Available Bit Rate (ABR) Source Control and Delay Estimation

Ping Zhou*and Charles Thompson
Center for Advanced Computation and Telecommunications
University of Massachusetts Lowell
One University Ave, Lowell, MA 01854, USA
ping.zhou@tellabs.com
charles_thompson2@uml.edu

Abstract

The problem of regulating the transmission rate of an available bit rate (ABR) traffic source in an ATM network is examined. Of particular interest is linear quadratic (LQ) rate regulation based on estimates of the round-trip propagation delay. The round trip delay is estimated using a nonlinear least mean square (NLMS) algorithm. Simulation results are used to demonstrate the method.

1. Introduction

Available Bit Rate (ABR) service defined in ATM standards [5]. is to be used to support applications that can accommodate excess network capacity. This is done by adjusting the source data transmission rate based on the available resources in the network. To do so, information of the current network status is returned to the source. This state-of-affairs results in a closed-loop feedback system. For feedback control schemes, the time-delays incurred in the feedback path and the temporal variation in the link capacity are considered to be problematic features. Therefore, successful ABR source rate control depends upon the effectiveness of the controller to overcome these delays and to adapt to the temporal variations in the excess bandwidth.

Developing a good rate control mechanism for ABR service is a challenging task and it has been the focus of many recent papers, especially in the ATM forum [1, 4, 6, 8]. These proposed schemes use explicit rate indication mechanism. They differ primarily in the way congestion is monitored and in how an explicit rate is computed. Most of these algorithms pose heuristic solutions to the problem. As such, these methods do not easily relate the resulting traffic features with the control action. Analytical studies have been presented for a proportional-derivative (PD) feedback controller for a congested node [2]. Controller parameters were determined using a pole-placement procedure.

In this work, the problem of regulating available bit rate (ABR) traffic is examined. The objective is to regulated the source rate based on the magnitude of the excess capacity. To this end, a LQ regulator is developed. In addition the problem of estimation of the time-delay in the control loop is examined. The paper is structured as follows. Section 2 describes the problem, the buffer and the controller model used. The control methodology is presented in section 3. Section 4 presents a delay estimation method based on the NLMS algorithm. Finally, in section 5 concluding remarks are given.

2. Problem Statement

For a given virtual circuit (VC), the focus is on controllers that maintaining a prescribed backlog level. The queue length is chosen as an indicator of congestion. The per-VC model is shown in Figure 1 where q is the buffer occupancy, q^0 is the target buffer occupancy, μ is the available service rate, D_f is forward delay, and D_b is backward delay.

In each time slot, the switch measures the queue size. Then a RM cell is sent back to the source and an new source rate $\xi(n)$ is calculated. This rate equals to the number of cells transmitted by the source in the nondimensional time interval $[n, n+1)$. At time n, the buffer occupancy is $q(n)$ and $\mu(n)$ is the available transmission capacity. The buffer occupancy at the next time step $q(n+1)$ is equal to the sum buffer occupancy $q(n)$ and the net number of cells that have entered the buffer $\xi(n) - \mu(n)$.

*Currently at Tellabs, Willington, MA 01887

0-7695-0912-6/00 $10.00 © 2000 IEEE

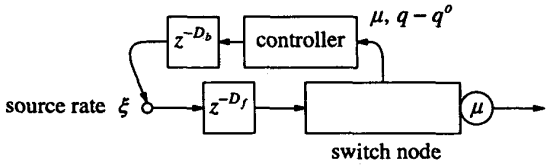

Figure 1. Virtual Circuit

$$q(n+1) = max[0,\ q(n) + \xi(n) - \mu(n)] \quad (1)$$

It is desired to control the rate based on the error between the actual and target buffer occupancy. The value q^0 is defined as the threshold value and the explicit rate is $r(n+1)$ is calculated following Benmohamed and Meerkov [1].

$$r(n+1) = max\left[0, r(n) - \sum_{j=0}^{1}\alpha_j(q(n-j) - q^0) - \sum_{k=0}^{D}\beta_k r(n-k)\right] \quad (2)$$

where $D = D_f + D_b$, α_j and β_k are parameters of the controller.

