The Effect of Uncertain Time-Variant Delays in ATM Networks with Explicit Rate Feedback: A Control Theoretic Approach

Mihail L. Sichitiu
Department of Electrical and Computer Engineering
North Carolina State University
Raleigh, NC 27695
mlsichit@eos.ncsu.edu

Peter H. Bauer*
Department of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556
pbauer@mars.ee.nd.edu

Kamal Premaratne*
Department of Electrical and Computer Engineering
University of Miami
Coral Gables, FL 33124
kamal@miami.edu

Abstract

A new, more realistic model for the Available Bit-Rate traffic class in ATM network congestion control with explicit rate feedback is introduced and analyzed. This model is based on recent results by Ekanayake, regarding discrete time models for time-variant delays. The discrete time model takes into account the effect of time-variant buffer occupancy levels of ATM switches, thus treating the case of time-variant delays between a single congested node and the connected sources. For highly dynamic situations, such a model is crucial for a valid analysis of the resulting feedback system. The new model also handles the effects of the mismatch between the resource management cell rates and the variable bit rate controller sampling rate as well as buffer and rate nonlinearities. A brief stability study shows that an equilibrium in the buffer occupancy is impossible to achieve in the presence of time-variant forward path delays. Stability conditions for the case of time-variant delays in the return path are presented. Finally, illustrative examples are provided.

1 Introduction

ATM networks can support a wide variety of traffic, diverse services, bandwidth requirements, and tolerance to message delay and loss [1]. One class of traffic is Available Bit Rate (ABR) which is a best effort class. A congestion control scheme is required to efficiently allocate the unused bandwidth of the link to the ABR traffic in order to improve network utilization. Two different congestion control mechanisms are provided in the ATM standard [1]. The first mechanism allows a switch to communicate its congestion status to the sources by using a single bit in the Resource Management (RM) cells. The second mechanism allows the switches to explicitly designate the cell transmission rate by modifying the ER (explicit rate) value of the RM cell. The advantage of the single bit approach is the implementation simplicity, although it has been shown to exhibit oscillatory behavior [2]. While the implementation of an explicit-rate switch is rather complex, it has the potential to achieve significantly improved performance compared to the binary mode switch, however the effectiveness of an explicit-rate switch is highly dependent on the determination method of the ER value.

Previous work [3–10] on explicit rate feedback of the ABR class of traffic in ATM networks deals with the analysis of the real feedback mechanism using a number of simplifying assumptions. These assumptions range from linear time-invariant system models with no delay [10] to linear time-invariant systems with constant delays [3–9]. In a more general framework the case of uncertain constant delays was considered in [11]. Some results even consider nonlinear effects such as the saturation of the buffer occupancy [5]. Even though most papers deal with the case of a single congested switch, there is some recent work where multiple congested switches were allowed [12].

In this paper, based on the work in [13], we develop a model for a rate-based congestion control system, considering rapidly changing buffer levels. This not only accounts for the real situation of time-variant delays between congested node and sources, but, as will be explained later, can also cope with the varying RM-cell rate and the resulting mismatch with the fixed controller cycle time. The presented models for

*This work was supported by NSF grants ANI 9726253 and ANI 9726247.

the source-switch link can, therefore, be thought of as a macroscopic model. Furthermore, we will also include the effects of the buffer and rate saturation nonlinearity without the simplifying assumption of linearization around an equilibrium point. The resulting time-variant linear feedback system model (nonlinearities are modeled through time-variant sector gains) is then analyzed with regard to its stability using stability theory for uncertain time-variant systems. It is shown that no stable equilibrium point exists, if the delays in the forward path are time-variant. This essentially means that, under time-variant delay conditions, set point control of the congested buffer sources cannot provide the desired queue occupancy. Therefore, set point control is an illusive goal, if forward path delays vary.

In Section 2 of this paper, we will introduce the new time-variant uncertain delay model for congestion control of ABR traffic in ATM networks. Section 3 briefly addresses the problem of stability and existence of equilibria for the developed models and Section 4 introduces examples in order to illustrate the results. Section 5 provides the conclusion and an outlook for future work.

2 The Time-Variant Delay Model

Throughout this paper, we make the following modeling assumptions:

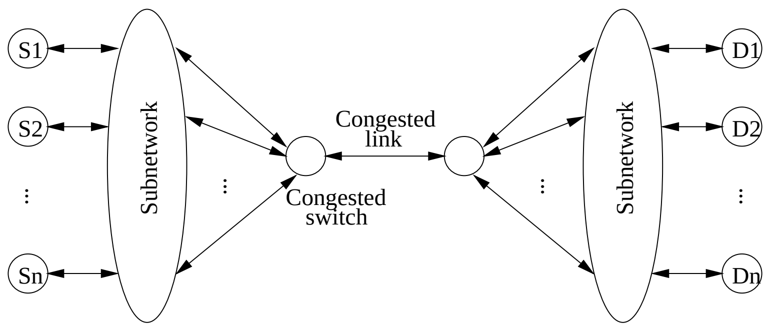


Figure 1: Single Congested Node ATM Network

- We consider a simple network with a single congested node (shown in Figure 1) and end to end RM cell routing.
- The number of sources trying to send cells through the same output link of the congested node is M.
- All sources are greedy and hence will always send at the maximum allowable rate.
- Bandwidth for the ABR traffic on the congested link is b_0.
- The variable bit-rate controller is located at the congested switch and uses a fixed sampling time T.
- The congested switch uses the Resource Management (RM) cells on the return path to inform the sources about the rate at which they should transmit. The delay these RM cells undergo from the congested node to the source will be time-variant in nature. These are referred to as return path delays.
- The effect of a rate change at the source is “felt” at the congested switch only after a time-variant delay, which is due to the buffer or queue delays of all switches (denoted as forward path delays) that the data has to pass before it arrives at the congested node.

Figure 2 depicts the case of a single source transmitting data through the congested switch. We will analyze the case of multiple sources later.

The two paths presented in Figure 2 are in reality one single communication link (or a chain of links and switches) but qualitatively they transport two different types of data. On the return path RM cells travel from the switch to the source. On the forward path the user data travels from the source through the congested switch. We need two different models corresponding to the two different quantities of interest: propagation of data volume in the forward path and propagation of a signal (rate request) in the return path. Both models will be formulated in discrete time with sampling period T, since this simplifies the analysis of the arising system.

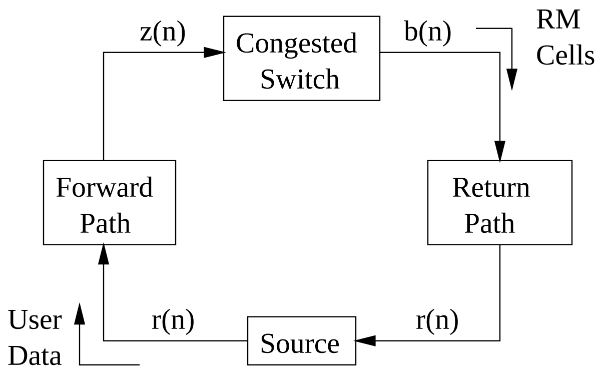


Figure 2: The single source case

2.1 The "Hold Freshest Sample" Model

We will assume that the switch computes the explicit rates using a linear controller. The output of the controller is written into the ER field of the RM cells in the return path and fed back to the sources.

