International Journal of Control and Automation
Vol. 9, No. 11 (2016), pp.77-92
http//dx.doi.org/10.14257/ijca.2016.9.11.07

Delay Based Robust H∞ Control Algorithm for TCP Networks Congestion Control to Disturbance Attenuation

Mahdi Alinaghizadeh Ardestani

Electrical and Computer Engineering Department, Qom University of Technology, Qom, Iran
ardestani@qut.ac.ir

Abstract

Many real-time applications use Internet as a media to transmit their data and need guarantee for correct data transmission and limited transmission delay. It requires efficient network traffic protocols to avoid or limit the level of congestion. In this paper we propose, design and analyze a Robust Delay Based Controller (RDBC) for traffic management on transmission control protocol (TCP). This controller is based on an output feedback controller for decreasing effect of delay and disturbance on performance of network. Linear Matrix Inequality (LMI) is used for optimize design objectives. The simulation results illustrate that the queue length in router reaches to desired value quickly with proposed RDBC. We use network simulator (NS2) to compare performance of RDBC with other AQM mechanisms Like RED, PI and PID. Queue length average and variance, bandwidth utilization, throughput and end-to-end delay are performance metrics that used for comparison.

***Keywords**: Network Congestion Control; Transmission Control Protocol (TCP); Robust Control; Output Feedback; Disturbance Attenuation*

1. Introduction

TCP is the dominant protocol in the Internet and many applications use it, therefore, increase Internet performance is very important. TCP congestion control is one of the key factors to increase Internet performance. There are several mechanisms to manage network congestion such as Slow Start, Additive Increase and Multiplicative Decrease (AIMD), Congestion Avoidance, Fast Retransmit and Fast Recovery are used by TCP. These mechanisms cannot prevent congestion because they detect congestion after packet drops or meaningful changes in end-to-end delay or throughput.

Active Queue Management (AQM) has been established for improving network congestion management. The main idea of AQM is to activate packet dropping at the routers' queue, before buffer overflow happens. The well-known AQM mechanism is Random Early Detection (RED) [1]. In RED, the arriving packets are dropped randomly with a probability that is proportional to its average queue length. The furthermore researches show, RED has some weaknesses, such as sensitivity to network-load variation and short-term and long-term queue behavior mismatch [2].

Many other objectives such as: high link utilization, low drop rate and low packet queuing delay are considered for researches. However, high link utilization and low packet queuing delay are two main objectives of queue management which often in conflict.

In fact, some AQM mechanisms such as GREEN [3], REM [4], or AVQ [5] minimize the queuing delay; tolerate low link utilization and packet loss. Besides, other AQM mechanisms such as BLUE [6] maximize the link utilization, tolerate queuing delays. Establish a balance between queuing delay and link utilization is the main philosophy of new AQMs.

The present work proposes a robust AQM; and explains how it achieves mentioned features. The proposed AQM is based on robust control theory that naturally has disturbance attenuation and robustness over model uncertainty. We use these basic properties of robust control theory to design more efficient AQM.

This Robust Delay-Based Controller (RDBC) is designed to increase network performance. Higher link utilization and lower jitter are two major parameters that we defined for network performance.

The remainder of the paper is organized as follows: Section 2 presents an overview of related researches on AQM. Section 3 introduces TCP modeling and robust control design for time delayed systems is discussed in section 4. In section 5, we design and evaluate a robust AQM controller. Section 6 presents the results analysis through simulation. Finally, conclusions follow in Section 7.

2. General Survey on AQM Approaches

Experimental researches show TCP congestion avoidance mechanisms are not sufficient to prevent congestion. Actually, the main weakness of these mechanisms is that they drop the packets only in overflows like Drop Tail mechanism [7] 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 changes by maintaining the average queue length and reduction of sequential packet drop.

Recently, multiple AQM mechanisms have been proposed in many papers and technical documents. In a direct manner, we can classify these mechanisms in four main categories [8]: the heuristics approaches, control theory - based approaches, optimization-based approaches and hybrid approaches. For congestion evaluation, each category used some metrics and did reaction. These metrics are different such as: queue length, buffer emptiness, buffer overflow, loss ratio, traffic input rates; or a mix of these metrics.

In this section we review AQM approaches and explain the philosophy of each category, the congestion estimation metrics that they used and methods of reaction.

2.1. Heuristics-based Methods

RED (Random Early Detection) is most popular and one of the first AQMs. RED prevents network congestion by controlling the average queue length. For this reason RED has some parameters like, minimum threshold, maximum threshold and probability of packet dropping. When the average is less than the minimum threshold, no packets are dropped. When the average exceeds the maximum threshold, all incoming packets are dropped. Between minimum threshold and maximum threshold, according to a probability function the arriving packets are dropped. This probability function is based on the average queue length.

Unfortunately, stability and efficiency of RED depends strongly on it parameter setting [9]. For this reason, several algorithms proposed to improve the basic RED algorithm (*e.g.*, ARED [10], SCRED [11], BLUE [6], GREEN [3] and AVQ [5]).

2.2. Control Theory-based Methods

Recently, control theoretic methods have been used in congestion control problem. These approaches can control the router queue length as a plant variable and balance between link utilization and queuing delay. Figure 1 illustrates the schematic of control loop and its components. In this figure, Q represents the instantaneous queue length, Q_{ref} represents the reference queue length value and the Process represents a combination of subsystems such as TCP sources, TCP receivers, routers and *etc*.

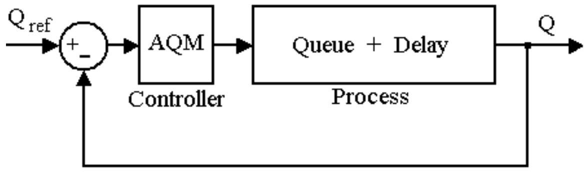

Figure 1. Feedback Control Modeling of Congestion Control with an AQM Algorithm

In [12] the TCP/AQM control system model was linearized and a proportional integral controller (PI) was proposed. There was shown that the PI controller has passable queue delay and link utilization. In other research activities, different improvements were added to the classical PI controller. The PD controller in [13] predicts the queue evolution which makes the control more reactive. The PIP controller in [14] improves the PI controller by adding a feedback compensation to reduce the sensitivity to network parameter changes.

