A CONTROL ENGINEER'S LOOK AT ATM CONGESTION AVOIDANCE

Charles E. Rohrs, Randall A. Berry, Stephen J. O'Halek

C.E. Rohrs is a Fellow of Tellabs Research Center and a Visiting Scientist at MIT.

Tellabs Research Center, 3740 Edison Lakes Pkwy, Mishawaka, IN 46545 (charlie@trc.tellabs.com)

R.A. Berry and S.J. O'Halek are graduate students at MIT.

Abstract -- Algorithms for controlling congestion are critical for the success of ATM networks. The various sample algorithms appearing in the literature have in common significant non-linearities that make these algorithms difficult to analyze using the tools of classical control theory. This paper reports on the beginnings of a research program that considers the ATM congestion control problem from the point of view of control theorist. A control scheme is developed that can be designed and analyzed using well established linear control theory. There is promise that as this approach is further developed it offers hope that analysis can assure reasonable behavior in the large scale system setting of ATM networks.

I. INTRODUCTION

A critical success factor for ATM networks is the resolution of a scheme to adequately control the flow of data into an ATM network so as to avoid congestion at the network switching points [1]. The ATM Forum is earnestly addressing this question and has defined protocols that enable the switches and sources to communicate congestion information [2]. It has also produced a series of sample congestion avoidance algorithms. These algorithms are not part of the standards being set but are included to demonstrate the viability of the protocols. The algorithms suggested to date fall into two categories – binary algorithms, *e.g.* the PRCA (proportional rate control algorithm) [3], which use a single bit to communicate congestion information from the switches to the sources, and multivalued algorithms such as those presented in [4],[5] and [6], which exchange much more information in both directions between the sources and the switches. While more recent work in the ATM Forum has concentrated on multi-valued schemes, this paper addresses only the more restricted binary schemes. Our attention is restricted to these schemes because we are looking to more fundamentally understand the behavior of these schemes and the role classical control theory can play in suggesting alternative schemes that can be more productively analyzed.

The binary schemes that have been presented to date as sample switch and source algorithms possess the characteristic that they contain control loops with significant non-linearities within their operating regions. Such schemes are difficult to analyze even in a single loop setting. In a network setting with many interacting loops, fully understanding the range of possible behaviors quickly becomes intractable. The mathematics of dynamic system theory suggests that the interaction of many non-linear feedback loops can produce unexpected and erratic behavior. While no such behavior has been demonstrated for the carefully crafted binary algorithms, we feel that it is a useful endeavor to examine if simple linear feedback loops can be designed in the context of the ATM congestion avoidance problem. If adequate feedback loops can be obtained, the expansion of the analysis to large interconnected systems becomes much simpler and more fruitful.

In this paper, we address the creation of source and switch algorithms that can be analyzed and designed using standard linear control system techniques that can be found in any standard undergraduate text in the subject, for example, [7]. We believe that this theory of control which has developed over many years has much to teach us about congestion avoidance algorithms and that this theory has been largely ignored. We believe manipulating the problem to make use of this theory can produce a number of benefits. Among the potential benefits are:

(1) simpler and/or more effective algorithms,

(2) better understanding of the current non-linear algorithms,

(3) better analysis techniques for large systems of interacting algorithms.

In this paper, we consider only the simple case where two sources are trying to send cells to the same output port of the same switch. Both sources are persistent in that they would like to send cells as fast as possible but are limited by the bandwidth of the output port of the switch. Each source should settle to a rate that is half the bandwidth of the output port.

The results of this paper provide only the first steps in using control theory to analyze and design alternative congestion avoidance algorithms. We are not yet at the point where we advocate a particular alternative algorithm. We present our thinking and our results as a contribution to the general understanding of the behavior of rate control algorithms and an opening to a different way of considering such algorithms. We hope that the lessons learned in addressing these issues from the viewpoint of a control engineer will enrich the work of those working on the problem of congestion control, most of whom have different backgrounds from ours.

II. A NON-LINEAR ALGORITHM

In this section we present a binary algorithm that is representative of those being considered in the ATM Forum. As algorithms are developed in the Forum small changes are made but, for our purposes, the algorithm given here summarizes the key ingredients of this class of algorithms. Again, note that we are only considering the binary algorithms with single bit feedback from the switches to the sources and not the multivalued algorithms with more sophisticated communication capabilities. The purpose of presenting this algorithm is to point out two characteristics that make the algorithm very difficult to analyze and that create behavior that is cause of concern. The algorithms are usually presented in pseudo-code as in [4]. Here, we present the algorithm somewhat less precisely and we fix the values of some parameters in order to simplify the exposition.

At each time each source i has a rate R_i at which it is sending cells. We assume that each source always has cells to send and would like to send them as fast as possible. The switch sees cells arriving for a particular output line at a rate R where R is the sum of the rates of cells being sent from all sources to the particular output line. The sources share a single switch output line of bandwidth B=3.54*10^5 cells/sec (150 Mbps). Every 32 cells the sources insert an RM (Resource Management) cell which is returned by the destination. The switch controls one bit in the RM cell to indicate congestion, C=1, or no congestion, C=0. We consider the case where the switch marks the RM cell on its way back to the source (Backward Explicit Congestion Notification or BECN). The feedback may be essentially instantaneous or may be subject to delay.

Switch Algorithm. The switch queues the cells until they can be delivered. The length, q, of the queue at successive time instants separated by Δ seconds is modeled by

$$q(n\Delta) = max\{q((n-1)\Delta) + (R((n-1)\Delta) - B)\Delta + n_q, 0\} \tag{1}$$

Equation (1) models the flow of cells through the queue as a continuously varying quantity. In reality, cells arrive and leave the actual queue at discrete times and the actual queue always contains an integer number of cells. The difference between the model of the queue in (1) and the real queue is accounted for by the noise source n_q.

