Proceedings of the 2001 Winter Simulation Conference
B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, eds.

FLUID MODEL FOR WINDOW-BASED CONGESTION CONTROL MECHANISM

Richard J. La

Mathematics of Communication Networks
Motorola Inc.
Arlington Heights, IL 60004, U.S.A.

ABSTRACT

We study the stability of two queueing delay-based congestion control algorithms, the (p, 1)-proportionally fair algorithm and the global optimization algorithm. We linearize the systems around the intended operating point and show that these algorithms are stable within a range of feedback delay. Based on these linearized systems we study the impact of various (cascade) compensators on the system. We show that the PID control improves the transient behavior of the system. We simulate both the linearized system and non-linear system and illustrate the validity of the analysis and improvement in the system behavior with compensators.

1 INTRODUCTION

Recently there has been much research activity on optimization-based rate control and active queue mechanisms. Kelly has modeled the problem of rate control as a problem of maximizing the aggregate utility of the users (Kelly 1997). He has made the fundamental observation that this problem can be decomposed into separate subproblems for the network and for the individual users (Kelly 1997). In the papers that build on this work, two classes of rate allocation mechanisms are proposed to achieve the system optimum (Kelly *et al.* 1998, Gibbens and Kelly 1998). Low *et al.* have approached the same problem, using the gradient-based algorithm (Low and Lapsley 1999, Low *et al.* 2000), and have proposed an active queue mechanism called Random Early Marking (REM), as a way of arriving at the system wide optimal solution. Kunniyur and Srikant have proposed a different AQM mechanism called adaptive virtual queue (AVQ) to solve the same system problem in combination with the same primal algorithm implemented at the end hosts (Kunniyur and Srikant 2001).

Hollot *et al.* have investigated the problem of designing an AQM mechanism from a control theoretic point, analyzed random early detection (RED) mechanism, and proposed a guideline for setting RED parameters (Hollot *et al.* 2001a). In another paper they have proposed two different AQM mechanisms: Proportional and Proportional-Integral controllers (Hollot *et al.* 2001b). They have shown that the Proportional controller exhibits better response time than the RED controller. Further, they have explained the limitations of the Proportional controller due to the coupling between the queue size and the marking probability, and propose the PI controller to decouple the queue size and the marking probability and to improve the steady state regulation errors. They have studied the stability of these controllers, using linearized models, and provided the guidelines for setting their parameters.

In this paper we first study the stability of the (p, 1)-proportionally fair algorithm proposed by Mo and Walrand (Mo and Walrand 2000) and the global optimization algorithm by La and Anantharam (La and Anantharam 2001), using a control theoretic approach with a model linearized around the intended operation points. We show that these algorithms are stable, which corroborates the convergence results in Mo and Walrand (2000) and La and Anantharam (2001). We then investigate the possibility of enhancing these algorithms by adding various controllers at the end hosts and study their performance. We show that a Proportional-plus-Integral-plus-Derivative (PID) controller can significantly improve the performance of the system.

2 BACKGROUND & ALGORITHMS

In this section we briefly explain the (p,1)-proportionally fair algorithm proposed by Mo and Walrand (Mo and Walrand 2000) and the global optimization algorithm by La and Anantharam (La and Anantharam 2001). These algorithms are window-based congestion control algorithms and use estimated queueing delay in order to adjust the congestion window size.

In a general network we consider there are a network with a set $\mathcal{J}$ of resources or links and a set $\mathcal{I}$ of users. Let C_j denote the finite capacity of link $j \in \mathcal{J}$. Each user has a fixed route $\mathcal{J}_i$, which is a non empty subset of $\mathcal{J}$. We define a zero one matrix A, where $A_{i,j} = 1$ if link j is in user i's route $\mathcal{J}_i$ and $A_{i,j} = 0$ otherwise. The throughput of a user i is denoted by x_i.

We present a fluid model of the network that describes the relationship between the window sizes, rates, and queue sizes. Throughout the paper we assume that the switches exercise the first-in-first-out (FIFO) service discipline. This model can be represented by the following equations:

$$\begin{aligned} A^T x - C &\leq 0 && (1)\\ Q(A^T x - C) &= 0 && (2)\\ X(AC'q + \bar{d}) &= w && (3)\\ x \geq 0, q &\geq 0, && (4) \end{aligned}$$

where

$$C = (C_1, ..., C_J)^T, q = (q_1, ..., q_J)^T,$$
$$\bar{d} = (\bar{d}_1, ..., \bar{d}_I)^T, w = (w_1, ..., w_I)^T,$$
$$X = diag(x), C' = diag(C_1^{-1}, ..., C_J^{-1}), Q = diag(q),$$

w_i is the congestion window size of user i, $\bar{d}_i$ is the round trip propagation delay of route $\mathcal{J}_i$ not including the queueing delay, and q_j denotes the backlog at link j buffer. For simplicity of analysis we assume that the buffer sizes are infinite. The first condition in (1) represents the capacity constraint. The second constraint says that there is backlogged fluid at a resource only if the total rate through it equals the capacity. The third constraint follows from that the window size of connection i should equal the sum of the amount of fluid in transit and the backlogged fluid in the buffers, i.e.,

$$w_i = x_i \cdot \bar{d}_i + q^i,$$

where q^i denotes connection i's total backlog in the buffers.