The PID controller as a strong controller in control theory was used in [15] and combines the action of the PI controller and the PD controller. A Self-Tuning PI controller based on pole placement (ST-PI-PP) has been proposed in [16]. In [17], it was introduced a sliding mode variable structure control, that permits a better stability and robustness than a PI controller by taking into account the model uncertainties and the number of active TCP connections.

Some research use H_∞ for AQM design. Robust discrete time controller is proposed in [18] and [19]. In [20] an AQM controller designed with loop shaping approach. Delay is another key parameter in H_∞ that considered in [21]. There are similar approach for AQM design in [22] and [23].

2.3. Optimization-based Methods

Another way to design AQM controllers is using optimization methods. The philosophy in such mechanisms is to find the optimal value of the dropping probability. A tradeoff between the minimization of the queuing delay and the maximization of the throughput is the key point to find this optimal value we have. For example, in REM, beside the queue length metric, the input rate was used as a congestion indicator [4]. This kind of mechanism permits maintaining a low level queue occupation compared to the previous ones, but it often suffers from low link utilization. Another mechanism is SFC that works with same method [24]. Moreover, in [25] an adaptive controller based on mathematical optimization is presented that is proven it has suitable robustness.

2.4. Hybrid Methods

The network model is strongly nonlinear and has many uncertainties. Moreover, many disturbances like non TCP flows are added to network. Therefore, mathematical dynamic modeling and parameter quantization are two major problems in heuristically and control theoretic approaches.

For surmount to these problems, different hybrid mechanisms have been proposed. FIPD is a fuzzy-based AQM that defines a congestion index based on the instantaneous queue length and uses it to indicate the degree of network congestion [26]. Also, adaptive sliding mode controller [27] and fuzzy based controller [28] are categorized in hybrid methods.

3. The Problem Statement and Preliminaries

In [12], a dynamic model of TCP was presented. Under some assumptions, this dynamic model was simplified as follows:

$$\dot{W}(t)=\frac{1}{\frac{q(t)}{C}+T_p}-\frac{W(t-R_0)}{\frac{q(t)}{C}+T_p}p(t-R_0) \tag{1}$$

$$\dot{q}(t)=\begin{cases} -C+\dfrac{N(t)}{\frac{q(t)}{C}+T_p}W(t) & q>0 \\ \max\left\{0,-C+\dfrac{N(t)}{R(t)}W(t)\right\} & q=0 \end{cases} \tag{2}$$

Where W is average TCP window size(packets), q is average queue length(packets), C link capacity (packets/sec), $R(t)=q(t)/C+T_p$ (secs) is round trip time, T_p is propagation delay (secs), N is load factor (number of TCP sessions) and p is mark probability of packet mark.

Suppose: $N(t)\equiv N$, $C(t)\equiv C$ and taking (W,q) 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 that:

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

$$\dot{q}=0\Rightarrow W_0=\frac{R_0C}{N};R_0=\frac{q_0}{C}+T_P$$

then

$$\begin{aligned}\delta\dot{W}&=-\frac{N}{R_0^2C}\left(\delta W(t)+\delta W(t-R_0)\right)-\frac{1}{R_0^2C}\left(\delta q(t)+\delta q(t-R_0)\right)-\frac{R_0C^2}{2N^2}\delta p(t-R_0)\\ \delta\dot{q}&=\frac{N}{R_0}\delta W(t)-\frac{1}{R_0}\delta q(t)\end{aligned} \tag{3}$$

Where $\delta W=W-W_0$, $\delta q=q-q_0$, $\delta p=p-p_0$ represent the perturbed variables about the equilibrium. We can change (3) to standard form of state space representation of a time delay system (5) if we set $x_1=\delta W$, $x_2=\delta q$ and $u=\delta p$:

$$\begin{aligned}\dot{x}(t)&=Ax(t)+A_d x(t-R_0)+B_h u(t-R_0)\\ z(t)&=Ex(t)\end{aligned} \tag{4}$$

with

$$A=\begin{bmatrix}-\frac{N}{R_0^2C} & -\frac{1}{R_0^2C}\\ \frac{N}{R_0} & -\frac{1}{R_0}\end{bmatrix}\quad A_d=\begin{bmatrix}-\frac{N}{R_0^2C} & \frac{1}{R_0^2C}\\ 0 & 0\end{bmatrix},\quad B_h=\begin{bmatrix}\frac{R_0^2C}{2N^2}\\ 0\end{bmatrix}$$

4. Robust Control for Time Delayed System

Robust control is a branch of control theory that explicitly deals with uncertainty in its approach to controller design. Robust methods aim to achieve robust performance and stability in the presence of bounded modeling errors and disturbance. Due to the inherent nature of the Internet, that has many uncertainties such as number of users, end to end delay and other parameters, we choose robust control to override these uncertainties.

Consider a class of time delay system of the form

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) + A_d x(t-d) + B_h u(t-h) + Dw(t) \\ z(t) &= Lx(t)\end{aligned} \tag{5}$$

Where $t \in \Re$ is the time, $x(t) \in \Re^n$ is the state vector, $w(t) \in \Re^q$ is the exogenous input vector (the disturbance), $u(t) \in \Re^m$ is the control input vector, $z(t) \in \Re^p$ is the output and d , h respected the amount delay in the state and at the input of the system.

As regards, there are delays in computer networks; the model that describes in (5) is more suitable for computer network dynamic model. The purpose of this research is to design a controller that despite the input delay and state delay, the final closed-loop system to be stable.

Given the time-delay system (5) with the output measurement $y(t) = Cx(t)$, now consider the problem of output feedback control by using a state observer-based control scheme. There are two state variables (queue length and window length) in (3). Note that the controller is implemented within the routers; the queue length parameter can be measured. Then state observer is necessary because window length must be estimated.

