Rate-Based Flow Controllers for Communication Networks in the Presence of Uncertain Time-varying Multiple Time-delays[1]

Pierre-François Quet[2] Banu Ataşlar[3] Altuğ İftar[4] Hitay Özbay[5] Shivkumar Kalyanaraman[6] Taesam Kang[7]

Abstract

An $\mathcal{H}^\infty$ based robust controller is designed for a rate-feedback flow-control problem in single-bottleneck communication networks. The controller guarantees stability robustness to uncertain time-varying multiple time-delays in different channels. It also brings the queue length at the bottleneck node to the desired steady-state value asymptotically and satisfies a weighted fairness condition. Lower bounds for stability margins for uncertainty in the time-delays and for the rate of change of the time-delays are derived. A number of simulations are included to demonstrate the time-domain performance of the controller. Trade offs between robustness and time-domain performance are also discussed.

Keywords: Communication networks, flow control, $\mathcal{H}^\infty$ control, time-delay systems, uncertain time-varying multiple time-delays.

1 Introduction

High speed data communication networks require resource management methods in order to provide good quality of service to its users. One typical resource management tool is flow control, which is aimed at avoiding traffic congestion by regulating the rate of data packets sent from the sources. This problem has been studied widely in computer networks and communications literature, see for example [9, 20, 33, 35] and their references.

[1]This work is supported in part by the National Science Foundation under grant numbers ANI-9806660, ANI-9809018, INT-9809903, by the Scientific and Technical Research Council of Turkey, and by the Korea Science and Engineering Foundation. An earlier version of this paper was presented at the 2000 American Control Conference.

[2]Dept. of Electrical Eng., Ohio-State University, Columbus, OH 43210, U.S.A.
e-mail: quet.1@osu.edu

[3]Dept. of Electrical and Electronics Eng., Anadolu University, Eskişehir 26470, Turkey.
e-mail: bataslar@anadolu.edu.tr

[4]Dept. of Electrical and Electronics Eng., Anadolu University, Eskişehir 26470, Turkey. Corresponding author.
e-mail: altug@mmf.mm.anadolu.edu.tr Fax: 90 222 323 9501.

[5]Dept. of Electrical Eng., Ohio-State University, Columbus, OH 43210, U.S.A.
e-mail: ozbay.1@osu.edu

[6]Dept. of Electrical, Computer, and Systems Eng., Rensselaer Polytechnic Institute, Troy, NY 12180, U.S.A.
e-mail: shivkuma@ecse.rpi.edu

[7]Hoseo University, School of Electrical Eng., San 29-1, Sechul-ri, Baebang-myun, Asan, Choongnam 336-795, S. Korea.
e-mail: tskang@dogsuri.hoseo.ac.kr

Feedback schemes used for flow control may be classified into two groups: "rate-based" and "window-based" (also called "credit-based") flow control. The rate-based control with explicit feedback is chosen as the standard flow control scheme in Asynchronous Transfer Mode (ATM) switching networks, by the ATM forum, [1]. The window-based flow control with loss-based or bit-based feedback is popular in end-to-end flow control in the Internet (eg: TCP [18]), though rate-based schemes are also being proposed recently [13].

All congestion control frameworks have three main components implemented at the source end-system, switches (or routers) and destination end-systems. In the case of ATM Available Bit Rate (ABR) service [1], the sources send a control cell once every N packets (called "cells") which can be used by switches to convey feedback. The control cells travel to the destination and are returned to the source in the same path. Feedback signal may be in the form of a single bit or an explicit rate value, and can be written in the forward or reverse direction of travel of the control cell. Many papers in the literature deal with the problem of designing the flow controllers at bottleneck nodes, see for example [2, 3, 6, 19, 27, 36, 40] and their references. ATM networks (and a competing technology MPLS [37]) are deployed in the core of today's Internet and therefore do not reach out to all end-systems, i.e., its control does not operate on end-to-end, short-lived flows, but on edge-to-edge, long-lived aggregate flows. As providers consider building future overlay networks on top of the Internet (where they may control both the "switches" and "end- or edge-systems"), they could consider an explicit rate-feedback framework.

In contrast, the *end-to-end* congestion control model in the Internet (eg: TCP) attempts to solve an optimization problem by decoupling the network problem of assigning bit-marks or losses (penalties or prices) from the source problem of utility maximization (eg: see [21, 17, 24, 23, 25, 26, 28] and references within). The various versions of TCP and drop/marking algorithms [29, 14, 10, 15, 12] can be captured by the optimization framework just described.

Our focus in the paper is on the explicit-rate feedback framework. A challenging aspect of flow control, as far as controller design in this framework is concerned, is the existence of time-delays in the data-flow. Since the controller is to be implemented at the bottleneck node, which regulates the data rates of the sources, a time-delay occurs between the time a command signal for a rate is issued and the actual time this rate is set (time-delay from the bottleneck node to the source node, *backward delay*). Furthermore, the effect of the new rate is seen only after a time-delay which is required for the data to reach the bottleneck (time-delay from the source node to the bottleneck node, *forward delay*). Therefore, the total delay in the process (from the control input to the regulated output) is the sum of these two delays, i.e., the *round-trip delay.* To further complicate the situation, these time-delays are usually uncertain and are time-varying. Furthermore, since there usually are more than one source affecting a bottleneck, there are multiple time-delays. There are several controller design methods for different classes of systems with time-delays, see for example [22, 30, 38] and their references. The techniques developed in [16, 39] are used in this paper.

One of the design goals considered here is *weighted fairness*, which means allocating different percentages of the available capacity to different sources. Thus, weighted fairness may be used as a pricing tool. Another design objective is *tracking*, which is to keep the queue size close to a certain desired size. By choosing this level sufficiently larger than zero and sufficiently smaller than the buffer size, nonlinear effects may also be avoided and the outgoing flow rate may be kept close to the full capacity (thus achieving the maximum utilization of the network). However, the most important design specification is *stability robustness* with respect to uncertainties in the values of time-delays in each flow path [8].

In a recent work, [31], an $\mathcal{H}^\infty$ based flow controller was designed for an explicit rate feedback based congestion control in high speed networks. Internally robust implementation of this controller was discussed in [32], where some simulations demonstrating the time-domain performance of the controller were also presented. One disadvantage of that controller, however, is that, it was obtained by equalizing the nominal time-delays in all the channels. This may result in an underutilization of the network if the difference between the time-delays of different sources is large. Another shortcoming was that, only time-invariant time-delay was considered. Therefore, there was no guarantee of robustness when the time-delays vary in time. In the present work, we alleviate both of these drawbacks. Precisely, we design a controller which is robust to uncertain time-varying multiple time-delays. A multi-variable approach is undertaken here, as opposed to the single-input single-output approach undertaken in [31, 32]. Therefore, different time-delays in different channels are dealt with appropriately. The controller forces the queue length at the bottleneck node to the desired steady-state value asymptotically and also satisfies a weighted fairness condition.

In summary, our focus in the paper is on the ATM-like explicit-rate feedback framework and not the end-to-end decoupled optimization framework. We aim to use the richer information available in the former model to effect robust control over a wider spectrum of objectives in spite of time-varying time-delays.

In the next section we derive the mathematical model of the system and consider robustness against time-varying multiple time-delays in different channels. Performance issues are discussed in Section 3. Robustness and performance conditions are combined in Section 4 to define a two-block optimization problem, which is solved in the same section. Time-domain performance of the controller is presented in Section 5 through a number of simulations. Robustness bounds, as well as the trade-off between robustness and time-domain performance are discussed in Section 6. Some concluding remarks are made in Section 7.

2 Mathematical Model

We consider the feedback system depicted in Figure 1, which consists of a bottleneck node, n source nodes feeding the bottleneck node, and a controller which is to be implemented at the bottleneck node. A queue may form at the bottleneck node, whose dynamics is described as:

$$\dot{q}(t) = \sum_{i=1}^{n} r_i^b(t) - c(t) \tag{1}$$

where $q(t)$ is the queue length at time t, $r_i^b(t)$ is the rate of data received at the bottleneck node at time t from the ith source node, and $c(t)$ is the outgoing flow rate from the bottleneck node at time t, which is equal to the capacity of the outgoing link at time t, unless $q(t) = 0$. In this model, the round-trip delay, $\tau_i(t)$, is defined as $\tau_i(t) = \tau_i^b(t) + \tau_i^f(t)$, where:

$\tau_i^b(t) := h_i^b + \delta_i^b(t)$: is the backward time-delay from the controller to the ith source node (the time-delay which occurs between the time a command signal for a rate is issued and the actual time this rate is set) where h_i^b is the nominal time invariant known backward delay and $\delta_i^b(t)$ is the time-varying backward time-delay uncertainty,

$\tau_i^f(t) := h_i^f + \delta_i^f(t)$: is the forward time-delay from the ith source node to the bottleneck node (the time-delay which is required for the data to reach the bottleneck node) where h_i^f is the nominal time invariant known forward delay and $\delta_i^f(t)$ is the time-varying forward time-delay uncertainty.