Mo and Walrand have shown in Mo and Walrand (2000) that the rates of the users as well as their queueing delays of the users are well defined given users' window sizes. This was also proved by Massoulie and Roberts in (Massoulie and Roberts 1999). Under the (p,1)-proportionally fair algorithm by Mo and Walrand (Mo and Walrand 2000) each connection has a target queue size $p_i > 0$ and attempts to keep a total of p_i packets at the switch buffers. Let $d_i(t)$, $w_i(t)$, and $x_i(t)$ denote the round trip delay with queueing delay, the congestion window size, and the rate of connection i at time t, respectively, where user i's queueing delay at time t is defined to be $\sum_{j \in \mathcal{J}_i} \frac{q_j(t)}{C_j}$ and $q_j(t)$ is the backlog at resource j at time t. Hence, $d_i(t) = \frac{w_i(t)}{x_i(t)} = \bar{d}_i + \sum_{j \in \mathcal{J}_i} \frac{q_j(t)}{C_j}$.. We use $x_i(t)$ to denote $x_i(w(t))$ when there is no confusion. Suppose that each connection i has a fixed target queue size p_i. Connection i updates its window size $w_i(t)$ according to the following differential equation

$$\frac{d}{dt} w_i(t) = -\kappa \frac{\bar{d}_i}{d_i(t)} \frac{s_i(t)}{w_i(t)},$$

where κ is some positive constant, and $s_i(t) = w_i(t) - x_i(t) \cdot \bar{d}_i - p_i$. Under this algorithm the window sizes converge to a unique point w^* such that for all $i \in \mathcal{I}$

$$w_i^* - x_i(w^*) \cdot \bar{d}_i = p_i,$$

where $x_i(w^*)$ is connection i's throughput that satisfy (1) through (4) when the window sizes are w^*. They show that at the unique stable point w^* of the algorithm the resulting allocation $x(w^*)$ is weighted proportionally fair. In other words, this algorithm achieves the solution to the *NETWORK* problem of maximizing weighted log functions with each user i's willingness to pay equal to p_i (Kelly 1997, Mo and Walrand 2000).

La and Anantharam in (La and Anantharam 2001) have proposed another algorithm, which allows users to dynamically adjust their target queue sizes based on the current congestion level in the network in order to solve the user problem of maximizing the net utility (Kelly 1997). Suppose that we denote the utility of user i by $U_i(x_i)$ when it receives a rate of x_i. We assume that these utility functions of the users satisfy the following assumption:

$$U_i'(x_i) + x_i \cdot U_i''(x_i) \geq 0, \;\; x_i \in [0, C^i],$$

where $C^i = \min_{j \in \mathcal{J}_i} C_j$. Under this algorithm each connection updates its window size $w_i(t)$ according to the following system of differential equations:

$$\frac{dw_i(t)}{dt} = -\kappa \cdot M_i(t) \cdot r_i(t) \tag{5}$$

where

$$M_i(t) = \frac{\bar{d}_i + \frac{U_i'(x_i(t))}{\lambda} + \frac{x_i(t) \cdot U_i''(x_i(t))}{\lambda}}{d_i(t)}, \tag{6}$$

$U_i'(\cdot) = \frac{d}{dx_i} U_i(\cdot)$, $U_i''(\cdot) = \frac{d^2}{dx_i^2} U_i(\cdot)$, λ is some positive constant, and

$$\begin{aligned} r_i(t) &= \frac{w_i(t) - x_i(t)\bar{d}_i - x_i(t)\frac{U_i'(x_i(t))}{\lambda}}{w_i(t)} \\ &= 1 - \frac{x_i(t)(\bar{d}_i + \frac{U_i'(x_i(t))}{\lambda})}{w_i(t)}. \end{aligned} \tag{7}$$

Note that user i's utility function appears in both $M_i(t)$ and $r_i(t)$. They have proved that the algorithm converges to the unique stable point $\tilde{w}$, where the resulting allocation $x(\tilde{w})$ is the solution to the *SYSTEM* problem of maximizing the aggregate utility of the users. One can see that the $(p,1)$-proportionally fair algorithm is a special case of this global optimization algorithm with the utility function of $p_i \cdot \log(\cdot)$.

3 STABILITY

In the analysis of algorithms in section 2 they assume that given the window sizes of the users, the rates and round trip times of the users are known immediately. In other words, given the window sizes, they assume that the users' rates are those that satisfy the conditions in (1) - (4). Hence, the delay in the feedback is not modeled in the analysis of convergence. In a real network users' rates need to be estimated, based on the current estimates of the round trip times, i.e., $x_i^{est}(t) = \frac{w_i(t)}{d_i^{est}(t)}$, where $x_i^{est}(t)$ and $d_i^{est}(t)$ are the estimated rate and estimated round trip time of user i at time t. In the case of a network with single link $d_i^{est}(t)$ can be approximated by $d_i(t - \frac{q(t-d_i(t))}{C})$, where $q(t)$ is the queue size at the link at time t and C is the capacity of the link.