Let the state-observer be described by:

$$\begin{aligned}\dot{\xi}(t) &= A_0\xi(t) + B_0[y(t) - \xi(t)] + C_d\xi(t-d) + C_h\xi(t-h) \\ u(t) &= G_0\xi(t)\end{aligned} \tag{6}$$

Where $\xi(t)$ is internal state variable of observer dynamic and the matrices A_0, B_0, C_h, G_0, C_d must be determined in design process. The targets of this controller design are: a good estimation of $x(t)$ by $\xi(t)$ and minimize $T_{zw}(s)$ (the transfer function of disturbance to output). Note that in (6) output of observer based controller is $u(t)$, the input of system (5).

Introducing the variables

$$e(t) := \xi(t) - x(t);\ x_a(t) := \begin{bmatrix} x^T(t) & e^T(t) \end{bmatrix}^T \tag{7}$$

Then the closed-loop system corresponding to (5) and (6)-(7) is given by the state model:

$$\begin{aligned}\dot{x}_a(t) &= A_a x_a(t) + B_a x_a(t-d) + C_a x_a(t-h) + D_a w_a(t) \\ z(t) &= L_a x_a(t)\end{aligned} \tag{8}$$

With

$$T_{zw}(s) = E_a\left[(sI - A_a) - \left(B_a e^{-ds} + C_a e^{-hs}\right)\right]^{-1} D_a \tag{9}$$

Let the matrices A_a, C_d, C_h be defined by

$$\begin{aligned}A_0 &= A + BG_0 + \gamma^{-2}DD^T P - L^{-1}G_0^T B^T P \\ C_a &= A_d; C_h = B_h G_0\end{aligned} \tag{10}$$

Such that

$$\begin{aligned}A_a &= \begin{bmatrix} A + BG_0 & BG_0 \\ A_0 - BG_0 - A & A_0 - BG_0 - B_0 C \end{bmatrix}; \\ B_a &= \begin{bmatrix} A_d & 0 \\ 0 & A_d \end{bmatrix}; \\ C_a &= \begin{bmatrix} B_h G_0 & B_h G_0 \\ 0 & 0 \end{bmatrix}; D_a = \begin{bmatrix} D \\ -D \end{bmatrix}; E_a = \begin{bmatrix} L & 0 \end{bmatrix}\end{aligned} \tag{11}$$

This describes a free time-delay system. The following theorem summarizes the desired result:

Theorem: The closed-loop system (9) is asymptotically stable and $\|T_{zw}\| \leq \gamma \,; \gamma > 0$ for d , $h \geq 0$ if there exist matrices $0 < Y^T = Y \in \mathrm{R}^{n\times n}$, $0 < X^T = X \in \mathrm{R}^{n\times n}$, $0 < Q_{dd}^T = Q_{dd} \in \mathrm{R}^{n\times n}$, $0 < Q_{hh}^T = Q_{dd} \in \mathrm{R}^{n\times n}$, $0 < Q_{dh}^T = Q_{dh} \in \mathrm{R}^{n\times n}$, $0 < Q_{hd}^T = Q_{hd} \in \mathrm{R}^{n\times n}$, $0 < Q_u^T = Q_u \in \mathrm{R}^{n\times n}$, $S_0 \in \mathrm{R}^{m\times n}$, $M_0 \in \mathrm{R}^{n\times s}$ and the scalar $\varphi > 0$ satisfying the LMIs:

$$W_1 = \begin{bmatrix} \Theta_1(Y) & A_d Y & \overline{B}_h S_0 & M \\ Y A_d^T & -Q_{dd} & 0 & 0 \\ S_o^T \overline{B}_h^T & 0 & -\overline{Q}_{dh} & 0 \\ M^T & 0 & 0 & -J_d \end{bmatrix} < 0 \tag{12}$$

And

$$W_2 = \begin{bmatrix} \Theta_2(X) & X N_0 & A_d X & \overline{D} \\ N_0^T X & -I & 0 & 0 \\ X A_d^T & 0 & -Q_u & 0 \\ \overline{D} & 0 & 0 & -U \end{bmatrix} < 0 \tag{13}$$

Where

$$\Theta_1(Y) = AY + YA^T + BS_0 + S_0^T B^T + Q_{dd} + Q_{hd}$$
$$A_e = A_0 + \gamma^{-2} D D^T Y^{-1}$$
$$\Theta_2(X) = A_e X + X A_e^T + Q_u + Q_{dd}$$
$$\overline{B}_h = \begin{bmatrix} B_h & B_h \end{bmatrix},\ \overline{Q}_{dh} = \begin{bmatrix} Q_{dh} & Q_{hd} \end{bmatrix},\ \overline{D} = \begin{bmatrix} D & C^T M_0^T \end{bmatrix}$$
$$N_0 N_0^T = \begin{bmatrix} \varphi I & Y^{-1}\left(BS_0 + S_0^T B^T\right) Y^{-1} \end{bmatrix}, U = \begin{bmatrix} \gamma^2 I & \varphi^{-1} I \end{bmatrix} \tag{14}$$

Moreover, the observer-based feedback controller is given by

$$\begin{aligned} \dot{\xi}(t) &= \left[A + BS_0 Y^{-1} + \gamma^{-2} D D^T Y^{-1} - X G_0^T B^T Y^{-1}\right]\xi(t) \\ &- M_0\left[y(t) - C\xi(t)\right] + A_d \xi(t-d) + B_h S_0 Y^{-1} \xi(t-h) \end{aligned} \tag{15}$$

$$u(t) = S_0 Y^{-1} \xi(t) \tag{16}$$

Proof: this theorem is proven in [29].

5. Controller Design

In this section, we design the Robust Delay-Based Controller (RDBC) and evaluate the performance and effectiveness of the controller and compare its performance with other congestion control mechanisms. We verify our propositions via simulations using the *ns2* [30] simulator.

We assume that there is a single bottlenecked router. We consider 5 scenarios for illustration of weakness and strength of each mechanism.

The RDBC mechanism is designed for a network with 60 TCP sessions, 100 millisecond RTT delay for a link with bandwidth 15Mbps and average packet size consider be 500 bytes. Topology of mentioned network is illustrated in Figure 2.