This noise source will be characterized by considering the actual queue size in the two source/one switch network when the input rates are both set at 1/2 the output bandwidth, B. In this case the model in (1), without n_q, specifies that the queue size should stay constant. Figure 1 shows the queue size of the actual queue for this situation. In this figure X_i represents the time cells arrive from source i. The x-axis is normalized by $2/B$ so that cells depart from the queue at 1/2, 1, . . . This figure shows that the actual queue size will vary depending on when X_1 and X_2 occur and when the queue is sampled. The noise, n_q, is modeled as zero mean white noise whose variance is equal to the variance of the queue in this situation.

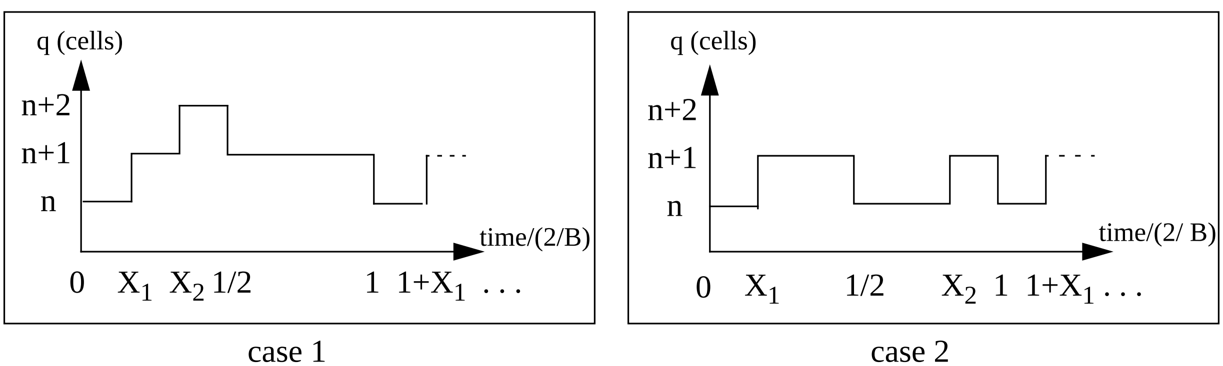

Fig. 1 The actual queue size, q, when the two input rates are both B/2 for two cases.

To calculate this variance it is assumed that X_1 and X_2 are independent of each other and equally likely to occur any where between 0 and 1, *i.e.* independent uniform random variables. The variance of the queue is calculated for each of four equally likely cases: (1) $X_1 < 1/2$, $X_2 < 1/2$, (2) $X_1 < 1/2$, $X_2 > 1/2$, (3) $X_1 > 1/2$, $X_2 < 1/2$, and (4) $X_1 > 1/2$, $X_2 > 1/2$. Cases 1 and 2 are shown in figure 1. Cases 3 and 4 can be thought of as cases 2 and 1 respectively with the origin shifted to the left by 1/2. Due to this symmetry, the variance for cases 1 and 4 and cases 2 and 3 will be the same, so really only two cases need to be considered.

CASE 1: Conditional on being in case 1, then X_1 and X_2 are independent uniform random variables on (0,1/2). We want to find the pmf of the queue in this case. This pmf can be written as

$$p(Q=q) = p(Q=q|X_1<X_2)p(X_1<X_2) + p(Q=q|X_2<X_1)p(X_2<X_1) \quad (2)$$

Since the X_i are independent and indentically distributed, then we know that

$$p(X_1<X_2) = p(X_2<X_1) = 1/2 \quad (3)$$

and

$$p(Q=q|\ X_1<X_2) = p(Q=q|X_2<X_1) \quad (4)$$

thus (2) reduces to

$$p(Q=q) = p(Q=q|X_1<X_2) \quad (5)$$

From fig. 1, the pmf for Q given X_1 and X_2 is given by:

$$p(Q=q|X_1=x_1, X_2=x_2, X_1<X_2) = \begin{cases} x_1, & q = n \\ x_2 - x_1 + 1/2, & q = n+1 \\ 1/2 - x_2, & q = n+2 \end{cases} \quad (6)$$

To get from (5) to (6), the pdf's $f(x_1|X_2=x_2, X_1<X_2)$ and $f(x_2|X_1<X_2)$ are needed. The first of these is given by

$$f(x_1|\ X_2 = x_2\ , X_1<X_2) = 1/x_2\ ,\ \ 0<x_1< x_2 \quad (7)$$

Bayes rule is used to find $f(X_2|X_1<X_2)$ as follows

$$f(x_2|X_1 < X_2) = \frac{p(X_1 < X_2|X_2 = x_2)f(x_2)}{p(X_1 < X_2)} \quad (8)$$

$$= \frac{(2x_2)(2)}{(1/2)}\quad \text{, if } 0<x_2<1/2 \quad (9)$$

multiplying (6) by (7) and (8) and then integrating over X_1 and X_2 yields the desired pmf

$$p(Q=q|\ X_1<X_2) = (1/6)\delta(q-n) + (2/3)\delta(q-n-1) + (1/6)\delta(q-n-2) \quad (10)$$

From this, the variance of Q for case 1 can be found to be 1/3, this is also be the variance of n_q for case 1.

