Robust Rate Control for ABR Sources

Eitan ALTMAN, Tamer BAŞAR and R. SRIKANT

Abstract— **We consider the design of explicit rate-based flow control for ABR sources in an ATM network. The goal is to share the available capacity "fairly" among many sources while maintaining queue length at a desired level at the bottleneck node. This problem is formulated as a stochastic control problem and rate-control mechanisms are obtained which stabilize the queue length even though different sources may have different round-trip delays to the bottleneck node. Various robustness properties of the solution are illustrated through simulation experiments.**

Keywords— **ATM, ABR, Rate Control, Team Theory, Stochastic Control**

I. Introduction

Reactive flow control and congestion control are widely used today in the emerging telecommunications superhighways. They are designed to regulate the transmission of information from a source to a destination so as to avoid unacceptably long delays, high loss rates and other effects of congestion which are undesirable for both the individual source as well as the entire network. Examples of such controls are the TCP/IP congestion control in the Internet [11], and the control of ABR (Available Bit Rate) traffic in ATM (as was standardized by the ATM forum [1]). Such control mechanisms are especially used in regulating "best effort" type traffic, such as data transfer; this type of traffic is often non real-time, and thus the rate of its transmission can be adapted to the state of the network. This is in contrast with guaranteed bandwidth services (which are typically required by real-time voice or video applications) which have the drawback that only a fixed number of sessions can be handled; when the entire bandwidth is used, new arriving sessions are blocked.

The situation we analyze in this paper is one arising in the control of ABR traffic, for which the control actions are taken by the network. More precisely, the rate at which a source is allowed to transmit is determined by the switches on the virtual circuit between the source and the destination. This gives rise to a theoretical framework of a centralized control, as the control objectives and policies can be determined by a single entity – the network.

INRIA B.P. 93, 06902 Sophia Antipolis Cedex, France. Tel: (33) 93 65 77 86; Fax: 93 65 77 65. Email: altman@sophia.inria.fr

Coordinated Science Laboratory and Department of Electrical and Computer Engineering, University of Illinois, 1308 West Main Street, Urbana, IL 61801, USA. Tel: (217) 333-3607; Fax: (217) 244-1653; Email: tbasar@decision.csl.uiuc.edu. Research supported by NSF Grants NSF ECS 92-20632, NSF ECS 93-12807, and a Grant from from Rome Labs (Air Force contract F30602-96C0156) through Scientific Systems Company, Inc.

Coordinated Science Laboratory and Departmnet of General Engineering, University of Illinois, 1308 West Main Street, Urbana, IL 61801, USA. Tel: (217) 333-2457; Fax: 217-244-1642; Email: rsrikant@uiuc.edu. Research supported by an NSF CAREER Award NCR 9701525 and a Grant from Rome Labs (Air Force contract F30602-96C0156) through Scientific Systems Company, Inc.

Unlike the situation of the Internet, where window-type flow control is used, the control mechanism that has been chosen by the ATM Forum Technical Committee of Traffic Management [1] for controlling ABR traffic is the rate control approach [8]. The behavior of the sources and destinations for ABR traffic has been standardized, as well as the way in which information and control commands are conveyed by the ATM switches to the sources and destinations. Yet, the specific control mechanism has not been specified and left to the designer. The purpose of this paper is to design rate-based flow control for ABR type architectures using control-theoretic tools.

In doing so, we are faced with a situation of action delays: it takes time from the moment control information is sent to a source, until an action is taken by it, and until subsequently that action affects the state of the switch that had initiated that information. This approach uses an idealized linear queue dynamics, already introduced and justified in [6], [7], [12], a quadratic cost structure, as was used in the context of a single source control design in [3], and a model for the available bandwidth as an autoregressive (AR) process driven by an arbitrary, finite variance independently distributed sequence. This work differs from our earlier work on rate control [4] in two important respects:

- Here we assume that both queue length and rate information is available to the switch. This simplifies the form of the controller and leads to significantly better performance.
- We allow the white noise process driving the autoregressive model for the interfering (VBR or CBR) traffic to be non-Gaussian. Specifically, we let the noise be any white noise process with a finite variance. But the variance itself is *not* needed to derive the optimal controller, although the system performance will depend on the variance. Thus, the controller is robust to the statistics of the interfering traffic beyond the mean.

Our approach is to first show that general control problems where a single controller controls several sources with different *action delays* can be transformed to equivalent decentralized team-control problems with different *information delays*. We then solve the flow control problem as a decentralized LQ (Linear Quadratic) team. Team problems are notoriously difficult to solve, and admit optimal or near-optimal solutions only in special cases (see [14] and references within). Thus our proposed solutions are possibly suboptimal in general, but are optimal in the case where all the sources have equal action delays. However, even in the case of unequal delays, the proposed rate controllers perform well as indicated by simulation results.

II. Mathematical model, and derivation of optimal decentralized flow controllers

The model we adopt in this paper is a natural extension of the one considered in [4]. The essential difference being that we allow the bottleneck node to have access to available rate information. More specifically, we consider here a discrete-time model, where a time unit corresponds to a round trip delay. Let q_n denote the queue length at a bottleneck link, and μ_n denote the effective service rate available for ABR traffic in that link at the beginning of the nth time slot. Let r_{mn} denote the effective source rate for source m ($m = 1, \ldots, M$) during the nth time slot. This effective source rate may actually be the outcome of an action taken by source m several time steps earlier, say at $n - d_m$, where d_m is a positive integer that stands for *action delay.* In our formulation, we will not recognize this delay explicitly, but instead (see below) include a delay factor in the information available to each source for the construction of transmission rates for that source. As shown in *Appendix 1*, these two formulations are equivalent, that is, to any problem that incorporates action delays there corresponds one with information delays, and *vice versa*.

