**Proceedings of the 40th IEEE
Conference on Decision and Control
Orlando, Florida USA, December 2001**

On the $\mathcal{H}^\infty$ Controller Design for Congestion Control in Communication Networks with a Capacity Predictor [1]

Pierre-François Quet [2] Sthanunathan Ramakrishnan [3] Hitay Özbay [4]
Shivkumar Kalyanaraman [5]

Abstract

In this paper we investigate the use of the outgoing link capacity information in the $\mathcal{H}^\infty$ controller design for rate based flow control in a communication network for the case of a single bottleneck node and a single source. In the previous works in this line of research it was assumed that the controller implemented at the bottleneck node has access to queue length information, and robust controllers were designed for queue management, under time varying time delay uncertainties. Here we assume that, besides the queue information, the controller has access to the outgoing link capacity. On top of the existing robust controller, we use an additional controller term, acting on the capacity information. We investigate optimal ways to design such additional controller term.

1 Introduction

A concern in the design of modern high speed data communication networks is the avoidance of traffic congestion. Flow control, by means of regulating the rate of data packets generated by the sources, aims at achieving this property, thus ensuring a good quality of service to the users of the network. This problem has been studied thoroughly in computer networks and communication literature, see for example [1, 2, 3, 4] and their references.

"Rate-based" flow control has been chosen by the ATM forum [5] as the control scheme in the Asynchronous Transfer Mode switching networks. In this scheme the congestion control is carried out at the bottleneck nodes, which compute and send the feedback signals (assigned rates) to the sources. Several papers in the literature deal with this problem, see for example [6, 7, 8, 9, 10, 11, 12] and their references.

In a recent work [13] an $\mathcal{H}^\infty$ based robust controller was designed solving such rate feedback flow control problem for the case of a network with a single bottleneck node and multiple sources. The controller was robust to uncertain time varying multiple time delays in different channels. The controller used the queue length information provided at the bottleneck node to force the queue length to a desired steady-state value. In the present paper, we explore a way to use the capacity of the bottleneck node (outgoing flow rate) in addition to the queue length information in the design of the controller for the case of a single source. We show that the use of this extra information brings the system to its steady-state more rapidly, and it leads to smaller steady-state tracking error.

2 Problem Formulation

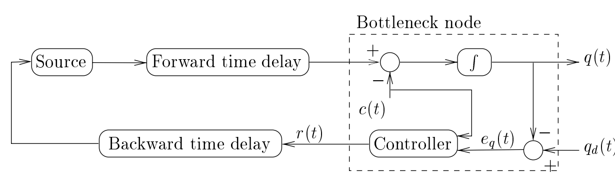

Figure 1: The feedback control system

Consider the single bottlenecked network with one source depicted in Figure 1. Let $q(t) \geq 0$ denote the queue length at the bottleneck node, and $r(t) \geq 0$ be the flow rate assigned by the controller. The capacity, $c(t)$, is the rate at which data is sent out from the node. A simple dynamical model of the system is

$$\dot{q}(t) = r(t-\tau) - c(t) \quad (1)$$

where τ is the return trip time delay between the source and the bottleneck node (addition of forward and backward delays). Assume that the time delay, τ, is constant and known (the discussion can be easily extended to time varying uncertain time delays, as in [13]). First,

[1] This work is supported in part by the National Science Foundation under grant numbers ANI-9806660, ANI-0073725.
[2] Dept. of Electrical Engineering, The Ohio-State University, Columbus, OH 43210, quet.1@osu.edu
[3] Dept. of Electrical, Computer, and Systems Eng., Rensselaer Polytechnic Institute, Troy, NY 12180, sthanu@networks.ecse.rpi.edu
[4] Dept. of Electrical Engineering, The Ohio-State University, Columbus, OH 43210, ozbay@ee.eng.ohio-state.edu
[5] Dept. of Electrical, Computer, and Systems Eng., Rensselaer Polytechnic Institute, Troy, NY 12180, shivkuma@ecse.rpi.edu

consider the single output case:

$$e_q(t) := q_d(t) - q(t)$$

where $q_d(t)$ is the desired queue length, and let the controller, K_1, determines $r(t)$ from $e_q(t)$, i.e.