The controller uses a fixed period to compute the new rates which will be inserted into the RM cells that travel on the return path. The generation of RM cells in general does not follow a fixed period (in absolute time) thus creating a rate mismatch between the controller rate and the RM cells rate. This rate mismatch may introduce a time-variant delay since an RM cell will not always be available at the time instant the controller computes its output (which is at a fixed rate of T^{-1}). The return paths RM cells also experience time-variant queueing delays in the intermediate switches. We will focus on the delay the RM cells encounter on the return path, and we will show how the model incorporates the delays caused by the rate mismatch.

The source adjusts its transmission rate to the one specified in the most recently received RM cell and continues to transmit at that rate until another RM cell arrives. Since the source "holds" the same rate until it receives "fresh" information, we will call this the "Hold Freshest Sample" (HFS) delay interface model (also called output variable delay in [13]).

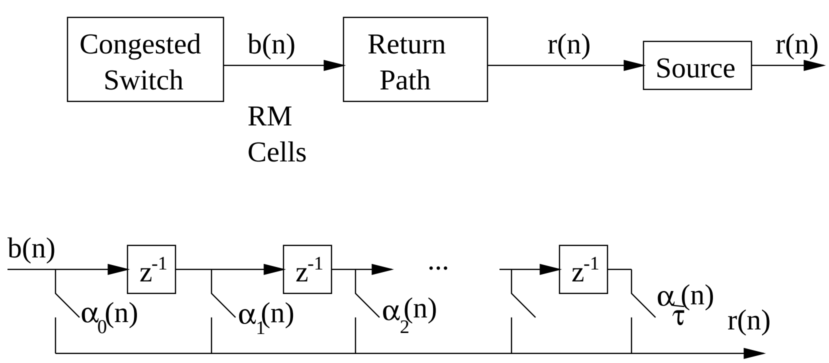


Figure 3: HFS model for the communication link: a tapped delay line with varying tap positions

Figure 3 depicts the HFS model for the return path. We denote with $b(n)$ the rate computed at the congested node at time instant n, with $r(n)$ the rate at which the source transmits at time instant n and with $\bar{\tau}$ (an integer multiple of the sampling period) the maximum delay encountered by an RM cell on the return path. The time-variant coefficients $\alpha_j(n)$ turn on and off exactly one switch at every time instant n. Which switch is turned on is determined by the "age" of the last received RM cell. Thus, if we have $r(n) = b(n - \tau(n))$ then:

$$\alpha_j(n) = \begin{cases} 1 & \text{if } j = \tau(n) \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Notice that by HFS definition, the coefficients $\alpha_j(n)$ cannot vary arbitrarily from one time instant to another [14]: the integer delay $\tau(n)$ is restricted by $\tau(n+1) \leq \tau(n) + 1$, and hence we have:

$$\alpha_j(n) = 1 \Rightarrow \alpha_k(n+1) = 0 \;\; \forall k > j+1 \tag{2}$$

Therefore, a sample that was available at time instant n will be held (reused) at time instant $n+1$, if no fresher sample has arrived at time $n+1$. Hence, the held sample ages by one time instant, which can be interpreted as a delay increase by one. Generally, samples age with time, leading to maximally a linear increase in delay over time. In the application at hand, the sample carries the explicit bit rate.

The delays encountered in ATM networks have four components: packet delays, transmission delays, processing delays and queueing delays. The first three categories are fairly constant, but the fourth category is the major source of time-variance.

For the queueing delays, the maximum and minimum delay ($\bar{\tau}$ and $\underline{\tau}$ respectively) are simple to compute: the minimum delay is zero corresponding to an empty tandem queue (all queues from switch to the source are empty), while the maximum delay occurs when all the queues from the switch to the source are at maximum occupancy.

Since the variation of the delay is due to the variation in the queue length in the switches between the congested switch and the source, we may be able to derive better bounds on the variation of the delays. This requires knowledge of at least bounds on the input/output rate of the buffers in between source and congested switch. Thus, if we bound the input rate $g(n)$ into a single queue by

$$0 < \gamma_1 g_0 \leq g(n) \leq \gamma_2 g_0 < \infty \tag{3}$$

where g_0 is the fixed rate at which the queue is depleted, then we have the following bound on the variation of the delay:

$$1 - \frac{1}{\gamma_1} \leq \tau(n+1) - \tau(n) \leq 1 - \frac{1}{\gamma_2} \tag{4}$$

In general the bounds in equation (4) are not integers and they should be interpreted as a long-term delay slope: if $1 - \frac{1}{\gamma_2} \leq \frac{q}{p}$ then $\tau(n+p) - \tau(n) \leq q$ where p and q are positive integers.

Of course the delay cannot decrease with more than the difference between the maximum and the minimum delay:

$$\max\{1 - \frac{1}{\gamma_1}, -(\bar{\tau} - \underline{\tau})\} \leq \tau(n+1) - \tau(n) \leq 1 - \frac{1}{\gamma_2} < 1 \tag{5}$$

For an excellent and comprehensive treatise of the subject, see the work of Cruz [15, 16].

As we mentioned earlier, in the general case there is a rate mismatch between the fixed controller rate and the time-variant RM cell rate.

There are two cases to be considered:

(a) the controller rate is greater than the RM cell rate at the switch

(b) the controller rate is smaller than the RM cell rate at the switch

Ideally, the controller rate should be chosen such that it is always lower than the RM cell rate. In case (a) the controller output is subsampled by the RM cells and transported to the source. In this case the controller output encounters an additional time-variant delay since the controller needs to wait for an RM cell to become available. This effect can be modeled by skipping samples on the delay line in Figure 3 resulting in a sawtooth delay time function. Case (b) can simply be handled by holding the last controller sample and repeatedly inserting it into the RM cell stream until the next sample becomes available. On the source side the most recent rate information of all the RM cells that arrives during one cycle is used. Therefore, this case does not introduce additional delays. These considerations already show that case (b) is preferable, i.e. the RM cell rate should, if possible, always be higher than the controller rate.

2.2 The Variable Bit Rate Model

The return path model in the previous Section describes the propagation of *explicit rate information* from the switch to the sources. We will now turn to the forward path model, which quantifies the propagation of *data volume* from the sources to the switch at any given time n. We assume that there are no cell losses on the communication channel. Such a model was presented in [13] as input variable delay.