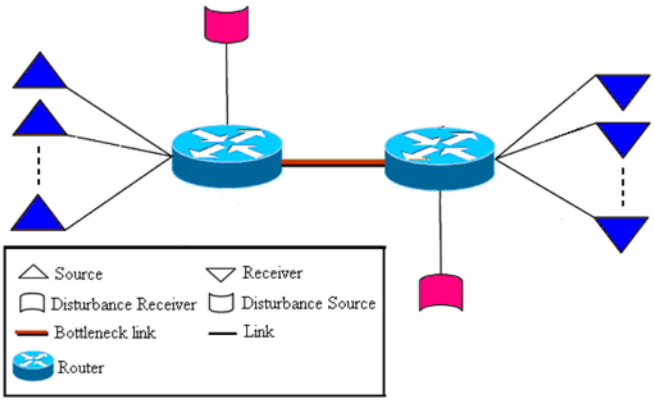

Figure 2. Simulation Topology

For this situation A, A_d, B_h and E in (5) are calculated as blew:

$$A=\begin{bmatrix}-1.6 & -0.0266\\ 600 & -100\end{bmatrix},\quad A_d=\begin{bmatrix}-1.60 & -0.02666\\ 0 & 0\end{bmatrix},\quad B_h=\begin{bmatrix}-195.312\\ 0\end{bmatrix}\quad \text{and}$$

$$E=D=\begin{bmatrix}0\\ 1\end{bmatrix}$$

Consider to equations (12), (13) and solving the LMIs, we have:

$$Y=\begin{bmatrix}0.0005 & 0.0061\\ 0.0061 & 7.8078\end{bmatrix},\ \gamma=0.0155 \text{ and } S_0=\begin{bmatrix}0.0013 & 0.1523\end{bmatrix}\times10^{-6}$$

$$X=\begin{bmatrix}0.0015 & 0.0615\\ 0.0615 & 6.0468\end{bmatrix}\times10^{8}$$

And then observer that was described in (15) and (16) modeled as below:

$$\dot{\xi}(t)=\begin{bmatrix}-0.0016 & -0.0000\\ 6.9121 & -0.5457\end{bmatrix}\times10^{3}\xi(t)-\begin{bmatrix}0.2034\\ 0.2034\end{bmatrix}\left[y(t)-\begin{bmatrix}0 & 1\end{bmatrix}\xi(t)\right]+\begin{bmatrix}-1.6005 & 0.0267\\ 0 & 0\end{bmatrix}\xi(t-d)$$

$$u(t)=\begin{bmatrix}0.2365 & 0.0018\end{bmatrix}\times10^{-5}\xi(t)$$

To implement this observer based controller in *ns2*, we have to discretize the observer model. For this reason bilinear transformation was used. Suppose T (sampling Time) is 1/60 second and $s=\frac{2}{T}\frac{z-1}{z+1}$ and $z=e^{sT}$ then the controller is:

$$K(z)=\frac{a_0+a_1z+a_2z^2+a_6z^6+a_7z^7+a_8z^8}{b_0+b_1z+b_2z^2+b_6z^6+b_7z^7+b_8z^8} \tag{17}$$

Where:

$a_0=-0.00051$, $a_1=-0.001135$, $a_2=-0.00062$, $a_6=3.402\times10^{-7}$, $a_7=-1.815\times10^{-7}$, $a_8=-5.217\times10^{-7}$

$b_0 = 1186$, $b_1 = 2757$, $b_2 = 1571$, $b_6 = -1.162 \times 10^5$, $b_7 = -3.001 \times 10^5$, $b_8 = 1.437 \times 10^5$

5.1. Stability Analysis

Stability analysis of a closed loop network with AQM has been discussed in [12] via some proposition. Consider the linear control system in Figure 3 where $P(s)$ is network transfer function, $C(s)$ represents a linear AQM controller and $\Delta(s)$ is high-frequency window dynamic like as below:

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

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

In [12] is a shown closed loop stability condition, via some proposition. Suppose V(s) were defined as follows:

$$V(s) = \frac{P(s)}{1 + P(s)C(s)e^{-sR_0}}$$

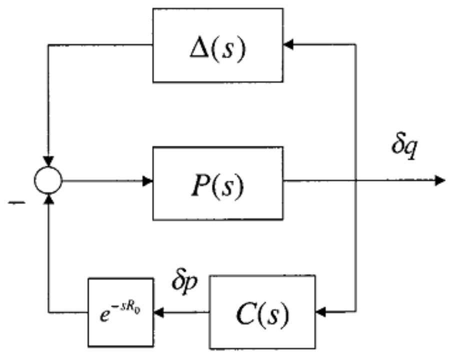

Figure 3. AQM Closed Loop Diagram

Illustrated system in Figure 3 is stable if:

- $C(s)$ stabilizes the delayed nominal plant $P(s)e^{-sR_0}$
- the high-frequency parasitic $\Delta(s)$ is gain-stabilized, *i.e.*, $|\Delta(j\omega)V(j\omega)| < 1; \forall \omega > 0$

Continues form of designed controller (17)(16) can be calculated as below:

$$K(s) = \frac{0.00028394(s^2 + 49.58s + 666.6)(s^2 + 44.21s + 1060)(s^2 + 37.3s + 2371)}{(s + 26.53)(s + 10.94)(s + 1.935)(s^2 + 27.04s + 968.5)(s^2 + 28.86s + 2839)}$$

With considering network parameters as R=250ms, N=60 and C=1250, we can calculate $P(s)$ and $\Delta(s)$ as below:

$$P(s) = \frac{13020.8333}{(s+1.536)(s+4)}, \quad \Delta(s) = 5.898 \times 10^{-5}\left(1 - e^{-0.25s}\right)$$

Error! Reference source not found. and Figure 5 illustrate bode diagram of open-loop transfer function $L(s) = P(s)K(s)$ and $|\Delta(j\omega)V(j\omega)|$.

