Controller Tuning for Active Queue Management Using a Parameter Space Method*

Murat Sağlam†, Sami Ezercan†, Suat Gümüşsoy‡ and Hitay Özbay‡

Abstract

In recent years, different mathematical models have been proposed for widely used internet control mechanisms. Simple low order controllers (such as PID which is easy to implement) are desired for network traffic management. In this paper, we discuss tuning of these linear controllers by using a parameter space method, which computes stability regions of a class of quasi-polynomials in terms of free controller parameters and PID controller parameters are found by minimization of mixed sensitivity function in stability region.

1 Introduction

Several active queue management (AQM) schemes supporting transmission control protocol (TCP) exist in literature [1, 2, 3, 4, 5]. Since simple low order controllers are desired in implementation, design of such controllers is considered in [2, 4, 6, 7, 8, 9] for AQM. In [2], the TCP avoidance mode is modeled by delay differential equations with nonlinearity and PI controller proposed for control mechanism. Although the controller design guarantee some robustness for parametric uncertainties, the high frequency dynamics are considered as parasitic. We will take into account the plant structure and design the PID controller without simplification (except linearization of TCP).

In this paper, tuning of these linear controllers are discussed such that overall system ensures robust stability and good performance. By using a parameter space method [10], we compute stability regions of a class of quasi-polynomials in terms of free controller parameters [11]. We find the optimal PID controller parameters by minimization of mixed sensitivity function in stability region.

The outline of paper as follows: In section 2, the mathematical model of AQM scheme and the linearized plant to be controlled is given [2, 6]. The stable region of PID parameters

*This work was supported in part by the National Science Foundation under grant ANI-0073725

†Department of Electrical and Electronics Engineering, Bilkent University, Bilkent, Ankara, TR 68000, Turkey; *email:* saglamt@ug.bilkent.edu.tr, sami@ug.bilkent.edu.tr

‡Department of Electrical Engineering, The Ohio State University, Columbus, OH 43210, USA *email:* gumussos@ee.eng.ohio-state.edu, ozbay@ee.eng.ohio-state.edu

achieving stable close loop is found and optimal parameter search resulting robust stability and good performance is proposed in section 3. Section 4 gives the simulation results and discusses robustness and performance of controller. The paper ends with section 5, concluding remarks.

2 Mathematical Model of AQM Scheme Supporting TCP Flows

We assume that the network configuration is same as in [2, 6], i.e., network has a single router receiving N TCP flows. We consider a model for the congestion avoidance mode only. TCP slow start and time out mechanisms are ignored. For the rest of the section, we reviewed the mathematical model for AQM Scheme in [6].

When one TCP flows interacting for single router, additive-increase multiplicative-decrease behavior of TCP has been modeled in [1] by difference equation

$$dW(t) = \frac{dt}{R(t)} - \frac{W(t)}{2} dN(t) \tag{2.1}$$

where $R(t) = \frac{q(t)}{C} + T_p$ and other variables are defined as:

$$\begin{aligned} q(t) &\doteq \text{queue length at router,} \\ W(t) &\doteq \text{congestion window size,} \\ R(t) &\doteq \text{round trip time delay,} \\ dN(t) &\doteq \text{number of marks the flow suffers,} \\ T_p &\doteq \text{propagation delay,} \\ C &\doteq \text{router's transmission capacity.} \end{aligned}$$

For N homogeneous TCP sources and one router, nonlinear model of AQM implementation is given in [2] as

$$\dot{W}(t) = \frac{1}{R(t)} - \frac{W(t)}{2}\frac{W(t-R(t))}{R(t-R(t))} p(t-R(t)) \tag{2.2}$$

$$\dot{q}(t) = \left[\frac{N(t)}{R(t)} W(t) - C\right]^+ \tag{2.3}$$

where $p(t)$ is the probability of packet mark used by AQM mechanism at the router and

$$[x]^+ \doteq \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases} .$$

The linearization of (2.2) and (2.3) about the operating point (R_0, W_0, p_0) is proposed in [2]. The operating point is defined by the derivations in [2] as

$$R_0 = \frac{q_0}{C} + T_p, \tag{2.4}$$

$$W_0 = \frac{R_0 C}{N}, \tag{2.5}$$

$$p_0 = \frac{2}{W_0^2}. \tag{2.6}$$

Note that linearization of (2.2) and (2.3) ignores the time-varying nature of the round-trip time delay $(t - R(t))$ and approximates as $(t - R_0)$.