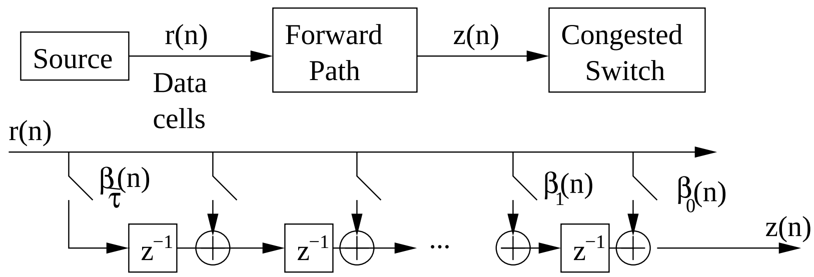


Figure 4: VBR model for the forward path

In the VBR model presented in Figure 4, $r(n)$ denotes the number of cells transmitted by the source between time instant $n-1$ and time instant n, $z(n)$ is the number of cells that arrive at the congested switch between time instant $n-1$ and time instant n. The time-variant coefficients $\beta_i(n)$ turn on exactly one "switch" (in Figure 4) at every time instant n. Which "switch" is turned on is determined by the average delay the cells transmitted between time instants $n-1$ and n will encounter. Thus, if the delay at time n is $\tau(n)$ then:

$$\beta_j(n) = \begin{cases} 1 & \text{if } j = \tau(n) \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

Since the delays on the forward path have the same nature as those on the return path, similar considerations as in the HFS case apply. If the total input rate $g(n)$ into a single queue (i.e. the sum of the rates of all the sources that feed into that queue) is bounded by

$$0 < \gamma_1 g_0 \leq g(n) \leq \gamma_2 g_0 < \infty \tag{7}$$

then the delays encountered by the packets of any input source will have a delay variation bounded by:

$$-1 < \gamma_1 - 1 \leq \tau(n+1) - \tau(n) \leq \gamma_2 - 1. \tag{8}$$

The bounds on the delay variations (4) and (8) are different, as the nature of the two delays is different. For example, the HFS delay cannot increase by more than one per time step (due to the holding action), while the VBR delay cannot decrease by more than one per time step (in order to assure that packet order is maintained). For detailed derivations of these bounds see [14, 17], or [15, 16].

The model shown in Figure 4 is a macroscopic model for the delays and their effects on the data rates. The delays can be viewed as the compounded delays generated in individual queues from the source to the switch, but other delay effects can also be modeled by this method.

2.3 Total System Model

Figure 5 depicts the total system model for the case of multiple sources. We denote with M the maximum number of sources that may connect to the congested switch at one time. T is the sampling period of the discrete time system, dictated by the controller cycle time.

The congested switch model consists of a finite buffer, a queue and a rate control component. The buffer receives incoming data from all sources. The data rates (bps) are converted to data (bits) by multiplication with the sampling period T. At each time instant n the buffer level $y_s(n)$ is equal to the buffer level at the previous time instant $y_s(n-1)$ plus the sum of all new data $\sum_{i=1}^{M} z_i T$ that is received from the M sources minus the forwarded data $b_0 T$. The buffer is subject to a saturation nonlinearity enforcing a strictly positive queue length and a finite buffer size. The switch computes the desired rates

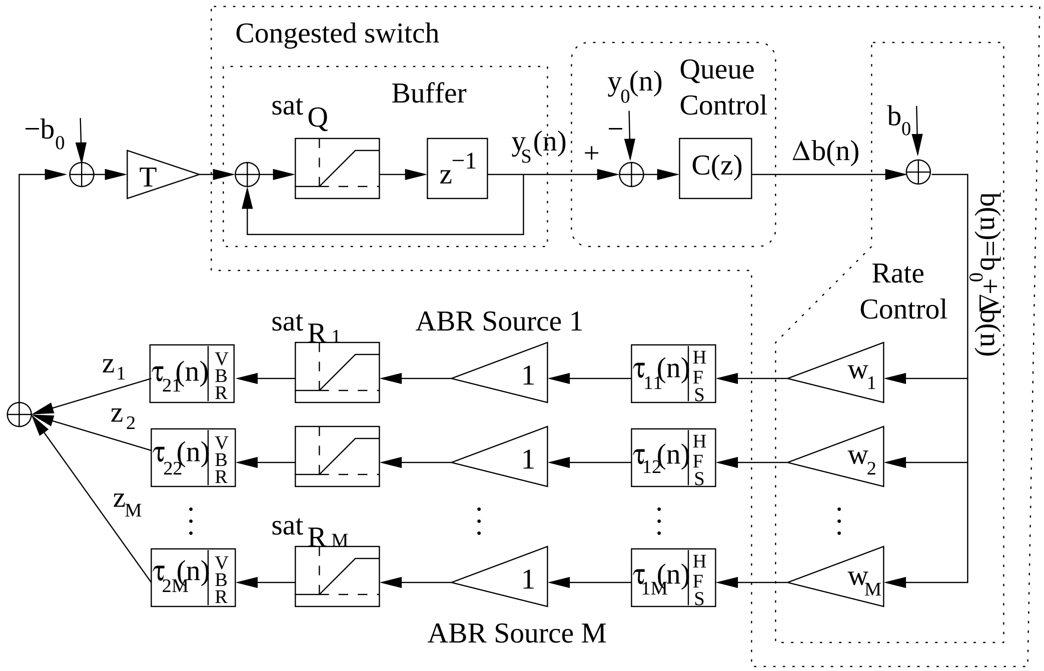


Figure 5: Total system model

for the sources (by subtracting the desired set-point from the current queue length, implementing the transfer function of the controller and dividing the total rate request according to the weights), such that the buffer length is kept at the desired set-point. The desired rates are sent to the ABR sources and experience HFS delays in the return paths. Each ABR source is assumed to be both compliant and greedy; thus the command received from the congested switch is immediately obeyed. Hence, each ABR source is represented by a unity gain. Data sent by the ABR sources is subject both to saturation nonlinearities (enforcing positive data rates and finite link bandwidths) as well as to the VBR delays in the forward path.

The weights $w_i(n)$ represent the "fair" share of the bandwidth allocated to source i and can be computed using a max-min fairness algorithm [9, 18]. The weights $w_i(n)$ vary with time as virtual circuits connect or disconnect from the congested switch; their sum is equal to one:

$$\sum_{i=1}^{M} w_i(n) = 1. \tag{9}$$

Notice that our proposed scheme (Figure 5) has max-min fairness which is inherited from the max-min algorithm employed in computing the weights w_i. At the same time it has the desirable property of constant complexity $O(1)$ with respect to the number of sources which is especially attractive if the number of sources is large.

The rate $b(n)$ is computed at the controller of the congested switch, and it represents the total desired incoming rate for the congested switch. The constant rate b_0 is the output bandwidth available for ABR traffic. Matching the incoming rate to the output rate assures a stable steady-state for the queue, but due to time-variance, rate-nonlinearities, weight changes, etc., the queue length can vary in the absence of queue control. The queue control component is denoted by $\Delta b(n)$, and it aims to stabilize the congested switch queue length to a fixed set point y_0. In addition, the queue control component can correct a number of non-ideal phenomena of the rate control system like saturation, quantization, cell-loss, etc.