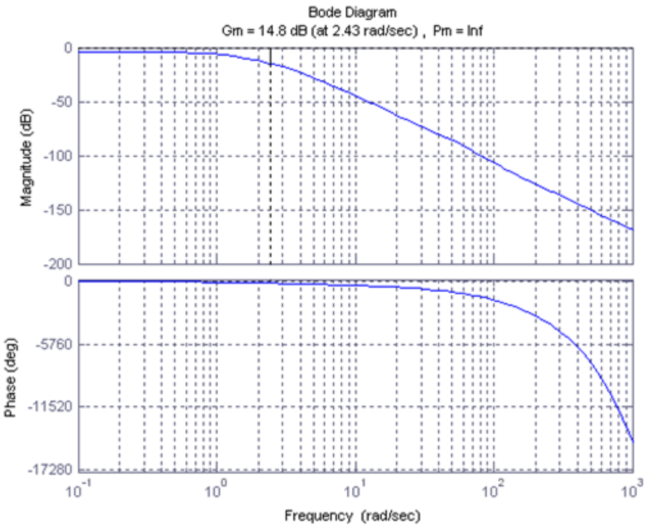

Figure 4. Bode Diagram of L(S).

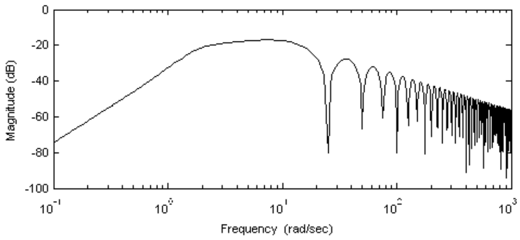

Figure 5. Bode Diagram of $|\Delta(j\omega)V(j\omega)|$

As it seems in **Error! Reference source not found.** and Figure 5 all of stability conditions are satisfied.

5.2. Sensitivity Analysis

In this section, sensitivity of total system respect to each basic parameters of network is evaluated. In the next figures (Figure 6 to Figure 8), it can be seems that RDBC can stabilize network with large range network parameter variation. RDBC can guarantee network stability when propagation delays increase to around 500ms (two times over design condition), link capacity increase to 10Mb/s and network sessions decrease to 18 sessions.

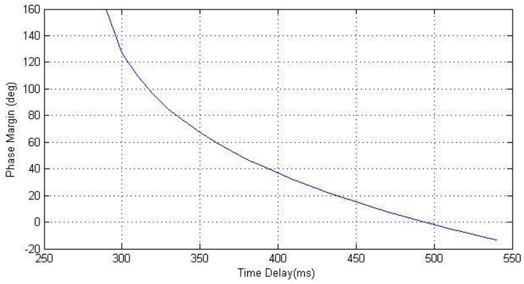

Figure 6. Phase Margin Diagram of Open Loop System For Delay Change

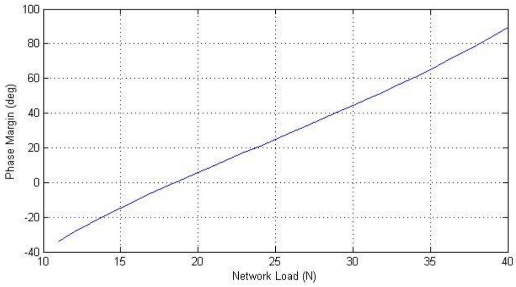

Figure 7. Phase Margin Diagram of Open Loop System for Network Load Change

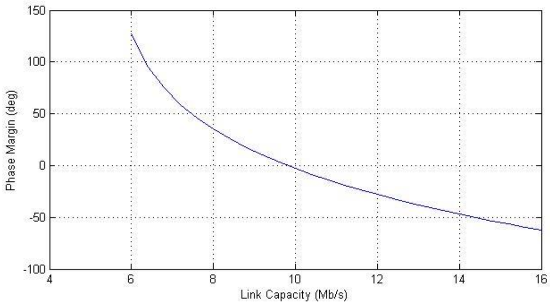

Figure 8. Phase Margin Diagram of Open Loop System for Link Capacity Change

6. Simulation

For evaluate performance of RDBC, there are three scenarios. These scenarios show robustness of RDBC against delay variation and different disturbances apply to network. The topology of simulation network is like Figure 2.

6.1. First Scenario

In the first scenario, it is assumed, the link delay changes from 120ms to 440ms and other conditions are like nominal conditions. Queue lengths corresponding to each link delay are illustrated in Figure 9 to Figure 12. In these simulations, the desired queue length is set to 250 packets for RDBC, PI and PID.

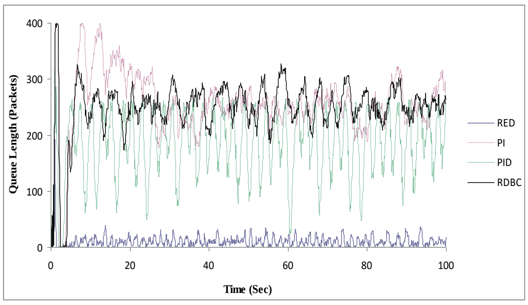

Figure 9. Queue Length For 120ms End To End Delay

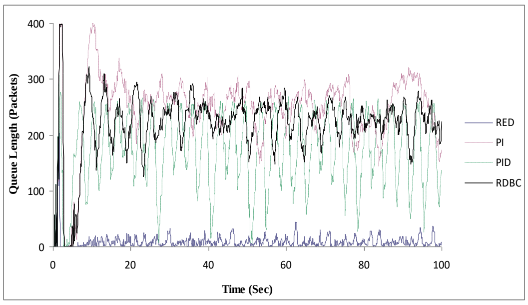

Figure 10. Queue Length For 240ms End To End Delay

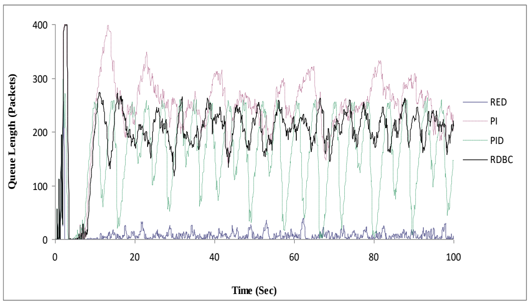

Figure 11. Queue Length For 340ms End To End Delay

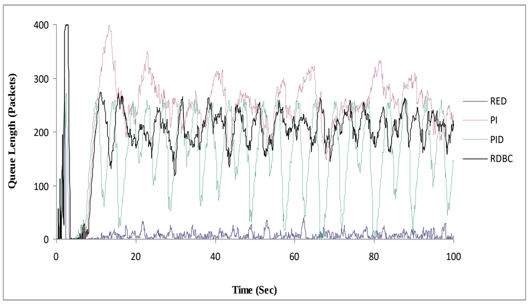

Figure 12. Queue Length For 440ms End To End Delay

In these figures, RDBC has been track the desired queue length. Other performance metrics are listed in Table 1.