Now, in terms of the notation introduced, the queue length evolves according to

$$q_{n+1} \;=\; q_n + \sum_{m=1}^{M} r_{mn} - \mu_n \,. \tag{1}$$

Since several sources with varying transmission rates may share the same bottleneck link, the service rate μ_n available to the M sources may change over time in an unpredictable way. The other sources (e.g., CBR and VBR traffic) may then be represented as some interference, which we model as generated by a p-dimensional stable AR process:

$$\mu_n \;=\; \mu + \xi_n \tag{2}$$

$$\xi_n \;=\; \sum_{i=1}^{p} \alpha_i\, \xi_{n-i} + \phi_{n-1}\,, \tag{3}$$

where μ is the constant nominal service rate (known to all sources), and α_i, $i = 1, \ldots, p$, are known parameters. $\{\phi_n\}_{n\geq 1}$ is a zero-mean *i.i.d.* sequence with finite (but arbitrary) variance, denoted by k^2.

The objective function to minimize (collectively by all M sources) is

$$J = \limsup_{N\to\infty} \frac{1}{N} E \left\{ \sum_{n=1}^{N} \Big[(q_n - Q)^2 + \sum_{m=1}^{M} \frac{1}{c_m^2} \left(r_{mn} - a_m\, \mu_n \right)^2 \Big] \right\} \tag{4}$$

where Q is the target queue length, c_m's are some positive constants, and $\sum_{m=1}^{M} a_m = 1$. The *first additive term* above represents a penalty for deviating from a desirable queue length. The *second additive term* is a measure of the quality with which the input rate for each source tracks a given fraction of the available service rate, where the c_m's are weighting terms that serve to prioratize relative importances of these individual terms (among different sources as well as collectively with respect to the first additive term). For example, if we desire "fair" sharing of the available bandwidth, we would choose

$$a_1 = a_2 = \cdots = a_M = \frac{1}{M}\,,$$

assuming that everything else is also symmetric for the sources.

The information available to source m at time n is I_{n-d_m}, where

$$I_n := \{q_1, q_2, \ldots, q_n\,; \mu_1, \mu_2, \ldots, \mu_n\} \tag{5}$$

and d_m's denote delays in the acquisition of queue length information. Note that a crucial assumption here is that the delay for each source is independent of n. In view of this, and without any loss of generality, we can take d_m's to be ordered in the following way:

$$0 \leq d_1 \leq d_2 \leq \cdots \leq d_M\,.$$

Hence,

$$r_{mn} \;=\; \tilde{g}_{mn}(I_{n-d_m}),\; n = 1, 2, \ldots, ; \quad m = 1, 2, \ldots, M,$$

where $\tilde{g}_{mn}$ is some measurable function, with respect to which J will be minimized.

We now introduce the new (appropriately shifted) variables

$$x_n \;:=\; q_n - Q \tag{6}$$

$$u_{mn} \;:=\; r_{mn} - a_m \mu \tag{7}$$

which will serve as the state and the control, respectively. The queue dynamics (1) can be re-written in terms of these quantities as:

$$x_{n+1} \;=\; x_n + \sum_{m=1}^{M} u_{mn} - \xi_n \tag{8}$$

$$\xi_{n+1} \;=\; \sum_{i=1}^{p} \alpha_i\, \xi_{n+1-i} + \phi_n \tag{9}$$

and the cost function J as:

$$J = \limsup_{N\to\infty} \frac{1}{N} E \left\{ \sum_{n=1}^{N} \underbrace{\left[(x_n)^2 + \sum_{m=1}^{M} \frac{1}{c_m^2} (u_{mn} - a_m\, \xi_n)^2 \right]}_{\ell_n(x_n, u_n)} \right\} \tag{10}$$

where ℓ_n denotes the instantaneous cost. Let

$$\tilde{u}_{mn} := u_{mn} - a_m\, \xi_n\,; \quad \tilde{u}_n := (\tilde{u}_{1n}, \ldots, \tilde{u}_{Mn})'$$

Then, the instantaneous cost takes the pure quadratic form

$$\ell_n(x_n, u_n) = \tilde{\ell}_n(x_n, \tilde{u}_n = (x_n)^2 + |\tilde{u}_n|_R^2 \tag{11}$$

where $|\cdot|_R$ denotes the Euclidean norm in an appropriate finite-dimensional vector space, weighted by R, and

$$R := \text{ diag } \left(\frac{1}{c_1^2}, \cdots, \frac{1}{c_M^2}\right). \tag{12}$$

Also, in terms of $\tilde{u}$, the dynamics for x become:

$$x_{n+1} = x_n + b' \, \tilde{u}_n \tag{13}$$

where

$$b' := (\, 1 \; 1 \; \cdots \; 1 \,) \tag{14}$$

Hence, if x_n (as well as ξ_n) were available at time n, we would have had a standard discrete-time linear-quadratic regulator problem, which admits the unique solution [5] :

$$\tilde{u}_n \;\; = \;\; -[R + bb's]^{-1} bs \, x_n \tag{15}$$

where s is the unique positive root of the algebraic equation

$$s \;\; = \;\; 1 + s[1 - b'(R + bb's)^{-1} bs] \tag{16}$$

This positive root can in fact be explicitly computed:

$$s = \frac{1 + \sqrt{1 + 4c^2}}{2}; \quad c^2 := \left(\sum_{m=1}^{M} \frac{1}{c_m^2}\right)^{-1} \tag{17}$$

The corresponding control $\tilde{u}$ renders the closed-loop queue dynamics asymptotically stable:

$$x_{n+1} = \frac{1}{2}\left(\frac{1 + 2c^2 - \sqrt{1 + 4c^2}}{c^2}\right) x_n \, .$$