CASE 2: This is the simpler of the two cases. For this case X_1 is uniform on (0, 1/2) and X_2 is uniform on (1/2,1) and again both are independent. The ordering of X_1 and X_2 is already fixed in this case. The pmf of Q, given X_1 and X_2, can be written, by looking at Fig. 1, as:

$$p(Q=q|X_1=x_1, X_2=x_2) = \begin{cases} x_1 + x_2 - 1/2, & q = n \\ 3/2 - x_1 - x_2, & q = n+1 \end{cases} \tag{11}$$

Multiplying this by the pdf's for X_1 and X_2 and integrating over these variables results in the unconditional pmf for Q:

$$p(Q=q) = (1/2)\delta(q-n) + (1/2)\delta(q-n-1) \tag{12}$$

Thus the variance of Q and n_q for this case is 1/4.

Since n_q is zero mean, the overall variance for n_q can be found by adding up conditional variances for each case weighted by the probability of that case. This results in a total variance of 0.292 for n_q.

The switch sets the congestion indicator of RM cells returning to the source if its queue length surpasses some threshold (in this case, q=500).

$$C = threshold(q - 500) \tag{13}$$

where the *threshold* function equals 1 if the argument is greater than or equal to 0 and the function equals 0 otherwise. This, of course, is a significant non-linearity in a control system.

Source Algorithm. The source changes its rate when either one of two events occurs:

1. It is time to send a new RM cell. At this time the source decreases its rate by a multiplicative scale factor
2. The source receives a returned RM cell with C=0. If this occurs the source raises its rate by more than enough to overcome the decrease incurred during the last condition 1 situation.

There is an aspect of this algorithm that makes it unamenable to analysis. The rate at which cells are being sent by a source is constantly changing; indeed, it is the variable being controlled. Since the times at which changes are being made in the rate depends on the rate itself, the time between successive sample instants is constantly changing making a discrete-time analysis very difficult. Also, the feedback mechanism is corrupted by the timing out aspect of condition 1 in the source algorithm. The RM cells are not being sent out and returned at a constant rate since the returning RM cells were generated some time ago when the source's rate may have been lower or higher. If the RM cells are going out faster than they are coming back there are times when rate decreases due to condition 1 are not offset even though there is no congestion in the switch.

A simulation of the switch and source algorithms described above is presented in Fig. 2. Two persistent sources are contending for a single switch's output port which has a bandwidth of $3.54*10^5$ cells/sec. The cell rates of the two sources are shown in Fig. 2a. Rather than converging to steady-state rates of $1.77*10^5$ cells/sec each, both source rates oscillate in unison between $0.5*10^5$ and $3.0*10^5$. The switch's queue length also oscillates between 300 and 600 cells as shown in Fig. 2b.

Also shown in Fig. 2b is the timing of the *C*=0 and *C*=1 RM cells as received back at a source. The timing out problem is seen most clearly at the top of the source's rate cycle. The source rate levels off while it is receiving only *C*=0 (no congestion) indicators. At this point the rate is high and new outgoing RM cells are being generated much faster than RM cells (which were generated at lower rate previously) are being returned. The result is many false slowdowns due to timing out. Notice that this effect is in fact stabilizing but that is makes accurate analysis very difficult.

.

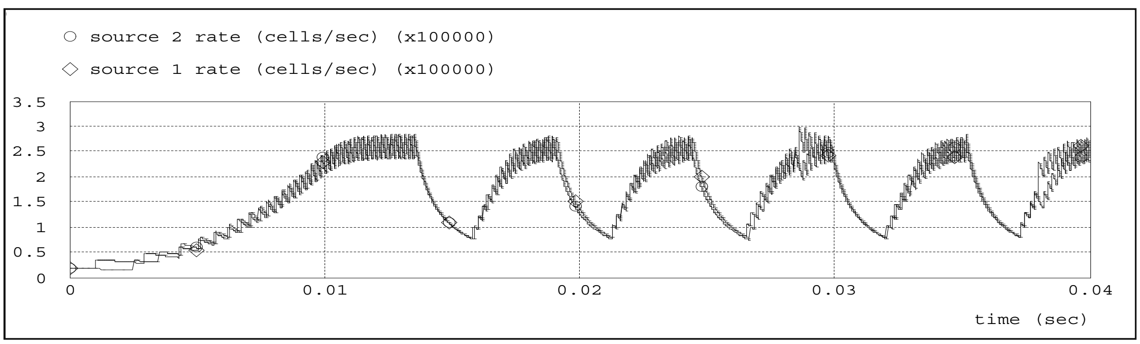

Fig. 2a The Rates of the sources.

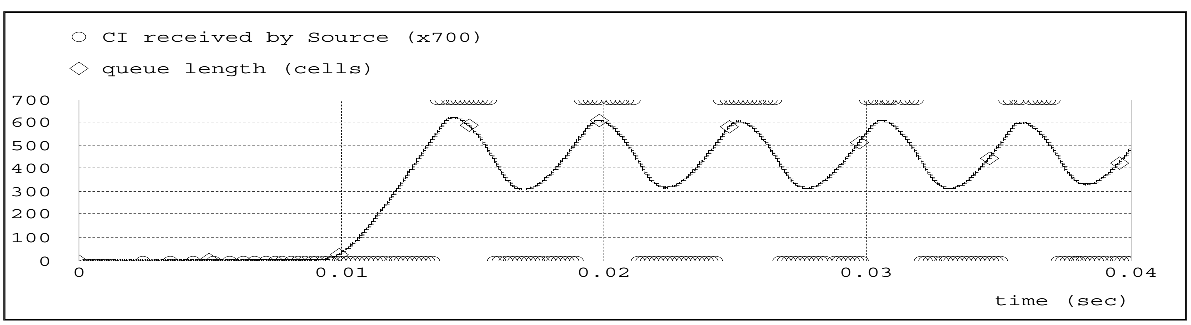

Fig. 2b The size of the queue and the *C* indicators on returning RM cells.

