, Dec. 2003.
[9] S. Ryu, C. Rump, and Q. Chunming, "A predictive and robust active queue management for Internet congestion control," *in Proc. ISCC 2003*, vol. 2, pp. 991–998, June 2003.
[10] R. Fengyuan, *et al.*, "Robust active queue management algorithm based on sliding mode variable structure control," *in Proc. INFOCOM'02*, pp. 64–79, June 2002.
[11] C. Wang, B. Li, Y. T. Hou, K. Sohraby, and Y. Lin, "LRED: A robust active queue management scheme based on packet loss ratio," *in Proc. INFOCOM 2004 Conf.*, vol. 1, pp. 1–12, March 2004.
[12] G. Zames and B. A. Francis, "Feedback, minimax sensitivity, and optimal robustness," *IEEE Trans. Autom. Control*, vol. AC-28, pp. 585–600, 1983.
[13] M. Green and D. J. N. Limebeer, *Linear Robust Control*, Prentice-Hall, Englewood Cliffs, NJ, 1995.
[14] "*ns-2* Network Simulator," Obtain via
[Online] http://www.isi.edu/nsnam/ns/

1 Introduction

Many applications use the Internet as a media for transmitting data between data centers and users. Major part of Internet traffic is TCP flows. Therefore, TCP congestion control is very important for network design. Various mechanisms are used by TCP to manage network congestion. These mechanisms are not sufficient to prevent congestion because they detect the congestion after packet drops or changes in throughput or in end-to-end delay.

For improving network congestion management, Active Queue Management (AQM) has been established. The main idea of AQM is to begin packet dropping at the routers' queue, before buffer overflow occurs. A well-known AQM mechanism is Random Early Detection (RED) [1].

Therefore, many active queue management algorithms were proposed not only designed to avoid congestion but also designed to reach many other objectives such as: high link utilization, low drop rate and low packet queuing delay. However, high link utilization and low packet queuing delay are the two main objectives of queue management which often are in conflict.

The present work proposes a robust AQM; and explains how it achieves the mentioned features. The proposed AQM is based on robust discrete-time control theory that naturally has disturbance attenuation and robustness over model uncertainty. This Robust Discrete-Time Controller (RDTC) is designed to increase network performance. Higher link utilization and lower jitter are the main parameters that we defined for network performance.

The rest of the paper is organized as follows: Section 2 gives an overview of some related work on AQM. Section 3 gives an overview of robust control theory and TCP modeling is discussed in section 4. Section 5 presents the performance evaluation and results' analysis through simulation. Finally,

DOI: 10.1587/elex.6.1548
Received August 26, 2009
Accepted October 01, 2009
Published November 25, 2009

conclusions follow in Section 6.

2 General survey on AQM approaches

In this section we present an overall view over AQM mechanisms. Unfortunately, the behavior of Internet shows TCP congestion avoidance mechanisms are not sufficient to prevent congestion. Actually, the main weakness of these mechanisms is in that they drop the packets only in overflows like Drop Tail mechanism [2] but one possible solution to overcome this weakness is to drop packets before overflows. Methods that follow this approach are named Active Queue Management (AQM). Compared to Drop Tail, AQM mechanisms have some advantages like lower delay by maintaining the average queue length and reduction of sequential packet drop.

Recently, multiple AQM mechanisms have been proposed in many papers. In a direct manner, we can classify these mechanisms in four main categories: heuristics methods, control theory based methods, optimization based methods and hybrid methods. For congestion evaluation, each category used some metrics and does reaction such as: queue length, buffer overflow, buffer emptiness, loss ratio, traffic input rates; or a combination of these metrics.

However, high link utilization and low packet queuing delay are the two main objectives of queue management which often are in conflict. In fact, some AQM mechanisms such as GREEN [3] and REM [4] that minimize the queuing delay tolerate low link utilization and packet loss. On the other hand, other AQM mechanisms such as BLUE [5] that maximize the link utilization tolerate high queuing delays. Establish a balance between queuing delay and link utilization is the main philosophy of new AQMs. Therefore, some AQMs such as SC-RED [7], ARED [8], and PD-RED [6] try to maintain the queue length between a minimum and a maximum threshold. Other AQMs such as PI [7] and PD [8] try to track the queue length at a desired reference value. Also PID [9], SMVS [10] and LRED [11] were proposed to avoid buffer overflow and buffer emptiness.

For some traffic such as VoIP which is sensitive to delay, buffer overflow prevention is an important issue. Moreover, for real-time applications, stability in queuing delays is beneficial, because they need to be able to estimate retransmission timeouts. In addition, preventing buffer emptiness permits optimal link utilization.