$$R(s) = K_1(s)E_q(s). \tag{2}$$

Note that the design objectives

(i) robust stability in the presence of delay uncertainty and

(ii) queue regulation (e.g. trying to make $\|e_q\|_2$ small)

can be combined to define a single cost function to be minimized:

$$\gamma(K_1) = \left\| \begin{bmatrix} W_1 S_1 \\ W_2 K_1 S_1 \end{bmatrix} \right\|_\infty$$

where $S_1 = (1 + K_1 P)^{-1}$, $P(s) = \frac{e^{-\tau s}}{s}$ and the weights W_1 and W_2 are determined from the uncertainty magnitude over the frequency range of interest, the weighted tracking error for queue regulation, and the relative importance of these two objectives. See, [14, 15, 13] for details.

Now, we modify the measured output to include the capacity information: $y(t) = \begin{bmatrix} e_q(t) \\ c(t) \end{bmatrix}$. In particular we let

$$R(s) = K_1(s)E_q(s) + K_{\text{new}}(s)C(s). \tag{3}$$

Clearly, in the new setting we can have smaller tracking error by using the capacity information. How much can we reduce the tracking error in the new setting, and what should be the optimal use of $c(t)$ information? More precisely, how do we choose K_{new} for a given fixed K_1 to improve the tracking performance? This is the problem studied in the present paper.

3 Performance Improvement with a Capacity Predictor

Let

$$r(t) = r_o(t) + \widehat{c}(t) \tag{4}$$

where $\widehat{c}(t)$ is a causal estimate of $c(t+\tau)$, and

$$R_o(s) = K_1(s)E_q(s) + K_2(s)E_c(s) \tag{5}$$

with $e_c(t) = c(t) - \widehat{c}(t-\tau)$. This leads to $\dot{q}(t) = r_o(t-\tau) - e_c(t)$. The predictor is a linear time invariant filter, which is part of the feedback controller, i.e.,

$$\widehat{C}(s) = K_c(s)C(s), \tag{6}$$

with the LTI filter K_c being the predictor. Suppose that the desired queue length and capacity are signals that are generated according to

$$C(s) = W_c(s)\widetilde{C}(s) \quad \text{and} \quad Q_d(s) = W_q(s)\widetilde{Q}(s)$$

where $\widetilde{c}(t)$ and $\widetilde{q}(t)$ are external signals (they may be considered as noise or reference generating finite energy signals), and W_c and W_q are proper filters putting a frequency weighting on these external signals. The predictor will be designed to minimize the following $\mathcal{H}^\infty$-cost function

$$J_\infty(K_c) := \sup_{\widetilde{c}\neq 0} \frac{\|e_c\|_2}{\|\widetilde{c}\|_2} = \|W_c(1 - DK_c)\|_\infty \tag{7}$$

where $D(s) = e^{-\tau s}$ is the transfer function of the pure delay system. We will come back to the issue of designing the $\mathcal{H}^\infty$ predictor $K_c(s)$. Next we discuss the design of K_2. Observe that in terms of the external signals the tracking error can be expressed as

$$E_q(s) = S_1(s) \begin{bmatrix} W_q(s) & M(s)\,\frac{1}{s}W_c(s) \end{bmatrix} \begin{bmatrix} \widetilde{Q}(s) \\ \widetilde{C}(s) \end{bmatrix}$$

where $M(s) = (1 - D(s)K_2(s))\,(1 - D(s)K_c(s))$. Note that when we do not use the capacity information in the controller we have $K_c(s) = K_2(s) = 0$, and in that case $M(s) = 1$. Let us define

$$\begin{aligned} S_{\text{old}}(s) &= S_1(s) \begin{bmatrix} W_q(s) & \frac{1}{s}\,W_c(s) \end{bmatrix} \\ S_{\text{new}}(s) &= S_1(s) \begin{bmatrix} W_q(s) & M(s)\,\frac{1}{s}\,W_c(s) \end{bmatrix}. \end{aligned}$$