$y(n)$ represents the length of the queue of the congested switch (i.e without considering the saturation effects). $y_s(n)$ is the queue length after the saturation is accounted for:

$$y_s(n) = sat_Q(y(n)) \tag{10}$$

where the saturation function is defined as follows:

$$sat_Q(y) = \begin{cases} 0 & \text{if} \quad y < 0 \\ y & \text{if} \quad 0 \le y \le y_{\max} \\ y_{\max} & \text{if} \quad y_{\max} < y \end{cases} \tag{11}$$

where $y_{\max}$ is the buffer capacity (in cells). The saturation nonlinearity can be represented by a sector description around a setpoint y_0:

$$sat_Q(y) = Q(n)(y - y_0) + y_0 \quad \text{where} \quad Q(n) \in [Q_{\min},\ 1] \tag{12}$$

(This is not a linearization and models the full dynamic range of the nonlinearity [13, 19]).

y_0 is the buffer set point. Ideally, at steady state, the buffer will have $y_s(n) = y_0$ which will ensure that the buffer will not overflow (losing data cells) or underflow (missing service opportunities).

Similar to the queue saturation rate, we model the saturation of the source rates by positive rates that are less than or equal to the available bandwidth. We define the rate saturation for the i^{th} source:

$$sat_{R,i}(r_i) = \begin{cases} 0 & \text{if} \quad r_i < 0 \\ r_i & \text{if} \quad 0 \le r_i \le R_{i,\max} \\ R_{i,\max} & \text{if} \quad R_{i,\max} < r_i \end{cases} \tag{13}$$

where r_i is the explicit rate feedback at source i. We can also use a sector description for the rate saturation around a rate equilibrium point $r_{0,i}$:

$$sat_{R,i}(r_i) = R_i(n)(r_i - r_{0,i}) + r_{0,i} \quad \text{where} \quad R_i(n) \in [R_{i,\min},\ 1] \tag{14}$$

The condition $R_{i,\min} > 0$ for some i is an essential condition for the stability of the system: if $R_{i,\min} = 0 \quad \forall i = 1, \ldots, M$, we have the possibility of an open loop system with an unstable plant (buffer). In practice $R_{i,\min} > 0$ is always satisfied, as links have strictly positive bandwidths.

The delays $\tau_{1,i}(n)$ and $\tau_{2,i}(n)$ correspond to the HFS/VBR models for the return paths and the forward path respectively, where the index i denotes the source with $i = 1, ..., M$. We assume that the delays are bounded:

$$0 \le \tau_{1,i}(n) \le \bar{\tau}_{1,i} \tag{15}$$

$$0 \le \tau_{2,i}(n) \le \bar{\tau}_{2,i} \tag{16}$$

Let us denote with $\alpha[j,i](n),\ j = 1, \ldots, \bar{\tau}_{1,i},\ i = 1, \ldots, M,\ n \ge 0$ the j^{th} time-variant coefficient α_j of the HFS model for the i^{th} return path at time instant n. Similarly, denote with $\beta[j,i](n),\ j = 1, \ldots, \bar{\tau}_{1,i},\ i = 1, \ldots, M,\ n \ge 0$ the time-variant coefficient β_j of the VBR model for the i^{th} forward path at time instant n. The coefficients $\alpha[j,i](n)$ and $\beta[j,i](n)$ are computed using the equations (1) and (6).

For the sake of brevity in what follows, we will not explicitly show the dependence of $\alpha[j,i]$,$\beta[j,i]$, w_i, Q and R_i on n.

We now use a state space representation for the time-variant $i-th$ delay path in terms of the weight w_i, the two time-variant delays $\tau_{1i}(n)$ and $\tau_{2i}(n)$, and the source rate nonlinearity corresponding to source i as shown in Figure 5 (state space descriptions for each of the individual delay types, i.e. HFS or VBR, can be found in [13, 14, 17]):

$$x_i(n+1) = A_i(n)x_i(n) + B_i(n)b(n) \tag{17}$$

$$z_i(n) = C_i(n)x_i(n) + D_i(n)b(n) \tag{18}$$

$$A_i(n) = \begin{pmatrix} 0 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 & 0 \\ 1 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 1 & \ldots & 0 & 0 & 0 & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & 0 & 0 & \ldots & 0 & 0 \\ R_i\beta[\bar{\tau}_{2,i},i]\alpha[1,i] & R_i\beta[\bar{\tau}_{2,i},i]\alpha[2,i] & \ldots & \ldots & R_i\beta[\bar{\tau}_{2,i},i]\alpha[\bar{\tau}_{1,i},i] & 0 & \ldots & 0 & 0 \\ R_i\beta[\bar{\tau}_{2,i}-1,i]\alpha[1,i] & R_i\beta[\bar{\tau}_{2,i}-1,i]\alpha[2,i] & \ldots & \ldots & R_i\beta[\bar{\tau}_{2,i}-1,i]\alpha[\bar{\tau}_{1,i},i] & 1 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ R_i\beta[1,i]\alpha[1,i] & R_i\beta[1,i]\alpha[2,i] & \ldots & \ldots & R_i\beta[1,i]\alpha[\bar{\tau}_{1,i},i] & 0 & \ldots & 1 & 0 \end{pmatrix} \tag{19}$$

$$B_i(n) = \begin{pmatrix} w_i \\ 0 \\ \vdots \\ 0 \\ w_i R_i \beta[\bar{\tau}_{2,i}, i]\alpha[0,i] \\ w_i R_i \beta[\bar{\tau}_{2,i} - 1, i]\alpha[0,i] \\ \vdots \\ w_i R_i \beta[1,i]\alpha[0,i] \end{pmatrix} \tag{20}$$

$$C_i(n) = \begin{pmatrix} R_i\beta[0,i]\alpha[1,i]; & R_i\beta[0,i]\alpha[2,i]; & \ldots; & R_i\beta[0,i]\alpha[\bar{\tau}_{1,i},i]; & 0; & \ldots; & 0; & 1 \end{pmatrix} \tag{21}$$

$$D_i(n) = \begin{pmatrix} R_i w_i \beta[0,i]\alpha[0,i] \end{pmatrix} \tag{22}$$

$x_i(n)$ corresponds to the state of the i^{th} delay line corresponding to the i^{th} source, including the RM cell path from the congested switch to the source and the data cell path from the source to the congested switch. $x_i(n)$ has the dimension $\bar{\tau}_{1,i} + \bar{\tau}_{2,i}$. $z_i(n)$ corresponds to the data cells sent by the i^{th} source that reach the congested switch between time instant $n-1$ and time instant n.