Fig.2 The simulation results for the non-linear algorithm.

Some analysis of this representative algorithm can be performed if this timing out problem is assumed away by only letting the source adjust at fixed time intervals. An adjustment mechanism using fixed intervals but retaining the spirit of the ATM Forum PRCA algorithm is

$$R_i(n\Delta) = (1-\alpha)^{C_1} R_i((n-1)\Delta) + C_0\beta \quad (14)$$

where C_0 is the number of C=0 RM cells received in time interval Δ,

C_1 is the number of C=1 RM cells received in time interval Δ,

α is the multiplicative decrease factor, and β is the additive increase factor.

A system very similar to this is analyzed in [8]. Even with the timing out problem removed, the system contains two non-linearities which cause it to exhibit steady-state oscillations similar to those demonstrated in Fig. 2. A typical cycle in the oscillation is described in [8] where it is made apparent that the cycles are fundamentally caused by the threshold non-linearity in (13).

In summary, the non-linearities in the algorithm of (1),(13)-(14) create a control system that results in large limit cycles. The binary control variable C does not anticipate changes in the queue. By the time C is used to indicate congestion the rates are large and growing. The congestion indication causes the rates to begin to decrease but the queue continues to grow until the sum of the rates falls below the outgoing bandwidth. The cycle is reversed on the downside. The two sources are synchronized in their oscillation since the nonlinear mapping of the single queue is driving them simultaneously in the same direction. The situation becomes worse when delay is added to the feedback.

III. USING PROBABILISTIC FEEDBACK TO CREATE A LINEAR CONTROL SYSTEM

The most serious problem in the system discussed above is the restriction that only a single bit can be fed back from the switch to the sources. If a many valued control variable could be used in the feedback loop then linear control could be applied and standard control design techniques could be used to create a system that behaves better. While we agree to live with the binary feedback restriction, we believe that it can be used more wisely. The system actually has many chances to send a congestion bit back to the source. It can create the effect of a many valued feedback variable by using probabilistic feedback.

Consider the situation where the switch would like to communicate the parameter p, a number between 0 and 1 to the source. It can use a pseudo-random number generator to set C=1 with probability p and C=0 with probability 1-p. The source can then recreate a noisy estimate of p by averaging over a number of successive C's. This is the scheme we suggest to turn the rate control system into an (almost) linear control system.

Source algorithm. In the interval between time (n-1)Δ and nΔ accumulate as C_1 the number of RM's received with C=1 and as C_0 the number of RM's received with C=0. Let

$$p_i((n-1)\Delta) = \frac{C_1}{C_0 + C_1} \quad (15)$$

At $n\Delta$, compute $R_i(n\Delta)$ as

$$\begin{aligned} R_i(n\Delta) &= \gamma R_i((n-1)\Delta) + (1 - p_i((n-1)\Delta))\beta - p_i((n-1)\Delta)\alpha \\ &= \gamma R_i((n-1)\Delta) - (\alpha + \beta) p_i((n-1)\Delta) + \beta \end{aligned} \quad (16)$$

Notice that this equation for the rate adjustment has an effect very similar to that of Eq. (14) but it has been modified to be linear in the control parameter p_i. There is still an exponential decrease in rate which can be overcome by an additive increase when there is little congestion, *i.e.*, when p_i is near zero.

Switch Algorithm. The queue length is still given by (1). The switch computes the feedback parameter p as the output of a linear time invariant operator working on q. The loop contains a summer and another low frequency pole at γ. The fundamentals of control theory indicate that a simple controller that feeds back a term proportional to the queue size and another term indicating the change in the queue size is appropriate for a good, stable system response. Such a controller is standard in control theory textbooks [7]. It is referred to as a proportional-derivative (PD) or a lead network controller. This controller can be expressed as

$$\begin{aligned} p(n\Delta) &= a(q(n\Delta) - q((n-1)\Delta)) + bq(n\Delta) \\ &= (a+b)q(n\Delta) - aq((n-1)\Delta) \end{aligned} \quad (17)$$

The parameter p is communicated to the source by marking C=1 to all sources with probability p if $0<p<1$. The variable p is allowed to saturate at 0 and 1. If the parameters a and b are chosen properly, p will not saturate and the effect will be a linear control system where each source receives a noisy estimate of the variable p. In particular, the parameter b relates the steady-state queue size to p and is important in allowing operation to remain in the linear region.

Notice that if the system operates in the saturation region, the behavior is much the same as the original system. We have, in effect replaced the hard switching non-linearity of (13) and Fig. 3a with the softer non-linearity of Fig. 3b. The softer non-linearity allows a region of linear control which can produce better performance. The remaining non-linearity provides a safety net against any instability in the linear region as any locally unstable trajectories are trapped in a stable limit cycle.

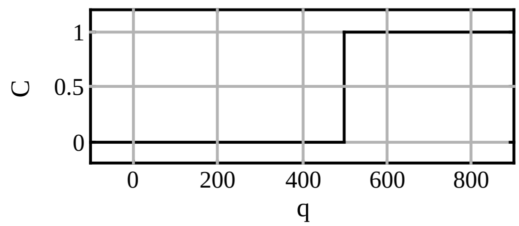

Fig.3a The hard non-linearity of the threshold function

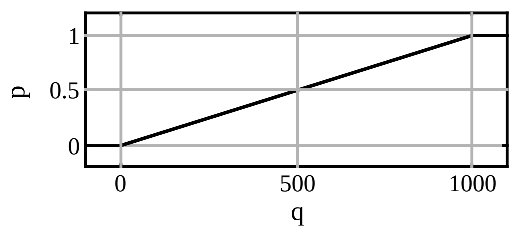

Fig.3b The softer non-linearity of probabilistic feedback

Fig. 3 Comparison of the non-linearities involved in the congestion control loops.