When the explicit rate is sent back to the source, the source sends the cells at the rate $\xi(n)$ the middle value among PCR, MCR and received explicit rate $r(n+1-D_b)$

$$\xi(n) = min[PCR,\ max[r(n+1-D_b), MCR]] \quad (3)$$

where, D_b is the backward delay, PCR is the peak cell rate, and MCR is the minimum cell rate.

3. LQ Regulation of the Source Rate

The controller is designed using the linearized version of the Eqns 1-3. Under the condition that $MCR = 0$, PCR equals infinity, and $r > 0$, the explicit rate and source rate are equal $\xi = r$. For a step change in available transmission capacity the source rate r should approach μ in the limit of n approaching infinity while q approaches q^0. Using Eqns 1-3 it can be shown that $\beta_D = -\sum_{k=0}^{D-1}\beta_k$. Under the aforementioned conditions, the simplified equations are

$$\hat{q}(n+1) = 2\hat{q}(n) - \hat{q}(n-1) + \hat{r}(n+1-D) - [\mu(n) - \mu(n-1)] \quad (4)$$

$$\hat{r}(n+1) = \hat{r}(n) + u(n) \quad (5)$$

where $\hat{r}(n) = r(n) - r(n-D)$, $\hat{q}(n) = q(n) - q^0$ and u is the control input. The regulator realization of the control input u is $u = -\underline{G}^T\underline{x}$. The gain and state vectors are defined as

$$\underline{G} = [\alpha_0,\ \alpha_1,\ \beta_0,\ \beta_1,\ \cdots,\ \beta_{D-1}]^T$$

and

$$\underline{x}(n) = [\hat{q}(n),\ \hat{q}(n-1),\ \hat{r}(n),\ \hat{r}(n-1),\ \cdots,\ \hat{r}(n+1-D)]^T$$

The LQ-regulator is obtained by minimizing the quadratic cost function J

$$J = \sum_{n=0}^{\infty}\underline{x}^T S\underline{x} + Pu^2 \quad (6)$$

subject to the constraints given by Eqns 4-5

$$\underline{x}(n+1) = A\underline{x}(n) + Bu(n) + C[\mu(n) - \mu(n-1)] \quad (7)$$

The matrix S is chosen suitably as a positive semi-definite weight matrix and P is a positive constant. Due to stochastic term $\mu(n) - \mu(n-1)$, the state vector may not be directly accessible for feedback. However, a separation principle can be applied to the solution of the regulator problem [9]. The regulator is designed by setting $\mu = 0$ and the optimal gains $\underline{G}$ are obtained from resulting problem. A Kalman filter can be used to estimate the state vector $\underline{x}$ in the presence of the noise signal imposed by the excess rate μ. The optimal gain $\underline{G}$ is expressed in terms of the solution of the algebraic Riccati equation W is

$$\underline{G} = \left(B^T W B + P\right)^{-1} B^T W A \quad (8)$$

where

$$W = S + A^T W\left(I - B\left(B^T W B + P\right)^{-1} B^T\right) WA$$

Once $\underline{G}$ is determined, the full nonlinear equations are used.

The results of computer simulations for the problem of controlling a single ABR source connected to a single node will be presented. The feedback coefficients for the regulator are obtained using the linearized system equations given by Eqns. 4 and 5. To obtain the feedback gains the matrix Riccati equation is iterated till a steady state solution is reached, $W(\infty)$. This result is used to obtain $\underline{G}(\infty)$. Finally, using the