Table 1. Performance Metrics for First Scenario

Delay	AQM Mechanisms	Mean Queue (Packets)	variance/Mean	Bandwidth Utilization (Kbps)	Throughput (Bytes/Secs)	End-to-End Delay (sec)
120	**RDTC**	235.51	5.80	4853.67	622128.6	272.74
	RED	14.63	12.21	4730.45	606329.4	76.57
	PI	267.21	11.39	4855.61	622384.4	301.14
	PID	201.69	346.88	4858.78	622795	242.23
240	**RDTC**	248.27	10.49	4807.19	615801.4	345.85
	RED	10.88	16.17	4709.36	603292	132.98
	PI	256.04	14.16	4816.10	616965.8	353.31
	PID	184.02	19.82	4826.61	618270.6	287.58
340	**RDTC**	218.25	15.72	4762.57	609754.4	370.63
	RED	9.20	24.86	4577.06	586012.4	181.72
	PI	245.77	19.11	4762.52	609754.6	396.96
	PID	170.44	26.98	4807.39	615528.2	326.19
440	**RDTC**	196.33	18.92	4724.03	604554.6	402.32
	RED	8.32	37.12	4468.60	571851	231.18
	PI	240.00	23.63	4742.39	606874.4	441.26
	PID	155.73	38.27	4745.07	607233	365.14

As it is shown in Table 1, RDBC tolerate network parameters changes, and stability of system is guaranteed.

6.2. Second Scenario

In this scenario, to evaluate disturbance rejection strength of each controller, we apply a CBR traffic as a disturbance to bottleneck. In this situation, CBR traffic rate is 1.5Mbps with 500 bite cell length. CBR traffic starts at first epoch of simulation and stops after 20 seconds, another time CBR starts at 50 and stops at 70 and finally it starts at 80 and continues until 100. The propagation delay changes from 120ms to 440ms. Figure 13 to Figure 16 illustrate the queue size at each propagation delay.

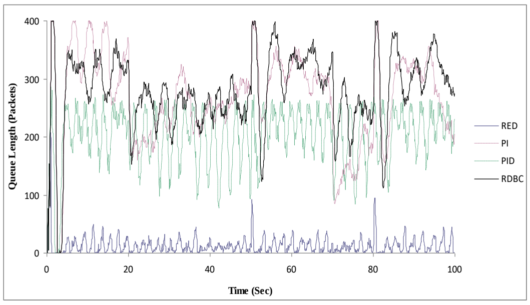

Figure 13. Queue Length for 120ms End to End Delay with 1.5 Mbps Disturbance

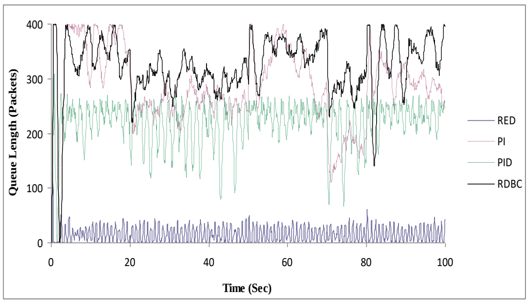

Figure 14. Queue Length for 240ms end to End Delay with 1.5Mbps Disturbance

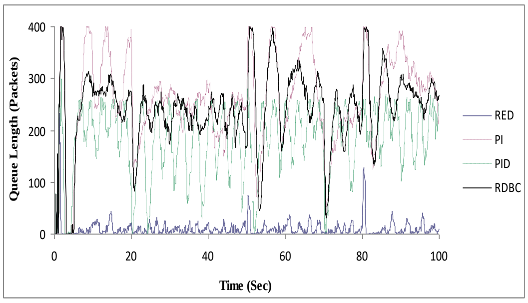

Figure 15. Queue Length For 340ms End To End Delay With 1.5Mbps Disturbance

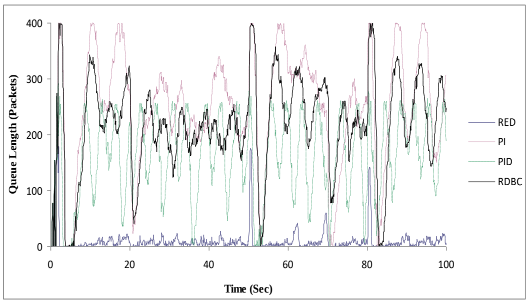

Figure 16. Queue Length For 440ms End To End Delay With 1.5Mbps Disturbance

In these figures, RDBC has an acceptable performance. Other performance metrics are listed in Table 2.

Table 2. Performance Metrics for Second Scenario

Delay	AQM Mechanisms	Mean Queue (Packets)	variance/Mean	Bandwidth Utilization (Kbps)	Throughput (Bytes/Secs)	End-to-End Delay (sec)
120	**RDTC**	323.26	9.49	4863.98	512909.2	341.32
	RED	15.48	12.59	4773.30	510829.2	76.57
	PI	285.28	17.98	4864.91	513251.4	304.87
	PID	218.35	8.00	4867.15	515131.4	251.30
240	**RDTC**	279.25	16.19	4847.89	509926.2	363.74
	RED	12.96	20.00	4697.97	494286.6	134.43
	PI	265.83	23.43	4843.19	508836.4	349.64
	PID	201.56	13.24	4836.74	509331	298.85
340	**RDTC**	242.48	22.38	4821.47	505356	384.15
	RED	10.75	30.46	4634.02	483199.8	182.74
	PI	259.39	28.95	4817.80	505030.8	397.90

	PID	181.80	25.36	4810.41	505120.8	332.18
440	**RDTC**	215.78	35.37	4789.53	501023.8	413.16
	RED	9.96	52.69	4440.70	457367.8	232.35
	PI	245.54	43.34	4806.82	503412.6	437.03
	PID	160.34	36.02	4761.83	497792.6	367.60

6.3. Third Scenario

In this scenario, the CBR traffic is decreased. In this situation CBR traffic rate is set to 3Mbps with 500 bite cell length. Again CBR traffic starts at first epoch of simulation and stops after 20 seconds, another time CBR starts at 50 and stops at 70 and finally it starts at 80 and continues until 100. The propagation delay changes from 120ms to 440ms. Figure 17 to Figure 20 illustrate the queue size and bandwidth utilization at each propagation delay.