Note that in this case the AR process (which is stable by our initial hypothesis) is completely decoupled from the queue dynamics. Further note that in spite of this decoupling, the original control u_n still has to use the information on ξ_n, since the inverse transformation from $\tilde{u}_n$ to u_n yields

$$u_n \;\; = \;\; \tilde{u}_n + \left(\begin{array}{c} a_1 \\ \vdots \\ a_M \end{array}\right) \xi_n. \tag{18}$$

The state z_n is, however, not available to u_{mn}, but only I_{n-d_m} is. What we have here then is a decentralized optimal control problem, but since the information is *nested* i.e., what is available to source m at time n is available to all sources $m' < m$), it is possible to obtain an optimal linear solution (linear in available queue length and rate information) to a finite horizon version of this problem (i.e., $N < \infty$) problem using dynamic programming. However, whether such a solution has an easily implementable structure is still an open problem. Therefore, we restrict our attention to certainty-equivalent controller, i.e., controllers where, in the above solution (18), we replace the queue length and rate with their estimates. However, even certainty-equivalent controllers are not easy to define for team problems since each source uses a different estimate of the queue length and rate. In Appendix 2, we show that certainty-equivalent controllers can be defined in many ways and each such controller could lead to a different value for the performance index. We present two certainty-equivalent controllers in the following subsections.

A. Controller 1

Here we present a specific form of a certainty-equivalent controller where we simply replace z_n by the best estimate for z_n in the expression for u_{mn} :

$$u^*_{mn} \;\; = \;\; g^*_{mn}(I_{n-d_m}) = -p_m \hat{x}^{m,n}_{n|n-d_m} + a_m \hat{\xi}^{m,n}_{n|n-d_m} \tag{19}$$

where p_m is the m^{th} component of $[R + bb's]^{-1} bs$ and $\hat{x}$, $\hat{\xi}$ are generated by

$$\begin{aligned}
\hat{x}^{m,n}_{j+1|n-d_m} &= \hat{x}^{m,n}_{j|n-d_m} + \sum_{i=1}^{M} \hat{u}^{m,n}_{i,j} - \hat{\xi}^{m,n}_{j|n-d_m} \qquad (20)\\
\hat{x}^{m,n}_{n-d_m|n-d_m} &= x_{n-d_m} \\
\hat{\xi}^{m,n}_{n-d_m+1|n-d_m} &= \sum_{i=1}^{p} \alpha_i \, \xi_{n+1-i-d_m} \\
\hat{\xi}^{m,n}_{n-d_m+2|n-d_m} &= \alpha_1 \, \hat{\xi}^{m,n}_{n-d_m+1|n-d_m} + \sum_{i=2}^{p} \alpha_i \, \xi_{n+2-i-d_m} \\
&\;\;\vdots \\
\hat{\xi}^{m,n}_{n-d_m+p|n-d_m} &= \sum_{i=1}^{p-1} \alpha_i \, \hat{\xi}^{m,n}_{n-d_m+p-i|n-d_m} + \alpha_p \xi_{n-d_m} \\
\hat{\xi}^{m,n}_{j|n-d_m} &= \sum_{i=1}^{p} \alpha_i \, \hat{\xi}^{m,n}_{j-i|n-d_m}, \; j > n - d_m + p \; (21)
\end{aligned}$$

and

$$\hat{u}^{m,n}_{i,n+\ell-d_m} := \begin{cases} -p_m \hat{x}^{m,n}_{n+l-d_m|n-d_m} + a_m \hat{\xi}^{m,n}_{n+l-d_m|n-d_m} & \text{if } \ell > d_i \\ u^*_{i,n+\ell-d_m} & \text{if } \ell \leq d_i \end{cases} \tag{22}$$

These are the recursive equations generating the estimators for the sources. In the above set of equations, $\hat{\xi}^{m,n}_{l|n-d_m}$ denotes Source m's estimate at time n of the value of ξ at time l based on I_{n-d_m}. A similar interpretation holds for $x^{m,n}_{l|n-d_m}$. While these equations have been presented in the language of decentralized control, we wish to emphasize that the rate control commands are issued by a centralized decision maker, namely, the node. However, due to the fact that the round-trip delays from the bottleneck node to different sources are different, this is equivalent controller to a decentralized control problem. Thus, from a decentralized control point of view, the certainty-equivalent simply replaces x_n and ξ_n in the expression for u_{mn} by Source m's estimate of the queue length and the available rate at the bottleneck node. In terms of the node's actions, this corresponds to an estimate of the queue length and rate information at a future time, where the future time is the current time plus the action delay for Source m.

The above algorithm is relatively easy to implement. The estimator algorithms are simple scalar operations and the scalar solution of the Riccati equation s has been obtained explicitly. The only matrix operation ivolved is the inversion of the matrix $(R + bb's)$. We can reduce this to simple scalar operations by applying the matrix inversion lemma [13, Appendix D].