3 Robust control

There are always differences between mathematical models used for design and the actual system. Also real systems are susceptible to external disturbance. Thus, robustness is most important in control system design. Typically, it is required to design a controller that will stabilize a plant, if it is not stable originally, and satisfy certain performance levels in the presence of disturbance signals, noise interference, plant parameter variations and unmodelled plant dynamics. These design objectives are best realized via the

DOI: 10.1587/elex.6.1548
Received August 26, 2009
Accepted October 01, 2009
Published November 25, 2009

IEICE Electronics Express, Vol.6, No.22, 1548–1555

feedback control mechanism.

The described subjects led to a substantial research effort to develop a theory that could explicitly address the robustness issue in feedback design. The first work in the development of the outgoing theory, now known as the H_∞ optimal control theory, was presented in the early 1980s by Zames and Francis [12]. In the H_∞ approach, the designer right from the beginning specifies a model of system uncertainty, such as additive perturbation and/or output disturbance.

Consider a generalized linear discrete-time system, described by equations

$$\begin{aligned} x_{k+1} &= Ax_k + B_1 w_k + B_2 u_k \\ z_k &= C_1 x_k + D_{11} w_k + D_{12} u_k \\ y_k &= C_2 x_k + D_{21} w_k + D_{22} u_k \end{aligned} \tag{1}$$

Where $x_k \in \Re^n$ is the state vector, $w_k \in \Re^{m_1}$ is the exogenous input vector (the disturbance), $u_k \in \Re^{m_2}$ is the control input vector, $z_k \in \Re^{p_1}$ is the error vector, and $y_k \in \Re^{p_2}$ is the measurement vector. The transfer function matrix of system will be noted by

$$P(z) = \begin{bmatrix} P_{11}(z) & P_{12}(z) \\ P_{21}(z) & P_{22}(z) \end{bmatrix} = \begin{bmatrix} A & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \tag{2}$$

The "H_∞ sub-optimal discrete-time control problem" is to find an internally stabilizing controller $K(z)$ such that, for a pre-specified positive value γ, where

$$\|F_l(P, K)\|_\infty < \gamma \tag{3}$$

Where $F_l(P, K)$ is the lower linear fractional transformation (LFT) on $K(z)$, which is equal to the closed loop transfer function $T_{zw}(z)$ from w to z:

$$T_{zw}(z) = F_l(P, K) = P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21}$$

$K(z)$ is calculated by solving two Riccati Equation [13].

4 TCP discrete time model

Based on queue theory, Hollot et al. constructs a dynamic model of TCP/AQM that can be described by the following coupled nonlinear differential equations [7]:

$$\dot{W}(t) = \frac{1}{R(t)} - \frac{W(t - R(t))}{R(t - R(t))} p(t - R(t)) \tag{4}$$

$$\dot{q}(t) = \begin{cases} -C + \dfrac{N(t)}{R(t)} W(t) & q > 0 \\ \max\left\{0, -C + \dfrac{N(t)}{R(t)} W(t)\right\} & q = 0 \end{cases} \tag{5}$$

Where W is average TCP window size (packets), $R(t) = q(t)/C + T_p$ is round trip time (secs), q is average queue length (packets), T_p is propagation delay

DOI: 10.1587/elex.6.1548
Received August 26, 2009
Accepted October 01, 2009
Published November 25, 2009

(secs), C is link capacity (packets/sec), N is load factor (number of TCP sessions) and p is the probability of packet mark.

Suppose that the number of TCP sessions and link capacity are constant. Taking (W, q) as state variable and p as input, the equilibrium (W_0, q_0, p_0) is then defined by $dW/dt = 0$ and $dq/dt = 0$. So we linearize (4) about the equilibrium, to obtain

$$\delta\dot{W} = -\frac{2N}{R_0^2 C}\delta W(t) - \frac{R_0 C^2}{2N^2}\delta p(t) \quad \text{and} \quad \delta\dot{q} = \frac{N}{R_0}\delta W(t) - \frac{1}{R_0}\delta q(t)$$

Where $\delta W = W - W_0$, $\delta q = q - q_0$, $\delta p = p - p_0$ represent the perturbed variables about the equilibrium. Let $x_1(t) = \delta W(t)$, $x_2(t) = \delta q(t)$ and $u(t) = \delta p(t)$ then (4) and (5) can be described as:

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{6}$$

Where:

$$x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}, \quad A = \begin{bmatrix} -\dfrac{2N}{R_0^2 C} & 0 \\ \dfrac{N}{R_0} & -\dfrac{1}{R_0} \end{bmatrix}, \quad B = \begin{bmatrix} -\dfrac{R_0 C^2}{2N^2} \\ 0 \end{bmatrix},$$
$$-p_0 \le u(t) \le 1 - p_0, \quad 0 \le p_0 \le 1$$

Under maintaining the controllability condition, we change (3) from continuous-time model to discrete-time model as follows:

$$x(k+1) = Gx(k) + Hu(k) \tag{7}$$

Where $G = e^{AT}$, $H = \left(\int_0^T e^{AT}\, dt\right) \times B$ and T is sampling period (Sampling period is the interval of sampling).