IV. ANALYSIS AND SIMULATION

The one switch, two sources cell transport system was analyzed and simulated with the switch implementing (17) and probabilistic tagging and the source implementing (15) and (16). As long as the system operates in the linear region a complete analysis is possible. Simulations confirm the expected results.

For analysis and design of the linearized control scheme (1) is added to model the queue. The difference between the variable p at the switch and the noisy estimates, p_i, of this variable computed at the sources is represented by additive noise, n_i.

$$p_i(n\Delta) = p(n\Delta) + n_i(n\Delta) \tag{18}$$

In a representative time interval Δ, a source receives N congestion indicators from the switch. These congestion indicators can be modeled as N independent indentically distributed binomial random variables taking on the value 1 with probability *p* and 0 with probability 1-*p*. The source algorithm (15) estimates *p* by computing the sample mean of these N random variable. The noise, n_i, represents the error made in this estimation. The errors in each interval are assumed independent so n_i can be represented as white noise with a variance given by:

$$\frac{1}{N}p(1-p) \tag{19}$$

The total rate that cells are entering the queue is represented as the sum of the rates that each source is sending at, *i.e.*

$$R(n\Delta) = R_1(n\Delta) + R_2(n\Delta) \tag{20}$$

Cells may be delayed arriving at the switch and RM cells may be delayed returning to the source. The approach taken here is not to specifically model these delays but to use control theory to analyze the largest delay for which the simple linear controller remains stable.

Equations (1), (16)-(18), and (20) provide a model with which the control system can be designed and analyzed. A block diagram of the model appears in Fig. 4.

Now that the model is specified, linear control theory can be applied. Linear control theory provides guidance in choosing the parameters so that design trade-offs are clear through analysis rather than the trial and error of simulation. The theory offers techniques to use the parameters to manipulate derived quantities such as bandwidth and phase margin to aid in the understanding of the following fundamental performance trade-offs:

1. A stable linearized controller can be designed.
2. The maximal round trip delay for which the linearized controller remains stable can be determined using the control loop bandwidth and phase margin. The lower the bandwidth and the larger the phase margin, the more delay can be tolerated.
3. The speed and the shape of the transient response can be determined. The larger the phase margin, the nicer the shape of the transient. The higher the bandwidth, the faster the

transient response.

4. Steady-state values of variables can be determined in terms of the parameters.
5. The effects of the noise sources on the variables can be analyzed. The smaller the bandwidth, the less effect the noise sources have on the variables.

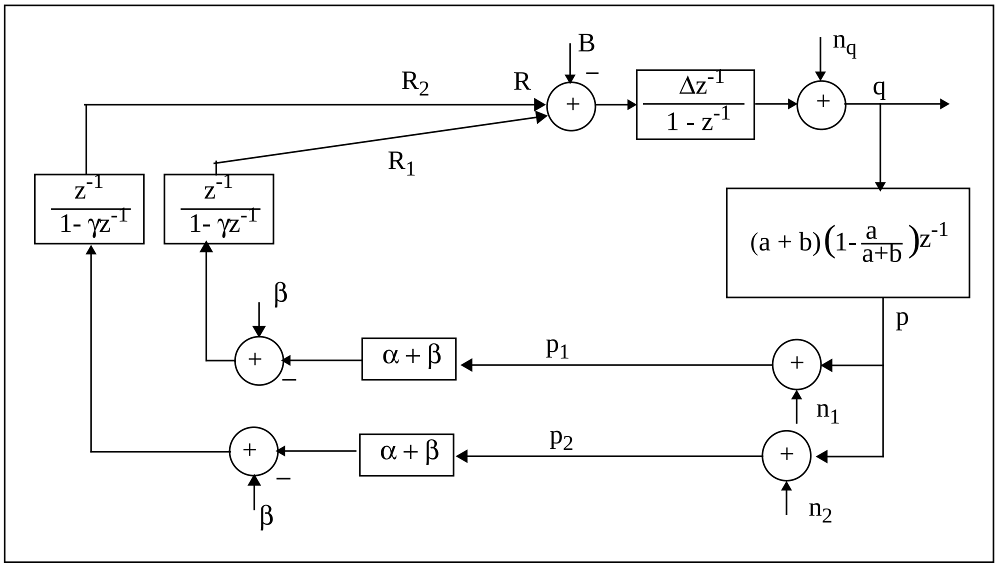

Fig. 4 Block diagram of the linear control system

We begin by examining the steady-state values. We analyze the case where the parameters are chosen so that the closed-loop system is stable and operates in the linear region, *i.e.*, that p remains between zero and one. Then, except for the noise sources, n_i and n_q driving the system, the system reaches a constant steady- state. With n_i=0 and n_q = 0, the steady-state values are:

From (1), (20) $$R = B \quad \text{and} \quad R_i = 1/2\,B \tag{21}$$

From (16) $$p = \frac{\beta - (1-\gamma)(B/2)}{\alpha + \beta} \tag{22}$$

From (17) $$q = p/b \tag{23}$$

From (22), the parameter $\beta/(\alpha+\beta)$ must be chosen larger than $(1-\gamma)B/(\alpha+\beta)$ to allow a positive steady-state p. From (16), smaller γ removes the effects of the initial rates more quickly and has the effect of forcing the two sources toward the same rate more effectively. We chose α=0 and γ=0.99 initially. To keep the queue size small (less than 100), (23) was used to guide the choice of b as b=0.01. Based on the expected steady-state rates of the sources, we chose our sampling time to be $\Delta=9*10^{-4}$ secs. This allows each source to see on the average 5 RM cells per sampling period. Increasing Δ reduces the size of the noise from n_1 and n_2. Increasing Δ also changes the frequency scale of the system and has effects similar to those produced by lowering the bandwidth.