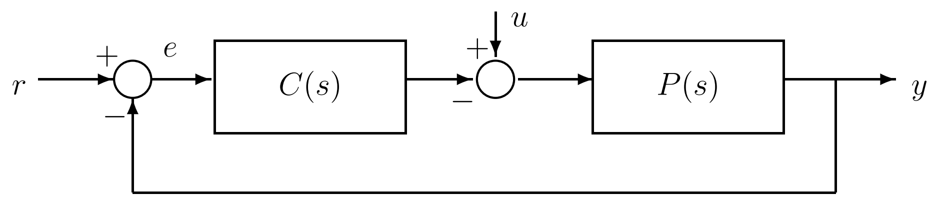


Figure 2. Overall system

The linearized model of the plant is given in [2] as follows:

$$P(s) = \frac{N_p(s)}{D_p(s)} = \frac{K e^{-R_0 s}}{W_0 R_0^2 s^2 + (W_0 + 1) R_0 s + 2 + R_0 s e^{-R_0 s}}. \tag{2.7}$$

where $K = \frac{N W_0^3}{2}$. Given the plant, P, a PI controller, C, is proposed in [2] and $\mathcal{H}^\infty$ controller is suggested in [6] for the feedback loop in Figure 2. We will consider in this paper a PID controller which has a better performance than a PI controller and simple structure for implementation compared to $\mathcal{H}^\infty$ controller. A method for tuning of PID parameters will be given in the next sections. Also, the high frequency dynamics of system (the delay term at the denominator of P) is considered as parasitic in PI controller design [2]. However, we will include these effects in the design of PID controller.

3 Tuning of PID Parameters

In this section, we will find the controller parameter space such that closed loop system is stable by using parameter space method [10]. This approach separates parameter space of PID controller into *stable* and *unstable* region. The stability of region is checked by direct method in [11]. The rest of section discusses finding optimal parameters in controller parameter space by numerical search algorithm. The optimal parameters achieves the minimum of mixed sensitivity function in admissible parameter values.

3.1 Finding PID Parameter Space

For the plant, P, the closed loop system in Figure 2 is stabilized by the PID controller,

$$C(s) = K_P + K_D s + \frac{K_I}{s}. \tag{3.8}$$

The triplet (K_P, K_I, K_D) stabilizes the overall system if and only if all the dominant roots of closed loop characteristic equation,

$$\begin{aligned} CE(s) &= sD_p(s) + (K_I + K_P s + K_D s^2)N_P(s) \\ &= (W_0 R_0 s^3 + (W_0+1)R_0 s^2 + 2s) + (KK_I + KK_P s + (KK_D + R_0)s^2)e^{-R_0 s}, \end{aligned} \tag{3.9}$$

lies on left hand side of the complex plane. The algorithm in [10] offers a parameter space approach to certain class of quasi-polynomials in the form of

$$G(s) = (r_0 + r_1 s + r_2 s^2)A(s) + B(s)e^{sL}, \quad L > 0 \tag{3.10}$$

where A and B are polynomials with degrees m and n respectively satisfying $n \geq m + 2$. It computes the stable (r_0, r_1, r_2) regions. It is not difficult to form the quasi-polynomial as the characteristic equation of our time delayed system in (3.9) as

$$\begin{aligned} r_0 &= KK_I, \\ r_1 &= KK_P, \\ r_2 &= KK_D + R_0, \\ A(s) &= 1, \\ B(s) &= W_0 R_0 s^3 + (W_0+1)R_0 s^2 + 2s, \\ L &= R_0. \end{aligned} \tag{3.11}$$

As explained in [10], a stable quasi-polynomial will be unstable only when a left half plane root transients to right half plane. Since K_P, K_D, K_I and h change continuously, characteristic equation also changes continuously. Thus, for some (K_P, K_D, K_I) triplet, some roots of (3.9) lie on imaginary axis. From these (K_P, K_D, K_I), we can form the stability boundaries in the parameter space. In [10], these crossings are classified into 3 cases and for our problem the boundaries can be found as:

1. *Real Root Boundary* (RRB), a root crosses imaginary axis at origin, i.e., $G(0) = 0$, the boundary is $r_0 = 0$ line, equivalently, $K_I = 0$.

2. *Infinite Root Boundary* (IRB) when a root crosses the imaginary axis at infinity, since $m = 0$ and $n = 3$, the quasi-polynomial is retarded type $(n > m + 2)$ and no infinite root boundary exists.