The $\mathcal{H}^\infty$ cost function to be minimized (for queue tracking) is

$$\sup_{\begin{bmatrix} \widetilde{q} \\ \widetilde{c} \end{bmatrix} \neq 0} \frac{\|e_q\|_2}{\left\| \begin{bmatrix} \widetilde{q} \\ \widetilde{c} \end{bmatrix} \right\|_2} = \|S\|_\infty.$$

In the old setting (where we have feedback from the queue only) $S = S_{\text{old}}$, and in the new setting where we use capacity information as defined above $S = S_{\text{new}}$. Given a frequency weighting $|W_x(j\omega)|$, defining the desired *performance improvement*, we want to design K_2 in such a way that

$$\frac{\|S_{\text{new}}(j\omega)\|}{\|S_{\text{old}}(j\omega)\|} \leq |W_x(j\omega)| \quad \text{for all } \omega. \tag{8}$$

For uniform performance improvement we would like to have $|W_x(j\omega)| < 1$ for all ω. In fact, the smaller the magnitude of W_x the better the new performance. Note that in the limit as $\omega \to \infty$ the ratio in the left hand side of (8) cannot be made strictly less than unity with a proper controller (this is due to the essential norm of the Hankel operator associated with sensitivity minimization problems for delay systems, see [16] for details). In practice we would be interested in sensitivity reduction at low frequencies, and we can live with slightly worse or about the same sensitivity magnitude at mid and high frequency range. In the following we first consider the capacity predictor design corresponding to the problem defined in (7). Then we use this result to design a *desired performance improvement* W_x. Equation (8) then defines an $\mathcal{H}^\infty$ control problem from which K_2 is determined.

3.1 Capacity predictor design

The $\mathcal{H}^\infty$ problem (7)

$$\gamma_c := \inf_{K_c \in \mathcal{H}^\infty} \|W_c(1 - DK_c)\|_\infty \tag{9}$$

with $W_c(s) = \frac{1}{s}$ has the optimal solution (see [16])

$$K_{\text{c,opt}}(s) = \frac{1 + (\frac{2\tau}{\pi})^2 s^2}{\frac{2\tau}{\pi}s + e^{-\tau s}}\ , \tag{10}$$

with

$$\gamma_c = \frac{2\tau}{\pi}.$$

Note that $K_{\text{c,opt}}$ is improper. In order to make the capacity predictor implementable we replace $K_{\text{c,opt}}$ by

$$K_c(s) = \frac{1}{1 + \alpha s}\ K_{\text{c,opt}}(s), \tag{11}$$

where $\alpha > 0$ is a design parameter. It is easy to check that $J_\infty(K_c) = \left(\frac{2\tau}{\pi} + \alpha\right)$. So α needs to be small compared to τ for capacity prediction error to be close to the minimum. But, when we use K_2 for performance improvement, such a choice may not be optimal, as we will see in the next section.

3.2 Performance improvement

We assume that

$$\frac{W_q(s)}{W_c(s)} = \frac{\rho}{1 + \beta s}$$

(where $\rho > 0$ and $\beta > 0$ are given parameters), which means that $c(t)$ may have larger high frequency content relative to $q_d(t)$ and consider W_x to be in the form

$$W_x(s) = \frac{\sqrt{\rho^2 + \delta^2}s}{1 + \sqrt{\rho^2 + \beta^2}s} \text{ for some } \delta > 0. \tag{12}$$

