DECENTRALIZED ROBUST FLOW CONTROLLER DESIGN FOR NETWORKS WITH MULTIPLE BOTTLENECKS

İnci Munyas **Özen Yelbaşı** **Altuğ İftar**

Department of Electrical and Electronics Engineering, Anadolu University, Eskişehir 26470, Turkey

{imunyas , oyelbasi , aiftar}@anadolu.edu.tr

Fax: 90 222 323 9501

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

Abstract

Decentralized rate-based flow controller design in multi-bottleneck networks is considered. An $\mathcal{H}^\infty$ problem is formulated to find decentralized controllers which are robust to time-varying uncertain multiple time-delays in different channels. A suboptimal solution to this problem is found and the implementation of the decentralized controllers at different bottleneck nodes is presented. Besides robustness, the controllers also satisfy tracking and weighted fairness requirements. A number of simulations are also included to illustrate the time-domain performance of the proposed controllers.

1 Introduction

A communication network should be able to deliver data packets with minimum loss and as soon as possible to the destination. To achieve this aim the resources of the network should be used efficiently. Flow control is one of the basic resource management tools in high-speed communication networks. Flow control always involves a direct feedback from the receiver (bottleneck nodes) to the sender (sources and the other nodes sending data to that node). The type of the flow control is determined by the type of the feedback scheme used; i.e., it can be either *rate-based* or *window-based*. However, for both cases, the existence of the time-delays in the data-flow makes the design of the controller a difficult task. Since the controller is to be implemented at the bottleneck node, the time a command signal for a rate is issued and the time this rate is set are different, which corresponds to the *backward delay* from the bottleneck node to the source node. Besides, another time-delay occurs which is the time needed for the data packets sent with the new rate to reach the bottleneck node; i.e. the *forward delay*, from the source node to the bottleneck node. The *round-trip delay* from the control input to the regulated output is the sum of these two delays. It should also be considered that these time-delays are usually uncertain and time-varying. Since there usually are more than one source and node connected to a bottleneck, these delays are multiple. Many approaches to the flow controller design problem have been presented in the literature (see $[1, 4]$ and references therein). Among these, the time-varying uncertainties in the time-delays have been explicitly considered in [4] and a rate-based flow controller have been designed, using $\mathcal{H}^\infty$ methods, which is robust to uncertain time-varying multiple time-delays in different channels. In [4], however, only the single-bottleneck case has been considered. The multi-bottleneck case has been considered in [1], where it has been shown that decentralized flow controllers can be designed to solve the same problem in this case. The controller derivation, however, has not been considered in [1]. In the present paper, we re-present the problem of designing decentralized rate-based flow controllers, which are robust against time-varying uncertainties in time-delays in different channels, in the case of multi-bottleneck networks. We show the derivation of the controllers and present their implementation. We also implement the proposed controllers and present a number of simulation results to show the time-domain performance of the proposed controllers in certain realistic cases.

2 Problem Statement

In this work, we consider a network which consists of n bottleneck nodes and n_i sources feeding the i^{th} bottleneck node. Note that, if any physical source sends data to more than one bottleneck node, this source may be considered as a different source for each bottleneck node for the purpose of controller design. We also assume that, besides the sources, each bottleneck can also send data through other bottlenecks; i.e., each bottleneck can also be a source for the other bottlenecks. The problem is to design a decentralized controller to be implemented at each bottleneck node, to regulate the data sending rates of the sources and the other bottleneck nodes to that node.

The dynamics of the queue length at the i^{th} bottleneck node can be described as

$$\dot{q}_i\left(t\right) = \sum_{k=1}^{n_i} r^b_{i,k}\left(t\right) + \sum_{k=1,\ k\neq i}^{n} \rho^b_{k,i}\left(t\right) - c_i\left(t\right) - \sum_{k=1,\ k\neq i}^{n} \rho^s_{i,k}\left(t\right) \quad , \tag{1}$$

where $q_i(t)$ is the queue length at the i^{th} bottleneck node at time t ($i = 1, 2, \ldots, n$); $r^b_{i,k}\left(t\right)$ is the rate of data received at the i^{th} bottleneck node from the k^{th} source of the i^{th} bottleneck node at time t ($i = 1, 2, \ldots, n$, $k = 1, 2, \ldots, n_i$); $\rho^b_{k,i}\left(t\right)$ is the rate of data received at the i^{th} bottleneck node at time t from the k^{th} bottleneck node ($i = 1, 2, \ldots, n$, $k = 1, 2, \ldots, n$, $i \neq k$); $c_i(t)$ is the outgoing flow rate, except for the flow going to the other bottleneck nodes, of the i^{th} bottleneck node at time t ($i = 1, 2, \ldots, n$); and $\rho^s_{i,k}\left(t\right)$ is the rate of data sent from the i^{th} to the k^{th} bottleneck node at time t ($i = 1, 2, \ldots, n$, $k = 1, 2, \ldots, n$, $i \neq k$). Since there is a time-varying backward delay, $\phi^b_{i,k}\left(t\right)$, between the k^{th} and the i^{th} bottleneck nodes, we have $\rho^s_{i,k}(t) = \rho_{i,k}(t - \phi^b_{i,k}(t))$, where $\rho_{i,k}(t)$ is the flow rate command at time t for the flow from the i^{th} to the k^{th} bottleneck node ($i = 1, 2, \ldots, n$, $k = 1, 2, \ldots, n$, $i \neq k$), which is to be computed (by the controller to be designed) at the k^{th} bottleneck node. Data receiving rates, $r^b_{i,k}(t)$ and $\rho^b_{k,i}(t)$, on the other hand, can be found as (see $[4, 1]$)