3. *Complex Root Boundary* (CRB) when a pair of complex conjugate roots crosses the imaginary axis, i.e., $G(j\omega) = 0$, then we can separate real and imaginary parts as,

$$\begin{bmatrix} 1 & -\omega^2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} r_0 \\ r_2 \end{bmatrix} + \begin{bmatrix} \omega((W_0 R_0^2 \omega^2 - 2)\sin R_0\omega - (W_0+1)R_0\omega\cos R_0\omega) \\ -\omega(W_0 R_0^2\omega^2 - 2)\cos R_0\omega - (W_0+1)R_0\omega\sin R_0\omega + r_1) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

If we fix $r_1 = r_1^*$, the solution of the above system of equation exists only for real zeros of ω_{gi} of

$$g(\omega) = \omega r_1^* + (W_0 R_0^2 \omega^2 - 2)\omega \sin R_0\omega - (W_0 + 1)R_0\omega^2 \cos R_0\omega. \tag{3.12}$$

Each positive zero corresponds a straight line as CRB in (r_2, r_0) plane with equation,

$$r_0 = \omega_{gi}^2 r_2 - \omega_g((W_0 R_0^2 \omega_g^2 - 2)\sin R_0\omega_g - (W_0 + 1)R_0\omega_g \cos R_0\omega_g). \tag{3.13}$$

Stability boundaries (RRB, IRB, CRB) can be found as explained above. For each r_1, we can find a region in (r_2, r_0) plane, and for various r_1 values, we can form a three dimensional region, Π_r. Any triplet in this region as PID parameters (i.e., $(r_0, r_1, r_2) \in \Pi_r$) ensures stable closed loop system. Note that actual PID parameters are not the triplet (r_0, r_1, r_2), but (K_P, K_I, K_D). After the region, Π_r, obtained, we can transform to another region, Π_K, for the triplet (K_P, K_I, K_D) by linear transformation in (3.11). Also, in order to check the inside of the region, Π_K, is stable or not, we used the method in [11], which is not discussed here.

3.2 Computation of Optimal PID Parameters

We aim to find the optimal PID parameters such that robust stability and good performance of closed loop system is guaranteed. It is well known that if a controller gives small $\mathcal{H}^\infty$ norm of mixed sensitivity function, overall system has robust stability and good performance. Since our controller has three parameters, (K_P, K_I, K_D), we can search the optimal triplet, $(K_{P,opt}, K_{I,opt}, K_{D,opt})$ in Π_K such that the mixed sensitivity function,

$$\Psi(K_P, K_I, K_D) = \sup_{\omega\in[0,\infty)} \left\{ |W_1(j\omega)S(j\omega)|^2 + |W_2(j\omega)T(j\omega)|^2 \right\}, \tag{3.14}$$

attains its minimum value. The weight functions W_1 and W_2 are finite dimensional terms as a design parameter for robustness and performance. The sensitivity, S, and complementary sensitivity, T, functions are defined as,

$$\begin{aligned} S(s) &= (1 + P(s)C(s))^{-1} \\ T(s) &= P(s)C(s)(1 + P(s)C(s))^{-1} \end{aligned}$$

where P and C are given in (2.7) and (3.8) respectively. Formally, we can defined the numerical search problem as: *Find the triplet,* $(K_{P,opt}, K_{I,opt}, K_{D,opt})$ *such that*

$$\Psi(K_{P,opt}, K_{I,opt}, K_{D,opt}) \le \Psi(K_P, K_I, K_D), \quad \forall\ (K_P, K_I, K_D) \in \Pi_K, \tag{3.15}$$

the inequality is satisfied.

4 Simulations

We simulated the nonlinear model defined by equations (2.2) and (2.3) for the dynamics of N TCP flows loading a router by using `simulink` and `MATLAB`. The numerical values of simulations are same as in [6]:

- Nominal values known to the controller: $N_n = 50$ TCP sessions, $C_n = 300$ packets/sec, $T_p = 0.2$ sec, by simple calculation $R_{0n} = 0.533$ sec and $W_{0n} = 3.2$ packets. Desired queue length is $q_0 = 100$ packets.

- Real values of plant: $N = 40$ TCP sessions, $C = 250$ packets/sec, $T_p = 0.3$ sec, by simple calculation $R_0 = 0.7$ sec and $W_0 = 4.375$ packets.