It can be shown that if K_2 satisfies

$$\left\| \left(\frac{\beta s + 1}{\alpha s + 1}\right) \left(1 - e^{-\tau s} K_2(s)\right) \right\|_\infty \leq \frac{\delta}{\frac{2\tau}{\pi} + \alpha} \tag{13}$$

then it also satisfies (8) with W_x defined in (12). Now, let's define

$$\gamma_{\min} = \inf_{K_2 \in \mathcal{H}^\infty} \left\| \left(\frac{\beta s + 1}{\alpha s + 1}\right) \left(1 - e^{-\tau s} K_2(s)\right) \right\|_\infty \tag{14}$$

and

$$\gamma(\delta) = \frac{\delta}{\frac{2\tau}{\pi} + \alpha}.$$

So, for a given set of parameters τ, α, β and ρ, (8) is satisfied if

$$\gamma_{\min} \leq \gamma(\delta). \tag{15}$$

Since we want δ to be as small as possible (to have the best performance improvement à la (12)), we are interested in the smallest δ that guarantees (15), which is

$$\delta_{\min} = \gamma_{\min} \left(\frac{2\tau}{\pi} + \alpha\right).$$

If $\alpha \leq \beta$ then $\frac{\beta s + 1}{\alpha s + 1}$ is a high pass filter and we have (see [16]) $\gamma_{\min} = \frac{\beta}{\alpha}$. In that case, a controller K_2 that achieves $\gamma_{\min}$ is $K_2 = 0$. If $\alpha > \beta$, then (see [16]) the optimal controller K_2 that achieves (14) is

$$K_{2_{\text{opt}}} = \left(\frac{1 - \gamma_{\min}^2 + (\gamma_{\min}^2 \alpha^2 - \beta^2)s^2}{\gamma_{\min}(1 + \beta s)(1 + \alpha s)}\right) \cdot \frac{1}{1 + e^{-\tau s} \frac{1 - \beta s}{\gamma_{\min}(1 + \alpha s)}} \tag{16}$$

with

$$\gamma_{\min} = \sqrt{\frac{1 + \left(\frac{\beta}{\alpha}\right)^2 x_\gamma^2}{x_\gamma^2 + 1}} \tag{17}$$

where x_γ is the unique solution of

$$\left(\frac{\tau}{\alpha}\right) x_\gamma + \tan^{-1}\left(\frac{\beta}{\alpha} x_\gamma\right) + \tan^{-1} x_\gamma = \pi.$$

In order to determine the best value of α, the last design parameter still to be chosen, it is interesting to look at the plot of $\frac{\delta_{\min}}{\beta}$ as a function of $\frac{\beta}{\alpha}$ and $\frac{\tau}{\alpha}$. Note that if $\alpha \leq \beta$, then

$$\frac{\delta_{\min}}{\beta} = \frac{2}{\pi}\frac{\tau}{\alpha} + 1, \tag{18}$$

and if $\alpha > \beta$, then

$$\frac{\delta_{\min}}{\beta} = \frac{\gamma_{\min}}{\frac{\beta}{\alpha}} \left(1 + \frac{2}{\pi}\frac{\tau}{\alpha}\right) \tag{19}$$

where $\gamma_{\min}$ is determined from (17), and it satisfies $\frac{\beta}{\alpha} < \gamma_{\min} < 1$. Figure (2) shows the plot of $\frac{\delta_{\min}}{\beta}$ as a function of $\frac{\beta}{\alpha}$ and $\frac{\tau}{\alpha}$. We clearly see the following compromise for the choice of α in order to have $\frac{\delta_{\min}}{\beta}$ small: α should be chosen large with respect to τ, but it should be chosen small with respect to β. Considering the performance of the capacity predictor (see section 3.1), we see a compromise here. In conclusion, if α is small with respect to τ and β we have $K_2 = 0$, which means that we use K_c only, and in this case the performance improvement is not very good (18). But if a large α value is used compared to β and τ, we have a non zero K_2 compensating for the loss in predictor performance, and achieving a sensitivity performance improvement quantified by (19).