$w_i(n)b(n)$ corresponds to the fair share of the total bandwidth $b(n)$ as it is computed by the controller of the congested switch at the time instant n. It includes both the queue control and the rate control components, and it represents an imperative command for the source to start transmitting at (or below) that new rate. With our assumption of greedy sources, they transmit at exactly that rate.

System in (17, 18) describes the behavior of source i and its data output z_i as perceived by the congested switch in response to the command $b(n)$. Each state corresponds to one delay as presented in the models of Fig. 3 and Fig. 4.

The linear time-invariant controller $C(z)$ can be chosen to be of the general form:

$$x_c(n+1) = A_c x_c(n) + B_c(y_s(n) - y_0(n)) \tag{23}$$

$$\Delta b(n) = C_c x_c(n) + D_c(y_s(n) - y_0(n)) \tag{24}$$

where by $x_c(n)$ we denote the state of the controller, $\Delta b(n)$ represents the queue control component of the bandwidth and $y_0(n)$ represents the set point for the queue length. $x_c(n)$ has the dimension N_c; input and output are scalars.

The controller used neither has to be linear nor time-invariant. In fact the model is sufficiently general to accommodate time-variant (adaptive, sliding mode) or nonlinear controllers. However, linear time-invariant controllers are simple to design and, as shown in Section 4, provide adequate performance for this application.

We can express the entire closed loop system in Figure 5 in state space form as follows:

$$x(n+1) = A(n)x(n) + B(n)\begin{pmatrix} b_0(n) \\ y_0(n) \end{pmatrix} \tag{25}$$

$$y_s(n) = Cx(n) + D\begin{pmatrix} b_0(n) \\ y_0(n) \end{pmatrix} \tag{26}$$

where:

$$A(n) = \begin{pmatrix} Q + QT\left(\sum_{i=1}^{M} D_i\right) D_c & QT\left(\sum_{i=1}^{M} D_i\right) C_c & QTC_1 & QTC_2 & \ldots & QTC_M \\ B_c & A_c & 0 & 0 & \ldots & 0 \\ B_1 D_c & B_1 C_c & A_1 & 0 & \ldots & 0 \\ B_2 D_c & B_2 C_c & 0 & A_2 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ B_M D_c & B_M C_c & 0 & 0 & \ldots & A_M \end{pmatrix} \tag{27}$$

$$B(n) = \begin{pmatrix} QT\sum_{i=1}^{M} D_i - QT & -QT\left(\sum_{i=1}^{M} D_i\right)D_c \\ 0 & -B_c \\ B_1 & -B_1 D_c \\ B_2 & -B_2 D_c \\ \vdots & \vdots \\ B_M & -B_M D_c \end{pmatrix} \tag{28}$$

$$C = \begin{pmatrix} 1, & 0, & \ldots, & 0 \end{pmatrix} \tag{29}$$

$$D = \begin{pmatrix} 0, & 0 \end{pmatrix} \tag{30}$$

The state vector $x(n)$ is composed by the queue length, the state of the controller and the state of the delay lines: $x(n) = (y_s(n) \quad x_c(n) \quad x_1(n) \quad x_2(n) \ \ldots \ x_M(n))^T$. The state vector $x(n)$ has the dimension $1 + N_c + \sum_{i=1}^{M}(\bar{\tau}_{1,i} + \bar{\tau}_{2,i})$.The inputs are the available bandwidth $b_0(n)$ and the queue set point $y_0(n)$, while the output is the queue length $y_s(n)$ which is the best measure of the performance of the control system. Ideally, the output should track the input $y_0(n)$ with zero steady state error, should not saturate and should reject any disturbances due to changes in $w_i(n)$ as quickly as possible. Since the overall feedback system is of type 1, it always tracks step inputs which is what one wants in practice (i.e. buffer set-point).

The system model given by equations (25-30) is the key for analysis, design and simulation of the congestion control scheme. However, the switch is only required to execute code that corresponds to the two boxes labeled "queue control " and "rate control" in Figure 5. In the simplest case, this would require one subtraction $(y_S(n) - y_0(n))$ plus one multiplication with the gain G of a proportional controller for the queue control component. The complexity of the rate control component is independent of the controller chosen and given by M multiplications with the weights. Equations (25-30) are only used to design the controller $C(z)$ and evaluate stability and performance of the congestion control scheme, which is done off-line in the design stage. Since the ideal controller time step is larger than the RM cell spacing, there is plenty of time to perform the required operations. In case a more complex controller is desired, for example, an order h IIR controller, only $2h$ multiplications and $2h + 1$ additions will be needed at every time step. It is unlikely that this can become a computational burden even for a modest general purpose microprocessor.

The output of the controller is independent of the number of flows: it only takes the total queue length into consideration. After the computation, the output of the controller is multiplied with the weights w_i which does depend on the number of flows. However, the multiplication of the controller output with a weight w_j only needs to be performed when an RM cell for flow j passes by, and thus, the switch only has to perform a multiplication per RM cell. This should not present any problems, even in low end switches. Keeping track of the appropriate w_i is essential in *any* ATM congestion control scheme, as w_i provides max-min fairness to all flows. Efficient schemes to compute w_i exist [9], and storing the proper weights should not be a problem either, because ATM switches have to store per-flow information (e.g. virtual circuit maps, minimum/peak bandwidth for each virtual circuit, etc).

For the case of constant forward path delays, it is easily shown by using a minimal state space representation with only one forward and one return delay chain, that all instantaneous A-matrices in (27) that can possible appear in the multi-source case are convex combinations of all A-matrices that occur in the single source case. The proof rests on the facts that $\sum w_i = 1$ and that all HFS delays have the same maximum delay $\bar{\tau}_1$, which is not a restriction. The convex matrix polytope to be checked has $\bar{\tau}_1 + 1$ vertices, if nonlinearities are not considered. As will be seen in the next Section, this fact makes the controller design independent of the number of sources. Therefore, the choice of the controller depends only on the maximum delay that can be encountered in the forward and return paths. At design time, multiple controller gains corresponding to different maximum delays can be computed and uploaded into the switch. Once the switch is installed in the network, the maximum occurring delay determines which of the stored controller coefficient sets (possibly in the form of a look up table) is used. Of course, improved performance can be obtained by using a variety of time-variant controllers such as time-variant Smith predictors, adaptive controllers or sliding mode control.

3 Set Point Control and Stability

Using the models derived in the previous Section, we will now show that set-point control is an illusive goal in congestion control systems with time-variant forward path delays. Stability of the congestion control system (in Bounded Input Bounded Output (BIBO) sense) is, however, still a vital requirement for proper response behavior, since it assures that the buffer occupancy trajectories stay within certain bounds. If BIBO stability is not ensured, buffer oscillations can easily lead to a loss of data and inefficient link utilization.