The above data will be used to see the performance of overall system. In order to analyze the robustness of closed loop system with respect to variations in the network parameters, the following scenario is considered: outgoing link capacity, C, is a normally distributed random signal with mean 250 packets/sec and variance 50 added to a pulse of period 60sec, amplitude 60 packets/sec. The number of TCP flows N is a normally distributed random signal with mean 45 and variance 30 added to a pulse of period 20 sec and amplitude 10. The propagation delay T_p is a normally distributed random signal with mean 0.8 sec and variance 0.05 sec added to a pulse of period 20 sec and amplitude 0.2 sec. The controllers have the following values known to them: $C = 300$ packets/sec, $N = 50$, $T_p = 0.7$sec and desired queue length is $q_0 = 100$ packets.

4.1 Tuning PID Parameters

For the given network parameters, we can write the characteristic equation from (3.10),

$$\begin{aligned} G(s) &= (r_0 + r_1 s + r_2 s^2)A(s) + B(s)e^{sL} \qquad (4.16) \\ &= (r_0 + r_1 s + r_2 s^2) + (1.706s^3 + 2.239s^2 + 2s)e^{0.533s} \\ &= (819.2K_I + 819.2K_P s + (819.2K_D + 0.533)s^2) + (1.706s^3 + 2.239s^2 + 2s)e^{0.533s} \end{aligned}$$

We will work with (r_0, r_1, r_2) triplet and calculate the PID parameters, (K_P, K_D, K_I), at the end. It is clear that for this plant $m = 0$ and $n = 3$. Also note that $B(s)$ does not contain any constant term. Therefore, we do not encounter any infinite root boundary (IRB) and have always a real root boundary (RRB) which is $r_0 = 0$.

In order to determine complex root boundaries (CRB), we should first decide, over which interval we should sweep fixed r_1. If we acquire for (3.12), considering the values for C, N, W_0, R_0, we obtain Figure 1. The interval, in which maximum number of ω_{gi} is produced, can be better observed when we look in the interval, $\omega \in [0, 120]$ as in Figure 2. As we easily observe from Figure 2 that it is enough to sweep r_1 between $[-2, 8]$ in our problem. Therefore we obtained the boundary lines for stability on the $(r_0, r2)$ plane for each fixed