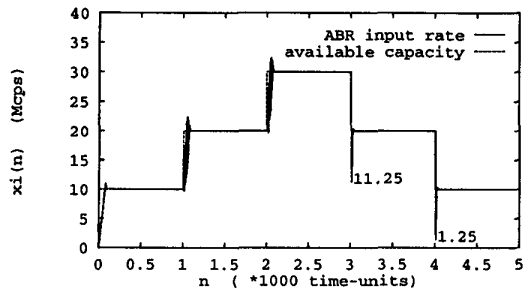

Figure 2. $\xi(n)$ for P=200 and D=6

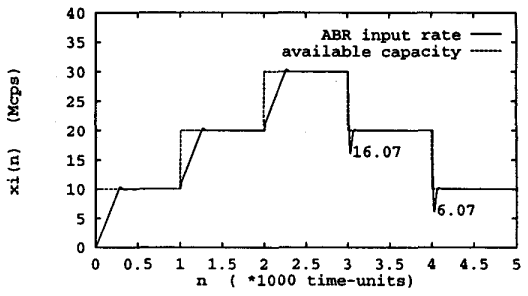

Figure 3. $\xi(n)$ for P=20,000 and D=6

computed gains the explicit rate $\xi(n)$ and the buffer occupancy $q(n)$ are determined using Eqns 1-3. The maximum and minimum rate, PCR and MCR, are ∞ and 0 respectively.

In this simulation, the magnitude of the available capacity μ will be varied with time. This variation will allow us to examine how well the ER tracks the available capacity. The available link capacity μ will be comprised of a series of step functions of successive amplitudes 10, 20, and 30 mega cells per second (Mcps), followed by a decrease to 20 and then 10 Mcps. The value of μ varies every 1000 time-units. The value of the threshold q^0 is set to 10 cells. To minimize the error $q(n) - q^0$ the weight matrix S is chosen such that $S_{11} = 1$ is the only nonzero value. The rate of convergence of rate and buffer occupancy can be controlled by adjusting the relative magnitude of P from 200 to 20,000. The round trip delay is held constant $D = 6$. The rate of the ABR source in each simulation is initialized to zero.

Figure 2 shows the simulation results of ABR rate for $P = 200$. The time required for the source rate ξ to reach μ is about 115 time-units. The overshoot of the explicit rate is 24.0% of μ while undershoot of the source rate are -87.5% of μ when the service rate decreases. The buffer occupancy increases occur during service rate decreases. In the case shown the maximum buffer occupancy of 83 cells occurs during the undershoot period.

Figure 3 shows the simulation result for the case where $P = 20,000$. Increasing the value of P slows the response time of the rate regulator. Hence the time required for the source rate to reach μ is 400 time-units. The overshoot of the source rate is 3.75% of μ and -39.3% when undershoots occurs. In the case shown the maximum buffer occupancy of 113 cells occurs during the undershoot period.

Hence as P is made smaller in value, the system exhibits more oscillations but requires less time to convergence to μ. When the available capacity decreases, the buffer occupancy overshoots are smaller with the smaller values of P. This is because weighting coefficient P is the penalty on the control input signal $u(n)$. Making P larger will result in a decrease of the magnitude of $u(n)$. This choice will cause the system to react slower when the available capacity changes value. Thus the convergence time will be longer. This implies that the system needs a longer time to let the buffer occupancy recover to the threshold, and therefore the overshoot of q will be larger.

The delay D is the most important parameter that effects the system's behavior. With the increase in the delay D, the system becomes harder to control and the controlled variables exhibit greater oscillation, longer convergence times and larger overshoot in the input rate and buffer occupancy. If the source rate could respond to the service rate perfectly, as shown in Figure 4, a phase shift of D time-units would be present.

When the available capacity increases, the decreased magnitude of the buffer occupancy is Δq_u, which equals $(\mu_2 - \mu_1)D$, or decreases to zero when Δq_u is greater than or equal to q^0. If the explicit rate increases properly, the queue size will increase to the threshold value q^0. When the available capacity decreases, the phase shift due to the delay causes most of the overshoot in $q(n)$. This overshoot is $\Delta q_o = (\mu_2 - \mu_3)D = D\Delta\mu$. So the larger the delay, the larger are the overshoot and buffer occupancy.

The control algorithms will reduce or increase the input rate to allow $q(n)$ to recover to the threshold value q^0. No matter how fast and how good a control algorithm is, it can not reduce the overshoot of the buffer occupancy below Δq_o.

