Adaptive Congestion Control in ATM based Networks: Quality of Service and High Utilisation#

Andreas Pitsillides[1] and Jim Lambert[2]

[1] Dept. of Computer Science, University of Cyprus, P.O. Box 537, Nicosia, Cyprus.
e-mail: cspitsil@turing.cs.ucy.ac.cy

[2]Laboratory for Telecommunication Research, School of Biophysical Sciences and Electrical Engineering, Swinburne University of Technology, P. O. Box 218, Hawthorn, Melbourne, 3122, Australia. e-mail: jfl@swin.edu.au

Abstract:

To improve network utilisation and robustness for ATM networks there is increasing evidence that reactive congestion controls are necessary. However the large bandwidth-delay product and the difficulty in traffic modelling inherent in ATM networks makes design of effective reactive controls difficult. In this paper, we use adaptive (feedback and feedforward) predictive control techniques, and despite the (large) propagation delays avoid congestion at high server utilisation while maintaining quality of service at prescribed levels. Our method overcomes the difficulty of the large propagation delays by using adaptive traffic prediction calculated from measurements of the net input traffic at an ATM buffer. Effective prediction is possible due to the high level of correlation present in some VBR traffic. The sensitivity to traffic modelling is addressed by using adaptive (predictive) control.

Bounds on operating conditions are derived analytically, and evaluated using simulation. A comparative evaluation is also presented.

Keywords: ATM networks, Congestion Control, Flow Control, Connection Admission Control (CAC), Adaptive (Feedback and Feedforward) Predictive Control.

1: Introduction

The demand for higher transmission capacity and higher bandwidth to the user by multiservice and multimedia applications, has put Asynchronous Transfer Mode (ATM) based architectures at the forefront of several competing switch technologies. ATM has been designed to support various classes of multimedia traffic with different bit rates and Quality of Service (QoS) requirements. Due to the unpredictable fluctuations and burstiness of traffic flows within multimedia networks congestion can occur frequently. Hence, it is necessary to design appropriate congestion control mechanisms to ensure the promised QoS are met. Designing effective congestion control techniques and developing traffic models for multimedia networks have been difficult because of the nature of the multimedia traffic and future networking scenarios.

In the networking literature several congestion mechanisms have been proposed and their performances have been reported. For tractability, a number of works previously reported base the control policy mainly on analytical steady state queueing models (mostly derived from performance evaluation studies), or on simplistic models (e.g. on-off type models). Furthermore due to the large propagation delay in comparison to the buffer dynamics (high bandwidth-delay product) a large number of publications advocate open-loop preventive control (mainly for B-ISDN) which reside at the network edge and aim to avoid the occurrence of congestion. On the other hand, a number of reactive control schemes have been reported, but the proposed control policy is often heuristically motivated. For example, control methods based on Forward or Backward Explicit Congestion Notification (FECN or BECN), or their variants, and Explicit Rate (ER) based schemes (such as EPRCA and ERICA), have recently become popular for flow control in ATM based networks. Not many studies have been presented which make use of "formal" control theory in the derivation of the control strategy.

The general structure of preventive control frameworks is described in the ITU recommendation I.371 [1]. It consists of Connection Admission Control (CAC) [2], [3], [4], [5], and Usage Parameter Control (UPC) [6]. When a new call request is made, the user is required to inform the network about the traffic characteristics (the 'contract', which contains traffic descriptors such as mean bit rate, peak bit rate, and possibly burstiness of traffic and others) and the desired QoS of the connection. It is then the responsibility of the CAC to decide whether to accept or reject the new connection. It is

---

[#] An earlier version of this paper was presented at INFOCOM'94.