There are a number of results in the literature [20, 21] concerning the control of a plant through a communication network while taking into account the variation of the delays. In [20, 21] the plant to be controlled is a buffer which closely matches the problem discussed in this paper, but the stability criterion employed in these papers requires an FIR controller. Also, the sufficient stability conditions were often shown to be conservative. In this paper we will pursue another avenue: we will use a necessary and sufficient stability condition presented in [22].

The system model is time-variant with the matrices $A(n)$ and $B(n)$ depending on the particular combination of delays $\tau_{1,i}(n)$ $\tau_{2,i}(n)$ and on the presence of queue and/or source rate saturation.

At first we will introduce a Lemma, that essentially shows that the overall system cannot stay at the equilibrium point if the delays $\tau_{2,i}(n) \quad i = 1, \ldots, M$ in the forward path are varying.

Lemma 1 *The system in (25),(26) cannot have a stable, non-zero equilibrium if the delay trajectories on the forward path are time-variant, i.e. there is at least one delay $\tau_{2,i}(n)$ such that $\tau_{2,i}(n) \neq const.$*

The proof can be found in [23] and is omitted for the sake of brevity. A brief interpretation of the result follows: Consider the model for a VBR delay in Figure 4. Even if the input rate to the VBR delay is constant, changes in the VBR delay can cause the output rate to vary significantly. The varying output directly feeds into the buffer of the switch (integrator), thus disturbing any previously existing buffer occupancy equilibrium.

Comments:

- This Lemma shows that set-point control for congestion control systems is an illusive goal in the presence of time-variant forward path delays.
- The phenomenon mentioned above does not occur if the time-variant delay is on the HFS side which is easily confirmed via Figure 3. When the system is at steady-state, the value of $r(n)$ in Figure 3 is also guaranteed to be at steady-state irrespective of the value of $\alpha_j(n)$.
- A similar result (as in Lemma 1) can be formulated for the case when $w_i = w_i(n)$. Even though $\sum_{i=1}^{M} w_i(n) = 1 \quad \forall n \geq 0$ the time-variant delays $\tau_{1,i}(n)$ will result in the actual rate splitting between the sources to occur at different times if $\tau_{1,i}(n) \neq \tau_{1,j}(n), \quad i \neq j$. Hence, the sum of all rates $\sum_{i=1}^{M} z_i(n) \neq b_0$, and there will be no equilibrium, i.e. $y_s(n) \neq y_0$.
- Stability of the congestion control system in (25),(26) in the BIBO sense is guaranteed, if $A(n)$ in (25) is exponentially stable [24].

Since time-variant delays in the forward path cause the system to have no equilibrium, and the delays in the return path are more critical (due to abrupt delay variations caused by the mismatch of varying RM cell spacing and the fixed controller sampling time), we will address the case of time-invariant uncertain delays in the forward path, time-variant uncertain delays in the feedback path and time-invariant weight distribution. A constant forward path delay that is uncertain (i.e. takes values in an interval) is often a sufficiently good description for slowly varying forward path delays [25]. Typically the weights are piecewise constant, which justifies such an analysis.

In order to address stability of the desired equilibrium point in the overall system in (25) and (26), we will use the following equilibrium point:

$$b(n) = b_0 \tag{31}$$
$$z_i(n) = w_i b_0 = z_{i,0} \quad (\text{assuming } w_i b_0 \leq R_{i,\max}) \tag{32}$$
$$y_s(n) = y_0(n) = y_0 \tag{33}$$

A new system description around the equilibrium point is obtained by letting $x_0 + \hat{x}(n) = x(n)$ in (25), where $x_0 = [y_0 \; x_{c0} \; x_{1,0} \ldots x_{M,0}]^T$. This yields the zero input system description:

$$\hat{x}(n+1) = A(n)\hat{x}(n) \tag{34}$$

with $A(n)$ given in (27) and the delays in the forward path fixed.

Denote by $\mathcal{P}$ the polytope of all matrices $A(n)$ for all combinations of delays $0 \leq \tau_{1,i} \leq \bar{\tau}_{1,i}$, with buffer and link saturation nonlinearities $Q \in \{Q_{\min},\ 1\}$ and $R_i \in \{R_{i\,\min},\ 1\}$ $\ i = 1, \ldots, M$:

$$\begin{aligned} \mathcal{P} = conv\,\{A(\tau_{1,1}; \ldots; \tau_{1,M}; Q; R_1; \ldots; R_M)\} \\ Q \in \{Q_{\min},\ 1\} \\ R_i \in \{R_{i\,\min},\ 1\} \\ 0 \leq \tau_{1,i} \leq \bar{\tau}_{1,i} \\ i = 1, \ldots, M \end{aligned} \tag{35}$$

where *conv* denotes the convex hull of the arguments.

Theorem 2 *The system*

$$\hat{x}(n+1) = A(n)\hat{x}(n), \quad A(n) \in \mathcal{P} \tag{36}$$

is globally asymptotically stable, iff $\exists$ *a finite* k*, such that:*

$$\|T^{-1} V_{i_1} \cdot \ldots \cdot V_{i_k} T\| \leq \gamma < 1, \quad \forall (i_1, \ldots, i_k) \in \{1, \ldots, N\}^k \tag{37}$$

where V_i *are the exposed vertices of polytope* $\mathcal{P}$*,* $\|\cdot\|$ *is any induced matrix norm and* T *is any invertible matrix with the same dimension as the vertex matrices* V_i*.*

The proof can be found in [23, 26] and is based on a result in [22]. It also guarantees exponential stability [27]. It is well known [22, 26] that for a time-variant system with A-matrices taken from a matrix polytope, exponential stability of the entire polytope is equivalent to exponential stability of a time-variant system with A matrices taken only from the set of vertex matrices. With the observation made in the previous Section, that the occurring matrices of the multi-node case can be expressed as convex combinations of the vertex matrices in the single node case, the stability check complexity is independent of the number of connecting sources.

The choice of the matrix T is critical in reducing the complexity of the test. A good algorithm to find the transformation matrix T is presented in [26].

4 Examples

We will provide two examples to illustrate our results. In the first example (the case of a single source node), we will demonstrate the non-existence of a buffer occupancy equilibrium in the presence of time-variant forward path delays. We also illustrate that in contrast, time-variant return path delays allow for such an equilibrium. Furthermore, we demonstrate network behavior if one designs the system without checking stability.