5 Simulation

In this section we present the simulation results of Robust Discrete-Time Controller (RDTC) proposed. Also, we compare its performance with RED, PI and PID mechanisms. We verify our propositions via simulations using the *ns2* [14] simulator. The topology that will be used is shown in Fig. 1. The RDTC mechanism is designed for a sample network with 60 TCP sessions, about 240 millisecond end to end delays, 15 Mbps bandwidth for each links

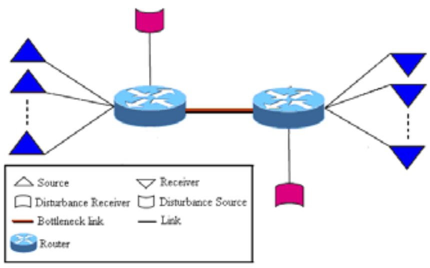

Fig. 1. Simulation Topology

DOI: 10.1587/elex.6.1548
Received August 26, 2009
Accepted October 01, 2009
Published November 25, 2009

IEICE Electronics Express, Vol.6, No.22, 1548–1555

and average packet size is considered to be 500 bytes. The maximum queue level is set to 400 packets.

For this situation A and B in (6) are calculated as follow:

$$A = \begin{bmatrix} -0.5556 & 0 \\ 250 & -4.1667 \end{bmatrix}, \quad B = \begin{bmatrix} -468.75 \\ 0 \end{bmatrix}$$

And then G and H in (7) for a sampling frequency of 170 are:

$$G = \begin{bmatrix} 0.9967 & 1 \\ 4.3518 & 0.9758 \end{bmatrix}, \quad H = \begin{bmatrix} -2.7529 \\ -6.2846 \end{bmatrix}$$

To change network model to (1) we assume that disturbance is as traffic added to queue without any consideration of AQM mechanism. We assume disturbance traffics are CBR traffics that have priority for transmission and have an effect on other TCP flows that are limited to our AQM discipline. Comparing (1) and (7) we have $A = G$, $B_2 = H$, $B_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ and $y_k = \delta q = \begin{bmatrix} 0 & 1 \end{bmatrix} x_k$ is the output of the system.

By solving two Riccati Equations, the H_∞ controller is calculated as below [13]:

$$K(z) = \frac{a + bz + cz^2}{d + fz + z^2}$$

$K(z)$ is depending on the value of γ in (3) and for the purpose of our simulation we put $\gamma = 6$ then controller parameters are calculated as below:

$$a = 0.0000082,\ b = 0.4961954,\ c = 0.3764782,\ d = -0.0000830$$
$$\text{and } f = 0.8584047$$

In this simulation, we evaluate performance of AQM mechanisms when propagation delay increases to 360 ms and disturbance (CBR traffic) applies to network. Notice that we set desired queue length at 250 packets. In this case CBR traffic rate is 10 Mbps with 500 bite cell length and starts at time 0 and stops at time 20, another time CBR starts at 50 and stops at 70 and finally it starts at 80 and continues to 100. Fig. 2 and Fig. 3 illustrate the queue size and bandwidth usage at each control mechanism.

Robustness of RDTC is shown exactly in Fig. 2. Network parameter variation and disturbance attenuation are the two major bases for robust control theory which are satisfied in this simulation. However PI, PID and RED try to preserve queue length in the desired value but their performance are not comparable with RDTC.

Although, PI and PID have good bandwidth utilization in comparison to RED, but RDTC has the best performance. RDTC present a significant disturbance rejection when we change delay as a key parameter in network.

Although in this simulation the propagation delay is set to 360 ms (50% above the design condition), another simulations show that in around 10 times propagation delay RDTC has good performance in throughput, jitter and bandwidth utilization rather than other AQM mechanisms.