Lemma 1: Suppose that matrices A, C and $A+B'C^{-1}B$ are invertible, then

$$(A+B'C^{-1}B)^{-1} = A^{-1} - A^{-1}B'(BA^{-1}B'+C)^{-1}BA^{-1}.$$

■

Applying the matrix inversion lemma, we obtain

$$(R+bb's)^{-1} = R^{-1} - R^{-1}b(b'R^{-1}b+\frac{1}{s})^{-1}b'R^{-1}.$$

Since R is a diagonal matrix,

$$R^{-1} = diag(c_1^2, c_2^2, \ldots, c_M^2),$$

and

$$b'R^{-1}b = \sum_{i=1}^{M} c_i^2.$$

Thus, for $i=j$, the $(i,j)^{\text{th}}$ element of $(R+bb's)^{-1}$ is given by

$$c_i^2 - \frac{sc_i^4}{1+s\sum_{j=1}^{M} c_j^2},$$

and, for $i \neq j$, it is given by

$$-\frac{sc_i^2 c_j^2}{1+s\sum_{j=1}^{M} c_j^2}.$$

Thus,

$$\begin{aligned} p_m &= c_i^2 - \frac{sc_i^4}{1+s\sum_{j=1}^{M} c_j^2} - \frac{s\sum_{j=1,j\neq m}^{M} c_m^2 c_j^2}{1+s\sum_{j=1}^{M} c_j^2} \\ &= \frac{c_m^2}{1+s\sum_{j=1}^{M} c_j^2}. \end{aligned}$$

B. Controller 2

First, we represent the optimal controller for the deterministic problem (15) in a different manner as follows: By noting that $(R+bb's)$ is a positive definite matrix, it follows that it has a Cholesky decomposition of the form $U'U$, where U is a non-singular, upper-triangular matrix [10]. Thus, (15) can be rewritten as

$$U\tilde{u}_n = -\tilde{x}_n, \qquad (23)$$

where $\tilde{x}_n := (U')^{-1}bsx_n$. Let U_{ij} represent the $(i,j)^{\text{th}}$ element of U and q_m represent the m^{th} element of $(U')^{-1}bs$. Then, it is easy to see that (23) can be written as

$$\tilde{u}_{mn} = -\frac{1}{U_{mm}}\left(\sum_{j=m+1}^{M} U_{mj}\tilde{u}_{jn} + q_m x_n\right). \qquad (24)$$

Thus,

$$\begin{aligned} u_{mn} = &-\tfrac{1}{U_{mm}}\left(\sum_{j=m+1}^{M} U_{mj}u_{jn} + q_m x_n\right) \\ &+ \left(a_m + \tfrac{1}{U_{mm}}\sum_{j=m+1}^{M} U_{mj}a_j\right)\xi_n. \end{aligned} \qquad (25)$$

Thus, certainty-equivalent Controller 2 simply replaces x_n and ξ_n in the expression for u_{mn} by Source m's estimate of the queue length and the available rate at the bottleneck node while using the new representation (25) for the controllers:

$$\begin{aligned} u^*_{mn} &= g^*_{mn}(I_{n-d_m}) \\ &= -\frac{1}{U_{mm}}\left(\sum_{j=m+1}^{M} U_{mj}u_{jn} + q_m\hat{x}^{m,n}_{n|n-d_m}\right) \quad (26) \\ &\quad + \left(a_m + \frac{1}{U_{mm}}\sum_{j=m+1}^{M} U_{mj}a_j\right)\hat{\xi}^{m,n}_{n|n-d_m} \quad (27) \end{aligned}$$

where $\hat{x}$, $\hat{\xi}$ are generated by

$$\begin{aligned} \hat{x}^{m,n}_{j+1|n-d_m} &= \hat{x}^{m,n}_{j|n-d_m} + \sum_{i=1}^{M}\hat{u}^{m,n}_{i,j} - \hat{\xi}^{m,n}_{j|n-d_m}, \quad (28) \\ & \qquad j = n-d_m, n-d_m+1, \ldots, n-1 \\ \hat{x}^{m,n}_{n-d_m|n-d_m} &= x_{n-d_m} \\ \hat{\xi}^{m,n}_{n-d_m+1|n-d_m} &= \sum_{i=1}^{p}\alpha_i\,\xi_{n+1-i-d_m} \\ \hat{\xi}^{m,n}_{n-d_m+2|n-d_m} &= \alpha_1\,\hat{\xi}^{m,n}_{n-d_m+1|n-d_m} + \sum_{i=2}^{p}\alpha_i\,\xi_{n+2-i-d_m} \\ &\vdots \\ \hat{\xi}^{m,n}_{n-d_m+p|n-d_m} &= \sum_{i=1}^{p-1}\alpha_i\,\hat{\xi}^{m,n}_{n-d_m+p-i|n-d_m} + \alpha_p\xi_{n-d_m} \\ \hat{\xi}^{m,n}_{j|n-d_m} &= \sum_{i=1}^{p}\alpha_i\,\hat{\xi}^{m,n}_{j-i|n-d_m},\ j > n-d_m+p, \quad (29) \end{aligned}$$

for $\ell > d_i$,

$$\begin{aligned} &\hat{u}^{m,n}_{i,n+\ell-d_m} := \\ &-\tfrac{1}{U_{mm}}\left(\sum_{j=m+1}^{M} U_{mj}u_{jn} + q_m\hat{x}^{m,n}_{n+l-d_m|n-d_m}\right) \\ &+ \left(a_m + \tfrac{1}{U_{mm}}\sum_{j=m+1}^{M} U_{mj}a_j\right)\hat{\xi}^{m,n}_{n+l-d_m|n-d_m} \end{aligned} \qquad (30)$$

and, for $\ell \le d_i$,

$$\hat{u}^{m,n}_{i,n+\ell-d_m} := u^*_{i,n+\ell-d_m} \qquad (31)$$

The computational complexity for Controller 2 is a little more than Controller 1. However, we show through simulations that it performs better than Controller 2, when one has good estimates of the action delay. Specifically, the implementation off Controller 2 requires the computation of a Cholesky decomposition of the matrix $(R+bb's)$. The special structure of the problem allows us to explictly compute this decomposition. The expression for the $(i,j)^{th}$ entry of U is given below:

$$U_{ii} = \sqrt{\frac{1+\sum_{j=1}^{i} c_j^2}{c_i^2(1+s\sum_{j=1}^{i-1} c_j^2)}}.$$

For $j > i$,

$$U_{ij} = \frac{s\sqrt{c_i^2}}{\sqrt{(1+s\sum_{j=1}^{i} c_j^2)(1+s\sum_{j=1}^{i-1} c_j^2)}}.$$

Since U is upper triangular, $U_{ij} = 0$ for $j < i$.