In this section, using the fluid model for the window-based congestion control mechanism described in section 2, we adopt a similar linearized model as in Hollot *et al.* (2001a) and Kunniyur and Srikant (2001) and study the stability of the single bottleneck network *with a delay in the feedback* of the queueing delay. Consider a single link with capacity C. Suppose that there are $N \geq 1$ users that share the link with a common round trip propagation delay of d, i.e., $\bar{d}_i = d$ for all $i \in \mathcal{I}$.

3.1 The $(p,1)$-Proportionally Fair Algorithm

We first analyze the $(p,\ 1)$-proportionally fair algorithm with $p_i = p$ for all $i \in \mathcal{I}$. At the equilibrium or unique stable point of the algorithm, which we denote by $\underline{W}^* = \{W_i^*, i \in \mathcal{I}\}$, we have

$$\sum_i \frac{W_i^*}{R^*} = C, \tag{8}$$

$$x_i^* = \frac{W_i^*}{R^*} = \frac{C}{N}, \tag{9}$$

$$R^* = d + \frac{q^*}{C} = d + \frac{p \cdot N}{C}. \tag{10}$$

We let

$$\delta W(t) = W(t) - W^*,$$
$$\delta q(t) = q(t) - q^* = q(t) - p \cdot N,$$

where W^* denotes the window size of the users at the equilibrium point. Note that all users have the same window size at the equilibrium from (9) and (10). The linearized model can be written as

$$\delta \dot{W}(t) = -\frac{\gamma(R^* - d)^2}{pR^{*2}} \delta W(t) - \frac{\gamma \cdot d}{R^{*2} C} \delta q(t - R^*), \tag{11}$$

$$\delta \dot{q}(t) = \frac{N}{R^*} \delta W(t) - \frac{1}{R^*} \delta q(t), \tag{12}$$

where $\gamma = \frac{\kappa \cdot d}{R^*}$. The dynamics of the linearized system with the $(p,1)$-proportionally fair algorithm are illustrated in Figure 1.

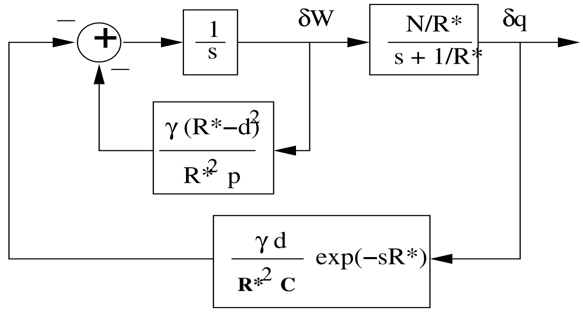

Figure 1: Block Diagram of the Network with $(p,1)$-Proportionally Fair Algorithm

One can simplify the block diagram in Figure 1 and show that it is equivalent to the following (Nise 1992):

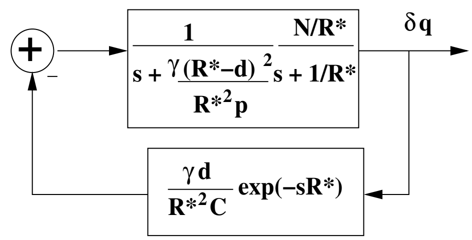

Figure 2: Simplified Block Diagram of the Network with $(p,1)$-Proportionally Fair Algorithm

From the simplified block diagram in Figure 2 we can find the poles of the closed-loop transfer function. If we ignore the delay in the feedback for the moment, we can easily show that these poles are in the left half-plane. In fact the poles are given as the roots of the following quadratic equation:

$$s^2 + \left(\frac{\gamma(R^* - d)^2}{R^{*2} p} + \frac{1}{R^*} \right) s + \frac{\gamma}{R^{*2}} \left(\frac{(R^* - d)^2}{R^* p} + \frac{Nd}{R^* C} \right).$$

Since all these variables are positive, one can show that the roots of the equations all lie in the left half-plane. Therefore, the system is stable, which corroborates the convergence results in Mo and Walrand (2000).

We now discuss the system with a delay in the feedback. The roots of the characteristic equation (or the poles of the closed loop transfer function), which can be obtained by taking Laplace transform, are continuous functions of the parameters. Hence, one can see that the roots of the characteristic function are continuous functions of the feedback delay $R^* = d + \frac{p \cdot N}{C}$. This tells us that by computing the poles of the closed loop transfer function we can find the smallest feedback delay R at which one of the roots lies on the imaginary axis, if there exists such R. From (11) and (12) we can find the characteristic equation

$$s + A + \frac{B \cdot C}{s + D} e^{-sR^*} = 0 \ ,$$

where

$$A = \frac{\gamma (R^* - d)^2}{p R^{*2}}, \ B = \frac{\gamma \cdot d}{R^{*2} C}, \ C = \frac{N}{R^*}, \text{ and } D = \frac{1}{R^*} \ .$$

This is equivalent to solving

$$\frac{B \cdot C}{(s + A)(s + D)} e^{-sR^*} = 1 \ . \tag{13}$$

The solutions to (13) need to satisfy the following two conditions:

$$\left| \frac{B \cdot C}{(jw + A)(jw + D)} \right| = 1 \tag{14}$$

and

$$\angle \frac{B \cdot C}{(jw + A)(jw + D)} e^{-jwR^*} = (2k+1)\pi, \ k \in \mathcal{Z} \ . \tag{15}$$