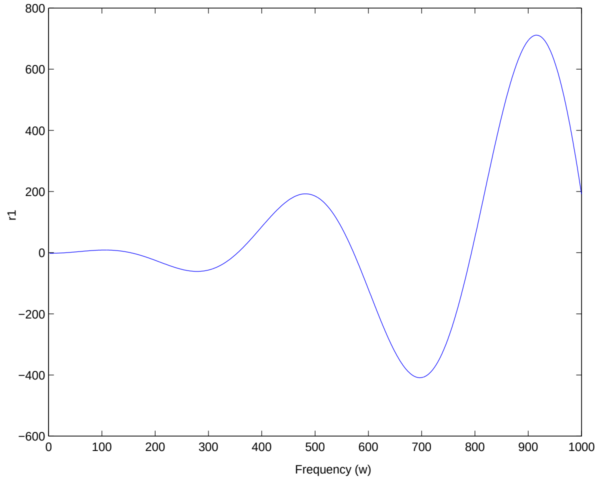


Figure 1: Plot of $g(\omega)$

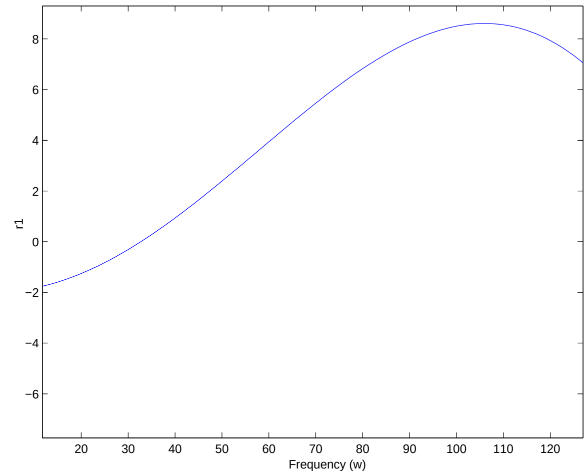


Figure 2: Plot of $g(\omega)$ zoomed

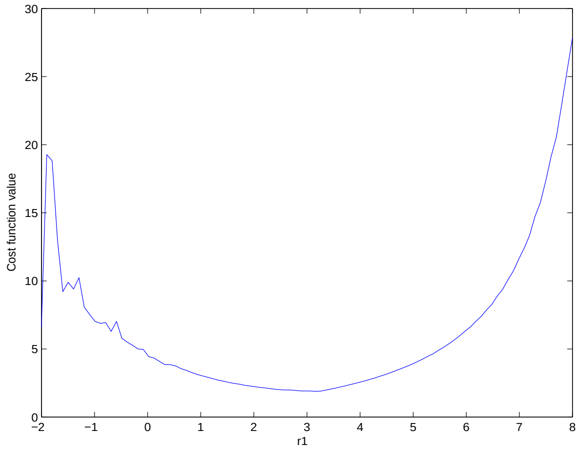


Figure 3: Plot of Ψ

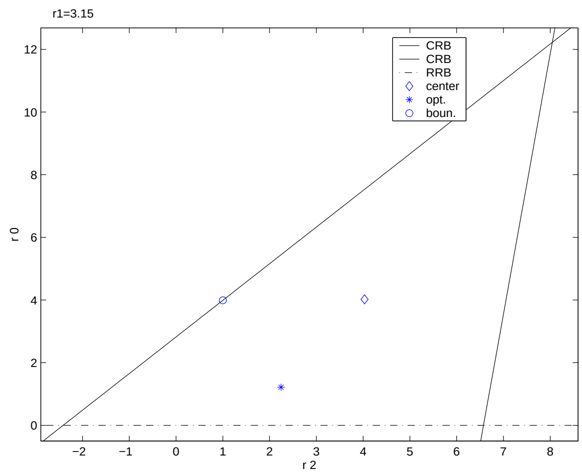


Figure 4: Plot of the region when $r_1^* = 3.15$

r_1. These boundary lines yield a polygon in which we have stability. After sweeping r_1 and combining all polygons, we obtain the stability space for controller parameters.

In order to find the optimal PID parameters, we define the cost function, Ψ, with weight functions,

$$\begin{aligned} W_1(s) &= \frac{1 + 0.01s}{0.01 + s}, \\ W_2(s) &= s + 1. \end{aligned}$$

In Figure 3, for fixed $r_1 \in [-2, 8]$, the minimum value of cost function in (r_0, r_2) plane is given. The minimum value is achieved at $r_{1,opt} = 3.15$. Figure 4 shows the controller parameter space in (r_0, r_2) plane when $r_1 = 3.15$. The optimal point is found as $r_{1,opt} = 3.15$, $r_{2,opt} = 2.2460$ and $r_{0,opt} = 1.2189$. These normalized values correspond to the $K_{P.opt} = 3.845\ 10^{-3}$, $K_{D,opt} = 2.091\ 10^{-3}$ and $K_{I,opt} = 1.48\ 10^{-3}$. The location of optimal, center and one of the boundary points can be seen in Figure 4.

4.2 Performance and Robustness of System

After we found the optimal point, we need to simulate the performance of our controller on the plant presented above. In [6], the performance of $\mathcal{H}^{\infty}$ and PI controller is compared. Using the simulation parameters of [6](given above), we obtained Figure 5, from which the comparison between our PID, $\mathcal{H}^{\infty}$ and PI_1 and PI_2 controllers can be verified.

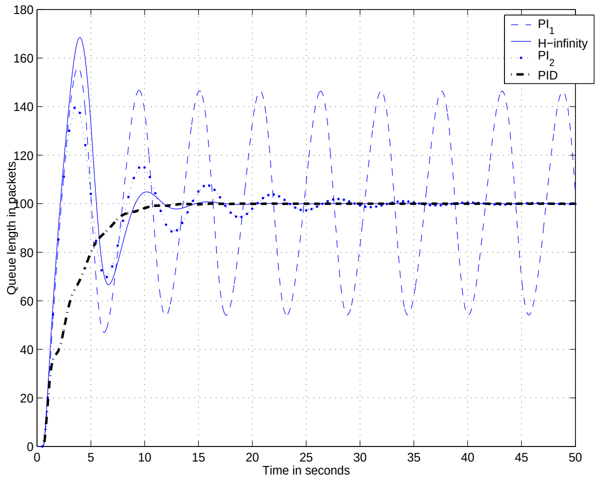


Figure 5: Performance comparison of PID, $\mathcal{H}^{\infty}$, PI_1 and PI_2 controllers