III. Simulation Results

A. Comparative Performance of the Controllers

c	$E(x^2)$	$E(u_1^2)$	$E(u_2^2)$	$E(u_3^2)$	J
10.0	2.69	0.89	0.63	0.49	8.68
1.0	3.18	0.84	0.60	0.49	11.44
0.1	7.19	0.38	0.48	0.5	52.36

TABLE I

Controller 1 with $d_1 = 0, d_2 = 5, d_3 = 10$, $a_1 = a_2 = a_3 = 1/3$, $c_1 = c_2 = c_3 = c$, $p = 2$, $\alpha_1 = \alpha_2 = 0.4$, Gaussian ϕ_n, $k^2 = 1$, Simulation Run Length $N = 1000$

c	$E(x^2)$	$E(u_1^2)$	$E(u_2^2)$	$E(u_3^2)$	J
10.0	0.01	1.48	1.18	0.49	3.68
1.0	0.80	1.01	0.82	0.50	2.46
0.1	6.00	0.38	0.47	0.50	36.59

TABLE II

Controller 2 with $d_1 = 0, d_2 = 5, d_3 = 10$, $a_1 = a_2 = a_3 = 1/3$, $c_1 = c_2 = c_3 = c$, $p = 2$, $\alpha_1 = \alpha_2 = 0.4$, Gaussian ϕ_n, $k^2 = 1$, Simulation run length $N = 1000$

In Tables I and II, we have presented the performance of the two controllers for various values of the problem parameters. In the results shown here, as well as many other simulation experiments that we have performed, Controller 2 always performs better than Controller 1. We do not have a proof that is indeed always the case, but our experimental results strongly suggest so. The tables also show that one can control the queue length fairly tightly around a nominal level Q, by increasing c^2. Also, as be expected, the mean values of x, u_1, u_2 and u_3 were all observed to be close to zero. This shows that the controller is fair in the long-term. The fact that the variances of u_i for the three sources are of approximately the same magnitude also shows that the fluctuation from fair allocation is about the same for all sources, despite the different action delays.

B. White Noise Process

We have observed that the certainty-equivalent controllers use the best-estimate of the state variables consistent with the control policies of all the other users. An interesting feature of these controllers is that both the control gains and estimator equations are independent of the variance of the white noise process $\{\phi_n\}$ which drives the dynamics of the AR process $\{\xi_n\}$. Furthermore, the resulting value of the cost function J, which does depend on the variance, is independent of any higher-order statistics of the white noise process. This has been proved in [4] for the special case where the controllers have access only to the queue length information, but not the rate information. However, the same argument applies here too.

We repeat the experiments in Section III-A with a uniform distribution for ϕ_n. The results, shown in Tables III and IV, indicate that the overall performance is unaffected by the statistics of the white noise process beyond the variance.

c	$E(x^2)$	$E(u_1^2)$	$E(u_2^2)$	$E(u_3^2)$	J
10.0	2.72	0.90	0.62	0.48	8.85
1.0	3,21	0.85	0.60	0.49	11.65
0.1	7.12	0.38	0.49	0.49	51.30

TABLE III

Controller 1 with $d_1 = 0, d_2 = 5, d_3 = 10$, $a_1 = a_2 = a_3 = 1/3$, $c_1 = c_2 = c_3 = c$, $p = 2$, $\alpha_1 = \alpha_2 = 0.4$, Uniform ϕ_n, $k^2 = 1$, Simulation Run Length $N = 1000$

c	$E(x^2)$	$E(u_1^2)$	$E(u_2^2)$	$E(u_3^2)$	J
10.0	0.01	1.41	1.09	0.48	3.29
1.0	0.81	1.02	0.79	0.49	2.43
0.1	5.97	0.37	0.48	0.49	36.10

TABLE IV

Controller 2 with $d_1 = 0, d_2 = 5, d_3 = 10$, $a_1 = a_2 = a_3 = 1/3$, $c_1 = c_2 = c_3 = c$, $p = 2$, $\alpha_1 = \alpha_2 = 0.4$, Uniform ϕ_n, $k^2 = 1$, Simulation run length $N = 1000$

C. Unknown Delays

We have not addressed the issue of queue-length stability here. We say that the queue is *stable* if its steady-state variance is finite. It can be proved that the controllers that we have presented here are stable. This proof is not included here due to space limitations and will be presented elsewhere. However, the stabilizing property of the controller is valid only when we have perfect knowledge of the delay. In a real network, delays would be unknown. It can also be shown that the controller would be stabilizing for a large range of delay values if the controller gain is small, i.e., p_m is small which would correspond to large penalties on the control action in the cost function. In other words, c_i^2 should be small. Again this result is not included here, but we show simulation results to justify this. Table V presents the performance of Controller 1 when the delay is underestimated and Table VI presents the results for Controller 2. In both cases, for sufficiently small values of c_i^2, the system is stable.