accepted only if the requested QoS can be satisfied without affecting the QoS of existing connections. Once a connection is accepted, the UPC polices the traffic characteristics to ensure that they do not exceed those specified at connection establishment. Such a preventive control framework, however, is not adequate to achieve the objectives of traffic control. Studies reveal that, for very bursty traffic such as LAN-to-LAN interconnection whose peak rate is comparable to the link speed, the achievable utilisation is very small unless detailed knowledge on the traffic characteristics is available [7]. Such knowledge may include burst length, burst and silence length distributions, or even a description of any correlation between successive bursts and silences. In most cases, these traffic descriptors are unknown at connection establishment. Even if they are known, it is inconceivable that they can be accurately policed. Note that the effectiveness of policing units has been questioned for what is seemingly a straightforward task of controlling, or policing, the peak rate [8]. Furthermore, the variability of traffic inside the network becomes independent of the variability of traffic at the network edge (e.g. [9] has shown this to be true for moderate to high network utilisation). Thus, cell clustering within the network may cause congestion which cannot be prevented by network edge preventive controls. Also another main problem of control is the sensitivity of buffer occupancy to uncertainty in traffic modelling. It is well known that open loop CAC schemes are sensitive to either overallocation of connections, or actual excess flow above the agreed parameters. For example, the scheme of [10] (based on cell loss probability) is discussed in [11], where it is shown that: linearisations of the connection acceptance surface should not be used to simplify the CAC algorithm; a very conservative CAC and policing scheme has to be used since only a few percent overallocation, or policing ineptness, will result in a very high probability of loss, at least as predicted by the bufferless approximation model. These arguments suggest that additional controls are necessary to handle congestion. Other notable preventive control schemes have been proposed, such as (open loop) rate based flow control [12], and the Fast Reservation Control Protocol (FRP) [13] which handles burst level congestion by controlling the admission of bursts, but similar arguments can be applied to justify the need for additional reactive controls.

The weakness of open loop ("preventive") control is the assumed existence of an accurate model. Feedback on the other hand can be effective in spite of model inaccuracies and can also compensate for disturbances [14]. A number of feedback based control schemes (derived mainly using intuition) have been proposed for B-ISDN. Examples include: end-to-end window based flow control [15], end-to-end binary feedback [16], network edge rate control [17], credit-based control [18], end-to-end ECN based (forward or backward) flow rate control [19], [20], and Explicit Rate (ER) based, such as EPRCA (Enhanced Proportional Rate Control Algorithm) [21] and ERICA (Explicit Rate Indication for Congestion Avoidance) [22]. (Note that since this paper was written, many new algorithms, as well as variants of existing ones have appeared, see e.g. [23]). Mostly, these schemes use heuristically motivated (simplistic) control laws and (due to the realisation of the complexity of the traffic models) do not model the behaviour of traffic flows and their influence on network behaviour (and as a result the delivered QoS). Additionally, end-to-end controls can suffer from a large bandwidth-delay product.

The main obstacle to using feedback control is the large bandwidth-delay product. Predictive model based control, on the other hand, is expected to handle large bandwidth-delay products by predicting the system behaviour (ahead of the propagation delay) and taking early action (rather than waiting for feedback to start taking corrective action, as for example in ECN based control systems). But these schemes require rather more accurate models than feedback based control. Thus the effectiveness of these schemes relies on whether an accurate enough model of the system behaviour can be found. Several predictive control based schemes have been proposed. Notable examples include [24], [25], [26], [27], [28]. However, most proposed schemes use simplistic models (which may not be accurate enough to predict system behaviour over long horizons) and offer no adaptivity (prediction is based on fixed parameter models). Thus these schemes cannot handle unforeseen or changing traffic or network dynamics. Also relationship with QoS offered to user is not explicitly modelled, hence QoS cannot be explicitly influenced or controlled.

Our work differs significantly from any of the above schemes because: i) it focuses on the dynamics of the net input process (the net input traffic at an ATM buffer), whose time dynamics are much greater than the buffer time dynamics (due to the much smaller timescales that the buffer operates), and are comparable to the propagation delays. (Alternatively stated, the high level of correlation in some traffic enables effective traffic prediction over long intervals, comparable to the propagation delays). Note that there is increasing evidence that feedback based on the network queue alone is too sensitive to the network traffic conditions; (ii) the quality of service required by the users is used in the design of the control scheme; (iii) we consider a controllable (such as ABR) together with an uncontrollable (such as real time constant and variable bit rate) traffic mix competing for the finite buffer and link resources; (iv) we integrate connection admission control with flow rate control, thus taking into account their interactions (v) we use adaptive feedback together with adaptive feedforward. Whilst feedback, even without an accurate model of the system, can compensate for disturbances (e.g. variations in traffic behaviour) it must wait for the feedback from the system to start taking corrective action (i.e. system has already been affected). On the other hand, feedforward can compensate for disturbances before they actually hit the system, but to do that it requires an accurate model of the system behaviour at all times. Combining feedback and feedforward, and offering adaptivity by using on-line modelling techniques (i.e. real time development of system model), intents to address the weaknesses of both schemes and resolve the problem of uncertainty [29] (examples of uncertainty include: model inaccuracies, traffic disturbances, and noisy feedback); and vi) being of the adaptive predictive type the control scheme is able to tolerate fairly long control intervals (hence overhead due to the exchange of control messages is minimised).