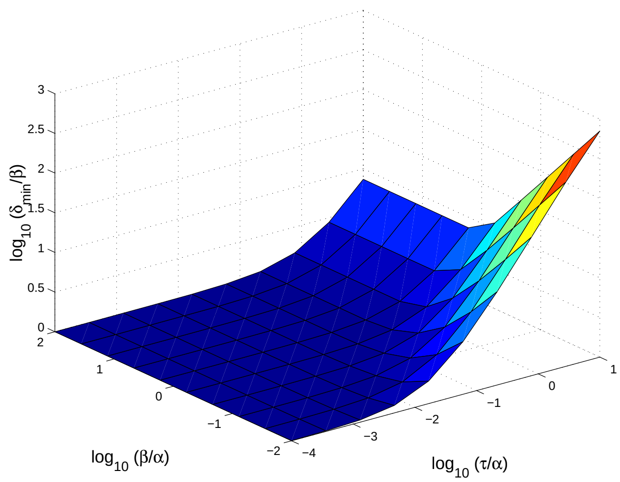

Figure 2: $\frac{\delta_{\min}}{\beta}$ as a function of $\frac{\beta}{\alpha}$ and $\frac{\tau}{\alpha}$

4 Implementation issues

In this section we are concerned about the implementation of K_c and $K_{2_{\text{opt}}}$ defined in (11) and (16).

4.1 Implementation of $K_{2_{\text{opt}}}$

Note that $K_{2_{\text{opt}}}$ in (16) can be written as

$$K_{2_{\text{opt}}}(s) = \frac{\gamma_{\min}^2\alpha^2 - \beta^2}{\gamma_{\min}\alpha\beta}\frac{1}{1+H(s)} \tag{20}$$

where

$$H(s) = \frac{\left(\frac{1}{\alpha}+s\right)\left(\frac{1}{\beta}+s\right) - \left(\frac{x_\gamma^2}{\alpha^2}+s^2\right)}{s^2+\frac{x_\gamma^2}{\alpha^2}} + \frac{1}{\gamma_{\min}\alpha\beta}\frac{1-\beta^2 s^2}{s^2+\frac{x_\gamma^2}{\alpha^2}}\mathrm{e}^{-\tau s} \tag{21}$$

is a finite impulse response filter which impulse response $h(t)$ satisfies

$$h(t) = \begin{cases} \left(\frac{1}{\alpha}+\frac{1}{\beta}\right)\cos\left(\frac{x_\gamma}{\alpha}t\right) + \left(\frac{1}{\beta x_\gamma} - \frac{x_\gamma}{\alpha}\right)\sin\left(\frac{x_\gamma}{\alpha}t\right) \\ \qquad \text{for } t \in [0,\tau] \\ 0 \text{ for } t > \tau \end{cases}. \tag{22}$$

4.2 Implementation of K_c

The capacity predictor K_c defined in (11) can be written as

$$K_c(s) = \left(\frac{2\tau}{\pi\alpha}\right)\frac{1}{1+F_c(s)} \tag{23}$$

where

$$F_c(s) = \frac{\left(\frac{s}{\alpha} - \left(\frac{\pi}{2\tau}\right)^2\right) + \frac{\pi}{2\tau}\left(\frac{1}{\alpha}+s\right)e^{-\tau s}}{\left(\frac{\pi}{2\tau}\right)^2 + s^2} \tag{24}$$

is a finite impulse response filter which impulse response $f_c(t)$ satisfies

$$f_c(t) = \begin{cases} \frac{1}{\alpha}\cos\left(\frac{\pi}{2\tau}t\right) - \frac{\pi}{2\tau}\sin\left(\frac{\pi}{2\tau}t\right) & \text{for } t \in [0,\tau] \\ 0 \quad \text{for } t > \tau \end{cases}. \tag{25}$$

Note that the above finite impulse response filters can be digitally implemented easily by making use of their impulse response expression.

4.3 Block diagrams of the controller

From (4), (5) and (6) the assigned rate R can be expressed as

$$R(s) = K_1(s)E_q(s) + \left(K_2(s) - e^{-\tau s}K_2(s)K_c(s) + K_c(s)\right)C(s), \tag{26}$$

which can be implemented as shown in Figure 3. In fact, because of the symmetry in (26), K_2 and K_c can be interchanged in this figure.