$$r^b_{i,k}(t) = \begin{cases} (1-\dot{\delta}^{rf}_{i,k}(t)) r_{i,k}(t-\tau^f_{i,k}(t)), & t-\tau^f_{i,k}(t) \geq 0 \\ 0, & t-\tau^f_{i,k}(t) < 0 \end{cases} \tag{2}$$

$$\rho^b_{i,k}(t) = \begin{cases} (1-\dot{\delta}^{\rho f}_{i,k}(t)) \rho_{i,k}(t-\phi^f_{i,k}(t)), & t-\phi^f_{i,k}(t) \geq 0 \\ 0, & t-\phi^f_{i,k}(t) < 0 \end{cases} \tag{3}$$

where $r_{i,k}(t)$ is the flow rate command at time t for the flow from the k^{th} source of the i^{th} bottleneck node to the i^{th} bottleneck node ($i = 1, 2, \ldots, n$, $k = 1, 2, \ldots, n_i$), which is to be computed (by the controller to be designed) at the i^{th} bottleneck node; $\tau^f_{i,k}(t)$ is the forward time-delay from the k^{th} source of the i^{th} bottleneck node to the i^{th} bottleneck node at time t ($i = 1, 2, \ldots, n, k = 1, 2, \ldots, n_i$) and $\delta^{rf}_{i,k}(t)$ is its time-varying uncertain part (i.e., $\tau^f_{i,k}(t) = h^{rf}_{i,k} + \delta^{rf}_{i,k}(t)$, where $h^{rf}_{i,k}$ is the time-invariant nominal part); and $\phi^f_{i,k}(t)$ is the forward time-delay from the i^{th} to the k^{th} bottleneck node at time t ($i = 1, 2, \ldots, n$, $k = 1, 2, \ldots, n$, $i \neq k$) and $\delta^{\rho f}_{i,k}(t)$ is its time-varying uncertain part (i.e., $\phi^f_{i,k}(t) = h^{\rho f}_{i,k} + \delta^{\rho f}_{i,k}(t)$, where $h^{\rho f}_{i,k}$ is the time-invariant nominal part).

The round-trip delay at time t for the flow from the k^{th} source of the i^{th} bottleneck node to the i^{th} bottleneck node is given as $\tau_{i,k}(t) = \tau^b_{i,k}(t) + \tau^f_{i,k}(t) = h^r_{i,k} + \delta^r_{i,k}(t)$, where $h^r_{i,k}$ is the time-invariant nominal part and $\delta^r_{i,k}(t)$ is the time-varying uncertain part. Similarly, the round-trip delay at time t for the flow from the i^{th} to the k^{th} bottleneck node is given as $\phi_{i,k}(t) = \phi^b_{i,k}(t) + \phi^f_{i,k}(t) = h^\rho_{i,k} + \delta^\rho_{i,k}(t)$, where $h^\rho_{i,k}$ is the time-invariant nominal part and $\delta^\rho_{i,k}(t)$ is the time-varying uncertain part. It is assumed that the uncertainties satisfy the following;

$$\left|\delta^r_{i,j}(t)\right| < \delta^{r+}_{i,j}\,, \quad \left|\dot{\delta}^r_{i,j}(t)\right| < \beta^r_{i,j}\,, \quad \left|\dot{\delta}^{rf}_{i,j}(t)\right| < \beta^{rf}_{i,j}\,, \tag{4}$$

$$\left|\delta^\rho_{i,k}(t)\right| < \delta^{\rho+}_{i,k}\,, \quad \left|\delta^{\rho b}_{i,k}(t)\right| < \delta^{\rho b+}_{i,k}\,, \quad \left|\dot{\delta}^\rho_{i,k}(t)\right| < \beta^\rho_{i,k}\,, \quad \left|\dot{\delta}^{\rho f}_{i,k}(t)\right| < \beta^{\rho f}_{i,k}\,, \tag{5}$$

for all t, for some $\delta^{r+}_{i,j} > 0$, $0 < \beta^{rf}_{i,j} < \beta^r_{i,j} < 1$, $0 < \delta^{\rho b+}_{i,k} < \delta^{\rho+}_{i,k}$, $0 < \beta^{\rho f}_{i,k} < \beta^\rho_{i,k} < 1$ ($i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, n_i$, $k = 1, 2, \ldots, n$, $k \neq i$).

It should be noted that, in a real application, there also exist some *hard constraints*, such as non-negativity constraints and upper bounds on the queue lengths and on the data rates. However, in this work, we will assume that these hard constraints are always satisfied.

It can be shown that (see $[1, 2]$) our system is captured by the fictitious system shown in Figure 1, where

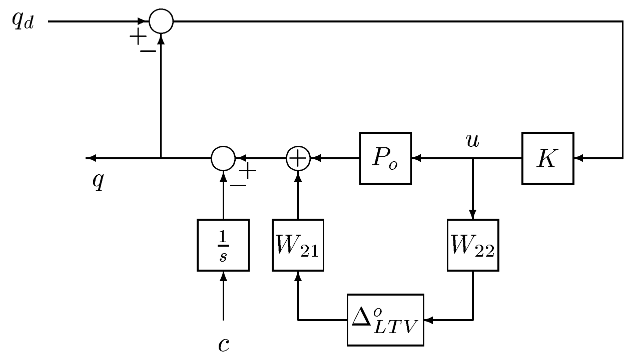

Figure 1: Fictitious system.