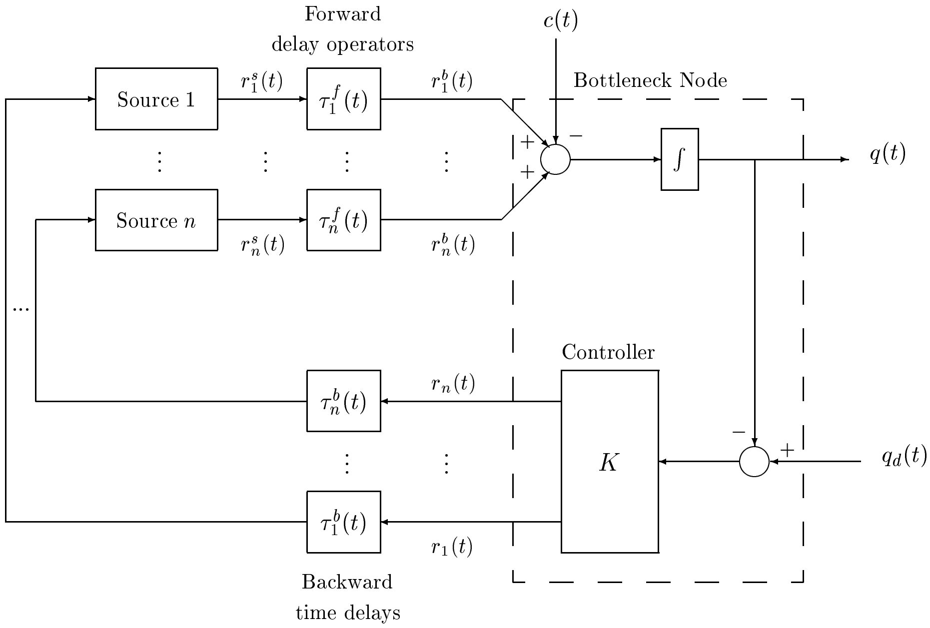

Figure 1: The feedback control system.

Thus, it is assumed that the total nominal time-delay is defined as $h_i := h_i^b + h_i^f$ and the total time delay uncertainty is defined as $\delta_i(t) = \delta_i^b(t) + \delta_i^f(t)$.

Under these assumptions, to determine $r_i^b(t)$, we write the total amount of data received at the bottleneck node from the ith source node by time t:

$$\int_0^t r_i^b(\varphi)d\varphi = \begin{cases} \int_0^{t-\tau_i^f(t)} r_i^s(\varphi)d\varphi, & t - \tau_i^f(t) \geq 0 \\ 0, & t - \tau_i^f(t) < 0 \end{cases} \tag{2}$$

where

$r_i^s(t) := r_i(t - \tau_i^b(t))$: the rate of data sent from the ith source node at time t,

$r_i(t)$: the rate command for the ith source node issued at the controller at time t.

By taking the derivative of both sides of (2), the *forward delay operator* equations are obtained as

$$r_i^b(t) = \begin{cases} (1 - \dot{\delta}_i^f(t))r_i(t - \tau_i(t)), & t - \tau_i^f(t) \geq 0 \\ 0, & t - \tau_i^f(t) < 0 \end{cases} \tag{3}$$

where we made the substitution $r_i^s(t - \tau_i^f(t)) = r_i(t - \tau_i(t))$ and $\dot{\tau}_i^f(t) = \dot{\delta}_i^f(t)$.

We assume that $\frac{d}{dt}(t - \tau_i^f(t)) > 0$, then $\dot{\tau}_i^f(t) < 1$ or $\dot{\delta}_i^f(t) < 1$. If this is not true, a packet sent out from the source node at a particular time may reach the bottleneck node before another packet

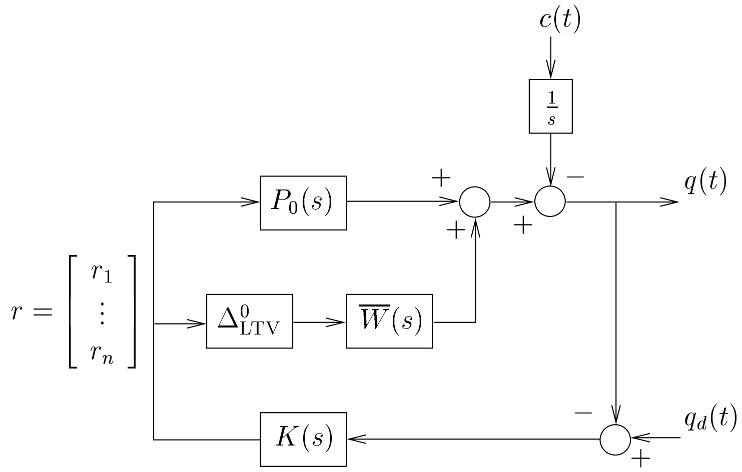

Figure 2: Fictitious system.

which was sent at an earlier time. Furthermore, $\delta_i(t)$ and $\delta_i^f(t)$ are assumed to satisfy

$$|\delta_i(t)| < \delta_i^+ \ , \quad \left|\dot{\delta}_i(t)\right| < \beta_i \ , \quad \left|\dot{\delta}_i^f(t)\right| < \beta_i^f \tag{4}$$

for some known bounds $\delta_i^+ > 0$ and $0 \le \beta_i^f \le \beta_i < 1$. It is also assumed that the queue size always remains between zero and the buffer capacity and $0 \le r_i(t) \le d_i$, where d_i is the limit at which source i can send data.

The above described system is captured by (see Appendix A for details) the uncertain system shown in Figure 2, with

$$P_o(s) = \frac{1}{s}\left[e^{-h_1 s} \cdots e^{-h_n s}\right] \ , \tag{5}$$

$$\Delta^o_{LTV} = \begin{bmatrix} \begin{bmatrix} \Delta^o_{1,1} \\ \Delta^o_{1,2} \end{bmatrix} & & 0 \\ & \ddots & \\ 0 & & \begin{bmatrix} \Delta^o_{n,1} \\ \Delta^o_{n,2} \end{bmatrix} \end{bmatrix} , \tag{6}$$

$$\bar{W}(s) = \left[\bar{W}_1(s) \cdots \bar{W}_n(s)\right] \ , \quad \bar{W}_i(s) = \begin{bmatrix} \frac{e_{i,1}}{s} & e_{i,2} \end{bmatrix} \ , \tag{7}$$

where $e_{i,1} := \frac{\beta_i + \beta_i^f}{\sqrt{1-\beta_i}}$, $e_{i,2} := 2\delta_i^+$, and $\Delta^o_{i,j}$ ($i = 1 \cdots n$ and $j = 1, 2$) are arbitrary LTV systems with norm less than 1. The induced $\mathcal{L}^2$-norm of Δ^o_{LTV} is then less than $\sqrt{2}$.

To find a fixed LTI controller $K(s)$ which robustly stabilizes the system shown in Figure 2, let us consider the following coprime factorizations of the nominal plant in $\mathcal{H}^\infty$:

$$P_0(s) = \frac{1}{s}\left[e^{-h_1 s} \ \cdots \ e^{-h_n s}\right] = N(s)M^{-1}(s) = \tilde{M}^{-1}(s)\tilde{N}(s) \tag{8}$$

where $N(s) = \tilde{N}(s) = \dfrac{1}{s+\epsilon}\left[e^{-h_1 s} \cdots e^{-h_n s}\right]$, $M(s) = \dfrac{s}{s+\epsilon} I_n$, and $\tilde{M}(s) = \dfrac{s}{s+\epsilon}$, where I_n denotes the $n \times n$ identity matrix and $\epsilon > 0$ is arbitrary. Now a parameterization of all controllers $K(s)$ which stabilize $P_0(s)$ can be obtained in terms of $Q \in \mathcal{H}^\infty$, [41]:

$$K(s) = [X(s) + M(s)Q(s)]\,[Y(s) - N(s)Q(s)]^{-1} \tag{9}$$

where $X \in \mathcal{H}^\infty$ and $Y \in \mathcal{H}^\infty$ satisfy the Bezout identity: $\tilde{M}(s)Y(s) + \tilde{N}(s)X(s) = 1$. Since $\lim_{s\to 0} \tilde{M}(s) = 0$, to satisfy the Bezout identity we must have $\lim_{s\to 0} \tilde{N}(s)X(s) = 1$, equivalently, $\epsilon^{-1}\,[1 \;\cdots\; 1]\, X(0) = 1$. Thus, we choose

$$X(s) = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} \epsilon \tag{10}$$

where real numbers α_i are such that $\sum_{i=1}^{n} \alpha_i = 1$. The entries of $X(s)$, α_i's, bring in an extra freedom in choosing the controller. Later in the paper we will show that this freedom may be used in satisfying the weighted fairness condition. Once $X(s)$ is chosen as in (10), we must choose

$$Y(s) = \tilde{M}^{-1}(s)\left[1 - \tilde{N}(s)X(s)\right] = \frac{s+\epsilon}{s} - \frac{\epsilon}{s}\sum_{i=1}^{n} \alpha_i e^{-h_i s}\,. \tag{11}$$

By using the small gain theorem (e.g. see [41]), the closed-loop system shown in Figure 2 is robustly stable for all $\|\Delta^0_{\mathrm{LTV}}\| < \sqrt{2}$ (hence, the actual closed-loop system is robustly stable for all time-delay variations satisfying (4)) if $K(s)$ stabilizes $P_0(s)$ (i.e., $K(s)$ is as given in (9) with $Q \in \mathcal{H}^\infty$) and $||K(I + P_o K)^{-1}\bar{W}||_\infty \le \frac{1}{\sqrt{2}}$. This condition is satisfied if

$$||WKS||_\infty \le 1 \tag{12}$$

with $S(s) = (I + P_o(s)K(s))^{-1} = \tilde{M}(s)\,[Y(s) - N(s)Q(s)]$ and $W(s) = w(s)I_n$, where

$$w(s) = \sqrt{2}\left(\frac{1}{s}\sqrt{\sum_{i=1}^{n} e_{i,1}^2} + \sqrt{\sum_{i=1}^{n} e_{i,2}^2}\right)\,.$$

3 Performance Issues

In this section we will consider some performance issues related to the nominal plant.