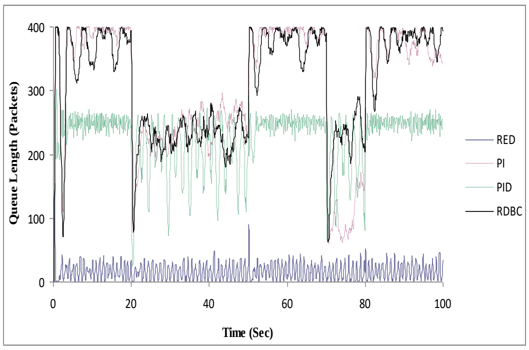

Figure 17. Queue Length for 120ms end to End Delay with 3Mbps Disturbance

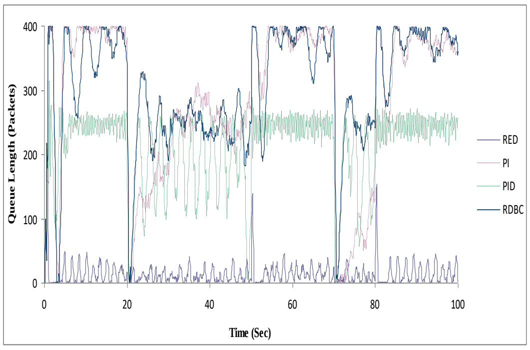

Figure 18. Queue Length for 240ms end to End Delay with 3Mbps Disturbance

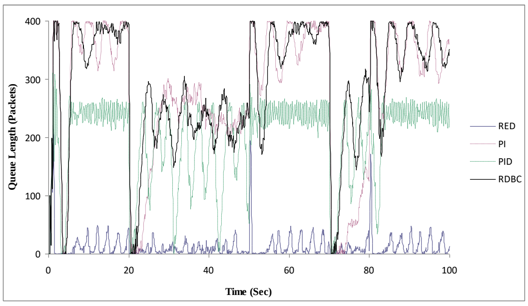

Figure 19. Queue Length for 340ms End To End Delay With 3Mbps Disturbance

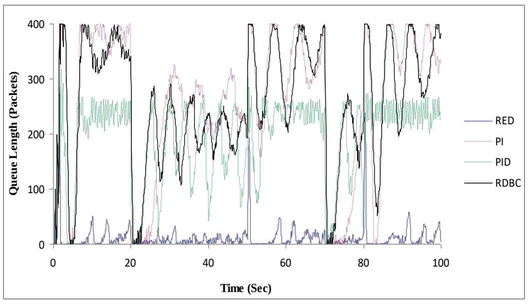

Figure 20. Queue Length for 440ms End To End Delay With 3Mbps Disturbance

Table 3. Performance Metrics for First Scenario

Delay	AQM Mechanisms	Mean Queue (Packets)	variance/Mean	Bandwidth Utilization (Kbps)	Throughput (Bytes/Secs)	End-to-End Delay (sec)
120	**RDTC**	311.46	21.63	4870.74	409394.2	305.08
	RED	16.16	12.18	4798.01	423882.2	f76.70
	PI	306.46	33.26	4870.77	408921.6	295.22
	PID	226.01	8.23	4870.70	418378.2	249.91
240	**RDTC**	305.98	26.22	4861.50	509926.2	364.84
	RED	13.95	24.22	4760.37	405243	134.29
	PI	288.85	50.04	4857.18	404353.8	337.25
	PID	212.30	16.65	4862.62	413382.2	294.84
340	**RDTC**	287.02	38.68	4827.56	399620.8	397.69
	RED	12.79	38.51	4658.71	386364.2	183.61
	PI	273.84	55.19	4814.11	397731.4	379.78
	PID	198.53	25.73	4836.53	406359.2	335.05
440	**RDTC**	256.75	47.83	4814.54	397277.8	423.60
	RED	12.22	55.75	4565.03	370010	233.18
	PI	259.22	66.81	4759.41	388997.6	425.48
	PID	189.37	30.37	4802.62	398323	378.52

7. Conclusions

In this paper, we proposed an active queue manager (AQM) based on robust control theory, named “Robust Delay Based Controller” (RDBC). The main motivation for designing this controller was utilization of robust control and its advantages, such as robustness against model uncertainties and disturbance attenuation. In computer networks and Internet, there are many uncertainties such as number of users and networks load, that changes of them can unstable the network. Considering delay in input and state of network model, and design a controller based on these delays, guarantees that closed loop system with existence of delay, is stable. Following an intensive performance evaluation, RDBC not only in desired queue length tracking, but also in usage of maximum bandwidth and decrease end-to-end delay, has shown more robustness against networks parameters changes. Furthermore, RDBC does a far better disturbance attenuation comparing to the most AQM algorithms.