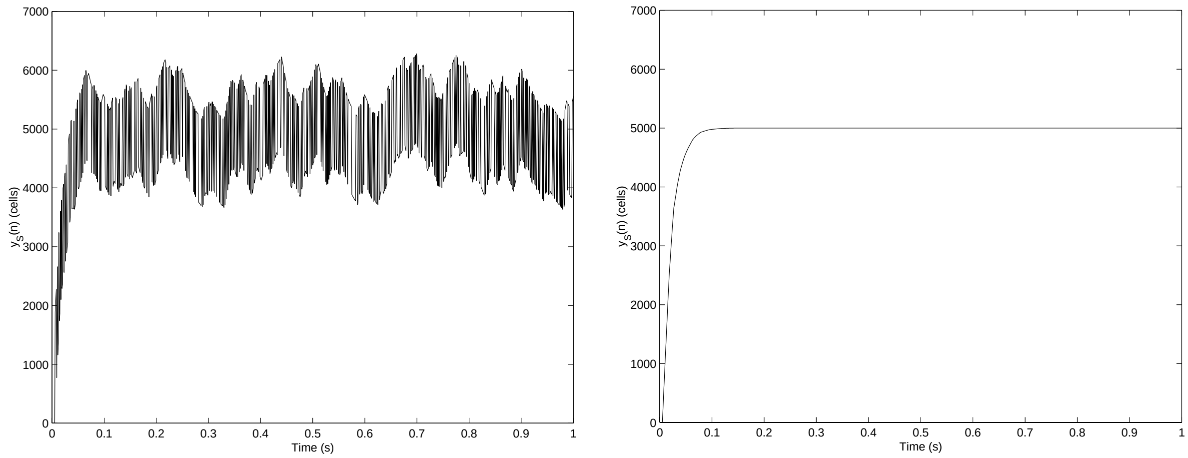
The second example illustrates network behavior for a large number of source nodes. The design illustrates stable buffer occupancy trajectories, even if the number of source nodes connecting increases over time.

4.1 The Case of a Single Source

In this Section we will consider the case of a congested switch with only one source. We will use the following parameters for our system:

- The bandwidth available for ABR traffic $b_0 = 1500$ cells/s.
- The maximum rate $R_{1,\max} = 2b_0 = 3000$ cells/s.
- The buffer length $y_{\max} = 10000$ cells.
- The buffer set point $y_0 = \frac{1}{2}y_{\max} = 5000$ cells.
- The controller cycle time $T = 1$ ms.
- The maximum delay on the return path $\bar{\tau}_{1,1} = 10$ ms.
- We used a proportional controller with a gain $-G$ (i.e. $\Delta b(n) = -G(y_s(n) - y_0(n))$).

Fig. 6(a) shows the buffer occupancy in the case of time-variant forward path (VBR) delays varying between 0 and $1ms$. Clearly, no equilibrium is reached, and the response is bounded (which is due to exponential stability of the origin of the feedback system with time-variant VBR delays). Fig. 6(b) shows the buffer occupancy trajectory $y_S(n)$ for time-variant return path delays between 0 and 10 ms,



(a) (b)

Figure 6: (a) Buffer level $y_S(n)$ with time-variant delay on the VBR side; (b) Buffer level $y_S(n)$ with the time-variant delay on the HFS side.

while keeping VBR delays in the forward path fixed at $1ms$. The desired equilibrium at $y_0 = 5000$ cells is clearly maintained.

We assume the forward delay is fixed at 1ms and the return delays vary between 0 and 10ms. In this case, the time-invariant analysis results in the following stability condition: $G \in (0,\ 136.48)$.

The sufficient stability condition for the time-variant case introduced in [21] yields the following stability condition: $G \in (0,\ 6.57)$.

To compute the maximum stabilizing proportional controller G for the linear time-variant case we constructed the system matrices (27-30) for each possible return path delay ($\tau_{1,1} \in [0, ..., 10]$). We determined a suitable transformation matrix T as outlined in [26]. Finally, we increased G until condition (37) in Theorem 2 was no longer satisfied for one of the possible matrix products of length k. The results are presented in Table 1.

Product length k	2	3	4	5	6	7	8	9
Controller gain $0 < G \leq$	3.92	21.67	31.23	34.42	36.66	37.58	38.31	38.68

Table 1: The case of a single source for the linear case: Stable proportional controller gains as a function of the product length k

A gain $G = 38.68$ is guaranteed to stabilize the system, while a gain $G > 136.48$ is guaranteed to make the system unstable. These results apply to the case for a maximum return path delay of 10ms ($\bar{\tau}_{1,1} = 10$). If the controller is designed for 10ms but the maximum delay is 20ms, the system can become unstable as shown in Fig. 7. In this situation, the buffer shows large occupancy oscillations, packets being dropped on a regular basis. Also, the rates allocated to the users by the controller will vary significantly. Clearly, the system should not be operating in this regime.

4.2 Multiple Sources

In this Section we will consider a more realistic example with 100 sources feeding into the same congested switch. We used the same parameters as in the previous example, and we fixed the delay in the forward path to 1ms (i.e. $\tau_{2,1}(n) = 1$). Since the same controller that stabilizes one source stabilizes multiple sources, a gain $G = 38.68$ will be used.

The system starts at equilibrium with 100 sources feeding into the congested switch. Ten additional sources join sequentially at 200ms intervals. The results of the simulation are shown in Fig. 8. As each source joins the switch, a small glitch can be observed. This is the effect of delays in updating the weights. When a new source is connected, the switch will compute new weights and send them to the sources (indirectly - by multiplying the computed rate by the weights). However the updated weights and the

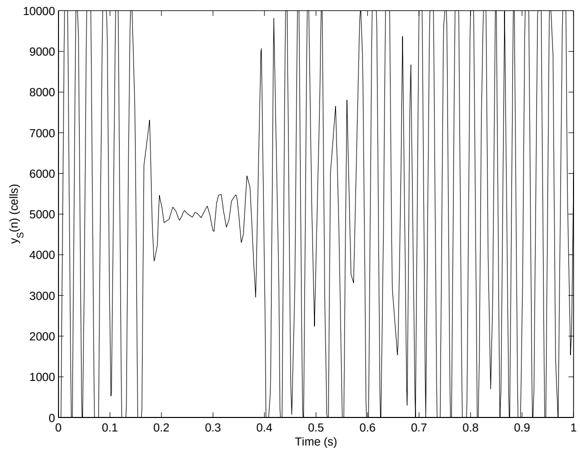


Figure 7: Unstable buffer level $y_S(n)$ with an unsuitable gain G.

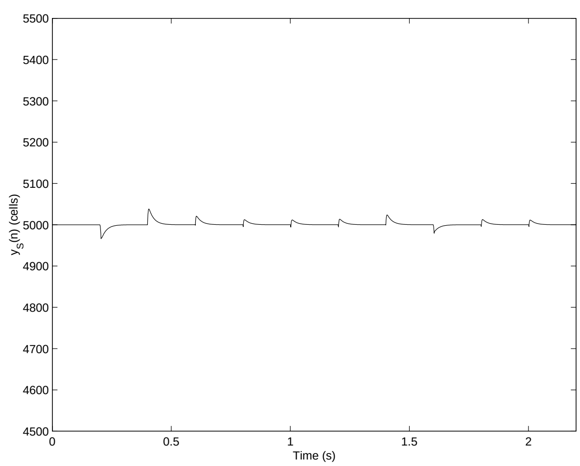


Figure 8: Buffer level $y_S(n)$ as 10 additional sources join the congested switch sequentially, at 200ms intervals

reaction of the sources to those updated weights are delayed (as any other command or reaction), and therefore, for a brief period of time, the sum of the weights may not be equal to one. After all switches are updated, the control scheme kicks in and brings the buffer to the desired set-point.