3.1 Tracking

One of the performance objectives of rate-based congestion control is to keep the queue size, $q(t)$, as close to its desired value, $q_d(t)$, as possible. In order to make steady-state tracking analysis, we assume that the limit $\lim_{t\to\infty} c(t) =: c_\infty$ exists, for example $c(t)$ can be a step-like function (more realistically, $c(t)$ may have infrequent sudden changes with small variations between such changes; thus, $c(t)$ may be approximated by a series of step functions). Consider $q_d(s) = \frac{1}{s}\hat{q}(s)$ where $\hat{q}$ is an arbitrary bounded energy signal. For example if $\hat{q}$ is a pulse of finite duration, then q_d is a saturating ramp. Considering the nominal plant, the tracking error, $e(t) = q_d(t) - q(t)$, satisfies the following frequency domain identity: $e(s) = S(s)\,\frac{1}{s}\,[\widehat{q}(s) + c(s)]$. Since K is a stabilizing controller, and P_0 has a pole at $s = 0$, we have that $S(0) = 0$, and by the final value theorem the steady state value of the error is

$$e_{ss} = \lim_{s\to 0}\left([e^{-h_1 s}\ \cdots\ e^{-h_n s}]K(s)\right)^{-1} s\,[\widehat{q}(s) + c(s)] = \left([1\ \cdots\ 1]K(0)\right)^{-1} c_\infty$$

(note that the signal $\widehat{q}(t)$ is assumed to have finite energy, so its final value is zero). Thus, at least one of the entries of the controller must have a pole at $s = 0$, in order to have zero steady state error.

3.2 Weighted fairness

Note that the rate feedback signals are given by $[\,r_1(s)\ \ \cdots\ \ r_n(s)\,] = K(s)S(s)\,\frac{1}{s}\,(\widehat{q}(s)+c(s))$. It may be desired to give different steady-state weights (say $\kappa_1, \ldots, \kappa_n$) to different sources (these weights may be determined, e.g., according to a pricing policy). That is, it is desired that $\lim_{t\to\infty} r_i(t) = \kappa_i c_\infty$, where real numbers $\kappa_i > 0$ are such that $\sum_{i=1}^{n}\kappa_i = 1$. This means that the entries, $K_i(s)$, of the controller $K(s)$ must satisfy

$$\lim_{s\to 0} K_i(s)\left([e^{-h_1 s}\ \cdots\ e^{-h_n s}]K(s)\right)^{-1} = \kappa_i\ , \qquad i = 1,\ldots,n\ . \tag{13}$$

If the steady-state weights must be distributed equally among the sources, we take $\kappa_i = \frac{1}{n}$, $i = 1,\ldots,n$.

3.3 Transient response

Besides the steady-state behavior, it is also desired to control the transient response of the system. For this purpose the $\mathcal{H}^\infty$ norm of the weighted sensitivity function can be taken as the cost to be minimized (this corresponds to worst energy minimization for the tracking error, see for example [11, 16]). More precisely, the problem is to

$$\text{minimize} \quad \|W_s\,S\|_\infty \tag{14}$$

over all controllers K stabilizing P_o, where W_s is the sensitivity weighting filter. By examining the formula given in [39], it is seen that the poles of this filter appear as the poles of the open-loop system P_0K. Since P_0 has one pole at $s = 0$, in order for K to have a pole at $s = 0$ (which was found to be a requirement for tracking) W_s must have double poles at $s = 0$. Thus we take:

$$W_s(s) = \frac{1}{s^2}. \tag{15}$$

4 An $\mathcal{H}^\infty$ Optimization Problem

We can combine the robust stability (12) and nominal performance (14) conditions to define a two-block $\mathcal{H}^\infty$ optimization problem:

$$\inf \left\| \begin{bmatrix} W_s\,S \\ WKS \end{bmatrix} \right\|_\infty =: \gamma_{\rm opt} \tag{16}$$

where the infimum is taken over all K stabilizing P_0 subject to the weighted fairness condition (13). Note that, due to the choice of W_s in (15), the tracking condition (that K must have a pole at $s=0$) will be satisfied automatically.

The following observations will be used to determine an upper bound for γ_{opt} in terms of solutions to n decoupled $\mathcal{H}^\infty$ problems. We begin with

$$\begin{aligned} W_s(s)S(s) &= W_s(s)\tilde{M}(s)\left[Y(s)-N(s)Q(s)\right] \\ &= W_s(s)\frac{s}{s+\epsilon}\left[\frac{s+\epsilon}{s}-\frac{\epsilon}{s}\sum_{i=1}^{n}\alpha_i e^{-h_i s}-\frac{1}{s+\epsilon}\sum_{i=1}^{n}e^{-h_i s}Q_i(s)\right] \\ &= \sum_{i=1}^{n}\alpha_i W_s(s)M_i(s)\left[Y_i(s)-N_i(s)Q_i(s)\right] \end{aligned} \tag{17}$$

where

$$Y_i(s) := \frac{s+\epsilon}{s}-\frac{\epsilon}{s}e^{-h_i s}\ , \qquad N_i(s) := \frac{1}{\alpha_i(s+\epsilon)}e^{-h_i s}\ , \qquad M_i(s) := \frac{s}{s+\epsilon}\ , \tag{18}$$

and $Q_i(s)$ is the i^{th} element of $Q(s)$. Let us also define $X_i(s) := \alpha_i\epsilon$, which satisfies $M_i(s)Y_i(s)+N_i(s)X_i(s)=1$. Then

$$\begin{aligned} W(s)K(s)S(s) &= W(s)\left[X(s)+M(s)Q(s)\right]\tilde{M}(s) \\ &= \sum_{i=1}^{n}W_i(s)\left[\alpha_i\epsilon+\frac{s}{s+\epsilon}Q_i(s)\right]\frac{s}{s+\epsilon} \\ &= \sum_{i=1}^{n}W_i(s)\left[X_i(s)+M_i(s)Q_i(s)\right]M_i(s) \end{aligned} \tag{19}$$

where $W_i(s)$ is a $n\times 1$ vector consisting of $w(s)$ at the i^{th} row and 0's elsewhere.

Using (17) and (19), problem (16) can be rewritten as:

$$\inf_{Q\in\mathcal{H}^\infty}\left\|\sum_{i=1}^{n}\begin{bmatrix}\alpha_i W_s\left[Y_i-N_iQ_i\right]M_i \\ W_i\left[X_i+M_iQ_i\right]M_i\end{bmatrix}\right\|_\infty =: \gamma_{\rm opt} \tag{20}$$

where the infimum is taken over all $Q\in\mathcal{H}^\infty$, subject to the weighted fairness condition (13).

Let

$$\inf_{Q_i\in\mathcal{H}^\infty}\left\|\begin{bmatrix}n^{-1}W_s\left[Y_i-N_iQ_i\right]M_i \\ (n\alpha_i)^{-1}w\left[X_i+M_iQ_i\right]M_i\end{bmatrix}\right\|_\infty =: \gamma_i \tag{21}$$

where the infimum is taken over all $Q_i\in\mathcal{H}^\infty$, subject to the weighted fairness condition (13). Clearly, $\gamma_{\rm opt}\leq\sum_{i=1}^{n}n\alpha_i\gamma_i$. Since it is very complicated, if not impossible, to find an optimal solution to problem (20) subject to (13), we propose a suboptimal solution $Q(s)=[Q_1(s)\ \cdots\ Q_n(s)]^T$, where each $Q_i(s)$ satisfies (21) such that the weighted fairness condition (13) is also satisfied. For this, we will first find a solution to (21) for each i without considering the condition (13); and then show that, by a proper choice of α_i, this solution also satisfies the weighted fairness condition (13).

Let us define

$$C_i(s) := [X_i(s) + M_i(s)Q_i(s)]\,[Y_i(s) - N_i(s)Q_i(s)]^{-1} \tag{22}$$

so that

$$Q_i(s) = [M_i(s) + C_i(s)N_i(s)]^{-1}\,[C_i(s)Y_i(s) - X_i(s)]\ . \tag{23}$$

Then (21) can be rewritten as:

$$\inf_{C_i \text{ stabilizing } P_i} \left\| \begin{bmatrix} n^{-1}W_s\,[1+P_iC_i]^{-1} \\ (n\alpha_i)^{-1}wC_i\,[1+P_iC_i]^{-1} \end{bmatrix} \right\|_\infty =: \gamma_i \tag{24}$$

where $P_i(s) := N_i(s)M_i^{-1}(s) = \frac{1}{\alpha_i s}e^{-h_i s}$. As shown in Appendix B, the solution to problem (24) is

$$C_i(s) = \frac{n\alpha_i\gamma_i}{2\sqrt{2\sum_{j=1}^n \left(\delta_j^+\right)^2}} \left(\frac{sh_i - k_i}{sh_i}\right) \frac{1}{1+F_i(sh_i)} \tag{25}$$

where $F_i(sh_i)$ corresponds to a finite impulse response (FIR) filter of duration h_i (thus, it can easily be implemented in discrete-time with h_i/T_s states, where T_s is the sampling period) and k_i and γ_i are constants to be calculated. The details of the computation of F_i, k_i, and γ_i are given in Appendix B.

The controller can now be determined as (see Appendix C) $K(s) = [K_1(s)\ \cdots\ K_n(s)]^T$, where

$$K_i(s) = \frac{C_i(s)}{1+C_i(s)P_i(s)} \left(1 - \sum_{j=1}^n \alpha_j\,\frac{P_j(s)C_j(s)}{1+P_j(s)C_j(s)}\right)^{-1}\ . \tag{26}$$

The determined controller may be implemented as shown in Figure 3.

Now it remains to be shown that the parameters α_i used in $X(s)$ can be chosen so that the resulting controller satisfies the weighted fairness condition (13). Using (9), it can be shown that (13) can be written as:

$$\lim_{s\to 0} \frac{\alpha_i\epsilon + \frac{s}{s+\epsilon}Q_i(s)}{\sum_{l=1}^n e^{-h_l s}[\alpha_l\epsilon + \frac{s}{s+\epsilon}Q_l(s)]} = \kappa_i\ . \tag{27}$$

However, since $Q_i \in \mathcal{H}^\infty$ (thus $\lim_{s\to 0} Q_i(s)$ is finite) for all $i = 1,\ldots,n$ and $\sum_{l=1}^n \alpha_l = 1$, the left hand side simply reduces to α_i. Thus, to satisfy the weighted fairness condition, we simply need to choose: $\alpha_i = \kappa_i$, $i = 1,\ldots,n$.