The condition on magnitude in (14) implies

$$\frac{B \cdot C}{\sqrt{w^2 + A^2}\sqrt{w^2 + D^2}} = 1$$

or

$$w^4 + (A^2 + D^2) w^2 + (A^2 D^2 - B^2 C^2) = 0 \ . \tag{16}$$

A positive, real solution to (16) exists only if $B^2C^2 - A^2D^2 \geq 0$ and is given by

$$w(\gamma, R^*, N) = \sqrt{\frac{-E + \sqrt{E^2 + 4 \cdot F}}{2}} \ , \tag{17}$$

where $E = A^2 + D^2 = \frac{\gamma^2 (R^* - d)^4}{p^2 R^{*4}} + \frac{1}{R^{*2}}$ and $F = B^2C^2 - A^2D^2 = \frac{\gamma^2 N^2}{R^{*6} C^2}(2R^*d - R^{*2})$. The condition on angle yields

$$wR^* + \arctan\left(\frac{w}{A}\right) + \arctan\left(\frac{w}{D}\right) = (2k+1)\pi \ , \ k \in \mathcal{Z}. \tag{18}$$

Since $w(\gamma, R^*, N, p)$ is an increasing function of R^* (and equivalently d), the smallest R that solves (18) tells us when at least one of the roots first hits the imaginary axis. Therefore, the system is stable for all round trip delays smaller than the smallest solution to (18) because the poles of the closed-loop transfer function stay in the lef half-plane.

3.2 The Global Optimization Algorithm

We assume that users have the same utility function $U(\cdot)$. At the equilibrium of the global optimization problem, which we denote by $\underline{\tilde{W}} = \{\tilde{W}_i, i \in \mathcal{I}\}$, we have

$$\sum_i \frac{\tilde{W}_i}{\tilde{R}} = C, \tag{19}$$

$$\tilde{x}_i = \frac{\tilde{W}_i}{\tilde{R}} = \frac{C}{N}, \tag{20}$$

$$\tilde{R} = d + \frac{\tilde{q}}{C} = d + U'(\frac{C}{N}), \tag{21}$$

where $U'(x) = \frac{d}{dx}U(x)$. With a little abuse of notation we let

$$\delta W(t) = W(t) - \tilde{W},$$
$$\delta q(t) = q(t) - \tilde{q} = q(t) - U'(\frac{C}{N}) \cdot C,$$

where $\tilde{W}$ denotes the window size of the users at the equilibrium, i.e., $\tilde{W} = \frac{C}{N}(d + U'(\frac{C}{N}))$. Again, from (20) and (21) every user has the same window size at the equilibrium. From above one can write the linearized model as

$$\delta \dot{W}(t) = \frac{\mu \cdot U''(\frac{C}{N})}{\tilde{R}^2} \delta W(t) - \frac{\mu}{\tilde{R}^2}\left(\frac{d}{C} + \frac{U'(\frac{C}{N})}{C} + \frac{U''(\frac{C}{N})}{N}\right)\delta q(t - \tilde{R}), \tag{22}$$

$$\delta \dot{q}(t) = \frac{N}{\tilde{R}} \delta W(t) - \frac{1}{\tilde{R}} \delta q(t), \tag{23}$$

where $\mu = \kappa \frac{d + U'(\frac{C}{N}) + \frac{C}{N} U''(\frac{C}{N})}{\tilde{R}}$. The dynamics of the network are illustrated in Figure 3.

Ignoring the delay in feedback for the moment one can show that the poles of the closed-loop transfer function lie

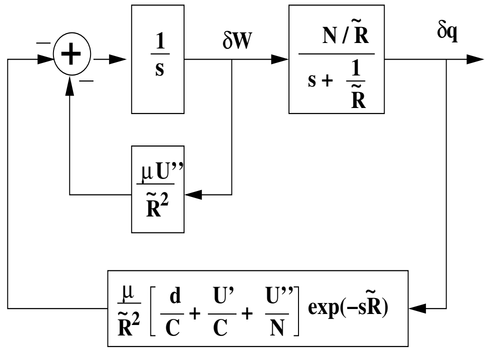

Figure 3: Block Diagram of the Network with the Global Optimization Algorithm

in the left half-plane from the assumption on the utility functions in section 2. This corroborates the stability result in La and Anantharam (2001).

In the case with a delay, similarly as in the previous case, we can find the characteristic function from (22) and (23)

$$s + A' + \frac{B'A'}{s + D'} e^{-s\tilde{R}} = 0 \ , \qquad (24)$$

where

$$A' = -\frac{\mu \cdot U''(\frac{C}{N})}{\tilde{R}^2} \ , C' = \frac{N}{\tilde{R}} \ , D' = \frac{1}{\tilde{R}} \ , \text{and}$$
$$B' = \frac{\mu}{\tilde{R}^2} \left(\frac{d}{C} + \frac{U'(\frac{C}{N})}{C} + \frac{U''(\frac{C}{N})}{N} \right) \ .$$

A solution to (24) satisfies the following conditions:

$$\left| \frac{B' \cdot C'}{(jw + A')(jw + D')} \right| = 1 \qquad (25)$$

or

$$w^4 + (A'^2 + D'^2) w^2 + (A'^2 D'^2 - B'^2 C'^2) = 0 \qquad (26)$$

and

$$\angle \frac{B'C'}{(jw + A')(jw + D')} e^{-jw\tilde{R}} = (2k+1)\pi \ , \ k \in \mathcal{Z} \ . \ (27)$$