5 Conclusion

This paper introduced a time-variant delay model for the ABR option of ATM networks. The introduced congestion control system is capable of modeling time-variant communication delays between a single congested node and several sources (in both directions), rate and buffer non-linearities, RM-cell losses as well as the mismatch between time-variant RM cell periods and the controller cycle time. To the author's knowledge, the presented approach is the first dynamical system model offering this high level of modeling accuracy and detail. The model was analyzed for stability of a chosen equilibrium point (typically given as a nominal buffer occupancy level). A unique advantage of the presented analysis method is the fact that its complexity is independent of the number of sources. It was proved that time-variant forward path delays do not allow for an equilibrium of the congestion control system. This shows that set point control in congestion control systems is generally an illusive goal. In the case where the delays in the forward path can be modeled as time-invariant, an equilibrium exists and its stability can be analyzed using Theorem 2.

References

[1] ATM Forum, "The ATM forum trafic management specifications version 4.0," Apr. 1996.

[2] C. E. Rohrs, R. A. Berry, and S. J. O'Halek, "Control engineer's look at ATM congestion avoidance," *Computer Communication*, vol. 19, pp. 226–234, Mar. 1999.

[3] A. Kolarov and G. Ramamurthy, "A control-theoretic approach to the design of an explicit rate controller for ABR service," *IEEE/ACM Transactions on Networking*, vol. 7, no. 5, pp. 741–753, 1999.

[4] C. E. Rohrs and R. A. Berry, "A linear control approach to explicit rate feedback in ATM networks," in *Proc. IEEE Infocom*, pp. 277–282, 1997.

[5] H. Zhang, O. W. Yang, and H. Mouftah, "Design of robust congestion controllers for ATM networks," in *Proc. 16th IEEE Annual Conference on Computer Communications INFOCOM*, vol. 1, pp. 302–309, Apr. 1997.

[6] P. Narváez and K. Siu, "Optimal feedback control for ABR service in ATM," in *Proc. International Conference on Network Protocols*, pp. 32–41, 1997.

[7] S. Mascolo, D. Cavendish, and M. Gerla, "ATM rate based congestion control using a Smith predictor: An EPRCA implementation," in *Proc. INFOCOM'96*, vol. 2, pp. 569–576, Mar. 1996.

[8] S. Mascolo, "Smith's principle for congestion control in high speed ATM networks," in *Proc. of 36th Conference on Decision and Control*, pp. 4595–4600, Dec. 1997.

[9] H. Ohsaki, M. Murata, and H. Miyhara, "Designing efficient explicit-rate switch algorithm with max-min fairness for ABR service class in ATM networks," in *Proc.1997 IEEE International Conference on Communications, ICC.*, vol. 1, pp. 182–186, 1997.

[10] H. M. Choi, K. E. Nygard, and R. J. Vetter, "Counter-based congestion avoidance scheme for ABR traffic in ATM networks," in *Proceedings of the 1998 IEEE International Performance, Computing and Communications Conference*, pp. 284–290, Feb. 1998.

[11] L. Benmohamed and S. M. Meerkov, "Feedback control of congestion in packet switching networks: The case of a single congested node," *IEEE/ACM Transactions on Networking*, vol. 1, Dec. 1993.

[12] L. Benmohamed and S. M. Meerkov, "Feedback control of congestion in packet switching networks: The case of multiple congested nodes," *International Journal of Communication Systems*, vol. 10, pp. 227–246, Sept. 1997.

[13] M. M. Ekanayake, *Robust Stability of Discrete Time Nonlinear Systems*. PhD thesis, University of Miami, 1999.

[14] P. H. Bauer, M. L. Sichitiu, and K. Premaratne, "On the nature of the time-variant communication delays," in *Proc. of the IASTED Conference on Modeling, Identification and Control*, (Innsbruck, Austria), pp. 792–797, Feb. 2001.

[15] R. L. Cruz, “A calculus for network delay, part I: Network elements in isolation,” *IEEE Transactions on Information Theory*, vol. 37, pp. 114–131, Jan. 1991.

[16] R. L. Cruz, “A calculus for network delay, part II: Network analysis,” *IEEE Transactions on Information Theory*, vol. 37, pp. 132–141, Jan. 1991.

[17] M. L. Sichitiu, *Computer Network Control: Models, Stability and Controller Design.* PhD thesis, University of Notre Dame, 2001.

[18] A. Arulambalam, X. Chen, and N. Ansari, “Allocating fair rates for available bit rate service in ATM networks,” *IEEE Communications Magazine*, pp. 92–100, Nov. 1996.

[19] D. Liu and A. N. Michel, *Dynamical systems with saturation nonlinearities.* Springer-Verlag, 1994.

[20] P. H. Bauer, M. L. Sichitiu, and K. Premaratne, “Controlling an integrator through data networks: Stability in the presence of unknown time-variant delays,” in *Proc. 1999 IEEE International Symposium on Circuits and Systems*, vol. 5, pp. 491–494, 1999.

[21] P. H. Bauer, M. L. Sichitiu, and K. Premaratne, “Closing the loop through communication networks: The case of an integrator plant and multiple controllers,” in *Proc. 38th IEEE Conference on Decision and Control*, pp. 2180–2185, Dec. 1999.

[22] P. H. Bauer, K. Premaratne, and J. Durán, “A necessary and sufficient condition for robust asymptotic stability of time-variant discrete systems,” *IEEE Transactions on Automatic Control*, vol. 38, pp. 1427–1430, Sept. 1993.

[23] M. L. Sichitiu, P. H. Bauer, and K. Premaratne, “The effect of uncertain time variant delays in ATM networks with explicit rate feedback,” in *American Control Conference (ACC’2001)*, pp. 4537–4543, 2001.

[24] A. P. Molchanov, “Lyapunov functions for nonlinear discrete-time control systems,” *Automat. Remote Control*, vol. 48, no. 6, pp. 728–736, 1987.

[25] P. H. Bauer, M. L. Sichitiu, K. Premaratne, and A. Hagen, “Stability of variable bit rate control systems for DSNs: Time-invariant versus time-variant delay models,” in *Proc. Eighteenth IASTED International Conference Modeling, Identification and Control (MIC’99)*, pp. 510–513, Feb. 1999.

[26] M. L. Sichitiu and P. H. Bauer, “Stability of discrete time-variant linear delay systems and applications to network control,” in *Proc. of International IEEE Conference on Electronics, Circuits, and Systems (ICECS’01)*, pp. 985–989, Sept. 2001.

[27] A. Bhaya and F. Mota, “Equivalence of stability concepts for discrete time-varying systems,” *International Journal of Robust and Nonlinear Control*, vol. 4, pp. 725–740, 1994.