References

[1] S. Floyd and V. Jacobson, “Random Early Detection Gateways for Congestion Avoidance,” IEEE/ACM Transaction on Networking, vol. 1, no. 4, **(1993)**, pp. 397-413.
[2] T. Eguchi, H. Ohsaki and M. Murata, “On control parameters tuning for active queue management mechanisms using multivariate analysis”, in: Proceedings of Symposium on Applications and the Internet, January, **(2003)**, pp. 120–127.
[3] B. Wydrowski and M. Zukerman, “GREEN: an active queue management algorithm for a self-managed Internet”, in: Proceeding of IEEE International Conference on Communications ICC’2002, vol. 4, **(2002)**, pp. 2368–2372.
[4] S. Athuraliya, S. H. Low, V. H. Li and Y. Qinghe, “REM: Active queue management”, Network, IEEE, vol.15, no.3, **(2001)**, pp. 48-53.
[5] S.S. Kunniyur and R. Srikant, “An adaptive virtual queue (AVQ) algorithm for active queue management”, IEEE/ACM Transactions on Networking, vol. 12, no. 2, **(2004)**, pp. 286–299
[6] W. Feng, D. Kandlur, D. Saha and K. Shin, “BLUE: A New Class of Active Queue Management Algorithms”, Tech. Rep., UM CSE-TR-387-99, **(1999)**.
[7] B. Braden, “Recommendations on queue management and congestion avoidance in the internet”, IETF RFC 2309, **(1998)**.
[8] Y. Hadjadjaoul, A. Mehaoua and C. Skianis, “A fuzzy logic-based AQM for real-time traffic over internet”, Computer Networks, No. 51, **(2007)**, pp. 4617–4633

[9] H. Ohsaki, M. Murata and H. Miyahara, "Steady state analysis of the RED gateway: stability, transient behavior, and parameter setting", in: IEICE Transactions on Communications E85-B, **(2002)**, pp. 107–115.
[10] S. Floyd, R. Gummadi and S. Shenker, "Adaptive RED: An algorithm for increasing the robustness of RED's Active Queue Management", http://www.icir.org/floyd/papers/adaptiveRed.pdf, **(2001)**.
[11] W.-C. Feng, D.D. Kandlur, D. Saha and K.G. Shin, "A selfconfiguring RED gateway", in: Proceedings of INFOCOM'99 Conference, vol. 3, **(1999)**, pp. 1320–1328.
[12] C.V. Hollot, V. Misra, D. Towsley and G. Weibo, "Analysis and design of controllers for AQM routers supporting TCP flows", IEEE Transactions on Automatic Control, vol. 47, no.6, **(2002)**, pp.945–959.
[13] S. Jinsheng, C. Guanrong, K. King-Tim, S. Chan and M. Zukerman, "PD-controller: a new active queue management scheme", in: Proceedings of IEEE GLOBECOM'03, vol. 6, **(2003)**, pp. 3103–3107.
[14] Z. Heying, B. Liu and D. Wenhua, "Design of a robust active queue management algorithm based on feedback compensation", in: Proceedings of ACM/SIGCOMM, **(2003)**, pp. 277–285.
[15] S. Ryu, C. Rump and Q. Chunming, "A predictive and robust active queue management for Internet congestion control", in: Proceedings of ISCC 2003, vol. 2, **(2003)**, pp. 991–998.
[16] C. Qiang and O.W.W. Yang, "A ST-PI-PP controller for AQM router", IEEE International Conference on Communications, **(2004)**, pp. 2277–2281.
[17]R. Fengyuan, "Robust active queue management algorithm based on sliding mode variable structure control", in: Proceedings of INFOCOM'02, **(2002)**, pp. 64–79.
[18] M. A. Ardestani and M. T H. Beheshti, "A robust discrete-time controller for delay sensitive applications", IEICE Electron Express, vol. 6, no. 22, **(2009)**, pp. 1548-1555.
[19] Cire Sall, El Houcine and Chakir El Aloui, "Design of a robust digital controllers for congestion control in Internet", Simulation Modeling Practice and Theory, no. 19, **(2011)**, pp. 301-313.
[20] S. M. Mahdi Alavi and M. J. Hayes, "Robust active queue management design: A loop-shaping approach", Computer Communications, vol.32, no.2, **(2009)**, pp. 324-331.
[21] S. Manfredi, M. di Bernardo and F. Garofalo, "Design, validation and experimental testing of a robust aqm control", Control Engineering Practice, vol.17, no.3, **(2009)**, pp. 394-407.
[22] J. Yuanwei, H. Wang, G. M. Dimirovski and L. Xiaoping, "Robust h-infinity control for uncertain time-delay tcp/aqm network system", 47th IEEE Conference on Decision and Control, **(2008)**, pp. 1428-1433.
[23] C.-K. Chen, Y.-C. Hung, T.-L. Liao and J.-J. Yan, "Design of robust active queue management controllers for a class of tcp communication networks", Information Sciences, vol.177, no.19, **(2007)**, pp. 4059-4071.
[24] Y. Gao and J.C. Hou, "A state feedback control approach to stabilizing queues for ECN-enabled TCP connections", in: Proceedings of INFOCOM 2003, vol. 3, **(2003)**, pp. 2301–2311.
[25] J. Wang, L. Rong and Y. Liu, "A robust proportional controller for aqm based on optimized second-order system model", Computer Communications, vol.31, no.10, **(2008)**, pp. 2468-2477.
[26] F. Yanfei, R. Fengyuan and L. Chuang, "Design of an active queue management algorithm based fuzzy logic decision", in: Proceedings of ICCT 2003 Conference, vol. 1, **(2003)**, pp. 286–289.
[27] W. Hongwei, M. Bo, J. Yuanwei and L. Xiaoping, "An adaptive fuzzy sliding mode control for aqm systems", in American Control Conference, vol. 2008, **(2008)**, pp. 4445-4450.
[28] W. Liu, S. Zhang, M. Zhang and T. Liu, "A fuzzy-logic control algorithm for active queue management in ip networks", Journal of Electronics, vol.25, no.1, , **(2008)** pp. 102-107.
[29] M. Mahmoud, "Robust Control and Filtering for Time-Delay System", Marcel Dekker, New York, **(2000)**.
[30] "ns-2 Network Simulator," Obtain via http://www.isi.edu/nsnam/ns/.

Author

Mahdi Alinaghizadeh Ardestani, he is currently a professor in Department of Electrical and Computer Engineering at Qom University of Technology. He received B.S. degree from KNTU in 2000, the M.S. and Ph.D. degrees from Tarbiat Modares University in 2003 and 2009 respectively. His current research interests are network control, robotics and multi agents.