$q := \begin{bmatrix} q_1 & \cdots & q_n \end{bmatrix}^T$, where superscript T denotes the transpose, $q_d := \begin{bmatrix} q_{d,1} & \cdots & q_{d,n} \end{bmatrix}^T$, where $q_{d,i}$ is the desired queue length at the i^{th} bottleneck node ($i = 1, \ldots n$), $c := \begin{bmatrix} c_1 & \cdots & c_n \end{bmatrix}^T$, $u := \begin{bmatrix} r^T & \rho^T \end{bmatrix}^T$, where $r := \begin{bmatrix} r_1^T & \cdots & r_n^T \end{bmatrix}^T$, where $r_i := \begin{bmatrix} r_{i,1} & \cdots & r_{i,n_i} \end{bmatrix}^T$, $\rho := \begin{bmatrix} \rho_1^T & \cdots & \rho_n^T \end{bmatrix}^T$, where $\rho_i := \begin{bmatrix} \rho_{1,i} & \cdots & \rho_{i-1,i} & \rho_{i+1,i} & \cdots & \rho_{n,i} \end{bmatrix}^T$, $P_o(s) = \frac{1}{s}\bar{P}e^{-Hs}\hat{P}$ is the nominal plant, where $\hat{P} := \begin{bmatrix} I_m & 0 \\ 0 & \frac{1}{\sqrt{2}} I_{n(n-1)} \\ 0 & -\frac{1}{\sqrt{2}} I_{n(n-1)} \end{bmatrix}$, where I_k denotes the $k \times k$ identity matrix and $m := \sum_{i=1}^n n_i$, $H := \mathrm{diag}(\bar{h})$, where $\bar{h} := \begin{bmatrix} h^r & h^\rho & h^{\rho b} \end{bmatrix}$, where $h^r := \begin{bmatrix} h^r_1 & \cdots & h^r_n \end{bmatrix}$, where $h^r_i := \begin{bmatrix} h^r_{i,1} & \cdots & h^r_{i,n_i} \end{bmatrix}$, $h^\rho := \begin{bmatrix} h^\rho_1 & \cdots & h^\rho_n \end{bmatrix}$, where $h^\rho_i := \begin{bmatrix} h^\rho_{1,i} & \cdots & h^\rho_{i-1,i} & h^\rho_{i+1,i} & \cdots & h^\rho_{n,i} \end{bmatrix}$, $h^{\rho b}$ and $h^{\rho b}_i$ has the same structure as h^ρ and h^ρ_i, respectively. $\bar{P} := \begin{bmatrix} \bar{P}_r & \sqrt{2}\bar{P}_\rho & \sqrt{2}\bar{P}_{\rho b} \end{bmatrix}$, where $\bar{P}_r := \mathrm{blockdiag}\left(\mathbf{1}_{n_1}, \ldots, \mathbf{1}_{n_n}\right)$, where $\mathbf{1}_k$ denotes the $1 \times k$ dimensional row vector of 1's, $\bar{P}_\rho := I_n \otimes \mathbf{1}_{n-1}$,

$$\bar{P}_{\rho b} := \begin{bmatrix} 0 & J_1 & J_1 & J_1 & \cdots & J_1 & J_1 \\ J_1 & 0 & J_2 & J_2 & \cdots & J_2 & J_2 \\ J_2 & J_2 & 0 & J_3 & \cdots & J_3 & J_3 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ J_{n-1} & J_{n-1} & J_{n-1} & J_{n-1} & \cdots & J_{n-1} & 0 \end{bmatrix},$$

where $J_i := \begin{bmatrix} 0_{1\times(i-1)} & 1 & 0_{1\times(n-i-1)} \end{bmatrix}$, $W_{22} := \begin{bmatrix} I_m & 0 \\ 0 & \frac{1}{\sqrt{2}} I_{n(n-1)} \\ 0 & \frac{1}{\sqrt{2}} I_{n(n-1)} \end{bmatrix}$, W_{21} is a weighting matrix which depends on the bounds in (4)–(5), K is the controller to be designed, and Δ^o_{LTV} represents an arbitrary linear time-varying system with $\mathcal{L}^2$-induced norm less than 1 (see [2] for details).

For the system shown in Figure 1 to be robustly stable for all $\|\Delta^o_{LTV}\| < 1$, K should stabilize P_o and

$$\left\| W_{22} K \left(I + P_o K\right)^{-1} W_{21} \right\|_\infty \leq 1 \tag{6}$$

should be satisfied. It can be shown that (see [2] for details)

(6) is satisfied if

$$\left\|\xi \hat{P} K\left(I+P_o K\right)^{-1}\right\|_{\infty} \leq 1 \tag{7}$$

where $\xi(s) := \frac{1}{s}\xi_1 + \xi_2$, where ξ_1 and ξ_2 are constants which depend on the bounds in (4)–(5) (see [2]).

Next, as in [4], to guarantee tracking ($\lim_{t\to\infty} q_i(t) = q_{d,i}$) and *good* transient response, we formulate the problem

$$\text{minimize} \quad \left\|W_1\left(I+P_o K\right)^{-1}\right\|_{\infty} \tag{8}$$

over all controllers K stabilizing P_o, where $W_1(s) = \frac{1}{s^2}$. Then, by combining the robust stability (7) and nominal performance (8) conditions we define the following two-block $\mathcal{H}^{\infty}$ optimization problem:

$$\inf_{K \text{ stabilizing } P_o} \left\| \begin{bmatrix} W_1 \ (I+P_o K)^{-1} \\ \xi \hat{P} K\,(I+P_o K)^{-1} \end{bmatrix} \right\|_{\infty} =: \gamma^{\text{opt}} . \tag{9}$$

3 Solution

To find a solution to the optimization problem (9), let $\hat{K} := \hat{P} K$ and note that $P_o K = \breve{P}\hat{K}$, where $\breve{P}(s) := \frac{1}{s}\bar{P}e^{-Hs} = \begin{bmatrix} P^r(s) & P^{\rho}(s) & P^{\rho b}(s) \end{bmatrix}$, where $P^r(s) := \frac{1}{s}\bar{P}_r e^{-H^r s}$, $P^{\rho}(s) := \frac{\sqrt{2}}{s}\bar{P}_{\rho} e^{-H^{\rho} s}$, $P^{\rho b}(s) := \frac{\sqrt{2}}{s}\bar{P}_{\rho b} e^{-H^{\rho b} s}$, where $H^r := \operatorname{diag}(h^r)$, $H^{\rho} := \operatorname{diag}(h^{\rho})$, and $H^{\rho b} := \operatorname{diag}(h^{\rho b})$. Also let

$$Q := \hat{K}\left(I + \breve{P}\hat{K}\right)^{-1} =: \begin{bmatrix} Q^r \\ Q^{\rho} \\ Q^{\rho b} \end{bmatrix} \tag{10}$$

where the partitioning of Q corresponds to the partitioning of $\bar{P}^T$.