In this paper, despite the propagation delays we aim to achieve high utilisation of resources while maintaining quality of service (QoS) in terms of user requirements (cell-loss, cell-delay and cell-delay-variation). We propose to achieve this by integrating Connection Admission Control (CAC) and flow control (generic functions for managing and controlling traffic and congestion in Broadband-ISDN [1]), and by making use of a novel control concept. We aim to use a combined feedback and feedforward adaptive predictive control system to maintain the QoS close to a reference QoS (target QoS value), irrespective of variations in traffic (the disturbance). We define two distinct groups of traffic: controllable (e.g. delay tolerant traffic, such as Available Bit Rate) and uncontrollable (e.g. delay intolerant traffic, such as variable bit rate video). This allows us to introduce the concept of network controllability. Controllability is achieved by simply bounding the uncontrollable traffic, which becomes the sole role of CAC. The controller manipulates the flow of controllable traffic into the network to regulate QoS based on a performance monitor which predicts the p^{th} percentile of the buffer occupancy distribution. Controllability guarantees that the network can be operated efficiently (theoretically at 100% utilisation) and still provide the user with tightly regulated QoS. In other words, we exploit the delay tolerance of a certain group of traffic (controllable traffic) in order to guarantee delay and loss for another group of traffic (the uncontrollable traffic). Note that loss (within the network) for the controllable traffic is also guaranteed. Of course one may also exploit the loss tolerance of certain services, e.g. by supplementing the proposed control strategy with a lower level selective cell discarding policy operating at the buffer (a simple cell discarding policy based on the cell loss priority bit may suffice, but it may be beneficial to investigate more advanced cell discarding policies, e.g. [30]). The supplementary selective discard approach is beyond the scope of this paper; its details and formal study are not considered here; we briefly discuss this approach in the simulation section.

The paper is organised as follows. Section 2 provides some definitions, and introduces the control concept and the calculation of the feedback signal. Section 3 presents the problem formulation and the proposed solution. In section 4, using both analysis and simulation we evaluate the performance of the proposed scheme, and also present a comparative evaluation with four other representative schemes. In section 5 our concluding remarks appear. Finally appendices I and II provide some of the proofs.

2: Preliminaries

Network controllability and the controllable and uncontrollable groups of traffic

We define the concepts of (strict) controllability, controllable traffic and uncontrollable traffic as follows:

> *Definition 1:* A system is **controllable** if its control signal can influence the system behaviour, so that the desired output state of the system can be reached in finite time. Furthermore **strict controllability** is obtained if all desired output states are reachable in finite time.

Note: Our definition is less formal than that in the literature on control systems theory, and is possibly closer to reachability [31].

> *Definition 2:* **Controllable traffic** is the group of traffic that has agreed to have its cell-transmission-rate controlled by a network control system, in accordance with the network state.

Note: Ideal candidates for controllable traffic are any delay tolerant connections such as Available Bit Rate (ABR). No user declaration of traffic parameters is necessary for this group. Cell-loss performance can be guaranteed, but not end-to-end cell-transfer delay, or cell-delay-variation.

> *Definition 3:* **Uncontrollable traffic** is the group of traffic, that has not agreed to have its cell-transmission-rate controlled. It has agreed to limit its peak cell-rate only, as negotiated during connection admission.

Note: The only user declared parameter required for this traffic is the peak rate. The QoS can be guaranteed.

If the uncontrollable traffic is bounded then the network is made controllable, hence an effective congestion control policy is feasible.

Quality of service (QoS)

ATM [32] will offer statistical multiplexing and buffering within the network in order to allow effective use of resources. However buffering and multiplexing leads to cell-delay, cell-delay-variation (CDV), and possibly cell-loss.

The Quality of Service (QoS), as perceived by a network user, depends on cell-loss, cell-transfer-delay, and cell-delay-variation. The QoS target depends on the requirements of the individual connection (or connection mix). The achievement of the QoS target is determined by the service-rate and buffer-space resources allocated to the connection (or connection mix).