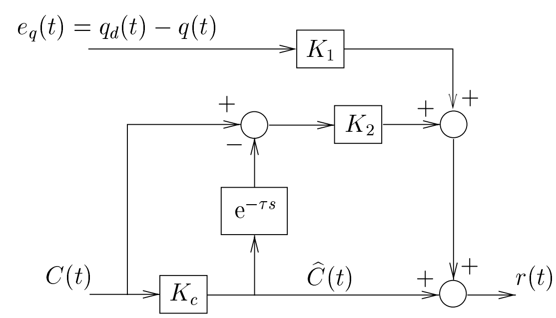

Figure 3: Overall controller implementation

5 Simulations

5.1 Improvement using the new controller

We would like to compare the control schemes defined by (2) and (3) by carrying out `simulink` simulations. The following parameters are used for the simulation:

$$\begin{aligned} c(t) &= 1000 + 100\sin(0.1t) & t \geq 0 \\ q_d(t) &= 100 & t \geq 0 \\ \tau &= 1, \quad \alpha = 1, \quad \beta = 10^{-3}. \end{aligned}$$

The saturation of the queue is taken into consideration in the simulation, $0 \leq q(t) \leq 1000$. Figure 4 shows the plots of the queue length and the source sending rate as a function of time for both cases. It can be seen that the system reaches the steady-state more rapidly when (3) is used, and that we can better reject the influence of variations in capacity on the queue length and source rate dynamics.

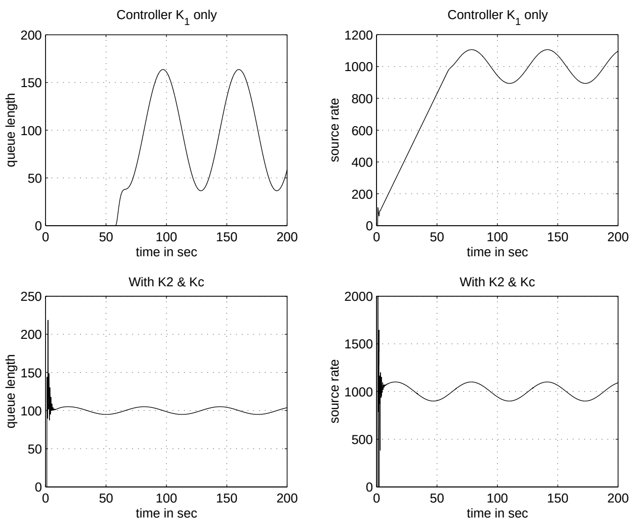

Figure 4: Performance improvement using the new controller

5.2 Use of different controllers

One could envision three different control schemes:

$$\begin{aligned}\text{Case 1}: \quad R_c(s) &= K_1(s)E_q(s) + C(s)\\ \text{Case 2}: \quad R_{\widehat{c}}(s) &= K_1(s)E_q(s) + \widehat{C}(s)\\ \text{Case 3}: \quad R(s) &= K_1(s)E_q(s) + K_{\text{new}}(s)C(s),\end{aligned}$$

the third case being the one discussed in Section 3. Let's examine what are the consequences of such choices on the dynamics of the system. See Figure 5 for the plots of $e_q(t)$ versus time t. In this study we

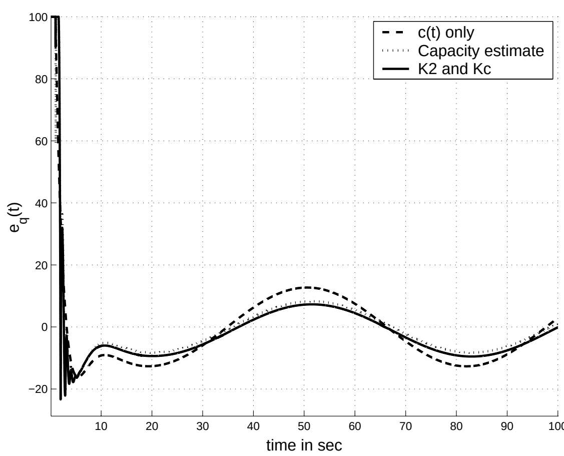

Figure 5: Queue tracking error for different controllers

used:

$$\begin{aligned}c(t) &= 1000 + 200\sin(0.1t) & t &\geq 0\\ q_d(t) &= 100 & t &\geq 0\\ \tau &= 1.\end{aligned}$$