Note that, P^r and P^{ρ} are block diagonal. Although $P^{\rho b}$ is not block diagonal, by permuting the columns of $I_{n(n-1)}$, we can find a non-singular matrix T, satisfying $T^T T = T T^T = I$, such that $\hat{P}^{\rho b} := P^{\rho b} T^T$ is block diagonal. Then, by defining $\hat{Q}^{\rho b} := T Q^{\rho b}$ (thus $P^{\rho b} Q^{\rho b} = \hat{P}^{\rho b}\hat{Q}^{\rho b}$), the optimization problem (9) can be rewritten as:

$$\begin{aligned} &\inf_{Q\in\mathcal{H}^{\infty}} \left\| \begin{bmatrix} W_1 \ \left(I - \breve{P}Q\right) \\ \xi Q \end{bmatrix} \right\|_{\infty} \\ &= \inf_{Q\in\mathcal{H}_{\infty}} \left\| \begin{bmatrix} W_1(I - P^r Q^r - P^{\rho} Q^{\rho} - \hat{P}^{\rho b}\hat{Q}^{\rho b}) \\ \xi Q \end{bmatrix} \right\|_{\infty} = \gamma^{\text{opt}} \end{aligned} \tag{11}$$

In order to obtain a decentralized structure for K, we impose block diagonal structures $Q^r = \operatorname{blockdiag}(Q_1^r, \ldots, Q_n^r)$, $Q^{\rho} = \operatorname{blockdiag}(Q_1^{\rho}, \ldots, Q_n^{\rho})$, $\hat{Q}^{\rho b} = \operatorname{blockdiag}(\hat{Q}_1^{\rho b}, \ldots, \hat{Q}_n^{\rho b})$, where the structures correspond to the structures of P^{rT}, $P^{\rho T}$, $\hat{P}^{\rho b\,T}$, respectively. On noting that $\|\hat{Q}^{\rho b}\|_{\infty} = \|T Q^{\rho b}\|_{\infty}$, problem (11) now decomposes into the following problems:

$$\inf_{\hat{Q}_i\in\mathcal{H}_{\infty}} \left\| \begin{bmatrix} W_1(1 - P_i^r Q_i^r - P_i^{\rho} Q_i^{\rho} - \hat{P}_i^{\rho b}\hat{Q}_i^{\rho b}) \\ \xi \begin{bmatrix} Q_i^r \\ 0 \\ 0 \end{bmatrix} + \xi \begin{bmatrix} 0 \\ Q_i^{\rho} \\ 0 \end{bmatrix} + \xi \begin{bmatrix} 0 \\ 0 \\ \hat{Q}_i^{\rho b} \end{bmatrix} \end{bmatrix} \right\|_{\infty} =: \gamma_i^{\text{opt}} \tag{12}$$

for $i = 1, \ldots, n$, where $\hat{Q}_i := \begin{bmatrix} Q_i^{rT} & Q_i^{\rho T} & \hat{Q}_i^{\rho b\,T} \end{bmatrix}^T$. Note that $\gamma^{\text{opt}} = \max_i(\gamma_i^{\text{opt}})$, which means that an optimal (respectively, suboptimal) solution to (11) (equivalently to (9)) is obtained by combining the optimal (respectively, suboptimal) solutions to the problems in (12).

The problems defined in (12) are similar to the problem considered in [4]. As discussed in [4], due to the multiple delays involved, there is no known approach to find optimal solutions to the problems in (12) (e.g., see [3]). Therefore, as it was done in [4], we will decompose the problems (12) into subproblems involving single delays and find a suboptimal solution to each problem in (12). For this, we introduce positive scalars, $\alpha_{i,j}^r$, $\alpha_{k,i}^{\rho}$, and $\alpha_{i,k}^{\rho b}$ ($i = 1, \ldots, n$, $j = 1, \ldots, n_i$, $k = 1, \ldots, n$, $i \neq k$), satisfying

$$\sum_{j=1}^{n_i} \alpha_{i,j}^r + \sum_{j=1, j\neq i}^{n} \alpha_{j,i}^{\rho} + \sum_{j=1, j\neq i}^{n} \alpha_{i,j}^{\rho b} = 1 , \qquad i = 1, \ldots, n . \tag{13}$$

The left-hand side of (12) can now be written as

$$\begin{aligned} \inf_{\hat{Q}_i\in\mathcal{H}_{\infty}} \Bigg\| &\sum_{j=1}^{n_i} \begin{bmatrix} W_1 \alpha_{i,j}^r (1 - \frac{1}{\alpha_{i,j}^r} P_{i,j}^r Q_{i,j}^r) \\ \xi Q_{i,j}^r \end{bmatrix} + \\ &\sum_{j=1, j\neq i}^{n} \begin{bmatrix} W_1 \alpha_{j,i}^{\rho} (1 - \frac{1}{\alpha_{j,i}^{\rho}} P_{j,i}^{\rho} Q_{j,i}^{\rho}) \\ \xi Q_{j,i}^{\rho} \end{bmatrix} + \\ &\sum_{j=1, j\neq i}^{n} \begin{bmatrix} W_1 \alpha_{i,j}^{\rho b} (1 - \frac{1}{\alpha_{i,j}^{\rho b}} P_{i,j}^{\rho b} Q_{i,j}^{\rho b}) \\ \xi Q_{i,j}^{\rho b} \end{bmatrix} \Bigg\|_{\infty} \end{aligned} \tag{14}$$