Definition 4: The **QoS target** is a set of upper bounds for the *cell-loss, cell-delay* and *cell-delay-variation*. A given QoS target is fully specified by the set

$$\text{QoS target}=\left\{P_{cell-loss}^{\max}, t_{cell-delay}^{\max}, t_{CDV}^{\max}\right\}$$

where $P_{cell-loss}^{\max}$ is the maximum tolerable probability of cell-loss (at the cell-buffer), $t_{cell-delay}^{\max}$ is the maximum tolerable cell-queueing-delay (note that all fixed delays, as e.g. propagation delay, packetisation and depacketisation delay, are subtracted from the maximum tolerable cell-transfer-delay provided by the user, to obtain the maximum tolerable cell-queuing delay), and $t_{CDV}^{\max}$ is the maximum tolerable cell-queuing-delay-variation.

Definition 5: The **QoS reference** is the number of buffer places expected to meet the QoS target. Note: A given QoS target is maintained by appropriate choice of the values of the cell-overflow constant ($E_{overflow}$) and the QoS reference value (y^{ref}), where $E_{overflow}$ is defined as a constant specifying the desired value of the expected cell-overflow (above y^{ref}) probability, and y^{ref} is given in terms of cell-buffer-places (in number of cells). The control system aims to maintain the cell-overflow probability above y^{ref} equal to $E_{overflow}$, the specified cell-overflow constant. Observe that the physical buffer-space may be greater than y^{ref}. (A visual representation of the control aim and the desired controlled buffer distribution are shown in Figure 1.) As a consequence of shaping the controlled buffer distribution (to equal the desired, see Figure 1), the cell-loss, cell-delay, and cell-delay-variation (QoS parameters) are maintained at prescribed levels. Of course, for controllable traffic the cell-delay and cell-delay-variation must include the contribution of *Q*, the queue of the controlled sources (see Figure 1 and Figure **2**).

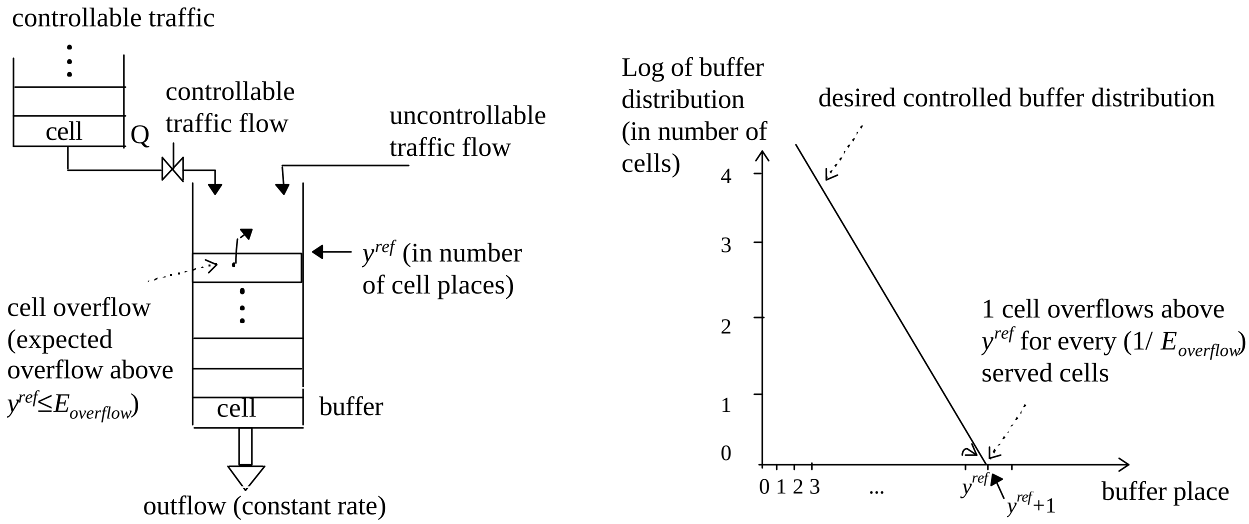

Figure 1. Visualisation of control aim and the desired controlled buffer distribution

The QoS target parameters (i.e. $\left\{P^{\max}_{cell-loss}, t^{\max}_{cell-delay}, t^{\max}_{CDV}\right\}$) can be mapped directly onto $E_{overflow}$ and y^{ref} as follows.

$$E_{overflow} = P^{\max}_{cell-loss}, \text{ and } y^{ref} = \left\lfloor \frac{\min(t^{\max}_{cell-delay}, t^{\max}_{CDV})}{t_{celltime}} \right\rfloor$$

where $t_{celltime}$ is the time taken to service a cell, i.e. it is equal to one celltime (e.g. for a 155 Mbit/sec link and a 53 byte cell, $t_{celltime}$ is about 2.6 μsec), and $\lfloor x \rfloor$ is the next integer toward $-\infty$.