5 Simulation Results

The closed loop system with the determined controller shown in Figure 3 is implemented in SIMULINK and the system is simulated for a number of different conditions. Nonlinear aspects of the system are also taken into account in these simulations: the queue length and all the rates must be non-negative, and the queue size cannot exceed 100 packets. In all the cases except for Case 2 we assumed that the

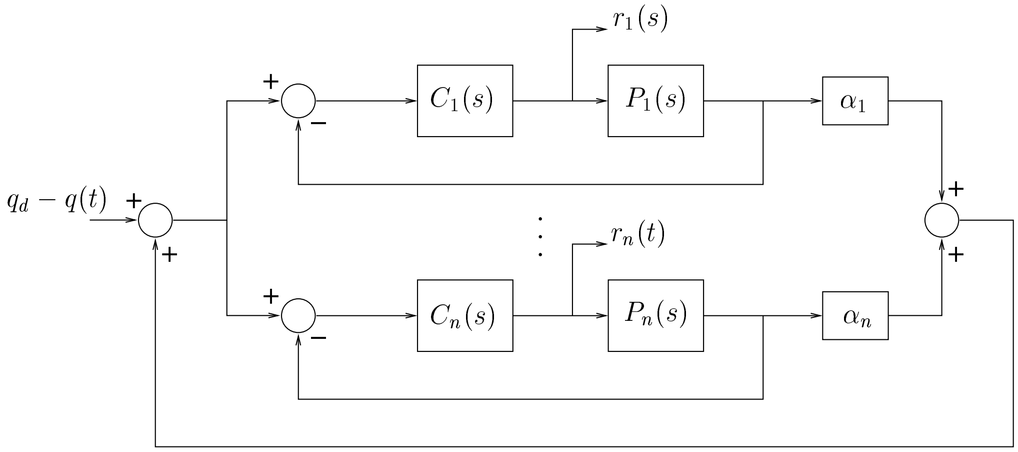

Figure 3: The implementation of the controller.

sources can supply data at a rate no more than 100 packets per second. The parameters $h_i = h_i^b + h_i^f$, κ_i, δ_i^+, β_i and β_i^f are design parameters used for the controller derivation and γ_i is the resulting $\mathcal{H}^\infty$ cost which is used in the controller implementation. The actual delays used in the simulations are $\tau_i^b(t) = h_i^b + \delta_i^b(t)$ and $\tau_i^f(t) = h_i^f + \delta_i^f(t)$. In all the cases the number of sources is assumed to be $n = 5$, the desired queue length is taken as $q_d = 30$ packets, and the capacity of the outgoing link is taken as 60 packets per second.

Case 1: The delays (in seconds), fairness weights, and the other controller parameters are as shown in Table 1. The plots of the queue length, $q(t)$, and the flow rates, $r_i^s(t)$, for each source are shown in Figure 4. In the period between 0 to about 20 seconds we note that the queue size is zero. This corresponds to the time needed for the sum of the rates $r_i^b(t)$ to exceed the capacity of the outgoing link at the bottleneck node. Note that the relative steady-state flow rates of the sources are equal to the relative fairness weights of these sources.

i	h_i^b	δ_i^b	h_i^f	δ_i^f	κ_i	δ_i^+	β_i	β_i^f	γ_i
1	0.9	$0.5\sin(\frac{2\pi}{50}t)$	0.1	$0.1\sin(\frac{\pi}{50}t)$	0.2	2	0.1	0.01	2.35
2	1.85	$0.2\sin(\frac{\pi}{50}t)$	0.15	$0.1\cos(\frac{\pi}{50}t)$	0.1	3	0.2	0.02	3.07
3	0.9	$0.5\cos(\frac{2\pi}{50}t)$	0.1	$0.05\sin(\frac{\pi}{100}t)$	0.4	2	0.1	0.01	2.35
4	1.88	$0.3\cos(\frac{2\pi}{50}t)$	0.12	$0.05\cos(\frac{\pi}{100}t)$	0.2	3	0.2	0.02	3.07
5	1.8	$0.4\sin(\frac{2\pi}{50}t)$	0.2	$0.05\sin(\frac{\pi}{50}t)$	0.1	3	0.2	0.02	3.07

Table 1: Parameters for Case 1

Case 2: Next we consider the same system as in Case 1, with the same delays, fairness weights, and other controller parameters. But in this case we assume that the sources can supply data at relatively lower rates. These rates are shown in Table 2 for each source. The resulting queue length and flow rates are shown in Figure 5. As observed from this figure, flow rates at sources 1, 3 and 5 are saturated. However, the controller can successfully redistribute the unused rates to the other two sources 2 and 4. Also note that the relative steady-state flow rates of the unsaturated sources are equal to the relative fairness weights of these sources. Although the controller is unaware of the saturation, it can still regulate the queue length, however, this regulation takes a little bit more time and a larger overshoot is observed in the queue length.

i	1	2	3	4	5
d_i (in packets/s)	10	20	20	20	5

Table 2: Rate limits for Case 2

Case 3: The delays are the same as in Case 1, the problem data is changed as shown in Table 3.

i	δ_i^+	β_i	β_i^f	γ_i
1	0.1	0.03	0.005	0.61
2	0.25	0.03	0.005	1.02
3	0.1	0.03	0.005	0.61
4	0.25	0.03	0.005	1.02
5	0.25	0.03	0.005	1.02

Table 3: Design parameters for Case 3

Case 4: The delays are the same as in Case 1, the problem data is changed as shown in Table 4.

i	δ_i^+	β_i	β_i^f	γ_i
1	6	0.7	0.2	4.39
2	8	0.7	0.2	5.33
3	6	0.7	0.2	4.39
4	8	0.7	0.2	5.33
5	8	0.7	0.2	5.33

Table 4: Design parameters for Case 4

The results for Cases 3 and 4 are shown in Figures 6 and 7, respectively. Comments on these results are made in the next section.

Although not easily noticable in the figures, steady-state oscillations exist in all the responses, due to the time-varying forward delay (precisely due to the term $(1-\dot{\delta}_i^f(t))$ appearing in (3)). The magnitude of these oscillations would be larger if the rate of change of the forward delay was larger and would be zero if this rate was zero (see [5]). These oscillations are unavoidable unless some information about the forward delay uncertainty is available to the controller.

We also did other simulations showing that the controller responds well to variations in the capacity of the outgoing link (see [5]). The response of the controller in this case may further be improved if a *capacity predictor*, which predicts future values of this capacity, is included in the controller (see [34]).

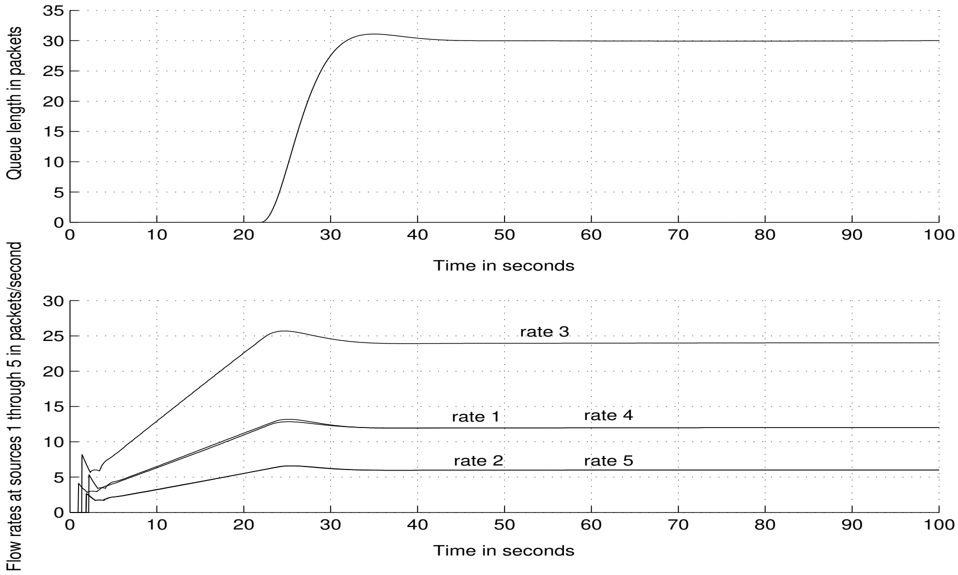

Figure 4: Simulation results for Case 1.

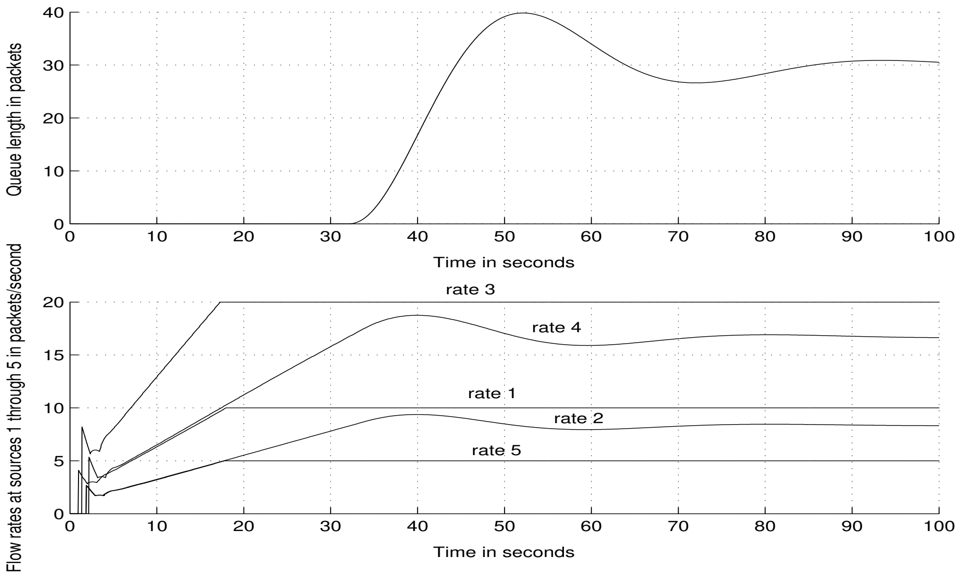

Figure 5: Simulation results for Case 2.