4. Delay Estimation

The major assumption thus far has been that the round-trip propagation delay is known *a priori*, and

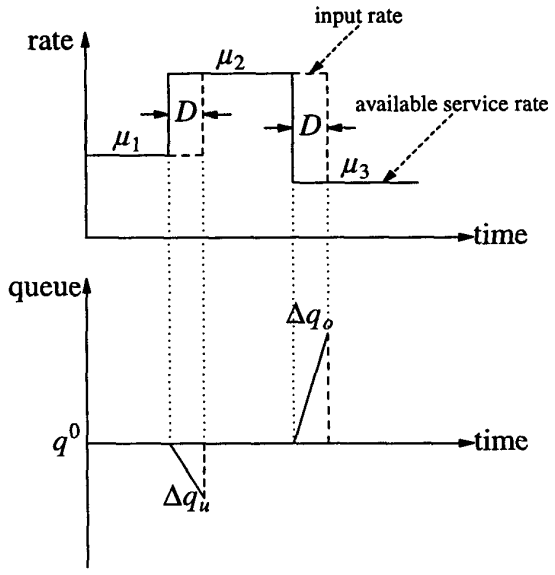

Figure 4. Buffer overshoot due to delay D

that this delay remains constant. In real networks, the delay is usually not known. In fact data may even be rerouted along a different path as a function of time. These path changes result in variations in the time-delay. In this section, a method for estimating the round-trip propagation delay is developed. To this end, the regulator will be used to compute the error between the actual and estimated time-delays.

Using Eqns 4 and 5 the buffer dynamics can be in a single equation. The actual round-trip delay is denoted by the variable D and the estimated delay used to design the controller is M time-units. When $M < D$, the system does yield a full rank characteristic polynomial. Thus, the system is not controllable. The system is considered robust to errors in the delay estimate if the poles can be held within the unit circle when the delay is varied over a prescribed range of values. The equation of $\hat{q}(n)$ that contains only the queue occupancy and the available transmission capacity variable

$$\begin{aligned}\hat{q}(n) &= 2\hat{q}(n-1) - \hat{q}(n-2) \\ &- \sum_{k=0}^{M} \beta_k \left[\hat{q}(n-k-1) - \hat{q}(n-k-2)\right] \\ &- \{\alpha_0 \hat{q}(n-D-1) + \alpha_1 \hat{q}(n-D-2)\} \\ &- \left\{\mu(n-1) - \mu(n-2) + \sum_{k=0}^{M} \beta_k \mu(n-k-2)\right\}\end{aligned} \quad (9)$$

Hence, the buffer occupancy equation is obtained

Table 1. Mean ans Standard Deviation of the Buffer Occupancy

Delay(Given) M	Delay(Actual) D	Mean m_q	Std. Dev. σ_q
2	2	99.996	3.39
2	5	99.872	13.75
2	10	101.154	58.13

where the only unknown parameter is the round-trip propagation delay D

Errors in the choice of the round-trip delay can cause instability in the system and/or large queueing delays. A single ABR source will serve as the input to a single node. The output rate of the source will be controlled by the ER calculated by a queue occupancy at a switch node. The available transmission capacity at the node is throttled by the cross-traffic generated by a single variable bit rate (VBR) source at the switch node [3]. The excess capacity is equal to $\mu(n) = max(VBR) - VBR(n)$ where $VBR(n)$ is the cell rate of the VBR source. The regulator is designed using the performance weight $S_{11} = 1$, and the other elements S_{ij} are set equal to zero. The performance weight of control input P is chosen equal to $2,000$. This number is chosen to make the explicit rate response to the change of service rate faster and to keep the buffer occupancy near the threshold. A buffer threshold q^0 of 100 cells allows for enough backlog to obtain the high link utilization. The regulator is designed using $M = 2$. The estimated delay is chosen to be smaller to ensure robust performance when the system is not fully controllable. The mean and standard deviation in the queue occupancy are shown in Table 1, Even though the mean value is also equal to the target buffer occupancy, the variance of the buffer occupancy is quite large. The range of variation is nearly equal to q^0. When the actual delay is 10 time-units, using a rate controller designed for a round-trip delay of 2 time-units yields an unstable system having roots equal to $1.03 \pm i0.145$.