DOI: 10.1587/elex.6.1548
Received August 26, 2009
Accepted October 01, 2009
Published November 25, 2009

IEICE Electronics Express, Vol.6, No.22, 1548–1555

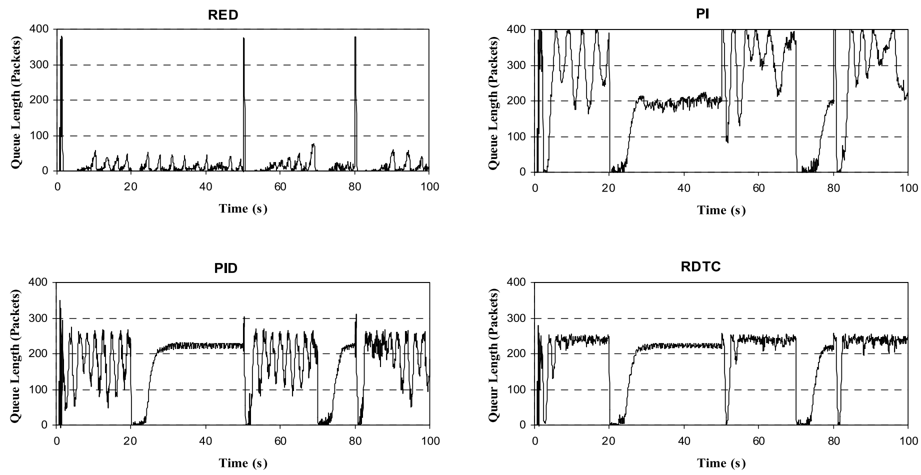

Fig. 2. Bottleneck queue length after applying disturbance with 360 ms delay

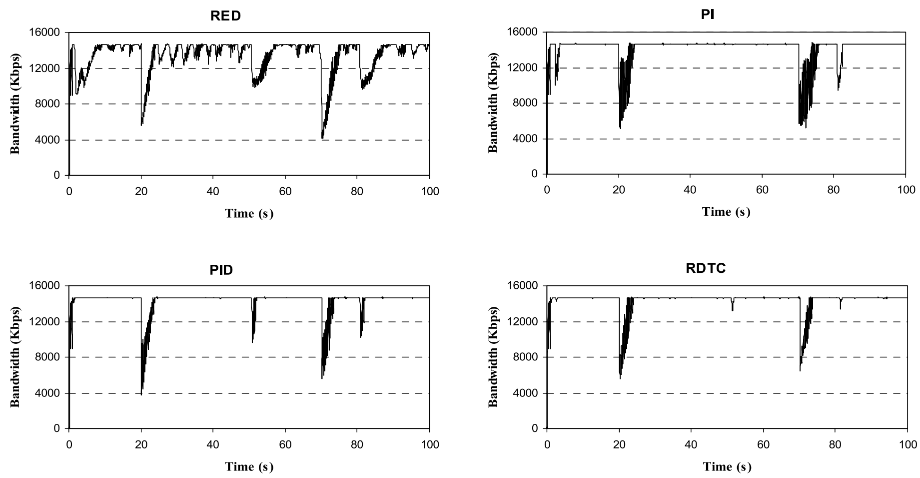

Fig. 3. Bottleneck bandwidth usage after applying disturbance with 360 ms delay

Table I. Network Throughput for each AQM

AQM mechanism	Throughput (Bytes/Sec)
RED	1779636
PI	1807197
PID	1845381
RDTC	1852860

Table I illustrates throughput of each mechanism. It shows the RDTC has the best performance in throughput.

In this simulation we measure end to end delay as a key parameter in network design. The average end to end delay in RED is 106 ms, RDTC is 163 ms, PI is 168 ms and PID is 150 ms. The reason of high average end to end delay in RDTC is that we set desired queue length (Q_{ref}) at 250 for higher resolution in our figures. The experiences show in RDTC the average end to end delay for lower desired queue lengths is better. For example if Q_{ref} set to 50 packets the average end to end delay is 111 ms and if Q_{ref} set to 50 packets the average end to end delay decrease to 103 ms. It shows that

DOI: 10.1587/elex.6.1548
Received August 26, 2009
Accepted October 01, 2009
Published November 25, 2009

IEICE Electronics Express, Vol.6, No.22, 1548–1555

RDTC can satisfy many performance metrics.

Another critical point is that this simulation is implemented over TCP/RENO, but other experiences show RDTC can work with other TCP congestion protocols as well as TCP/Reno.

6 Conclusion

In this paper, we proposed an active queue manager (AQM) based on robust control theory, named "Robust Discreet-Time Controller" (RDTC). The main motivation for designing this controller was the usage of robust control methods and its advantages, such as disturbance attenuation and the decrease of the sensitivity of the output of the system against model changes and uncertainties. Following an intensive performance evaluation, RDTC not only in tracking desired queue length, but also in usage of maximum bandwidth, has shown more robustness against network uncertainties. Furthermore, RDTC does a far better disturbance attenuation comparing to the most AQM algorithms.

Acknowledgments

This research is sponsored in part by Iran Telecommunication Research Center.

DOI: 10.1587/elex.6.1548
Received August 26, 2009
Accepted October 01, 2009
Published November 25, 2009