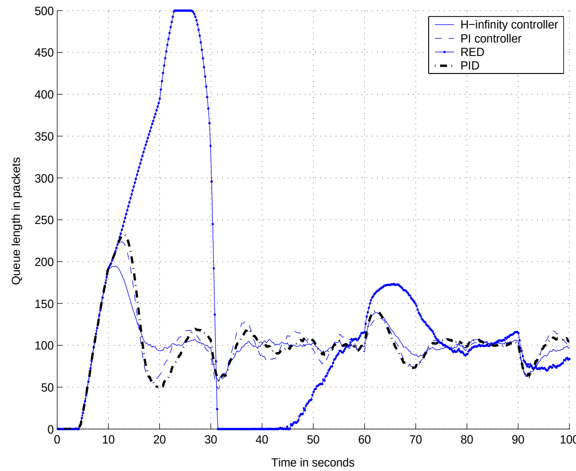


Figure 6: Robustness comparison of PID, $\mathcal{H}^{\infty}$, PI_1 and PI_2 controllers

Figure 5 reveals that PID controller responses better than other controllers. Although rise time is longer, settling time of PID is shorter than the other ones. Also note that there is no overshoot for the proposed PID controller.

For variation in network parameters as shown in Figure 7, robust performance of our global optimum point is obtained in Figure 6. We observe that the PID controller which we design using the method introduced in [10] has similar robust performances with other proposed controllers of [2, 6].

4.3 Remarks

1) Since PID controller design is based on linearization of nonlinear plant, we may encounter different points in the stable space which give us better performance and robustness. For example, in our simulations, the results of a PID controller with parameters $r_1 = 1$, $r_2 = 0.7016$ and $r_0 = 0.839$ ($K_P = 1.221\ 10^{-3}$, $K_D = 2.054\ 10^{-4}$ and $K_I = 1.024\ 10^{-3}$) are given in Figure 8 and 9. It can be seen that the controller has a better settling and rise time with an overshoot. However, the robust performance of optimal point is better than the point ($r_1 = 1$, $r_2 = 0.7016$ and $r_0 = 0.839$).

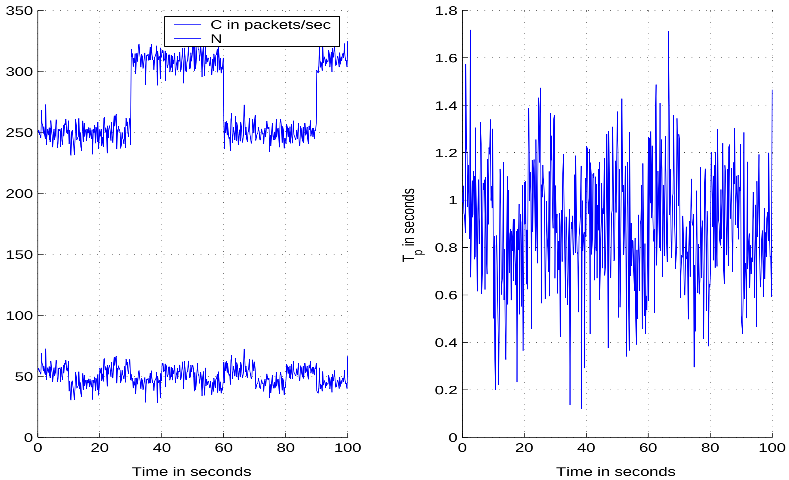


Figure 7: Values of C, N and T_p corresponding to Figure 6

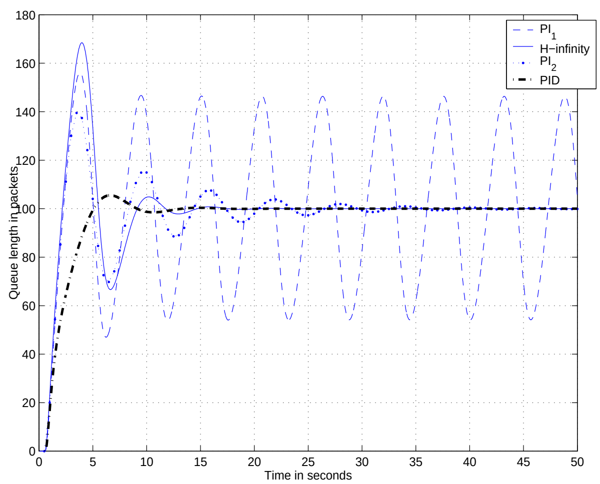


Figure 8: Performance comparison of when $r_1 = 1$, $r_2 = 0.7016$ and $r_0 = 0.839$