The stability can be assured by adjusting the performance index such that the designed controller is robust under time-delay uncertainty. Figure 5 shows the effect of the performance index on the pole having the largest magnitude. The five lines in the figure present the location of the pole when P equals 20, 200, 2×10^3, 2×10^4, and 2×10^5 respectively. The symbols track the pole location, left to right, when the actual delay D is equal to 2, 5, 10, 15, 20, 30, 40, and 50. The

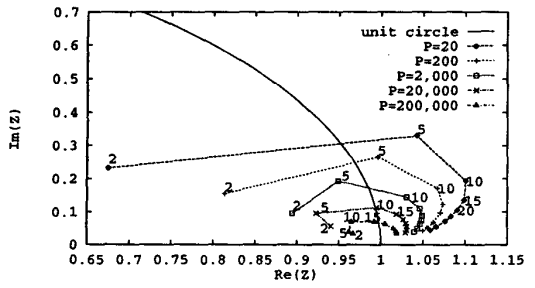

Figure 5. Effect of Performance Index on the Largest Pole, $M = 2$

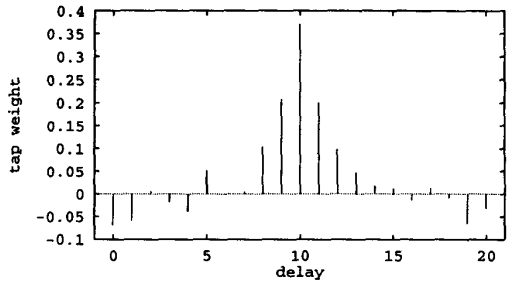

Figure 6. Tap Weights

controller is designed using an estimated delay $M = 2$. For each value of P, instability ensures when the error between the actual and estimated delay exceeds a threshold value. For higher values of P larger difference can be tolerated before onset of instability. Hence when the delay is uncertain, P should be large enough to assure robustness.

The simulation results illustrate that using an incorrect delay value impacts system performance. For small errors in the delay estimate, a robust controller can be designed to yield acceptable results. However, when larger errors occur, the time-delay must be estimated before a controller can be designed. Without estimation of the delay, the large variance in the buffer occupancy degrades the performance of the system. In such a case, cells will face longer waiting times in the queue or will be lost. For convenience, define

$$x_a(n) = \hat{q}(n) - 2\hat{q}(n-1) + \hat{q}(n-2) + \sum_{k=0}^{M} \beta_k[\hat{q}(n-k-1) - \hat{q}(n-k-2)] \quad (10)$$

$$x_e(n) = -[\alpha_0\hat{q}(n-1) + \alpha_1\hat{q}(n-2)] \quad (11)$$

$$v(n) = -[\mu(n-1) - \mu(n-2) + \sum_{k=0}^{M-1} \beta_k[\mu(n-k-2) - \mu(n-M-2)] \quad (12)$$

Thus, Eqn 9 can be expressed as follows,

$$x_a(n) = x_e(n-D) + v(n) \quad (13)$$

where $x_a(n)$ is equal to the sum of the D lagged delay version of $x_e(n)$ and the noise term $v(n)$. The time-shift operation $x_e(n-D)$ can be expressed as a linear convolution of x_e and an unknown filter $\underline{\hat{w}}(n)$. Using the filter $\underline{\hat{w}}(n)$, Eqn 13 can be rewritten as,

$$v(n) = x_a(n) - \sum_{k=0}^{D_{max}} \hat{w}_k(n)x_e(n-k) \quad (14)$$

Since x_a and x_e can be directly measured, the function $\underline{\hat{w}}$ can be determined using the Nonlinear Least-Mean-Square (NLMS) algorithm [7]. The number of taps used for $\hat{w}$ is equal to $D_{max}+1$ where the largest possible round-trip delay is D_{max}.