D. Random Delays

In addition to the delays being unknown, in a realistic network situation, the delays could be variable. We model

c	$E(x^2)$	$E(u_1^2)$	$E(u_2^2)$	$E(u_3^2)$	J
10.0	∞	∞	∞	∞	∞
1.0	∞	∞	∞	∞	∞
0.1	11.39	0.98	0.51	0.52	131.29

TABLE V

Controller 1 designed for $d_1 = 0$, $d_2 = 5$, $d_3 = 10$, $a_1 = a_2 = a_3 = 1/3$, $c_1 = c_2 = c_3 = c$, $p = 2$, $\alpha_1 = \alpha_2 = 0.4$, Gaussian ϕ_n, $k^2 = 1$. Actual Delays $d_1 = 10$, $d_2 = 5$, $d_3 = 0$. Simulation run length $N = 1000$.

c	$E(x^2)$	$E(u_1^2)$	$E(u_2^2)$	$E(u_3^2)$	J
10.0	∞	∞	∞	∞	∞
1.0	∞	∞	∞	∞	∞
0.1	11.46	1.07	0.55	0.52	133.04

TABLE VI

Controller 2 designed for $d_1 = 0$, $d_2 = 5$, $d_3 = 10$, $a_1 = a_2 = a_3 = 1/3$, $c_1 = c_2 = c_3 = c$, $p = 2$, $\alpha_1 = \alpha_2 = 0.4$, Gaussian ϕ_n, $k^2 = 1$. Actual Delays $d_1 = 10$, $d_2 = 5$, $d_3 = 0$. Simulation run length $N = 1000$.

this variability with a parameter γ which lies in $[0,1]$, and delays $\tilde{d}_1$, $\tilde{d}_2$ and $\tilde{d}_3$. For each Source i, we assume that with probability γ, the delay is

$$\min(\text{delay at previous time instant} + 1, \tilde{d}_i),$$

and with probability $(1-\gamma)$, the delay is

$$\min(\text{delay at previous time instant} + 1, d_i).$$

Thus, the delay varies randomly between the value d_i assumed by the controller and some unknown values $\tilde{d}_i$. Again, we observe from Tables VII and VIII that choosing the c_i^2 small ensure robustness against variable delay.

c	$E(x^2)$	$E(u_1^2)$	$E(u_2^2)$	$E(u_3^2)$	J
10.0	∞	∞	∞	∞	∞
1.0	∞	∞	∞	∞	∞
0.1	10.50	0.88	0.51	0.52	111.69

TABLE VII

Controller 1 designed for $d_1 = 0$, $d_2 = 5$, $d_3 = 10$, $a_1 = a_2 = a_3 = 1/3$, $c_1 = c_2 = c_3 = c$, $p = 2$, $\alpha_1 = \alpha_2 = 0.4$, Gaussian ϕ_n, $k^2 = 1$, $\tilde{d}_1 = 10$, $\tilde{d}_2 = 5$, $\tilde{d}_3 = 0$, $\gamma = 0.95$, Simulation run length $N = 5000$.

c	$E(x^2)$	$E(u_1^2)$	$E(u_2^2)$	$E(u_3^2)$	J
10.0	∞	∞	∞	∞	∞
1.0	∞	∞	∞	∞	∞
0.1	6.52	0.53	0.46	0.53	43.14

TABLE VIII

Controller 2 designed for $d_1 = 0$, $d_2 = 5$, $d_3 = 10$, $a_1 = a_2 = a_3 = 1/3$, $c_1 = c_2 = c_3 = c$, $p = 2$, $\alpha_1 = \alpha_2 = 0.4$, Gaussian ϕ_n, $k^2 = 1$, $d_1 = 10$, $d_2 = 5$, $d_3 = 0$, $\gamma = 0.95$, Simulation run length $N = 5000$.

IV. Discussion

We have presented two algorithms for rate control that are derived using tools from stochastic control. Specifically, we have shown that the rate-control problem is equivalent a so-called team problem, where each member of the team has access to different information, yet they try to minimize the same objective. Again we wish to emphasize that the decentralized nature of the problem arises in the mathematical formulation of the problem. The actual implementation is performed at a centralized entity, namely, the node. We have also shown that the formulation has built-in capabilities to be robust against unknown and variable action delays as shown in Sections III-C and III-D. There are several avenues of further research that we are currently investigating:

- RM (rate management) cells: Here we have assumed that each source receives feedback from the node from the controller at each time instant. Under the current ATM standards, an RM cell is generated every Nrm data cells, where Nrm is typically 32. The only exception is when the source's data rate becomes very low, in which case, other special mechanisms come into play. We believe that our controller can stabilize queue length even when the feedback mechanism is implemented using RM cells. In fact, our results on random delays suggest that the controller can be made robust even when the feedback is random.
- Unknown number of active users: In this formulation, we assume that the number of active users is known. If this is unknown, then one can use other work, such as in [9], to estimate the number of active users and use that as the parameter M in our formulation.
- Peak rate constraints and bursty sources: If a source cannot send the rate at which the node asks it to, either due to peak rate constraints or simply because it is not currently generating traffic at that rate, then one can use the ER field in the forward RM cell to indicate to the node the maximum rate at which the source can currently transmit. Then, this rate could then be subtracted from the available mean rate at the node, the number of active sources (M) can be reduced by one and the algorithm can proceed as usual.
- AR process: We have assumed that the parameters of the AR process are known. In practice, the node has to estimate this. Preliminary results indicate that the use of standard estimators such as recursive least-squares, in conjunction with our controller, performs well.
- Network model: We are currently incorporating all the above modifications into our basic algorithm and performing simulations on a "parking lot" configuration [8]. The results will be reported in a longer version of this paper.

V. Appendix 1: Equivalence of action and information delays

We prove in this *Appendix* the equivalence of dynamic team problems with two types of delays, namely:
(i) *information delay*: at time t the controller knows about the state of the system only until time $t-N_1$, where $N_1>0$ is the corresponding delay.
(ii) *action delay*: the action taken by the controller at time t influences the state of the system only at time $t+N_2$ where $N_2>0$ is the corresponding delay.