The dashed line corresponds to Case 1. The dotted line corresponds to Case 2 where $\alpha = 10^{-2}$. In this case, as we discussed in section 3.1, since only K_c is used, we are not constrained by the design of K_2 for the choice of α. We can thus take a very small α in order to minimize the predictor performance degradation due to the introduction of the filter $\frac{1}{1+\alpha s}$. The solid line corresponds to Case 3 with $\alpha = 10^2$ and $\beta = 10^{-3}$. It can be noticed that the last design exhibits a behavior similar to the one we have seen in Case 2. In other words, having $K_2 = 0$ and K_c with a small value of α, and having $K_2 \neq 0$ and K_c with a large value of α produce similar results. The reason for this phenomenon is the symmetry pointed out in Section 4.3, which has the interpretation that in Case 3 K_2 takes care of the prediction of $c(t)$, while the effect of K_c is small in that case due to large value of α.

Let's now examine the effects of tuning the parameter α on the performance of the system for the previous three cases. In the following we use:

$$\begin{aligned}\beta &= 10^{-1}, \quad \tau = 1,\\ q_d(t) &= 100 & t &\geq 0\\ c(t) &= 100\sin(0.1t) + \mathcal{N}(0,200) + w(t) & t &\geq 0\end{aligned}$$

where $\mathcal{N}(M,V)$ is a normally distributed random noise with mean M and variance V, $w(t)$ is obtained by passing the signal $Sat_+\left(e^{-0.1t}\mathcal{N}(0,10)\right)$ through the filter $\frac{80(1+\beta s)}{s}$, and the function Sat_+ saturates the negative values to zero. To compare the performance of different control schemes we examine the cost function

$$Z(t) = \sqrt{\int_0^t |e_q(\eta)|^2 d\eta}.$$

We have the following results:

- Case 1: $\lim_{t\to\infty} Z(t) = 250$
- Case 2:

	$\alpha = 10^{-2}$	$\alpha = 10^{-1}$	$\alpha = 1$
$\lim_{t\to\infty} Z(t)$	795	210	315

- Case 3:

	$\alpha = 10^{1}$	$\alpha = 10^{2}$	$\alpha = 10^{3}$
$\lim_{t\to\infty} Z(t)$	212	203	201

It can be noticed that for Case 3 the performance of the system keeps improving while α is increased, but for Case 2 we do not have this consistency, and the value of α needs more tuning. We are thus better off using the controller defined in Case 3.

5.3 `ns` simulations

Figure 6 shows a `ns` [17] simulation for the following configuration: out-going link capacity of 10 Mbps plus a uniform distribution from -10kbps to +10kbps, desired queue length of 20 packets, round-trip time delay of 200ms, packet size of 53 bytes and a sampling frequency of 1000 samples per round-trip time for the controller implementation. Note that in the simulation the rising time is around 3 seconds, which is due to a steady-state error between the estimated capacity and the actual one. This error is due to our digital implementation of (23). We noticed that if we increase the sampling frequency this error decreases and consequently the rising time is shorter, and it also decreases as the value of the outgoing link capacity is smaller.

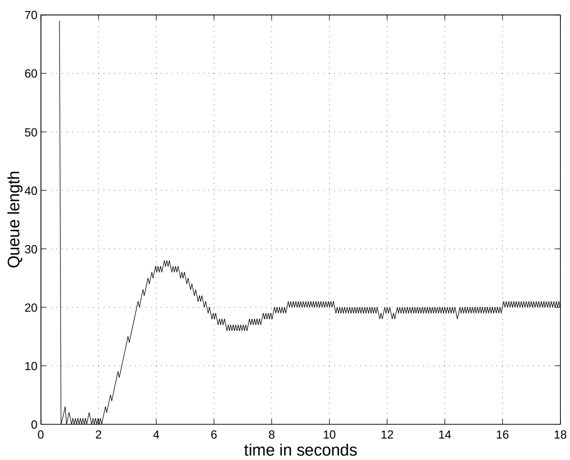

Figure 6: `ns` simulation

6 Concluding remarks

The performance improvement obtained by using the capacity information is quantified in this paper. The extra controller term acting on the capacity is added to the robust controller designed in earlier works, [13], and the improvement in the queue tracking is demonstrated via `simulink` simulations.