where $P_i^r =: \begin{bmatrix} P_{i,1}^r & \cdots & P_{i,n_i}^r \end{bmatrix}$, $P_i^{\rho} =: \begin{bmatrix} P_{1,i}^{\rho} & \cdots & P_{i-1,i}^{\rho} & P_{i+1,i}^{\rho} & \cdots & P_{n,i}^{\rho} \end{bmatrix}$, $\hat{P}_i^{\rho b} =: \begin{bmatrix} P_{i,1}^{\rho b} & \cdots & P_{i,i-1}^{\rho b} & P_{i,i+1}^{\rho b} & \cdots & P_{i,n}^{\rho b} \end{bmatrix}$, and Q_i^r, Q_i^{ρ} and $\hat{Q}_i^{\rho b}$ has the same structure as P_i^r, P_i^{ρ} and $\hat{P}_i^{\rho b}$ respectively (with P replaced by Q). Therefore, we define the following problems, each of which involves a single delay

$$\inf_{Q_{i,j}^r\in\mathcal{H}_{\infty}} \left\| \begin{bmatrix} W_1 \alpha_{i,j}^r (1 - \frac{1}{\alpha_{i,j}^r} P_{i,j}^r Q_{i,j}^r) \\ \xi Q_{i,j}^r \end{bmatrix} \right\|_{\infty} =: \gamma_{i,j}^r \tag{15}$$

$$\inf_{Q_{j,i}^{\rho}\in\mathcal{H}_{\infty}} \left\| \begin{bmatrix} W_1 \alpha_{j,i}^{\rho} (1 - \frac{1}{\alpha_{j,i}^{\rho}} P_{j,i}^{\rho} Q_{j,i}^{\rho}) \\ \xi Q_{j,i}^{\rho} \end{bmatrix} \right\|_{\infty} =: \gamma_{i,j}^{\rho} \tag{16}$$

$$\inf_{Q_{i,j}^{\rho b}\in\mathcal{H}_{\infty}} \left\| \begin{bmatrix} W_1 \alpha_{i,j}^{\rho b} (1 - \frac{1}{\alpha_{i,j}^{\rho b}} P_{i,j}^{\rho b} Q_{i,j}^{\rho b}) \\ \xi Q_{i,j}^{\rho b} \end{bmatrix} \right\|_{\infty} =: \gamma_{i,j}^{\rho b} \tag{17}$$

where $j = 1, \ldots, n_i$ for the problem defined in (15), $j = 1, \ldots, n$, $j \neq i$ for the problems defined in (16) and (17). Note that, a suboptimal solution to (12) can be obtained by combining optimal solutions of (15)–(17), since $\gamma_i^{\text{opt}} \leq \sum_{j=1}^{n_i} \gamma_{i,j}^r + \sum_{j=1, j\neq i}^{n} \gamma_{j,i}^{\rho} + \sum_{j=1, j\neq i}^{n} \gamma_{i,j}^{\rho b}$.

Using the results of [5], the optimal solution to each of the problems in (15)–(17) can be described as $Q_{i,j}^{\cdot} = \frac{C_{i,j}^{\cdot}}{1+\frac{1}{\alpha_{i,j}^{\cdot}} C_{i,j}^{\cdot} P_{i,j}^{\cdot}}$ and $C_{i,j}^{\cdot}(s) = \kappa_{i,j}^{\cdot}\left(\frac{s h_{i,j}^{\cdot} - k_{i,j}^{\cdot}}{s h_{i,j}^{\cdot}}\right) \frac{1}{1 + F_{i,j}^{\cdot}(s h_{i,j}^{\cdot})}$, where the superscript $\cdot$ represents r, ρ or ρb, where $F^{\cdot}_{i,j}$ is a finite impulse response filter and $\kappa^{\cdot}_{i,j}$ and $k^{\cdot}_{i,j}$ are constants to be calculated as in [4] (we can not show the details here due to space limitations). $Q^{\cdot}_{i,j}$'s are now substituted back to obtain Q. Once Q is found, from (10), we obtain $\hat{K} = \begin{bmatrix} \hat{K}^r \\ \hat{K}^\rho \\ \hat{K}^{\rho b} \end{bmatrix} = Q[I - \check{P}Q]^{-1}$. Then, our controller is obtained as $K = P^\dagger \hat{K}$, where $P^\dagger$ is a left inverse of $\hat{P}$. Using $P^\dagger = \begin{bmatrix} I_m & 0 & 0 \\ 0 & \sqrt{2}I_{n(n-1)} & 0 \end{bmatrix}$, we obtain

$$K = \begin{bmatrix} \hat{K}^r \\ \sqrt{2}\hat{K}^\rho \end{bmatrix} = \left[\begin{array}{ccc} K^r_{11} & & \\ \vdots & 0 & \\ K^r_{1n_1} & & \\ & \ddots & \\ & & K^r_{n1} \\ & 0 & \vdots \\ & & K^r_{nn_n} \\ K^\rho_{21} & & \\ \vdots & 0 & \\ K^\rho_{n1} & & \\ & \ddots & \\ & & K^\rho_{1n} \\ & 0 & \vdots \\ & & K^\rho_{(n-1)n} \end{array}\right] \tag{18}$$