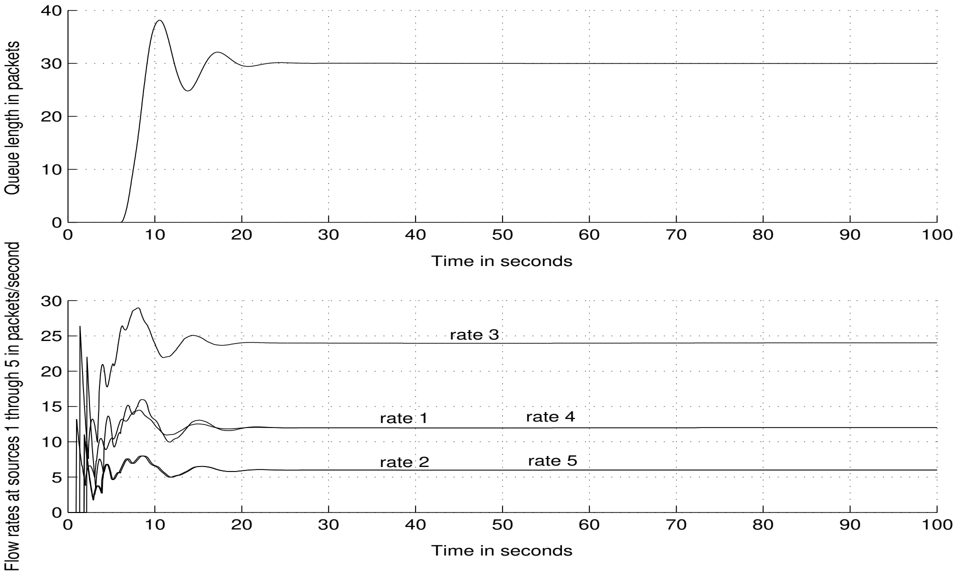

Figure 6: Simulation results for Case 3.

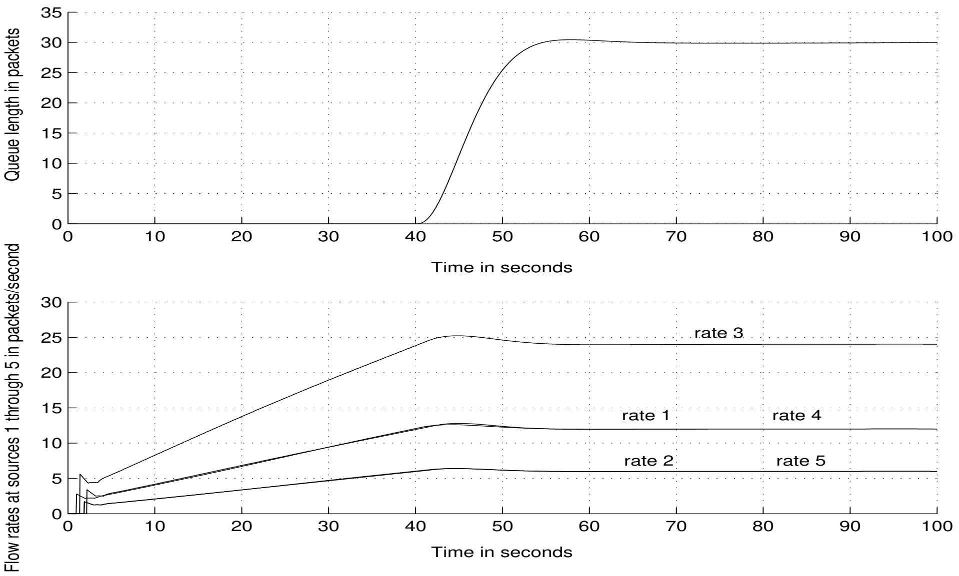

Figure 7: Simulation results for Case 4.

6 Stability Margins

The mixed sensitivity minimization problem (24) can be solved as described earlier for each fixed uncertainty level $(\beta_i, \beta_i^f, \delta_i^+)$. Figures 8, 9, and 10 show the smallest performance cost γ for several feasible values of β, β^f and δ^+. Here, for simplicity, we chose $n = 1$. However, since the subproblems are decoupled, the value of n only affects the numerical value of γ_i, the general shape of the graph is the same for $n > 1$. Also, we used $h = 1$ to normalize the delay.

Robust stability is achieved when (12) is satisfied. By using the arguments leading to (21), lower bounds (sufficient conditions) on the actual stability margins for $e_{i,1}$ (call it $e_{i,1,act}$) and for $e_{i,2}$ (call it $e_{i,2,act}$) must satisfy:

$$\sum_{i=1}^{n} e_{i,1,act}^2 = \frac{1}{\gamma^2} \sum_{i=1}^{n} e_{i,1}^2 \quad \text{and} \quad \sum_{i=1}^{n} e_{i,2,act}^2 = \frac{1}{\gamma^2} \sum_{i=1}^{n} e_{i,2}^2 \tag{28}$$

where $\gamma := \sum_{i=1}^{n} n\alpha_i\gamma_i$. We note that, for the case $n > 1$, there are infinitely many solutions to (28) and the system is robustly stable for any one of these solutions. These bounds on the actual stability margins are also depicted in Figures 8, 9, and 10 for various feasible uncertainty levels for the case $n = 1$. It is seen that, for a fixed β and β^f, $e_{2,act}$ increases with increasing δ^+ while $e_{1,act}$ decreases slightly. Similarly, for a fixed δ^+ and β^f, $e_{1,act}$ increases with increasing β while $e_{2,act}$ decreases slightly. The effect of larger β^f, although less noticable, is to slightly decrease both margins especially when β is close to 1. These figures indicate that, for large stability robustness margins $e_{1,act}$ and $e_{2,act}$, the uncertainty levels δ^+ and β must be chosen as large as possible. Such a choice, however, may adversely affect the time-domain performance of the controller. For small values of β and δ^+ as for the Case 3 of simulations, it is seen in Figure 6 that the response is oscillatory, but relatively fast. Whereas for large values of the same design parameters as for the Case 4 of simulations, it is seen in Figure 7 that the response is smoother, but takes much longer time to settle down. Similar results were also found in [32] for the case of time-invariant delays.

7 Conclusions

Robust controller design for a flow control problem in communication networks has been considered. A robust controller has been designed against uncertain time-varying multiple time-delays. The controller brings the queue length at the bottleneck node to the desired steady-state value asymptotically and also satisfies a weighted fairness condition. Stable implementation of the controller has also been presented. This implementation is depicted in Figure 3. As seen from this figure, the controller includes n integrators followed by delay elements (P_i's) and n blocks (C_i's) each of which include (as seen in (25)) a proportional plus integral (PI) term which is cascaded with a feedback block containing an FIR filter. Since digital imlementation of delays and FIR filters are relatively easy, the controller can easily be implemented without too much computational overhead, at the bottleneck node.

A number of simulations have been included to demonstrate the time-domain performance of the controller. Stability margins for uncertainty in the time-delays and for the rate of change of the time-delays have also been discussed and their lower bounds have been derived. It has been shown that there is a trade off between robustness and the time-domain performance. If the uncertainty levels are chosen high, then the system is highly robust to uncertain time-varying time-delays and a smooth, however, very slow response is obtained. On the other hand, if these levels are chosen low, the response

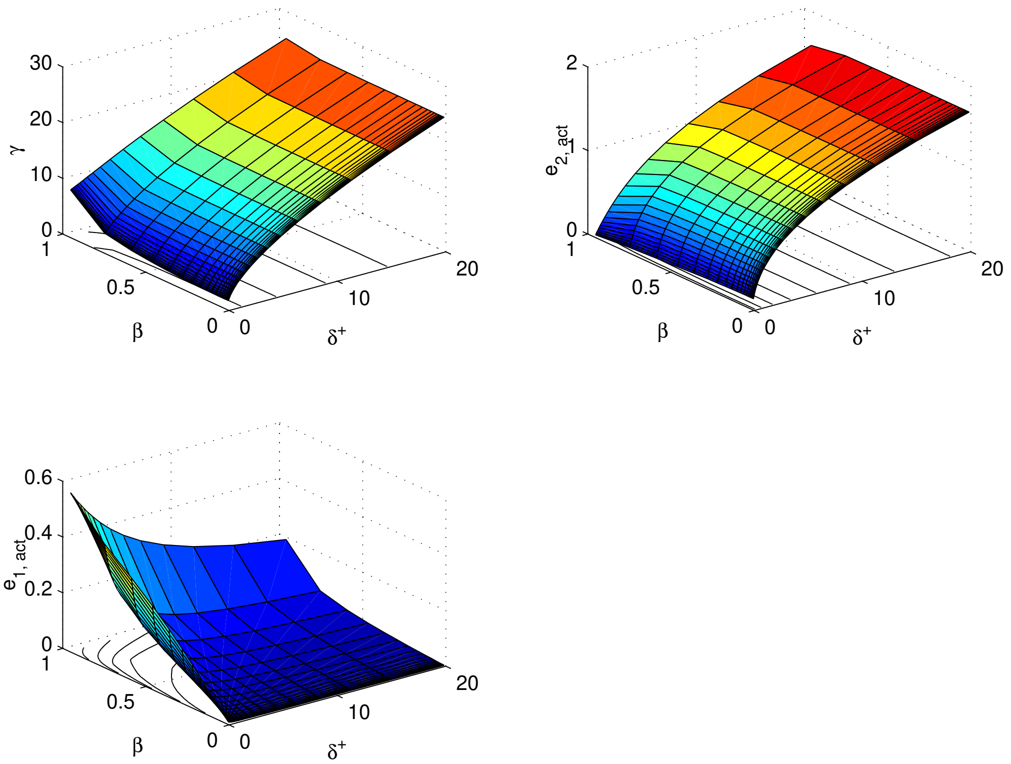

Figure 8: Performance cost and stability margins with $\beta^f = 0$.