A positive, real solution to (26) exists only if $B'^2 C'^2 - A'^2 D'^2 > 0$ and is given by

$$w(\mu, \tilde{R}, N) = \sqrt{\frac{-E' + \sqrt{E'^2 + 4 \cdot F'}}{2}} \ ,$$

where

$$E' = A'^2 + D'^2 = \frac{\mu^2 \left(U''(\frac{C}{N}) \right)^2}{\tilde{R}^4} + \frac{1}{\tilde{R}^2}$$

and

$$\begin{aligned} F' &= B'^2 C'^2 - A'^2 D'^2 \\ &= \frac{\mu^2}{\tilde{R}^6} \left(\frac{N\left(d + U'(\frac{C}{N})\right)}{C} + U''(\frac{C}{N}) \right)^2 \\ &\quad - \frac{\mu^2}{\tilde{R}^6} \left(U''(\frac{C}{N}) \right)^2 \ . \end{aligned}$$

The condition on angle yields

$$\begin{aligned} w\tilde{R} + \arctan\left(\frac{w}{A'}\right) + \arctan\left(\frac{w}{D'}\right) \\ = (2k+1)\pi \ , \ k \in \mathcal{Z} \ . \end{aligned} \qquad (28)$$

One can show that $w(\mu, \tilde{R}, N)$ is an increasing function of $\tilde{R}$ (and equivalently d given the utility function and N) for the utility functions that satisfy the assumption stated in section 2. The smallest R that solves (28) tells us when one of the roots first lies on the imaginary axis. Since the poles of the closed-loop transfer function remain in the left half-plane for all round trip delays smaller than the solution to (28), the system is stable. Therefore, given the choice of gain κ one can find the range of feedback delay for which system is stable.

4 CASCADE CONTROLLERS

The parameter κ in both the $(p, 1)$-proportionally fair algorithm and the global optimization algorithm determines the gain in the feedback loop. Thus, one can to some extent control the transient or steady-state behavior by changing this gain parameter. However, there is a limit on how much one can improve the transient or steady-state behavior just by changing the gain in the feedback-loop (Hollot *et al.* 2001b). One can increase the flexibility in the design of a desired transient or steady-state response by introducing a (cascade) compensator to the system. It is well known in control theory that one can improve the steady-state error of a feedback control system using cascade compensation without appreciably affecting the transient response (Nise 1992). An ideal integral compensation, which uses a pure integrator to place an open-loop, forward-path pole at the origin, increases the system type and reduces the steady-state error to zero (Hollot *et al.* 2001b, Nise 1992). On the other hand a pure differentiator, which places a zero to the open-loop, forward-path transfer function, can be added to improve the transient response.

In this and the following sections we describe various cascade controllers and investigate how these controllers can be implemented in conjunction with the algorithms in section 2 to improve the transient response and steady-state error of the algorithms. In the subsequent section we give a few numerical examples to show the performance of the system with the $(p,\, 1)$-proportionally fair algorithm. Since the $(p,1)$-proportionally fair algorithm is a special case of the global optimization algorithm with the utility function of $p_i \cdot \log(\cdot)$, the results can be easily extended to the global optimization algorithm.

4.1 The Proportional Controller

A proportional controller simply acts as an amplifier. The feedback signal generated in proportional control is the regulated output multiplied by some gain, which is the design parameter (Hollot *et al.* 2001b, Nise 1992). In our problem the regulated output is the queue size or the queueing delay at the link. Since both the $(p,1)$-proportionally fair algorithm and the global optimization algorithm use the estimated queueing delay to update the window size, it is equivalent to selecting the appropriate value for the parameter κ described in section 2 in order to adjust the gain parameter of the controller (La and Anantharam 2001). The performance of the proportional controller with different gain parameter values is demonstrated in section 5.2.

4.2 The Proportional-plus-Integral

Hollot *et al.* (Hollot *et al.* 2001b) have observed that in their model for stable operation of the Proportional controller it requires a shallow slope in the loss profile. In our problem this translates to a small value of κ in the $(p,\, 1)$-proportionally fair algorithm and the global optimization algorithm. This means slow response of the system.

Steady-state error can be improved by placing an open-loop pole at the origin since this increases the system type by one. In order to compensate for the change in the angular contribution of the open-loop poles due to the addition of a pole at the origin, one needs to add a zero close to the pole at the origin as well. Such a compensator with a pole at the origin and a zero close to the pole is called an *ideal integral compensator* (Nise 1992).

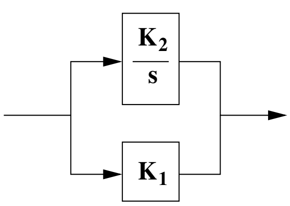

Figure 4: A PI Controller

A method of implementing an ideal integral compensator is shown in Figure 4. The value of the zero can be adjusted by varying $\frac{K_2}{K_1}$. Systems that feed the error forward to the plant are called *proportional control systems*. Systems that feed the integral of the error to the plant are called *integral control systems*. The controller, or compensator, is thus given the name *PI controller*, which stands for "Proportional plus Integral" (Nise 1992).