$$\hat{w}_i(n+1) = \hat{w}_i(n) + \frac{\lambda\left(x_a(n) - \sum_{k=0}^{D_{max}} \hat{w}_k(n)x_e(n-k)\right)}{\sum_{k=0}^{D_{max}} \|x_e(n-k)\|^2} \quad (14)$$

An experiment is simulated using Eqns 1-3. The actual round-trip delay set to $D = 10$, and the design delay $M = 2$. The excess capacity μ is modeled as Gaussian distributed random noise having a mean value equal to 1000 cells and variance equal to 1×10^4. The buffer threshold q_0 is set to 100 cells. The initial tap weight vector $\underline{\hat{w}}(0)$ is set to zero. The weight $S_{11} = 1$ and the remaining S_{ij} terms equal zero. A linearized equations were used to design the regulator. The weight of the control signal $P = 2 \times 10^3$. The relaxation parameter used in the NLMS algorithm is $\lambda = 2 \times 10^{-3}$ and $D_{max} = 30$. The controller was updated every 5 time units.

The tap weights $\hat{w}_i$ after 200 iterations is shown in Figure 6. The delay is determine by location of the maximum $\hat{w}_i$. The result differs from the ideal result which is a pulse. This is because the noise signal $v(n)$ is the weighted sum of previous excess rates. Hence, $v(n)$ is temporally correlated.

As mentioned earlier, simulations were carried out using VBR cross-traffic to modulate the excess capacity. In the cases examined, the round-trip D propagation delay varies as 5,10 and 20. The weight update

Table 2. Mean ans Standard Deviation of the Buffer Occupancy

Delay D	m_q w Est.	m_q w/o Est.	σ_q w Est.	σ_q w/o Est.
5	99.95	99.87	9.27	13.75
10	100.14	101.15	6.32	58.13
20	100.31	unstable	29.40	unstable

interval is 5 sampling time-units and the initial delay is $M = 2$. Table 2 presents the results of the buffer statistics with and without delay estimation. It is shown that delay estimation yields control results that are greatly improved. When the estimated delay $M = 2$ and the actual delay $D = 20$, the system is unstable. When the delay estimation is applied, the system adapts and converges to stability.

5. Conclusion

The simulation results demonstrate robustness to delay when the input weight P is large and the departures from the design delay are small. When the delay error is larger, the system can become unstable. It is shown that the round-trip propagation delay can be estimated using an NLMS algorithm. Large values of P should be chosen until the time delay is estimated. After the time-delay convergence is reached, the value of P can be decreased to obtain a faster response.

References

[1] A. Barnhart. Baseline performance using prca rate-control. *ATM Forum*, 94-0597, July 1994.

[2] L. Benmohamed and S. M. Meerkov. Feedback control of congestion in packet switching networks: The case of a single congested node. *Trans on Networking*, 1:693–707, 1993.

[3] K. Chandra and A. Reibman. Modeling one- and two-layer variable bit rate video. *IEEE/ACM Trans. on Networking*, 7:398–413, 1999.

[4] A. Charny, D. Clark, and R. Jain. Congestion control with explicit rate indication. *Proc. of IEEE ICC '95*, June 1995.

[5] T. Committee. Traffic management specification. *ATM Forum*, Af-tm-0056.000, April 1996.

[6] R. Jain, S. Kalyanaraman, and R. Viswanathan. The osu scheme for congestion avoidance using explicit rate indication. *ATM Forum*, 94-0883, September 1994.

[7] M. Montazeri and P. Duhamel. A set of algorithms linking nlms and block rls algorithms. *IEEE Trans. on Signal Processing*, 43:444–453, 1995.

[8] L. Roberts. Enhanced prca (proportional rate-control algorithm). *ATM Forum*, 94-0735R1, August 1994.

[9] M. Santina, A. Stubberud, and G. Hostetter. *Digital Control Systems*. Saunders College Publ., 1998.