The equivalence we will establish is in the following sense: A team problem with action delays between a central controller and several controlled units can be transformed into an equivalent team problem with no action delays but with different information delays.

Discrete-time control problem with action delays.

Considering a class of team problems more general than the one formulated in the main body of the paper, let there be M units (e.g. sources) controlled by a central controller. Define a vector $\theta=(\theta_1,...,\theta_M)$ of action-delays: the impact of a control decision taken by the central controller on the mth unit is affected only θ_m time units later, as will become clear in (32) below.

Let $\mathbf{X}$ be some state-space, $\mathbf{A}=(\mathbf{A}_1\times\mathbf{A}_2\times\ldots\times\mathbf{A}_M)$ a product action space, and $\mathcal{Y}$ the space of measurements. Let x_n be the state at time n, and u_{mn} the action taken by unit m at time n. We define the vector of "relevant actions" at time n as $w_n=(w_{1n},...,w_{Mn}):=(u_{1,n-\theta_1},u_{2,n-\theta_2},\ldots,u_{M,n-\theta_M})$. The state of the system at time $n+1$ will be determined by the state at time n, the relevant actions w_n at time n, and some other quantity (such as random noise), ϵ_n, taking values in some space $\mathcal{E}$:

$$x_{n+1}=f_n(x_n,w_n,\epsilon_n),\qquad n\in I\!N:=\{1,2,\ldots\},\tag{32}$$

where f_n is a given function $f_n:\mathbf{X}\times\mathbf{A}\times\mathcal{E}\rightarrow\mathbf{X}$.

The measurement y_n at time n is a function of the current system state, x_n, and some other quantity (such as random noise), v_n, taking values in some space $\mathcal{V}$; that is,

$$y_n=h(x_n,v_n),\qquad n\in I\!N,\tag{33}$$

for some functions $h_n:\mathbf{X}\times\mathcal{V}\rightarrow\mathcal{Y}$.

The composite control is a sequence of vectors $\{u_{mn},n\in I\!N,\ m=1,...,M\}$, where u_{mn} is the command sent to unit m at time n, which is a function of the available information:

$$u_{mn}=g_{mn}(\{y_j\},1\le j\le n).$$

A discrete-time team control problem with delayed information.

Now we consider the situation of M different controllers that are faced with a team problem. The information for controller m on the system state is acquired after a delay of θ_m time units.

We consider the same spaces $\mathbf{X},\mathbf{A}$ and $\mathcal{Y}$ of states, actions and measurements, respectively, as well as the spaces of driving sequences, $\mathcal{E}$ and $\mathcal{V}$. Let $\overline{x}_n$ be the state at time n, and $\overline{u}_{mn}$ the action taken by controller m at time n. Define the vector of actions at time n by $\overline{w}_n=(\overline{u}_{1n},\overline{u}_{2n},\ldots,\overline{u}_{Mn})$. The state of the system at time $n+1$ will be determined by the state and actions at time n and the state of the environment, ϵ_n:

$$\overline{x}_{n+1}=f_n(\overline{x}_n,\overline{w}_n,\epsilon_n),\qquad n\in I\!N,\tag{34}$$

where f_n and ϵ_n are defined as in (32). The measurement sequence $\{\overline{y}_n\}$ and the noise sequence $\{v_n\}$ are related to each other as in (33):

$$\overline{y}_n=h(\overline{x}_n,v_n),\qquad n\in I\!N.\tag{35}$$

The composite control is a sequence of vectors $\{u_{mn},n\in I\!N,m=1,...,M\}$, where u_{mn} is the command taken by unit m at time n, which is a function of the available information:

$$\overline{u}_{mn}=\overline{g}_{mn}(\{\overline{y}_j\},1\le j\le n-\theta_m),$$

with the convention adopted being that for $n<\theta_m$, $\overline{u}_{mn}$ is just a constant element of $\mathbf{A}_m$.

The equivalence between the two models

Theorem 1: Suppose that

$$\overline{x}_n=x_n,\quad n=1,2,...,n_0$$

where $n_0:=\max_i\theta_i$, and

$$w_{mn}=\overline{u}_{mn},\quad 1\le n<\theta_m,$$

and let $\overline{g}_{mn}=g_{mn}$ for all m and $n\ge n_0$. Then the same state, measurement and action trajectories are obtained in the two systems:

$$x_n=\overline{x}_n,\quad w_n=\overline{u}_n,\quad y_n=\overline{y}_n,\quad n=1,2,...$$

Proof: The assumptions of the Theorem imply that $y_j=\overline{y}_j$ for $1\le j\le n_0$. This, together with the fact that $\overline{g}_{mn}=g_{mn}$ implies that

$$w_{mn}=\overline{u}_{mn},\quad m=1,...,M\tag{36}$$

for $n=\theta_m$, which in turn implies that

$$x_n=\overline{x}_n,\qquad\text{and}\qquad y_n=\overline{y}_n\tag{37}$$

for $n=n_0+1$. Continuing by induction, (36) and (37) are then seen to hold for all n. ∎

When optimal control problems are considered, Theorem 1 can be used to transform a centralized control problem with different action delays into a team problem with different information delays, provided that the corresponding costs are defined in a compatible way; more precisely, if the cost to be minimized in the first problem is a function of the trajectories $\{x_n,w_n,n\in I\!N\}$, and the cost in the second problem is the same function of the trajectories $\{\overline{x}_n,\overline{w}_n,n\in I\!N\}$, then any optimal policy for one yields (through Theorem 1) an optimal policy for the other problem.