4.3 The Proportional-plus-Derivative

One can improve the transient response of a feedback control system by using *ideal derivative compensation*. With ideal derivative compensation a pure differentiator is added to the forward path of the feedback control system (Nise 1992). This results in the addition of a zero to the forward-path transfer function. We call an ideal derivative compensator a "Proportional plus Derivative (PD) controller" because the implementation consists of feeding the error (proportional) plus the derivative of the error forward to the plant. This is shown in Figure 5.

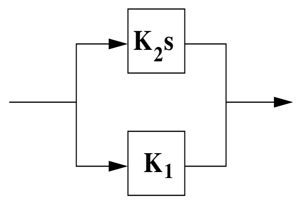

Figure 5: A PD Controller

4.4 The Proportional-plus-Integral-plus-Derivative

Using a combination of the PI controller and PD controller, one can improve both the steady-state and transient response of the system. One can either first improve the transient response by using PD control and then steady-state error of this compensated system by applying PI control or vice versa. One disadvantage of the first approach is that there will be slight degradation in the transient behavior after adding the PI control, and in the second case the steady-state response may be somewhat compromised. This can be achieved by the following combination of PI and PD controllers in Figure 6.

In order to implement any of the cascade compensators with the algorithms in section 2 one needs not only to estimate the queueing delay, but also to compute the difference in the successive queueing delay estimates and/or integrate the difference between the estimated number of queued packets and the target queue size, which is p_i and $x_i(t)\frac{U_i'(x_i(t))}{\lambda}$ in the $(p,\, 1)$-proportionally fair algorithm and the global optimization algorithm, respectively. This additional computation requires two subtractions and one addition and is done only once every round trip time. There-

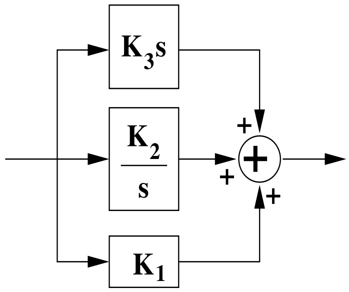

Figure 6: A PID Controller

fore, the computational requirement is not likely to be an issue.

5 NUMERICAL EXAMPLES

In this section we give a few numerical examples. The first simulation is run using *simulink* to investigate the behavior of the linearized models with the controllers described in section 4. The second simulation is run using *ns*-2 (Network Simulator) to evaluate the performance of the (p, 1)-proportionally fair algorithm with different control.

5.1 The Linearized Networks

In this subsection we simulate the linearized system with the (p,1)-proportionally fair algorithm with the controllers described in section 4 and compare their performance. We set $N = 100$, $d = 100$ ms, $p = 2$ packets, $C = 2,000$ packets/sec, and $\kappa = 0.5$. All users' window sizes are set to 0 at the beginning, and the queue is initially empty. At the equilibrium the feedback delay R^* is 0.2 seconds because the queueing delay is $\frac{p \cdot N}{C} = 0.1$ seconds. The regulated output, which is the queue size, is normalized by the capacity, i.e., the information fed back to the users is the queueing delay. The gains of the controllers are set to $K_1 = 100$, $K_2 = 10$, and $K_3 = 20$. These parameters of the controller should be selected based on the desired transient response, such as the percent overshoot and settling time, and the location of the open-loop zero close to the origin. We do not discuss the selection of the parameters in this paper. Interested readers may refer to (Nise 1992).

Figures 7 through 10 plot $\delta q(t)$ as a function of time and show the system response with various controllers. One can easily see that the transient response is better for PD and PID controllers than for P and PI controllers. The P and PI control exhibits much larger overshoot than PD and PID control. Clearly, this large overshoot is undesirable in a network since it leads to a larger variation in queue size and results in a temporary increase in round trip delay and potential congestion. Further, the settling time of PD and PID control is much smaller than that of P and PI control. This faster convergence to the equilibrium of PD and PID control is desirable especially with dynamic arrivals and departures of the users. If the convergence time is too large, the system may never converge to the equilibrium with the arrivals and departures of the users. One can also see that the steady-state error of P and PD control is non-zero, while that of PI and PID converges to zero. This is consistent with the claims in section 4.