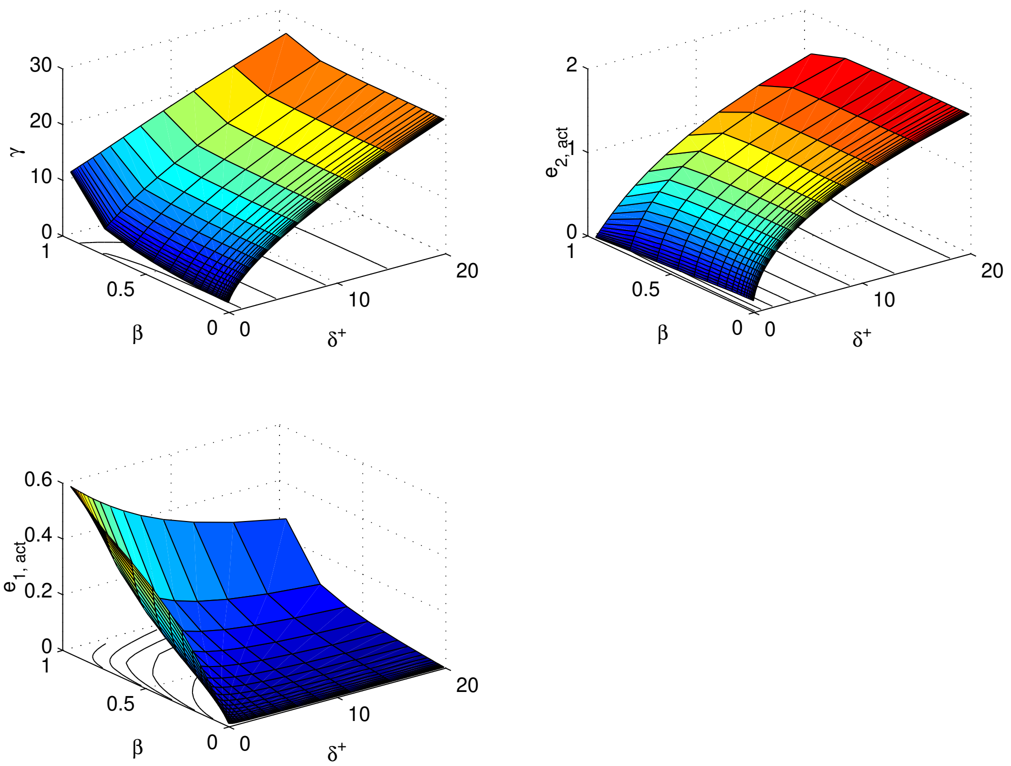

Figure 9: Performance cost and stability margins with $\beta^f = \frac{1}{2}\beta$.

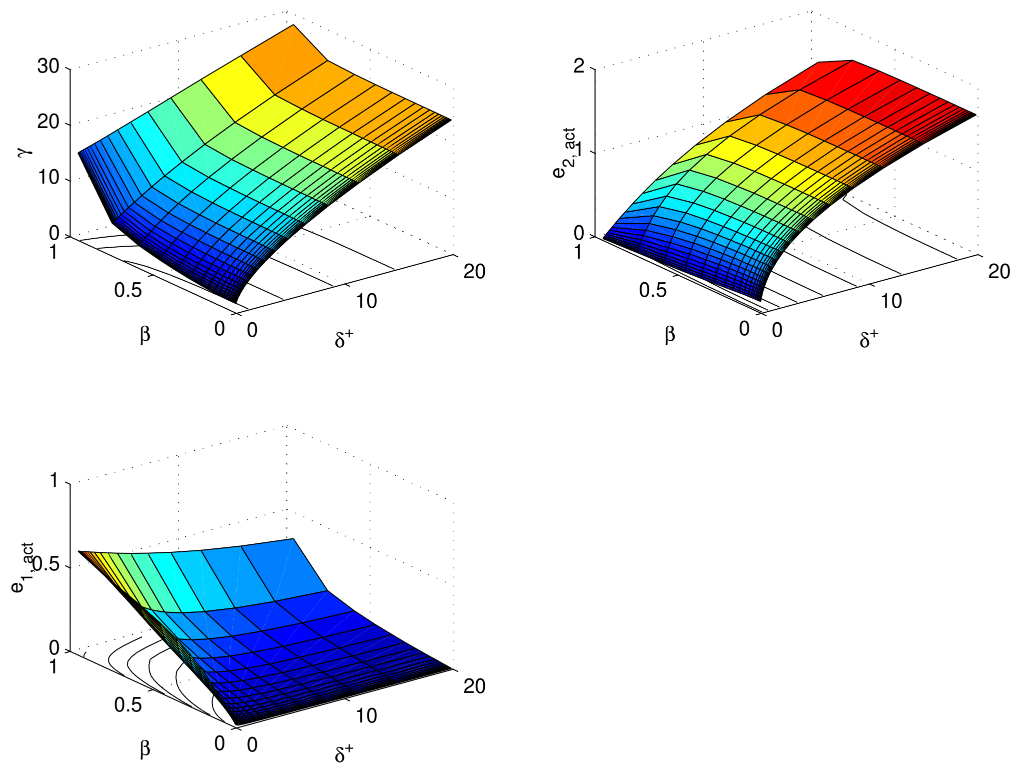

Figure 10: Performance cost and stability margins with $\beta^f = \beta$.

is much faster, but more oscillatory. Furthermore, for lower uncertainty levels, the actual stability margins are also lower, in general.

In this work we considered only the case of a single bottlenek node. Although the present work may be extended to the case of multiple bottlenecks, the extension is not trivial since the controllers to be implemented at different bottlenecks will interact. Initial results on this topic may be found in [7].

We also note that, in this work, we did not utilize the special structure of the uncertainty given in (6). This special structure may be taken into account by using the structured singular value approach, as demonstrated in [4].

Appendices

A Uncertainty Model

From (1) and (3), $q(t)$ is given by:

$$q(t) = \int_0^t \left[\sum_{i=1}^{n} (1 - \dot{\delta}_i^f(\nu)) r_i(\nu - \tau_i(\nu)) - c(\nu) \right] d\nu \; + q(0) \; . \tag{29}$$

Let

$$q_0(t) := \int_0^t \left[\sum_{i=1}^n r_i(\nu - h_i) - c(\nu) \right] d\nu \ + q(0) \tag{30}$$

and

$$\delta_q(t) := q(t) - q_0(t) = \sum_{i=1}^n \int_0^t \left[(1 - \dot{\delta}_i^f(\nu)) r_i(\nu - \tau_i(\nu)) - r_i(\nu - h_i) \right] d\nu \ . \tag{31}$$

Also let $\lambda_i := \nu - \tau_i(\nu) = \nu - h_i - \delta_i(\nu) =: f_i(\nu)$. Then

$$\frac{d\lambda_i}{d\nu} = 1 - \frac{d\delta_i}{d\nu} = 1 - g_i(\lambda_i) \tag{32}$$

where $g_i(\lambda) := \left.\frac{d\delta_i}{d\nu}\right|_{\nu = f_i^{-1}(\lambda)}$. Note that, the inverse function $\nu = f_i^{-1}(\lambda)$ exists since, by (4), $\frac{d\lambda_i}{d\nu} > 0$.

Now, noting that, from (32), $d\nu = \frac{d\lambda_i}{1 - g_i(\lambda_i)}$ and assuming that $\delta_i(0) = 0$, (31) can be rewritten as:

$$\begin{aligned}
\delta_q(t) &= \sum_{i=1}^n \left[\int_0^t (1 - \dot{\delta}_i^f(\nu)) r_i(\nu - \tau_i(\nu))\, d\nu - \int_{-h_i}^{t-h_i} r_i(\sigma)\, d\sigma \right] \\
&= \sum_{i=1}^n \left[\int_0^t (1 - \dot{\delta}_i^f(\nu)) r_i(\nu - \tau_i(\nu))\, d\nu - \int_{-h_i}^{t-h_i} r_i(\sigma)\, d\sigma \right. \\
&\qquad \left. + \int_{-h_i}^{t-h_i-\delta_i(t)} r_i(\sigma)\, d\sigma - \int_{-h_i}^{t-h_i-\delta_i(t)} r_i(\lambda_i)\, d\lambda_i \right] \\
&= \sum_{i=1}^n \left[\int_0^t (1 - \dot{\delta}_i^f(\nu)) r_i(\nu - \tau_i(\nu))\, d\nu - \int_{t-h_i-\delta_i(t)}^{t-h_i} r_i(\nu)\, d\nu \right. \\
&\qquad \left. - \int_0^t r_i(\nu - \tau_i(\nu)) \left[1 - g_i(\nu - \tau_i(\nu)) \right] d\nu \right] \\
&= \sum_{i=1}^n \left[\int_0^t \left[g_i(\nu - \tau_i(\nu)) - \dot{\delta}_i^f(\nu) \right] r_i(\nu - \tau_i(\nu))\, d\nu - \int_{t-h_i-\delta_i(t)}^{t-h_i} r_i(\nu)\, d\nu \right] \ .
\end{aligned} \tag{33}$$

We now have $\delta_q(t) = \sum_{i=1}^n \delta_q^i(t)$, where $\delta_q^i(t)$ is the output of the system shown in Figure 11, with $\Delta_{i,1}$ and $\Delta_{i,2}$ linear time varying (LTV) systems, M_{g_i} and $M_{\dot{\delta}_i^f}$ are the LTV systems defined by $p_i(t) = g_i(t)\, r_i(t)$ and $z_i(t) = \dot{\delta}_i^f(t)\, y_i(t)$, respectively, and $e_{i,j}$'s are constants to be specified later.