where

$$K^r_{i,j} = \frac{C^r_{i,j}}{1+\frac{1}{\alpha^r_{i,j}}C^r_{i,j}P^r_{i,j}} \left(\begin{array}{c} 1 - \sum_{k=1}^{n_i} \frac{C^r_{i,k}P^r_{i,k}}{1+\frac{1}{\alpha^r_{i,k}}C^r_{i,k}P^r_{i,k}} - \\ \sum_{k=1,k\neq i}^{n} \frac{C^\rho_{k,i}P^\rho_{k,i}}{1+\frac{1}{\alpha^\rho_{k,i}}C^\rho_{k,i}P^\rho_{k,i}} - \\ \sum_{k=1,k\neq i}^{n} \frac{C^{\rho b}_{i,k}P^{\rho b}_{i,k}}{1+\frac{1}{\alpha^{\rho b}_{i,k}}C^{\rho b}_{i,k}P^{\rho b}_{i,k}} \end{array} \right)^{-1} \tag{19}$$

$$K^\rho_{j,i} = \frac{\sqrt{2}C^\rho_{j,i}}{1+\frac{1}{\alpha^\rho_{j,i}}C^\rho_{j,i}P^\rho_{j,i}} \left(\begin{array}{c} 1 - \sum_{k=1}^{n_i} \frac{C^r_{i,k}P^r_{i,k}}{1+\frac{1}{\alpha^r_{i,k}}C^r_{i,k}P^r_{i,k}} - \\ \sum_{k=1,k\neq i}^{n} \frac{C^\rho_{k,i}P^\rho_{k,i}}{1+\frac{1}{\alpha^\rho_{k,i}}C^\rho_{k,i}P^\rho_{k,i}} - \\ \sum_{k=1,k\neq i}^{n} \frac{C^{\rho b}_{i,k}P^{\rho b}_{i,k}}{1+\frac{1}{\alpha^{\rho b}_{i,k}}C^{\rho b}_{i,k}P^{\rho b}_{i,k}} \end{array} \right)^{-1} \tag{20}$$

As seen from (18), the part of the controller for the i^{th} bottleneck node gets feedback only from q_i to regulate the queue length q_i by determining the flow rates $r_{i,j}$, $j = 1,\ldots,n_i$, and $\rho_{k,i}$, $k = 1,\ldots,n, k \neq i$. Therefore, the controller is composed of n decentralized controllers each of which can be implemented at the corresponding bottleneck node as shown in Figure 2. This controller stabilizes the nominal plant and makes the $\mathcal{H}_\infty$ norm of the matrix in (9) less than some $\tilde{\gamma}$ (an upper bound that can be found from the γ's of the suboptimal problems). Thus, as long as the hard constraints are satisfied, the controller stabilizes the actual plant for all variations of the time-delays

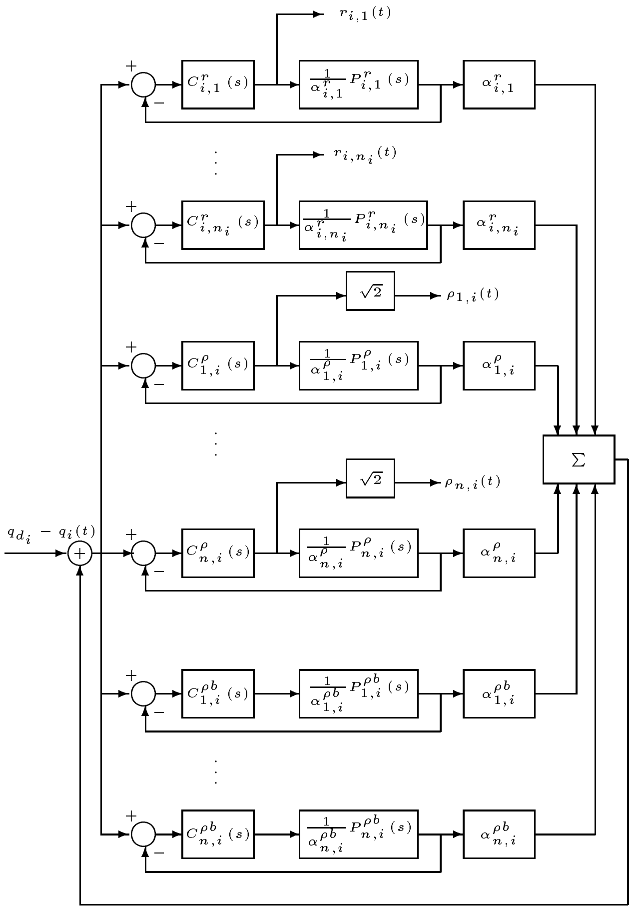

Figure 2: The implementation of the controller K_i.

satisfying $\|\delta^r_{i,j}(t)\| < \frac{\delta^{r+}_{i,j}}{\tilde{\gamma}}$, $\|\dot{\delta}^r_{i,j}(t)\| < \frac{\beta^r_{i,j}}{\tilde{\gamma}}$, $\|\dot{\delta}^{rf}_{i,j}(t)\| < \frac{\beta^{rf}_{i,j}}{\tilde{\gamma}}$, $\|\delta^\rho_{j,i}(t)\| < \frac{\delta^{\rho+}_{j,i}}{\tilde{\gamma}}$, $\|\delta^{\rho b}_{i,j}(t)\| < \frac{\delta^{\rho b+}_{i,j}}{\tilde{\gamma}}$, $\|\dot{\delta}^\rho_{j,i}(t)\| < \frac{\beta^\rho_{j,i}}{\tilde{\gamma}}$, and $\|\dot{\delta}^{\rho f}_{j,i}(t)\| < \frac{\beta^{\rho f}_{j,i}}{\tilde{\gamma}}$.