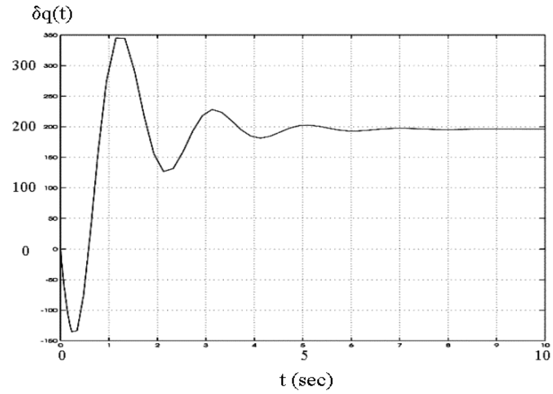

Figure 7: P Controller

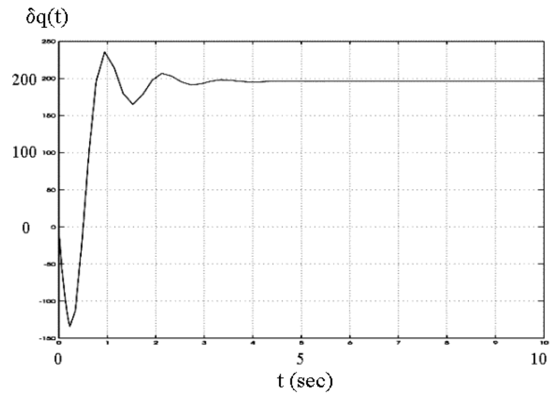

Figure 8: PD Controller

5.2 The *ns*-2 Simulation

We have run the simulation of a network with single link. The link is shared by 100 users. Each user has a round trip propagation delay that is randomly selected from [10 ms, 190 ms] with a mean of 100 ms. The first 25 users start randomly between 0 ms and 200 ms, another 25 users start randomly between 7.0 sec and 7.2 sec, and the other users start randomly between 12.0 sec and 12.2 sec. In the first simulation we have set the parameters in the *ns*-2 to the same settings as in the simulation in the previous subsection except for the round trip propagation delay of the users. Note that the users' propagation delays are assumed to be the same in the analysis of the linearized system and the simulation in the previous subsection. Since users measure the round trip time and estimate the queueing delay of the link, we implement the various controllers within the TCP protocol.

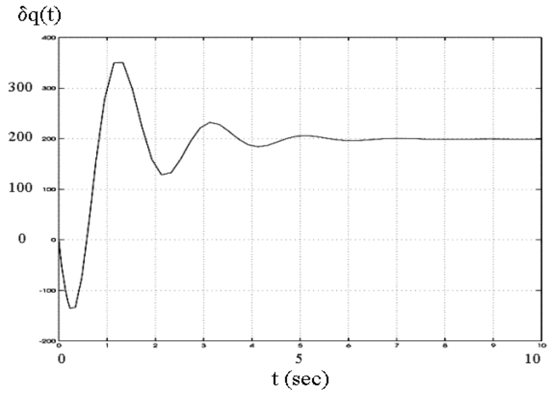

Figure 9: PI Controller

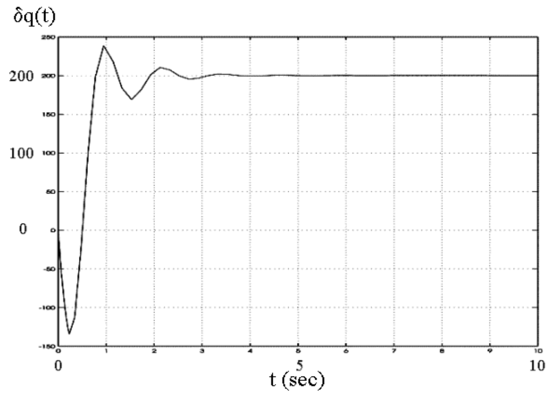

Figure 10: PID Controller

In other words, at the arrival of each acknowledgment (ACK) the user computes the derivative and/or integral necessary for the controller and use the information to update the congestion window size. The packet size is fixed at 1,000 bytes, and the capacity of the link is set to 15 Mbps. The topology of the simulated network is shown in Figure 11.

The evolution of the queue sizes are shown in Figures 12 and 13 with P and PID control, respectively. As one can see the system with PID controller implemented at the end users performs better in terms of overshoot and settling time. This is consistent with the simulation results from the subsection 5.1. The P controller results in a large overshoot after the second and third group of users start at $t = 7$ sec and $t = 12$ sec and requires 6-8 seconds to settle to a point close to the equilibrium, while the PID controller does not lead to a large overshoot and requires less than 3 seconds to settle to the equilibrium.

In order to find out how small the gain in the closed-loop path needs to be in the P control, so as to avoid a large overshoot, we have run the simulation with various κ values and observed that $\kappa = 0.1$, i.e., the gain of $\frac{1}{5}$ of the previous gain, leads to similar overshoot as in the PID control case. The evolution of the queue size is shown in

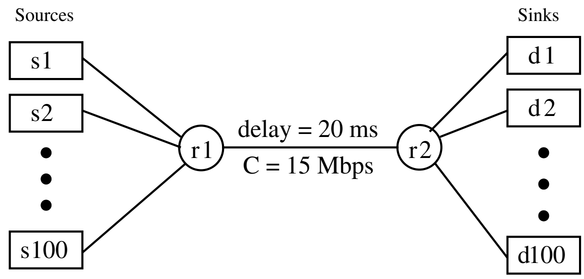

Figure 11: Topology of the Simulated Network

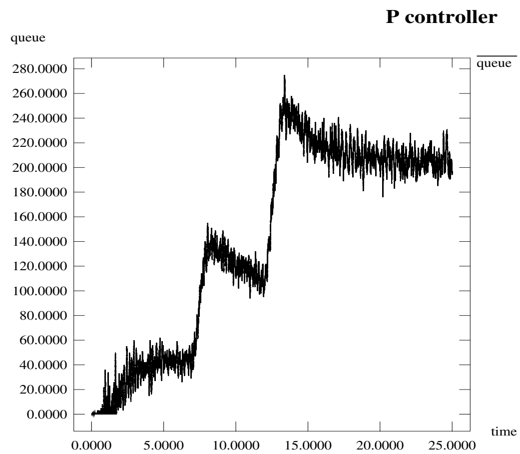

Figure 12: P Controller ($\kappa = 0.5$)

Figure 14. One can easily see that reducing the overshoot comes at the price of much slower response. With the smaller gain in the closed-loop path the system takes much longer to settle to the equilibrium.