Note that

$$\begin{aligned}
\int_0^\infty |y_i(t)|^2\, dt &= \int_0^\infty |x_i(t - \tau_i(t))|^2\, dt = \int_{-\tau_i(t)}^\infty |x_i(\lambda_i)|^2 \frac{d\lambda_i}{1 - g_i(\lambda_i)} \\
&= \int_0^\infty |x_i(\lambda_i)|^2 \frac{d\lambda_i}{1 - g_i(\lambda_i)} < \frac{1}{1 - \beta_i} \int_0^\infty |x_i(\lambda_i)|^2\, d\lambda_i
\end{aligned} \tag{34}$$

where we assumed $r_i(t) = 0$ for $t < 0$. This implies that the $\mathcal{L}^2$-induced norms of the delay blocks in $\Delta_{i,1}$ are less than $\frac{1}{\sqrt{1-\beta_i}}$. Thus, the $\mathcal{L}^2$-induced norm of $\Delta_{i,1}$ is less than $\left(\frac{\beta_i + \beta_i^f}{\sqrt{1-\beta_i}} \right) \frac{1}{e_{i,1}}$, since we have

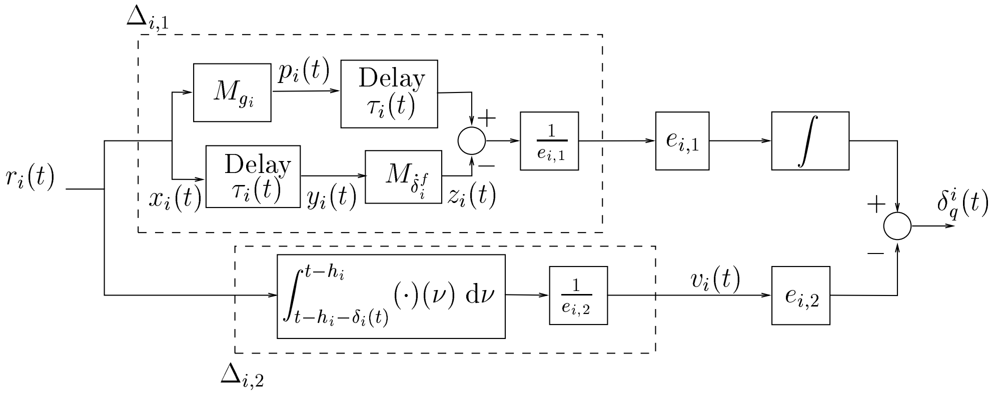

Figure 11: Model of the uncertain part of the system.

$|g_i| < \beta_i$ and $|\dot{\delta}_i^f| < \beta_i^f$. Then, defining $e_{i,1} = \frac{\beta_i+\beta_i^f}{\sqrt{1-\beta_i}}$, we show that the $\mathcal{L}^2$-induced norm of the LTV system $\Delta_{i,1}$ is less than 1.

Also, using the fact that $r_i(t) \geq 0$, we note that

$$\left\| \frac{1}{2\delta_i^+} \int_{t-h_i-\delta_i(t)}^{t-h_i} r_i(\nu)\, d\nu \right\|_2 < \left\| \frac{1}{2\delta_i^+} \int_{t-h_i-\delta_i^+}^{t-h_i+\delta_i^+} r_i(\nu)\, d\nu \right\|_2 \tag{35}$$

since $|\delta_i(t)| < \delta_i^+$. Therefore, by defining

$$\hat{v}_i(s) = \frac{1}{2\delta_i^+} \left(\frac{e^{-(h_i-\delta_i^+)s} - e^{-(h_i+\delta_i^+)s}}{s} \right) r_i(s) \tag{36}$$

we obtain that

$$\begin{array}{rcl} \|v_i\|_2 & < & \|\hat{v}_i\|_2 \\ & < & \left\| \frac{1}{2\delta_i^+} \left(\frac{1-e^{-2\delta_i^+ s}}{s} e^{-(h-\delta_i^+)s} \right) \right\|_\infty \|r_i\|_2 \\ & < & \|r_i\|_2 \end{array} \tag{37}$$

where by a slight abuse of notation we use $\hat{v}_i(s)$ and $r_i(s)$ to denote the Laplace transforms of time signals $\hat{v}_i(t)$ and $r_i(t)$. So, by defining $e_{i,2} = 2\delta_i^+$, we show that the $\mathcal{L}^2$-induced norm of the LTV system $\Delta_{i,2}$ is less than 1.

B Solution to the Optimization Problem

Let us define $\tilde{W}_i(s) := \dfrac{1}{n} W_s(s) = \dfrac{h_i^2}{n} \dfrac{1}{(h_i s)^2}$ and $\hat{W}_i(s) := \dfrac{s}{n} w(s) = \tilde{\beta}_i + \tilde{\delta}_i(h_i s)$, where $\tilde{\beta}_i = \sqrt{2}\, n^{-1} \sqrt{\sum_{i=1}^n e_{i,1}^2} = \sqrt{2}\, n^{-1} \sqrt{\sum_{i=1}^n \frac{\left(\beta_i+\beta_i^f\right)^2}{1-\beta_i}}$ and $\tilde{\delta}_i = \sqrt{2}\, (n h_i)^{-1} \sqrt{\sum_{i=1}^n e_{i,2}^2}$

$= 2\sqrt{2}\,(nh_i)^{-1}\sqrt{\sum_{i=1}^{n}\left(\delta_i^{+}\right)^2}$. We also define $\hat{P}_i(s) := \dfrac{1}{nh_i}P_i(s) = \dfrac{1}{n\alpha_i}\dfrac{1}{h_i s}e^{-h_i s}$ and $\hat{C}_i(s) := nh_iC_i(s)$, so that (24) can be rewritten as

$$\inf_{\hat{C}_i \text{ stabilizing } \hat{P}_i} \left\| \begin{bmatrix} \tilde{W}_i\left[1+\hat{P}_i\hat{C}_i\right]^{-1} \\ \hat{W}_i\hat{P}_i\hat{C}_i\left[1+\hat{P}_i\hat{C}_i\right]^{-1} \end{bmatrix} \right\|_\infty =: \gamma_i \tag{38}$$

which is now in terms of the normalized frequency $\hat{s}_i := h_i s$. In deriving (38) from (24) we also made use of the fact that $|e^{-jh_i\omega}| = 1$ for all $\omega \in \mathbb{R}$.

By defining $\hat{\beta}_i = \frac{n\tilde{\beta}_i}{h_i^2}$, $\hat{\delta}_i = \frac{n\tilde{\delta}_i}{h_i^2}$, and $\hat{\gamma}_i = \frac{n\gamma_i}{h_i^2}$, and applying the formulae given in [39] and [32], the optimal solution to (38) is found as:

$$\hat{C}_i(s) = \frac{n\alpha_i\hat{\gamma}_i}{\hat{\delta}_i}\left(\frac{\hat{s}_i - k_i}{\hat{s}_i}\right)\frac{1}{1+F_i(\hat{s}_i)} \tag{39}$$

where

$$F_i(\zeta) = \frac{(\zeta+k_i)(\zeta+a_i)(\zeta^2+b_i\zeta+c_i) - (\zeta^4 - \hat{\gamma}_i^{-2}) - \frac{\hat{\gamma}_i}{\hat{\delta}_i}e^{-\zeta}\zeta^2(\zeta-k_i)}{\zeta^4 - \hat{\gamma}_i^{-2}} \tag{40}$$

where

$$c_i := \sqrt{x_i}\ , \quad a_i := \frac{1}{c_i\hat{\delta}_i}\sqrt{1-\frac{\hat{\beta}_i^2}{\hat{\gamma}_i^2}}\ , \quad b_i := \sqrt{\frac{\hat{\beta}_i^2}{\hat{\delta}_i^2} + 2c_i - a_i^2}\ , \quad k_i := \frac{\rho_i - 1}{\sqrt{\hat{\gamma}_i}(\rho_i+1)}$$

where $x_i > 0$ is the unique positive root of

$$x_i^3 + \frac{1}{\hat{\gamma}_i^2}x_i^2 + \left(1-\frac{\hat{\beta}_i^2}{\hat{\gamma}_i^2}\right)\frac{\hat{\beta}_i^2}{\hat{\delta}_i^4}x_i - \frac{\left(1-\frac{\hat{\beta}_i^2}{\hat{\gamma}_i^2}\right)^2}{\hat{\delta}_i^4} = 0$$

and

$$\rho_i := \frac{e^{-\frac{1}{\sqrt{\hat{\gamma}_i}}}}{\hat{\delta}_i\left(\frac{1}{\sqrt{\hat{\gamma}_i}} + a_i\right)\left(\frac{1}{\hat{\gamma}_i} + c_i + \frac{b_i}{\sqrt{\hat{\gamma}_i}}\right)}\ .$$

The optimal $\mathcal{H}^\infty$ performance cost $\hat{\gamma}_i$ is determined as the largest root of the equation

$$1 - \frac{\hat{\gamma}_i}{\hat{\delta}_i}e^{-s}s^2\frac{(s-k_i)}{(s+k_i)(s+a_i)(s^2+b_is+c_i)}\bigg|_{s=j/\sqrt{\hat{\gamma}_i}} = 0\ . \tag{41}$$

A time domain realization of $F_i(\hat{s}_i)$ given in (40) can be obtained by noting that

$$F_i(\zeta) = E_i\left(e^{-\zeta}I - e^{-A_i}\right)(\zeta I - A_i)^{-1}B_i \tag{42}$$

where

$$A_i = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \hat{\gamma}_i^{-2} & 0 & 0 & 0 \end{bmatrix} \qquad B_i = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \qquad E_i = \frac{\hat{\gamma}_i}{\hat{\delta}_i} \begin{bmatrix} 0 & 0 & k_i & -1 \end{bmatrix}.$$

This in turn shows that the impulse response of $F_i(\zeta)$ is restricted to the time interval $[0\ ,\ 1]$, and hence, $F_i(h_i s)$ may be realized as a finite impulse response (FIR) filter of duration h_i, with impulse response

$$f_i(t) = \begin{cases} -\frac{1}{h_i} E_i e^{\frac{1}{h_i} A_i (t-h_i)} B_i & \text{for } 0 \le t < h_i \\ 0 & \text{otherwise} \end{cases} .$$

Solution to (24) can now be obtained as $C_i(s) = \frac{1}{nh_i}\hat{C}_i(s)$, which leads to (25).