Control theory tells us that the bandwidth and phase margin are determined by a transfer function called the control loop gain. A larger loop gain means a larger bandwidth. The control loop gain is given by

$$G(s) = \frac{2(\alpha+\beta)\Delta(a+b)z^{-2}\left(1-\frac{a}{a+b}z^{-1}\right)}{(1-\gamma z^{-1})(1-z^{-1})} \tag{24}$$

The frequency response of a typical loop gain is shown in Fig. 5. Such a pair of plots with the log magnitude and phase plotted separately versus the log frequency are called the Bode plots of the loop gain. For simple system such as those of interest here, the phase margin and bandwidth can be read directly from the Bode plots. The bandwidth is by convention considered to be the frequency range from zero to the frequency where the magnitude of the loop gain passes through unity (0 dB). This frequency is also called the crossover frequency. In Fig. 5 the crossover frequency is 500 rad/sec. The phase margin is the difference between the phase at crossover frequency and 180 degrees. In Fig. 5 the phase margin is 45 degrees.

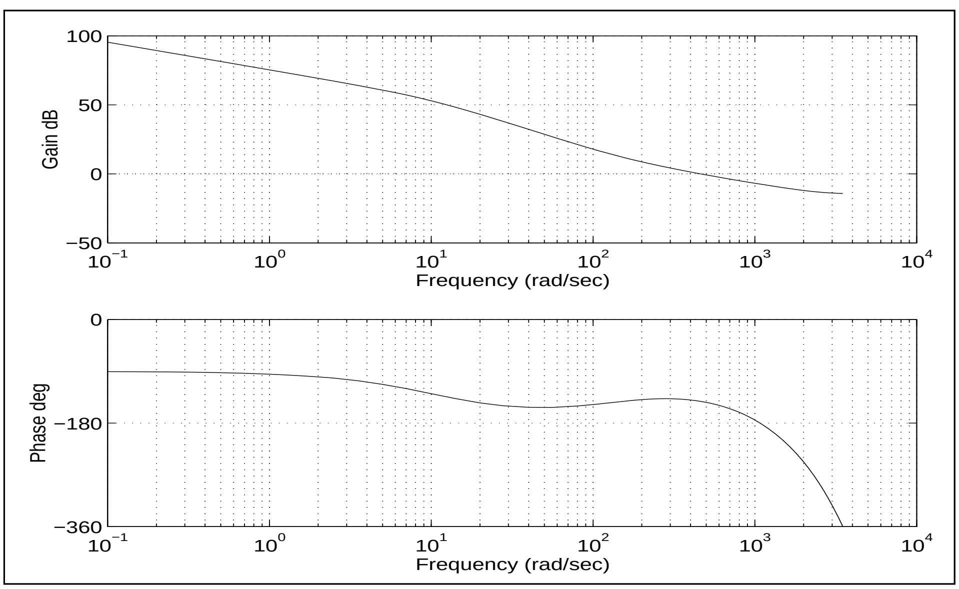

Fig. 5 Bode plots of loop gain for γ=0.99, a=0.0685, b=0.01, α=0, and $\beta=2.95*10^3$

If the loop gain is increased while the phase remains unchanged (by, for example, increasing β), the magnitude plot is shifted up, the bandwidth is increased and the phase margin is decreased. The opposite effect occurs if the loop gain is decreased. The phase margin is also influenced (in a more complex relationship) by the parameter *a*.

The parameters *a* and β are used to set the phase margin, the bandwidth and the steady-state probability *p*. Choosing a=0.0685 and $\beta=2.95*10^3$ creates the Bode plot of Fig. 5 with an adequate phase margin of 45 degrees, a bandwidth of 500 rad/sec and a steady-state p=0.4. The steady-state *q* is 40.

The amount of delay that can be handled in the linear region of this control system can be predicted using control theory. A pure delay adds a negative phase shift (a phase lag) to the Bode plots of the loop gain while leaving the magnitude plot unaffected. The phase lag associated with a delay of *T* seconds increases proportional to frequency, i.e., the phase lag is ωT rad. The linearized system remains stable until the phase lag contributed by the delay at the crossover frequency exceeds the phase margin of the control system. In our example the phase margin is 45 degrees or $\pi/4$ rads at a crossover frequency of 500 rad/sec. The maximum allowable round trip delay in the loop is T=π/(4*500)=1.5 msec. If the delay were larger than this the linearized system would be unstable; the system would be controlled by the saturation nonlinearity and it would behave much as the original non-linear system. Larger delays could be handled by either lowering the loop gain or increasing Δ.

Linear system theory provides the means to predict the effect the noise sources will have on the steady state values. Given the transfer function, $H(e^{j\omega})$, between a noise source and a variable of interest, then the variance in the variable is related to the variance of the white noise source by

$$var(variable) = \left(\int_{-\pi}^{\pi} \left| H(e^{j\omega}) \right|^2 d\omega \right) var(noise) \qquad (25)$$

We will call this scale factor the covariance response of the transfer function. There are three noise sources present in the system, one probabilistic noise, n_i, for each data source and the queue noise, n_q. These noise sources are independent and so the total variance seen in a variable is the sum of the variance in the variable due to each noise source.