6 CONCLUSIONS

We have investigated two queueing delay-based congestion control algorithms proposed in the past. We have shown that these algorithms are stable as long as the delay in feedback lies within some range, which depends on the system parameters. Various cascade compensators have been introduced and the improvement in both the transient behavior and steady-state response has been demonstrated.

We have simulated both the linearized systems and the non-linear systems. The simulation results demonstrate that these algorithms are stable with a delay in feedback. Moreover, it is shown that the PID controller does improve the transient behavior significantly in that the system does not overshoot as much and the settling time is significantly reduced.

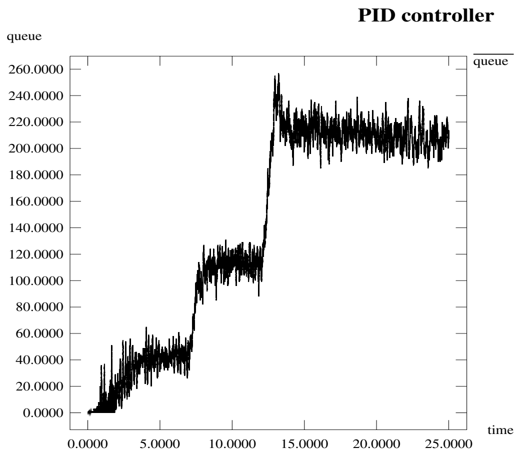

Figure 13: PID Controller ($\kappa = 0.5$)

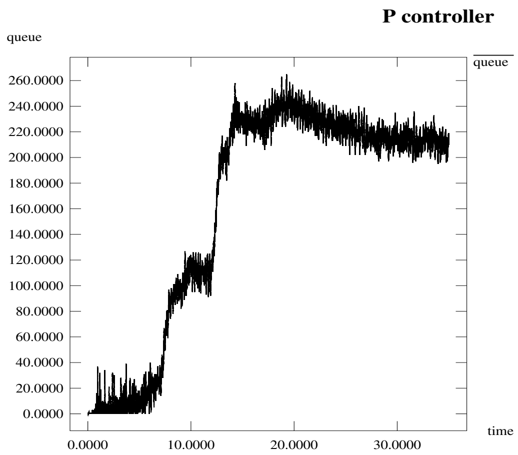

Figure 14: P Controller ($\kappa = 0.1$)

REFERENCES

Athuraliya, S., Low, S., and Lapsley, D. 2000. Random Early Marking. Available online via <http://www.ee.mu.oz.au/staff/slow/research> [accessed March 1, 2000].

Gibbens, R. J., Kelly, F. P. 1998. Resource pricing and the evolution of congestion control. Available online via <http://www.statslab.cam.ac.uk/ frank> [accessed June 20, 1999].

Hollot, C. V., Misra, V., Towsley, D., and Gong, W. 2001a. A control theoretic analysis of RED. In *Proceedings of the IEEE INFOCOM 2001*. Anchorage, Alaska:Institute of Electrical and Electronics Engineers.

Hollot, C. V., Misra, V., Towsley, D., and Gong, W. 2001b. On designing improved controllers for AQM routers supporting TCP flows. In *Proceedings of the IEEE INFOCOM 2001*. Anchorage, Alaska:Institute of Electrical and Electronics Engineers.

Kelly, F. P. Charging and rate control for elastic traffic. 1997. *European Transactions on Telecommunications* 8 (1):33-7.

Kelly, F. P., Maulloo, A., and Tan, D. 1998. Rate control for communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society* 49 (3):237-52.

Kunniyur, S., and Srikant, R. 2001. A time scale decomposition approach to adaptive ECN marking. In *Proceedings of the IEEE INFOCOM 2001*. Anchorage, Alaska: Institute of Electrical and Electronics Engineers.

La, R. J., and Anantharam, V. 2001. Window-based congestion control with heterogeneous users. In *Proceedings of the IEEE INFOCOM 2001*. Anchorage, Alaska: Institute of Electrical and Electronics Engineers.

Low, S., and Lapsley, D. 1999. Optimization flow control. *IEEE/ACM Transactions on Networking* 7 (6):861-74.

Massoulie, M., and Roberts, J. 1999. Bandwidth sharing: objectives and algorithms. In *Proceedings of the IEEE INFOCOM 1999*. New York, New York:Institute of Electrical and Electronics Engineers.

Mo, J., and Walrand, J. 2000. Fair end-to-end window-based congestion control. *IEEE/ACM Transactions on Networking* 8 (5):556-567.

Nise, N. S. 1990. *Control Systems Engineering*. CA Benjamin/Cummings Pub. Co.

AUTHOR BIOGRAPHY

RICHARD J. LA is a senior engineer at Motorola. He received a doctoral degree in electrical engineering and computer sciences from the University of California at Berkeley in May 2000. Since then he has been working in the Mathematics of Communication Networks group at Motorola.