C Stable Implementation of the Controller

Once $C_i(s)$ is obtained as given in (25), $Q_i(s)$ can be obtained from (23). However, in order to avoid an unstable realization of $Y_i(s)$, $Q_i(s)$ must be implemented after making some factorizations in (23). From (18), $Y_i(s) = (1 - N_i(s)X_i(s))M_i^{-1}(s)$. Thus, using (23),

$$\begin{aligned} Q_i(s) &= [M_i(s) + C_i(s)N_i(s)]^{-1}[C_i(s)(1 - N_i(s)X_i(s)) - X_i(s)M_i(s)]\, M_i^{-1}(s) \\ &= \bar{Q}_i(s)M_i^{-1}(s) \end{aligned} \tag{43}$$

where $\bar{Q}_i(s) := [M_i(s) + C_i(s)N_i(s)]^{-1}[C_i(s)(1 - N_i(s)X_i(s)) - X_i(s)M_i(s)]$.
Using $P_i(s) = N_i(s)M_i^{-1}(s)$,

$$\begin{aligned} \bar{Q}_i(s) &= \left[1 + C_i(s)N_i(s)M_i^{-1}(s)\right]^{-1} M_i^{-1}(s)\,[C_i(s)(1 - N_i(s)X_i(s)) - X_i(s)M_i(s)] \\ &= [1 + C_i(s)P_i(s)]^{-1}\left[C_i(s)M_i^{-1}(s)(1 - N_i(s)X_i(s)) - X_i(s)\right] \\ &= [1 + C_i(s)P_i(s)]^{-1}\left[C_i(s)M_i^{-1}(s) - (1 + C_i(s)P_i(s))X_i(s)\right] \\ &= \frac{C_i(s)}{1 + C_i(s)P_i(s)} M_i^{-1}(s) - X_i(s)\ . \end{aligned} \tag{44}$$

Once $Q_i(s)$, thus $Q(s)$, is obtained, the controller $K(s)$ can be obtained by using (9). We can avoid the unstable realization of the $Y(s)$ in a similar way. By substituting (11) into (9), and using the equalities $M(s)Q(s) = Q(s)\tilde{M}(s)$, $N(s)M(s) = \tilde{M}(s)\tilde{N}(s)$ and $N(s) = \tilde{N}(s)$, $K(s)$ is obtained as

$$\begin{aligned} K(s) &= [X(s) + M(s)Q(s)]\left[\tilde{M}^{-1}(s)\left(1 - \tilde{N}(s)X(s)\right) - N(s)Q(s)\right]^{-1} \\ &= [X(s) + M(s)Q(s)]\left[\left(1 - \tilde{N}(s)X(s)\right) - \tilde{M}(s)N(s)Q(s)\right]^{-1}\tilde{M}(s) \\ &= [X(s) + M(s)Q(s)]\,[1 - N(s)(X(s) + M(s)Q(s))]^{-1}\,\tilde{M}(s) \\ &= Z(s)\,[1 - N(s)Z(s)]^{-1}\,\tilde{M}(s) \end{aligned} \tag{45}$$

where

$$Z(s) := X(s) + M(s)Q(s) \ . \tag{46}$$

By substituting (44) into (43), and then (43) into (46),

$$Z(s) = \begin{bmatrix} \frac{C_1(s)}{1+C_1(s)P_1(s)} \\ \vdots \\ \frac{C_n(s)}{1+C_n(s)P_n(s)} \end{bmatrix} \tilde{M}^{-1}(s) \ . \tag{47}$$

Then, using (47) in (45), the controller is determined as $K(s) = [K_1(s) \ \cdots \ K_n(s)]^T$, where $K_i(s)$ is as given in (26).

References

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

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

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

[4] Ataşlar, B., H. Özbay, and A. İftar, "Comparison of $\mathcal{H}^\infty$ and μ-synthesis based flow controllers for high-speed networks with multiple time-delays," *Proc. of the American Control Conference,* Arlington, VA, June 2001, pp. 3787–3788.

[5] Ataşlar, B., P. 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," *Proc. of the American Control Conference,* Chicago, IL, June 2000, pp. 2804–2808.

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

[7] Biberović, E., A. İftar, and H. Özbay, "A solution to the robust flow control problem using J-spectral factorizations," *Proc. of the Conference on Decision and Control*, Orlando, FL, December 2001. (To appear).

[8] Blanchini, F., R. Lo Cigno, R. Tempo, "Control of ATM networks: Fragility and robustness issues," *Proc. of the American Control Conference*, Philadelphia, PA, June 1998, pp. 2847–2851.

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

[10] Brakmo, L.S. and Peterson, L.L., "TCP Vegas: End-to-end Congestion Avoidance on a Global Internet," *IEEE Journal on Selected Areas in Communication (JSAC)*, Vol. 13, No. 8, October 1995, pp. 1465–1480.

[11] Doyle, J., B. Francis and A. Tannenbaum, *Feedback Control Theory*, McMillan, New York, NY, 1992.

[12] Floyd, S., "TCP and Explicit Congestion Notification," *ACM Computer Communication Review*, Vol. 24, No. 5, October 1994, pp. 10-23.

[13] Floyd, S., M. Handley, J. Padhye, and J. Widmer, "Equation-Based Congestion Control for Unicast Applications," *Proc. of SIGCOMM*, August 2000.

[14] Floyd, S. and T. Henderson, "The NewReno Modification to TCP's Fast Recovery Algorithm," *IETF Internet Experimental RFC 2582*, April 1999.

[15] Floyd, S. and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, Vol. 1, No. 4, August 1993, pp. 397–413.

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

[17] Gibbens, R.J. and F.P. Kelly, "Resource pricing and the evolution of congestion control," *Automatica*, Vol. 35, 1999, pp. 1969-1985.

[18] Jacobson, V., "Congestion avoidance and control," *Proc. of SIGCOMM*, August 1988, pp. 314–329.

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

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

[21] Kelly, F.P., A. Maulloo, and D. Tan, "Rate control in communication networks: shadow prices, proportional fairness and stability," *Journal of the Operational Research Society*, Vol. 49, 1998, pp. 237-252.

[22] Kojima A., K. Uchida, and E. Shimemura, "Robust stabilization of uncertain time delay systems via combined internal-external approach," *IEEE Trans. on Automatic Control*, vol. 38 (1993), pp. 373–378.

[23] Kunniyur, S. and R. Srikant, "End-to-end congestion control: utility functions, random losses and ECN marks," *Proc. of INFOCOM 2000*, Tel Aviv, Israel, April 2000.

[24] Kunniyur, S. and R. Srikant, "Analysis and Design of an Adaptive Virtual Queue Algorithm for Active Queue Management," *Proc. of SIGCOMM 2001*, August 2001.

[25] Low, S.H., "A Duality Model of TCP and Queue Management Algorithms," *Proceedings of ITC Specialist Seminar on IP Traffic Measurement, Modeling and Management*, September 2000, Monterey, CA.

[26] Low, S.H. and D.E. Lapsley, "Optimization Flow Control, I: Basic Algorithm and Convergence," *IEEE/ACM Transactions on Networking*, Vol. 7, No. 6, Dec. 1999, pp. 861–875.

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

[28] Massoulie, L. and J. Roberts, "Bandwidth sharing: Objectives and algorithms," *Proc. of INFOCOM'99*, March 1999, pp 1395–1403.

[29] Mathis, M., J. Mahdavi, S. Floyd, and A. Romanow, "TCP Selective Acknowledgement Options," *IETF Internet RFC 2018*, October 1996.

[30] Niculescu, S-I., J. M. Dion, L. Dugard, "Robust stabilization for uncertain time delay systems containing saturating actuators," *IEEE Trans. on Automatic Control*, vol. 41 (1996), pp. 742–746.

[31] Özbay, H., S. Kalyanaraman, A. İftar, "On rate-based congestion control in high-speed networks: Design of an H^∞ based flow controller for single bottleneck," *Proc. of the American Control Conference*, Philadelphia, PA, June 1998, pp. 2376–2380.

[32] Özbay, H., T. Kang, S. Kalyanaraman, A. İftar, "Performance and robustness analysis of an $\mathcal{H}^\infty$ based flow controller," *Proc. of the Conference on Decision and Control*, Phoenix, AZ, December 1999, pp. 2691–2696.

[33] Parekh, A. K., and R. 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 (1994), pp. 137–150.

[34] Quet, P., S. Ramakrishnan, H. Özbay, S. Kalyanaraman, "On the $\mathcal{H}^\infty$ controller design for congestion control with a capacity predictor," *Proc. of the Conference on Decision and Control*, Orlando, FL, December 2001. (To appear).

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

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

[37] Rosen, E., A. Viswanathan, and R. Callon, "Multiprotocol Label Switching Architecture," *IETF Internet RFC 3031*, January 2001.

[38] Stepan, G., *Retarded dynamical systems: stability and characteristic functions*, Longman Scientific & Technical, Wiley, New York, 1989.

[39] Toker, O., and H. Özbay, "$\mathcal{H}^\infty$ Optimal and suboptimal controllers for infinite dimensional SISO plants," *IEEE Trans. on Automatic Control*, vol. 40 (1995), pp. 751–755.

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

[41] Zhou, K., J.C. Doyle, and K. Glover, *Robust and Optimal Control*, Prentice-Hall, 1996.