The transfer function from each noise source, n_i, to the queue length, *q*, is given by

$$\frac{q(z)}{n_i(z)} = \frac{-(\alpha+\beta)\Delta z}{z^3-(1+\gamma)z^2+(\gamma+2(\alpha+\beta)(a+b)\Delta)z-2(\alpha+\beta)\Delta a} \tag{26}$$

For the model parameters above, the covariance response from each probability noise, n_i, to the queue length, q, is then 234. Using N=5 and the steady state value of p = 0.4 in (19) estimates the variance for each n_i as 0.048. Multiplying this by the covariance response yields a variance in the queue of 11.2 cells^2 due to each n_i. Since the n_i's associated with each source are independent, the total variance in the queue due to the probabilistic noise is 22.4 cells^2. The transfer function from the queue noise, n_q, is

$$\frac{q(z)}{n_q(z)} = \frac{z^3-(1+\gamma)z^2+\gamma z}{z^3-(1+\gamma)z^2+(\gamma+2(\alpha+\beta)(a+b)\Delta)z-2(\alpha+\beta)\Delta a} \tag{27}$$

For the chosen parameters the covariance response from n_q to q is 1.6. The variance of n_q is 0.292, so the queue will have a variance of 0.47 cells^2 due to the queue noise. Thus the total variance in the queue will be 22.9 cells^2. The variance in the rates or any other variable can be found in the same manner.

As long as the variables remain in the linearized region, the entire response can be determined analytically. Simulations of the actual congestion control system were run to verify performance. The results of a simulation are in given in Fig. 6. The rates of the two sources are displayed in Fig 6a, where it is seen that the rates converge to near their steady-state values. The queue is well behaved as in steady-state it moves slightly around its steady-state value. The computed sample variance in the steady state queue during the simulation run is 22.2 cells^2, corresponding to a standard deviation of 4.7 cells, very close to that predicted above. The presence of the noise which enters the system in communicating p to the sources prevents a completely quiescent steady-state. Notice, however, as a general benchmark that the queue is kept to an order of magnitude smaller than the non-linear simulation of Fig. 2 and that the small remaining deviations from a constant steady-state are due to noise and not to systematic oscillations.

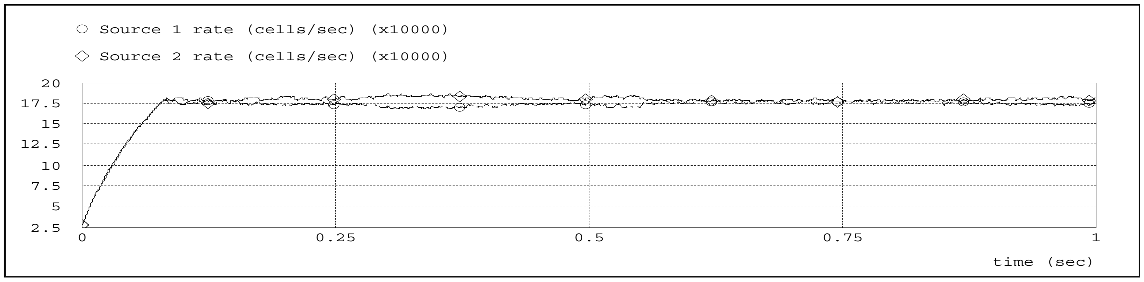

Fig 6a. The rates of the sources

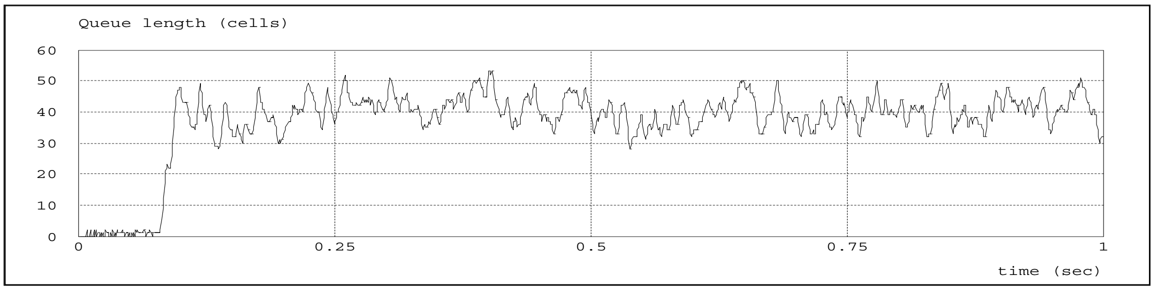

Fig. 6b. The size of the queue.

Fig. 6 Simulation of the probabilistic system with γ=0.99, a=0.0685, b=0.01, α=0, and $\beta=2.95*10^3$.

To demonstrate the trade-offs that are achievable using linear control theory the system was redesigned with the goal of reducing the effect of the noise from the communication of p. To achieve this goal we must reduce the bandwidth and accept the accompanying slowing of the transient response. The important point is that because the system can be analyzed we know the precise nature of the trade-offs involved. To decrease the bandwidth, the loop gain of (24) is lowered by decreasing β. The parameter a must be increased to protect the phase margin. Finally, in order to keep the steady state p in (22) at p=0.4, the value of γ must be increased making the rates of the two sources converge more sluggishly. We set γ=0.9998, a=0.4, b=0.01, α=0, and β=59.