References

[1] F. Bonomi and K. Fendick, "The rate-based flow control framework for the available bit rate ATM service," *IEEE Network*, pp. 25–39, March/April 1995.

[2] R. Jain, "Congestion control and traffic management in ATM networks: recent advances and a survey," *Computer Networks and ISDN Systems*, vol. 28, pp. 1723–1738, 1996.

[3] A. K. Parekh and G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: the multiple node case," *IEEE/ACM Trans. on Networking*, vol. 2, pp. 137–150, 1994.

[4] K. K. Ramakrishnan and P. Newman, "Integration of rate and credit schemes for ATM flow control," *IEEE Network*, pp. 49–56, March/April 1995.

[5] ATM Forum Traffic Management AF-TM-0056.000, "The ATM Forum Traffic Management Specification Version 4.0," April 1996.

[6] E. Altman and T. Başar, "Multi-user rate-based flow control: Distributed game-theoretic algorithms," in *Proc. of 36th Conference on Decision and Control*, (San Diego, CA), pp. 2916–2921, December 1997.

[7] E. Altman, T. Başar, and R. Srikant, "Multi-user rate-based flow control with action delays: a team-theoretic approach," in *Proc. of 36th Conference on Decision and Control*, (San Diego, CA), pp. 2387–2392, December 1997.

[8] L. Benmohamed and S. M. Meerkov, "Feedback control of congestion in packet switching networks: the case of a single congested node," in *IEEE/ACM Trans. on Networking*, vol. 1, pp. 693–707, 1993.

[9] J. Jaffe, "Bottleneck flow control," *IEEE Trans. on Communications*, vol. 29, pp. 954–962, 1981.

[10] S. Mascolo, D. Cavendish, and M. Gerla, "ATM rate based congestion control using a Smith predictor: an EPRCA implementation," in *Proc. of IEEE INFOCOM '96*, vol. 5, (San Fancisco), pp. 569–576, March 1996.

[11] C. E. Rohrs and R. A. Berry, "A linear control approach to explicit rate feedback in ATM networks," in *Proc. of INFOCOM '97*, (Kobe, Japan), pp. 277–282, 1997.

[12] Y. Zhao, S. Q. Li, and S. Sigarto, "A linear dynamic model for design of stable explicit-rate ABR control schemes," in *Proc. of INFOCOM '97*, (Kobe, Japan), pp. 283–292, 1997.

[13] B. Ataşlar, P.-F. Quet, A. İftar, H. Özbay, T. Kang, and S. Kalyanaraman, "Robust rate-based flow controllers for high-speed networks: The case of uncertain time-varying multiple time-delays," in *Proc. of the American Control Conference*, vol. 4, pp. 2804 –2808, 2000.

[14] H. Özbay, S. Kalyanaraman, and A. İftar, "On rate-based congestion control in high speed networks: design of an $\mathcal{H}^\infty$ based flow controller for single bottleneck," in *Proc. of the American Control Conference*, vol. 4, pp. 2376 –2380, 24-26 June 1998.

[15] H. Özbay, T. Kang, S. Kalyanaraman, and A. İftar, "Performance and robustness analysis of an $\mathcal{H}^\infty$ based flow controller," in *Proc. of the Conference on Decision and Control*, vol. 3, pp. 2691 –2696, 7-10 Dec. 1999.

[16] C. Foias, H. Özbay, and A. Tannenbaum, *Robust Control of Infinite Dimensional Systems: Frequency Domain Methods.* No. 209 in LNCIS, Springer-Verlag, 1996.

[17] S. Floyd, "`ns` network simulator." http://www-mash.cs.berkeley.edu/ns/, 1995.