VI. Appendix 2: Certainty-Equivalence in Team Problems

In this appendix, we provide an example to show certainty-equivalent controllers for team problems can be derived from many different representations of the solution to stochastic control obtained obtained by assuming all sources have access to all the random variables without any delay. Furthermore, each of these controllers could lead to a different value for the cost function. Consider the following stochastic optimization problem:

$$\min_{u_1,u_2} E\left\{x_1^2+u_1^2+u_2^2\right\},$$

where

$$x_1 = x_0+u_1+u_2-\xi_1,$$

$$\xi_1 = \alpha\xi_0+\phi.$$

The random variables ξ_0, x_0 and ϕ are mean zero, Gaussian random variables with variances $\sigma^2_{\xi_0}$, $\sigma^2_{x_0}$ and σ^2_{ϕ}, respectively. The minimization is performed over the space of functions $u_1=\mu(I_1)$ and $u_2=\mu(I_2)$ where the information available to Sources 1 and 2, I_1 and I_2, respectively, are yet to be specified.

First let us consider the following information structure:

$$I_1 = I_2 = \{x_0,\xi_0,\xi_1\}.$$

In this case, it is easy to verify that the optimal solution is given by

$$u_1 = u_2 = \frac{-1}{3}(x_0-\xi_1). \tag{38}$$

This solution can also be represented as

$$u_1 = \frac{-1}{2}(x_0+u_2-\xi_1), \tag{39}$$

$$u_2 = \frac{-1}{3}(x_0-\xi_1). \tag{40}$$

To see that (39)-(40) is indeed just another way to represent (38), one can simply substitute for u_2 from (40) in (39) and verify that the expression for u_1 in (38) is obtained.

Now let us consider a decentralized team problem where the information structures of Sources 1 and 2 are

$$I_1=\{x_0,\xi_0,\xi_1\}, I_2=\{x_0,\xi_0\}.$$

Suppose that we apply certainty-equivalence to this problem. Thus, we have to replace the ξ_1 in u_1 by $E(\xi_1|I_1)$ and ξ_1 in u_2 by $E(\xi_1|I_0)$. However, doing this in (38) and (39)-(40) yields different costs for the team problem.

Under Representation 1 (38), the certainty-equivalent solution is given by

$$u_1 = \frac{-1}{3}(x_0-\xi_1),$$

and

$$u_2 = \frac{-1}{3}(x_0-\alpha\xi_0).$$

The cost under this representation is

$$J = \frac{1}{3}\sigma^2_{x_0}+\frac{5}{9}\sigma^2_{\phi}+\frac{\alpha^2}{3}\sigma^2_{\xi_0}.$$

Using Representation 2 (39)-(40), the certainty-equivalent solution is given by

$$u_1 = \frac{-1}{2}(x+u_2-\xi_1),$$

and

$$u_2 = \frac{-1}{3}(x_0-\alpha\xi_0).$$

The cost under this representation is

$$J = \frac{1}{3}\sigma^2_{x_0}+\frac{1}{2}\sigma^2_{\phi}+\frac{\alpha^2}{3}\sigma^2_{\xi_0}.$$

Thus, the second representation yields a better certainty-equivalent controller. The controllers desribed in this paper, Controller 1 and Controller 2, correspond to the first and second representations, respectively, in this example. While we do not have a general proof that Controller 2 always outperforms Controller 1, this example demonstrates that this is true at least for two-stage problems, i.e., problems in which $N=2$.

References

[1] The ATM Forum Technical Committee, *Traffic Management Specification*, Version 4.0, 95-0013R8, April 1996.

[2] E. Altman, F. Baccelli and J. C. Bolot, "Discrete-time analysis of adaptive rate control mechanisms", *High Speed Networks and their performance*, H.G. Perros and Y. Viniotis Eds., North Holland, 121-140, 1994.

[3] E. Altman and T. Başar, "Optimal Rate Control for High Speed Telecommunication Networks", *Proc. 34th IEEE Conference on Decision and Control*, pp. 1389–1394, New Orleans, Louisiana, Dec. 1995.

[4] E. Altman, T. Başar, R. Srikant, "Flow Control in Communication Networks with Multiple Users and Action Delays: A Team-Theoretic Approach," *Proceedings of the 36th IEEE Conference on Decision and Control,* San Diego, CA, December 1997.

[5] B.D.O. Anderson and J.B. Moore, *Optimal Control: Linear Quadratic Methods*, Prentice-Hall, N.J., 1990.

[6] J-C. Bolot, "A self-tuning regulator for overload control in communication networks," *Proc. 31st Conf. on Decision and Control*, Tucson, AZ, pp. 1022-1023, Dec. 1992.

[7] J-C. Bolot and A. U. Shankar, "Analysis of a fluid approximation to flow control dynamics," *IEEE INFOCOM '92*, Florence, Italy, May 1992, pp. 2398-2407.

[8] R. Jain, "Congestion control and traffic management in ATM networks: recent advances and a survey, *Computer Networks and ISDN Networks*, November 1996.

[9] S. Kalyanaraman, R. Jain, S. Fahmy, R. Goyal, and B. Vandalore, "The ERICA Switch Algorithm for ABR Traffic Management in ATM Networks," *Preprint*, http://www.cis.ohio-state.edu/ jain/papers/erica.htm.

[10] R. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge Univ. Press, 1985.

[11] V. Jacobson, "Congestion avoidance and control", *ACM SIGCOMM '88*, Stanford, CA, 314-329, August 1988.

[12] S. Keshav, "A control-theoretic approach to flow control", *ACM SIGCOMM '91*, Zurich, Switzerland, September 1991.

[13] A. P. Sage, C. C. White, "Optimum Systems Control," *Prentice Hall*, Englewood Cliffs, NJ.

[14] R. Srikant and T. Başar,"Asymptotic solutions to weakly coupled stochastic teams with nonclassical information," *IEEE Transactions on Automatic Control,* February, 1992, 163–173.