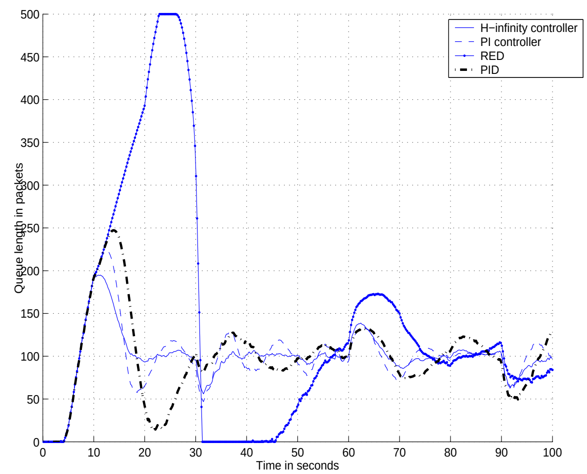
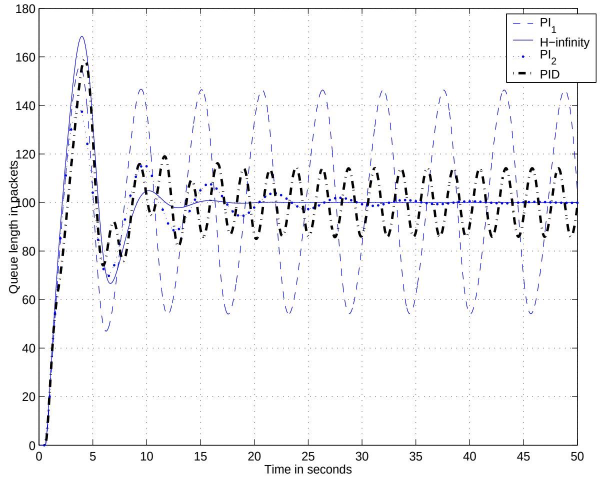


Figure 9: Robustness comparison of when $r_1 = 1$, $r_2 = 0.7016$ and $r_0 = 0.839$

2) For confirmation we did several performance simulations for the points, which lie on the center of the stability polygon, or on the boundary of the stability polygon (shown with diamond and circle symbol in Figure 4), which we think intuitively that, they yield stable and unstable responses, respectively.

Figure 10 and 12 are the response of the center and response of the boundary point respectively for the r1=3.15 polygon. We can see that stability is violated as we move to the boundary which is naturally expected. This violation can also be observed from the robust performance of the boundary in 13. The robust performance difference is very significant when we compare Figure 6 and 13. For the boundary, the robust response in queue length deviates in $[0, 250]$, unlikely for the optimal point, this deviation is in $[50, 140]$.



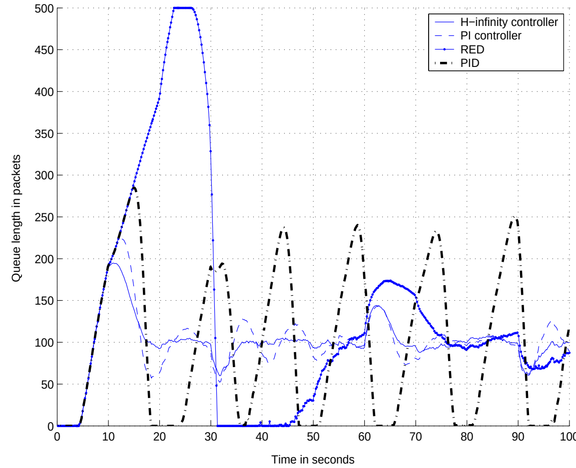
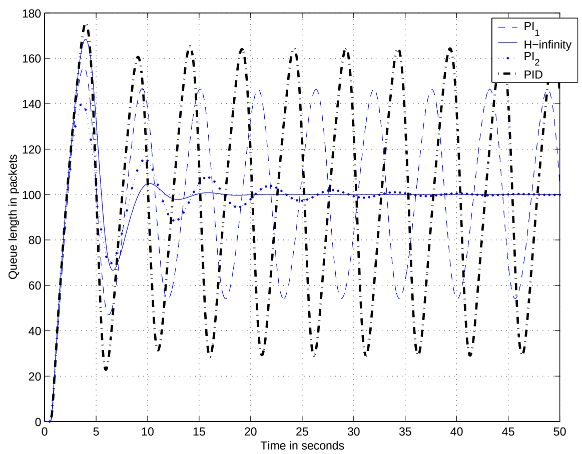