The bandwidth of the resulting system is now 50 rad/sec with a 60 degree phase margin resulting in robustness to delays in the loop up to 19 msecs. The covariance response from n_q to q, (26), is now 1.04 and the covariance response from each n_i to q, (27), is now 31.9. The noise variances are still the same, so the resulting variance of the queue is now predicted to be 3.36 cells^2, corresponding to a standard deviation of 1.8 cells. The simulation was run starting the rates near their steady-state values to avoid excessive simulation through the longer transient. The simulation results appear in Fig. 7. The most noticeable difference is the much reduced effect of the noise on the rates and the queue as predicted from the smaller closed-loop bandwidth.

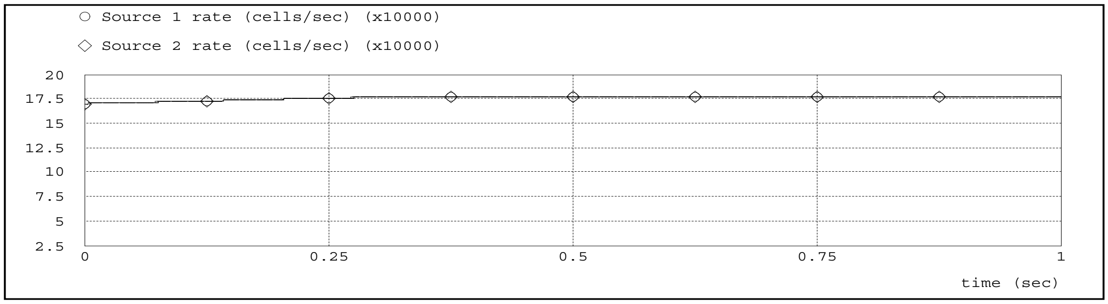

Fig 7a. The rates of the sources.

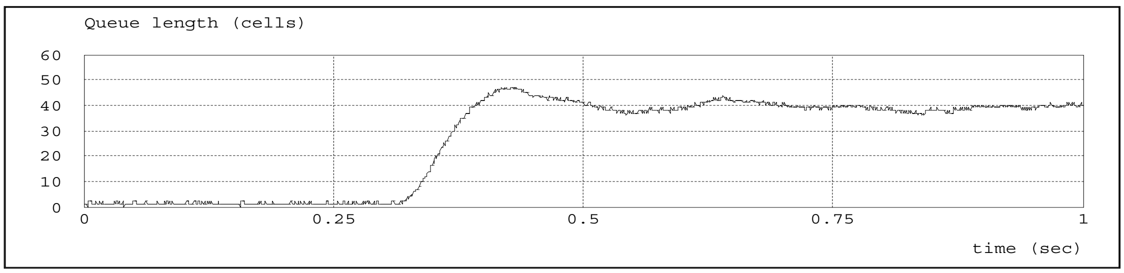

Fig 7b. The size of the queue.

Fig. 7 Simulation of the probabilistic system with γ=0.9998, a=0.4, b=0.01, α=0, and β=59.

There are a number of issues connected with this algorithm that remain unaddressed at this point. In order to properly set the gain of the controller, the number of active circuits using the same switch output must be known. This can be done either by the switch keeping a tally of the number of active users for each of its output ports or it can be addressed more directly using the advanced techniques of adaptive control. We leave this for further study. Also, the binary algorithm presented here suffers from the so called beat down problem [5] that was discovered in the ATM Forum's binary algorithm. The response to this problem in the ATM Forum was to create the class of multivalued algorithms. We believe that this is a useful approach and we are currently studying applying linear control techniques to multivalued algorithms. The availability of a multivalued feedback mechanism removes the noise sources, p_i, associated with the communication of the feedback parameter. We believe that linear control techniques are very useful tools for the design of these algorithms.

V. CONCLUSIONS

Probabilistic feedback has been used to create a congestion control algorithm which can be designed and analyzed using linear control theory. The viability of such an approach has been demonstrated in simulations that match theoretical predictions. The linear control technique can be used to eliminate the large steady-state oscillations from the results of the non-linear algorithms. The ability to understand the effects of parameters in this new system has also been demonstrated so that design trade-offs can be understood and enacted. The linear control techniques employed here apply *a fortiori* to the case where multivalued feedback is provided for directly.

Only a single simple case has been investigated to date. Much work is yet to be done before the approach given here is demonstrated as a viable alternative design. However, we feel the promise is clear. If the design can be achieved using analyzable linear systems, then a great deal more can be understood about the behavior of large interconnections of such systems.

REFERENCES

[1] Jain, R, "Congestion Control and Traffic Management in ATM Networks: Recent Advances and A Survey", to be published in *Computer Networks and ISDN Systems*.

[2] Closed-Loop Rate-Based Traffic Management," ATM Forum/95-0013R3, editing version.

[3] Closed-Loop Rate-Based Traffic Management," ATM Forum/94-0438R1, July 1994.

[4] Closed-Loop Rate-Based Traffic Management," ATM Forum/94-0438R2, September 1994.

[5] Siu, K. and H. Tzeng, "Intelligent Congestion Control for ABR Service in ATM Networks," *Computer Communication Review*, Vol. 24, No. 5 pp. 81-106, October. 1995.

[6] Barnhart, A.W., "Example Switch Algorithm for Section
5.4 of TM 5.4 of TM Spec.," ATM Forum/95-0195, February 10, 1995.

[7] Rohrs, C.E., J.L. Melsa, and D.G. Schultz, *Linear Control Systems*, McGraw-Hill. 1993.

[8] Yin, N. and M.G. Hluchyj, “On Closed-Loop Rate Control for ATM Cell Relay Networks,” *Proceedings, IEEE Infocom '94*, pp. 99-108. June 1994.