Furthermore, assuming that the hard constraints are satisfied, that all the delays converge to constant values, and that the outgoing flow rates converge to constant values ($\lim_{t\to\infty} c_i(t) = c_{i,\infty}$, $i = 1,\ldots,n$), the above controller also guarantees tracking, i.e., $\lim_{t\to\infty} q_i(t) = q_{d,i}$, $i = 1,\ldots,n$. It can also be shown that (we will not show the details here due to space limitations, see [4] for the development in the single-bottleneck case) the controller satisfies a *weighted fairness* condition, where weights are defined by the scalars introduced in (13). Specifically, assuming that the hard constraints are satisfied, that all the delays converge to constant values, and that the outgoing flow rates converge to constant values, we have

$$\lim_{t\to\infty} r_{i,j}(t) = \frac{\alpha^r_{i,j}}{\bar{\alpha}_i} \left(\lim_{t\to\infty} c_i(t) + \sum_{k=1,k\neq i}^{n} \lim_{t\to\infty} \rho_{i,k}(t) \right) , \tag{21}$$

for $j = 1, \ldots, n_i$, and

$$\lim_{t\to\infty} \rho_{j,i}(t) = \frac{\sqrt{2}\alpha^{\rho}_{j,i}}{\bar{\alpha}_i} \left(\lim_{t\to\infty} c_i(t) + \sum_{k=1,k\neq i}^{n} \lim_{t\to\infty} \rho_{i,k}(t) \right) , \tag{22}$$

for $j = 1, \ldots, n$, $j \neq i$, where, for $i = 1, \ldots, n$, $\bar{\alpha}_i := \sum_{j=1}^{n_i} \alpha^r_{i,j} + \sqrt{2} \sum_{j=1, j\neq i}^{n} \alpha^{\rho}_{j,i}$.

4 Examples

We have implemented the controller as derived above for various different networks and obtained simulations for many cases using SIMULINK. The hard constraints are also considered in the simulations. Here we show only a few examples due to space limitations. We consider a network with three bottleneck nodes. We have $n_1 = 2$, $n_2 = 3$, and $n_3 = 4$. Controller parameters and the nominal parts of the actual delays are as shown in Table 1. In the simulations, the uncertain parts of forward actual delays, $\delta^{rf}_{i,j}(t)$ and $\delta^{\rho f}_{i,j}(t)$, are taken as $0.03sin(\frac{\pi}{50}t)$, while the uncertain parts of backward actual delays, $\delta^{rb}_{i,j}(t)$ and $\delta^{\rho b}_{i,j}(t)$, are taken as $0.3sin(\frac{\pi}{50}t)$. The desired queue lengths ($q_{d,i}$'s) for the three bottleneck nodes are the same and equal to 50 packets. The capacities of the outgoing links (c_i's) are the same and equal to 100 packets/s for all the cases except for case 3. Figures 3, 4 and 5 are the simulation results for the cases 1, 2 and 3, respectively. For all cases, the plots in (a), (b) and (c) represent the queue lengths and flow rates versus time (in s) of the sources of bottleneck nodes 1, 2 and 3, respectively, while (d) represents the flow rates versus time (in s) between the bottleneck nodes.

Case 1: The queue size at bottleneck node 1 is zero between 0 and 50 s, this period extends to about 65 s for bottleneck nodes 2 and 3. These time ranges define the period needed for the sum of the incoming flows to exceed the capacity of the outgoing link at the bottleneck node. Note that, the steady–state flow rates satisfy fairness conditions (21)–(22).

Case 2: In this case, data supplying rates of the sources are limited as shown in Table 2. The flow rates at sources 12, 23 and 34 are saturated. Note that the controllers successfully redistribute the unused rates to the remaining sources, controllers still regulate the queue lengths in a little bit more time and a larger overshoot is observed in each of the queue lengths.

Case 3: In this case the outgoing flow rate at the first bottleneck node, c_1, switches between 60 packets/s and 40 packets/s as a square-wave of period 200 s, while $c_2 = c_3 = 100$ packets/s. As seen from the simulation results in Figure 5, a sudden decrease in the outgoing flow rate causes a significant increase in the corresponding queue length, a sudden increase causes a sharp decrease. These sudden changes in c_1 are also sensed by the other nodes and their queue lengths and flow rates are affected to some extend but not as directly and sharply as that of node 1.

5 Conclusion

We have considered decentralized rate-based flow controller design in multi-bottleneck networks. Specifically, we have considered the $\mathcal{H}^\infty$ problem which was set forth in [1] and solved this problem doing some manipulations and using the methods of [5]. We have also presented the implementation of the derived controller. We briefly discussed the robustness and performance (tracking and fairness) properties of the controller and presented a number of simulations.

i,j	h^{rf}_{ij}	h^{rb}_{ij}	$h^{\rho f}_{ij}$	$h^{\rho b}_{ij}$	α^{r}_{ij}	α^{ρ}_{ij}	$\alpha^{\rho b}_{ij}$	δ^{r+}_{ij}	$\delta^{\rho +}_{ij}$	$\delta^{\rho b+}_{ij}$	β^{r}_{ij}	β^{ρ}_{ij}	$\beta^{\rho b}_{ij}$	β^{rf}_{ij}	$\beta^{\rho f}_{ij}$
$1,1$	0.1	1	$--$	$--$	0.1	$--$	$--$	2	$--$	$--$	0.1	$--$	$--$	0.01	$--$
$1,2$	0.2	2	1.1	1.1	0.2	0.2	0.1	3	2	1	0.2	0.1	0.05	0.02	0.05
$1,3$	$--$	$--$	1.1	1.1	$--$	0.12	0.1	$--$	2	1	$--$	0.1	0.05	$--$	0.05
$2,1$	0.1	1	1.1	1.1	0.1	0.3	0.05	2	2	1	0.1	0.1	0.05	0.01	0.05
$2,2$	0.2	2	$--$	$--$	0.2	$--$	$--$	3	$--$	$--$	0.2	$--$	$--$	0.02	$--$
$2,3$	0.2	2	1.1	1.1	0.3	0.08	0.05	2	2	1	0.2	0.1	0.05	0.01	0.05
$3,1$	0.1	1	1.1	1.1	0.1	0.2	0.05	2	2	1	0.1	0.1	0.05	0.01	0.05
$3,2$	0.2	2	1.1	1.1	0.15	0.1	0.05	3	2	1	0.2	0.1	0.05	0.02	0.05
$3,3$	0.2	2	$--$	$--$	0.2	$--$	$--$	2	$--$	$--$	0.1	$--$	$--$	0.01	$--$
$3,4$	0.2	2	$--$	$--$	0.25	$--$	$--$	3	$--$	$--$	0.2	$--$	$--$	0.02	$--$