Figure 10: Performance comparison of central point

Figure 11: Robustness comparison of central point



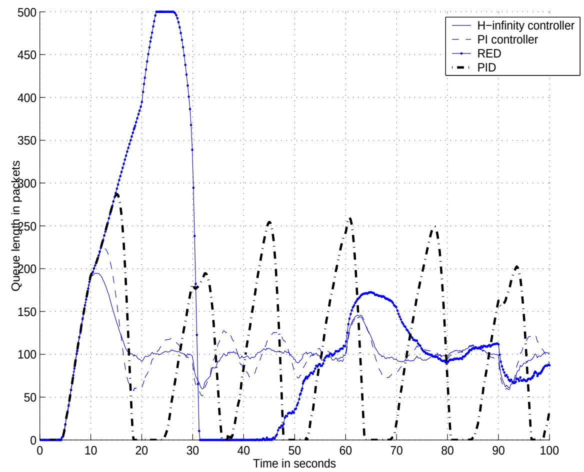


Figure 12: Performance comparison of boundary point

Figure 13: Robustness comparison of boundary point

5 Concluding Remarks

We proposed a PID controller for robust AQM control scheme supporting TCP flows. Tuning algorithm for PID controller is given based on [10, 11] and numerical search algorithm for minimization of mixed sensitivity cost function. We compared our controller performance and robustness with other controllers studied in [2, 6]. For the application on AQM supporting TCP flows, we obtained relatively good performances compared to RED, PI_1 and PI_2 controllers by achieving fast transients and low oscillatory behavior.

References

[1] V. Misra, W. B. Gong, and D. Towsley, "Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED," *Proc. of ACM/SIGCOMM*, pp. 151–160, 2000.

[2] C. V.Hollot, V. Misra, D. Towsley, and W. B. Gong, "Analysis and design of controllers for AQM routers supporting TCP flows," *IEEE Trans. on Auto. Cont.*, vol.47, pp.945–959, June 2002.

[3] F. Kelly, *Mathematical modelling of the Internet" in Mathematics Unlimited - 2001 and Beyond (B. Engquist and W. Schmid, eds.)*, Springer-Verlag, Berlin, 2001.

[4] S. Mascolo, "Modeling the internet congestion control as a time delay system: a robust stability analysis," *Proc. of 4th IFAC Workshop on Time Delay Systems*, Rocquencourt, France, September 2003.

[5] H-F Reynaud, C. Kulcsar, R. Hammi, "In communication networks there are delays, delays, ...and delays," *Proc. of 4th IFAC Workshop on Time Delay Systems*, Rocquencourt, France, September 2003.

[6] P-F. Quet, and H. Ozbay, "On the Design of AQM Supporting TCP Flows Using Robust Control Theory," *Proc. of the 42nd IEEE Conf. on Decision and Cont.*, Maui, Hawaii, December 2003.

[7] Y. Fan, F. Ren, C. Lin, "Design a PID Controller for Active Queue Management," *Proc. of the 8th IEEE Symp. on Computers and Communications*, Kemer, Antalya, Turkey, June-July 2003, pp.985–990.

[8] S. Ryu, C. Rump, C. Qiao, "A Predictive and robust Active Queue Management for Internet Congestion Control," *Proc. of the 8th IEEE Symp. on Computers and Communications*, Kemer, Antalya, Turkey, June-July 2003, pp.991-998.

[9] E. Plasser, T. Ziegler, "An Analytical RED function design guaranteeing stable system behavior," *Proc. of the 8th IEEE Symp. on Computers and Communications*, Kemer, Antalya, Turkey, June-July 2003, pp.1012-1017.

[10] N. Hohenbichler, J. Ackermann, "Computing stable regions in parameter spaces for a class of quasipolynomials," *Proc. of 4th IFAC Workshop on Time Delay Systems*, Rocquencourt, France, September 2003.

[11] N. Olgac, R. Sipahi, "Direct method for analyzing the stability of neutral type LTI time delayed systems," *Proc. of 4th IFAC Workshop on Time Delay Systems*, Rocquencourt, France, September 2003.