Table 1: Controller parameters

i,j	$1,1$	$1,2$	$2,1$	$2,2$	$2,3$	$3,1$	$3,2$	$3,3$	$3,4$
$d_{i,j}(packets/second)$	30	30	50	50	45	45	40	45	40

Table 2: Rate limits for Case 2

Acknowledgement

Programs used in the simulations are based on the programs previously developed by Taesam Kang, Pierre-François Quet, and Banu Ataşlar.

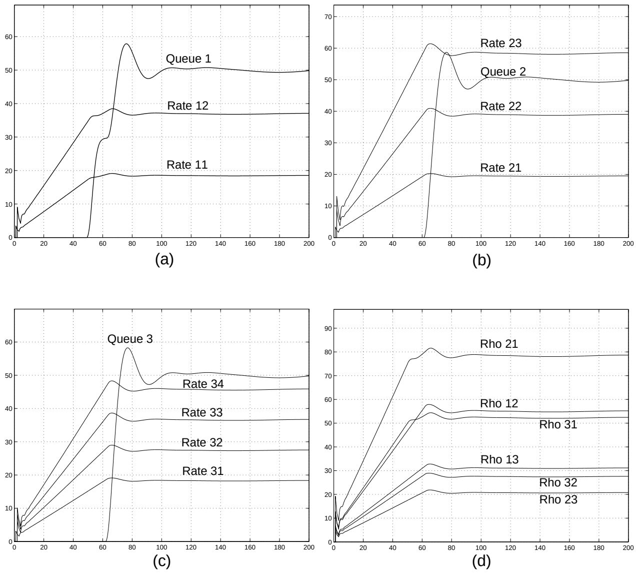

Figure 3: Simulation results for Case 1.

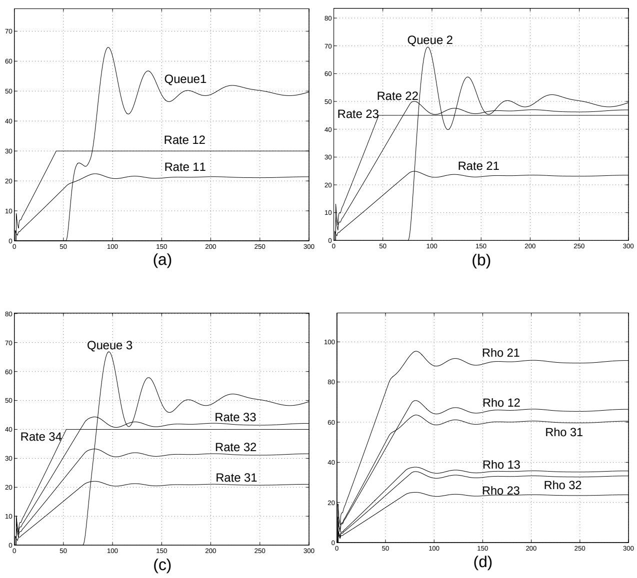

Figure 4: Simulation results for Case 2.

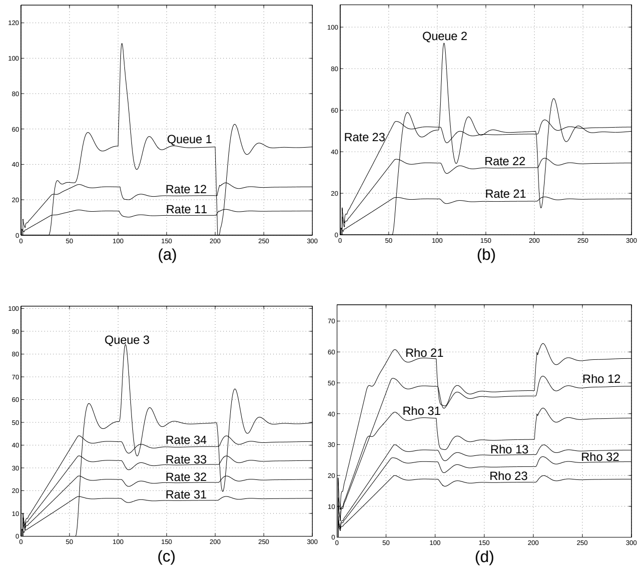

Figure 5: Simulation results for Case 3.

References

[1] E. Biberović, A. İftar, and H. Özbay, "A solution to the robust flow control problem for networks with multiple bottlenecks," in *Proceedings of the IEEE Conference on Decision and Control*, (Orlando, FL), pp. 2303–2308, Dec. 2001.

[2] E. Biberović, "Flow control in high-speed data communication networks," Master's thesis, Anadolu University, Eskişehir, Turkey, 2001. (In Turkish).

[3] G. Meinsma and H. Zwart, "On $\mathcal{H}^\infty$ control for dead-time systems," *IEEE Transactions on Automatic Control*, vol. 45, pp. 272–285, 2000.

[4] P. Quet, B. Ataşlar, A. İftar, H. Özbay, S. Kalyanaraman, and T. Kang, "Rate-based flow controllers for communication networks in the presence of uncertain time-varying multiple time-delays," *Automatica*, vol. 38, pp. 917–928, 2002.

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