Example: Assume that for a specific OD connection mix the QoS target is specified by the set {10^{-9}, 75 μsec, 50μsec} and that the OD path spans a single node served at a rate of 155 Mbit/sec. The value of $E_{overflow} = 10^{-9}$, and $y^{ref} = 19$ cell-buffer-places [expected to accommodate $(1 - E_{overflow}) \times 100\%$ of the served cells]. ? ? ?

Control concept

Our control concept is shown in Figure 2 (for clarity only one controllable source is shown). By manipulating the flow of controllable traffic, the control system aims to maintain the QoS feedback signal close to the QoS reference value, irrespective of variations in the bounded uncontrollable traffic (the disturbance). This is feasible since the network is controllable, therefore the desired output state can be reached. Thus the network can be operated efficiently (theoretically at 100% utilisation) and still provide the user with a tightly controlled QoS as determined by the reference value. The formal role of CAC is to preserve the bound on uncontrollable traffic and hence enforce controllability. In practice, the CAC policy may be more aggressive [33], as e.g. by using the "effective bandwidth" [5], but then (strict) controllability is not maintained, since one cannot presume that all users will have accurate knowledge of their parameters at connection setup time (e.g. the triplet [peak rate, mean rate, burst duration] required for the calculation of the effective bandwidth [5]) and that effective policing of possible violations from the declared parameters can be guaranteed [6], [7], [8]. Additionally one cannot assume that the effective bandwidth is always conservative (for example see [34], where it is shown that: the underlying assumptions of the approximations are not always correct; and that for less bursty than Poisson sources, the number of sources that can be supported are overestimated).

Remarks:

1) once an uncontrollable traffic connection is accepted into the network its transmission-rate cannot be regulated (controlled) by the network controls.

2) The uncontrollable traffic connections can transmit at any rate (in a bursty or otherwise fashion) limited only by their peak rate constraint.

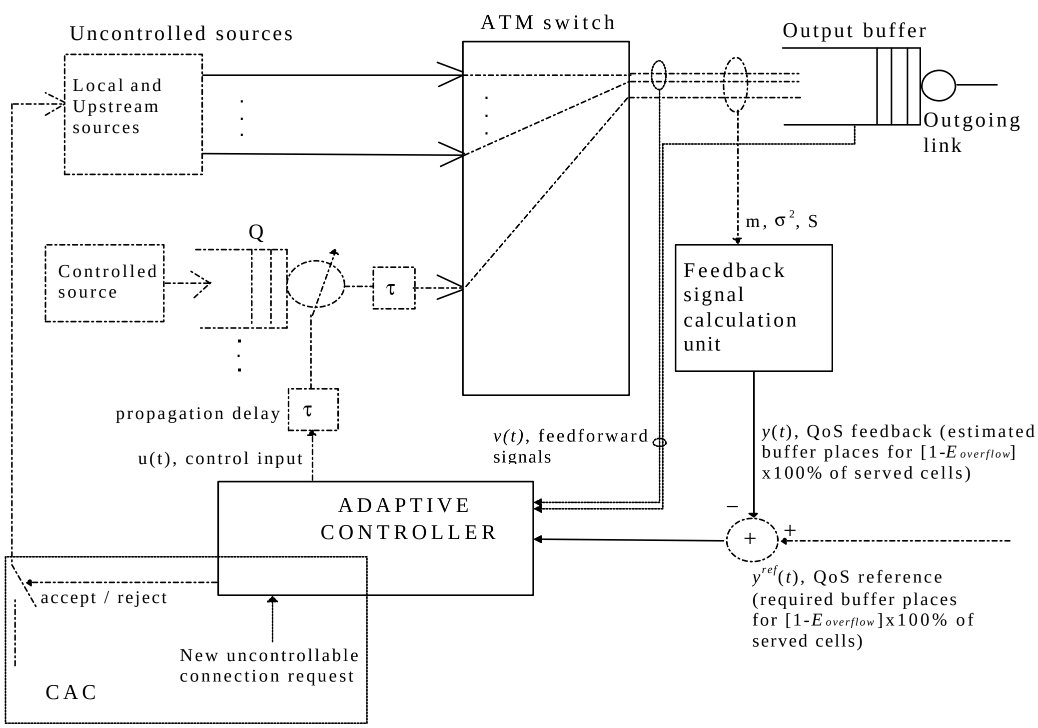

Figure 2. The control concept

A single adaptive regulator is shown, located at a node with direct responsibility for one outgoing link. We envisage one controller per outgoing link. Various control options are available, for example:

1) The controller calculates the maximum total cell rate for controllable traffic (i.e. the expected leftover bandwidth, over the duration of the next control interval), taking into account a representative value (e.g. it could be an “upper”

bound) of the propagation delay and QoS target for all sources using the switch. The maximum total cell-rate for controllable traffic can then be shared by all controllable (e.g. ABR) sources using this link in a "fair" manner. The design of a suitable algorithm that allocates this available cell-rate among all competing sources, taking into account the issue of fairness, is beyond the scope of the current paper; the interested reader can refer to the vast literature in the field, e.g. [35], [36], [37].

2) Exert direct control over all controllable sources, taking into account their individual propagation delay and QoS target value. The controllable buffers Q can be located at the premises of the customer.
3) A compromise between schemes 1 and 2, where the allowed cell rate is calculated for a few representative propagation delay and QoS target zones (or domains).

Note that this switch can be viewed as an Explicit Down Switch (EDS), as proposed by the ATM Forum [21], capable of calculating the Explicit Rate (ER) for each controllable source. The implementation of option 1 is simpler than that of option 2. Option 2 necessitates a change in the control structure for every connection or disconnection affecting this switch (i.e. the number of calculated control inputs changes), whereas for option 1 the controller structure does not change (unless the maximum propagation delay, or the QoS target, used to calculate the prediction is not representative of the current connection mix). Option 3 is a compromise between the two schemes, where a few representative propagation delay and QoS target zones are used to calculate the allowed cell rate (for controllable traffic) for each zone. The controller structure need not change as frequently as that for option 2. An investigation of the "best" control option is beyond the scope of this paper.

For simplicity in this paper we consider: a single controllable queue Q located at a distance equivalent to a propagation delay of τ from the switch; Q buffer size is infinite; and that the controllable source is saturated. Control is exercised directly on the outgoing rate of queue Q.

Dynamic system model

To capture the dynamic behaviour (required for effective network control [38], [39]) we use the integrating form of the ARMAX model which is suitable for adaptive control.

Mathematical formulation: Define: y, the feedback of estimated QoS (see feedback discussion below); y^{ref}, the desired reference value for y; u, the calculated control effort (i.e. allowed cell rate over the next control interval or allowed flow rate from controllable sources); v_j, the feedforward signal, where v_1 is the uncontrollable flow and v_2 the buffer length (averaged over a control interval); and ε, unmeasurable disturbances and equation errors of unspecified character.

A general ARMAX model with multiple feedforward signals, and multiple control signals is formulated as

$$y(t)+a_1 y(t-1)+\ldots+a_{na} y(t-na)=\varepsilon_i\, b_0^i u^i(t-k_i)+b_1^i u^i(t-k_i-1)+\ldots+b_{nb^i}^i u^i(t-k_i-nb^i)+$$

$$\varepsilon_j\, g_0^j n^j(t-k_j)+g_2^j n^j(t-k_j-1)+\ldots+g_{ng^j}^j n^j(t-k_j-ng^j)$$

and in shorthand notation as (for simplicity only a single control signal is shown)

$$A\left(q^{-1}\right)\Delta y(t)=B\left(q^{-1}\right)\Delta u(t-k)+\underset{j}{\varepsilon}\ \Gamma_j\left(q^{-1}\right)\Delta v_j(t-k_j)+e(t) \qquad (1)$$

where: q^{-1} is the backward shift (or delay) operator defined as $q^{-1}y(t)\ ?\ y(t-1)$; $\Delta=1-q^{-1}$ is the differencing operator, i.e. $\Delta y(t)=y(t)-y(t-1)$; $A(q^{-1})$, $B(q^{-1})$, and $\Gamma_j(q^{-1})$ are polynomials in q^{-1} of appropriate orders *na*, *nb*, and ng_j, respectively (i.e. $A(q^{-1})=1+a_1q^{-1}+\ldots+a_{na}q^{-na}$, $B(q^{-1})=b_0+b_1q^{-1}+\ldots+b_{nb}q^{-nb}$, $\Gamma_j(q^{-1})=g_0^j+g_1^jq^{-1}+\ldots+g_{ng_j}^jq^{-ng_j}$); t is the discrete sample time in increments of the sample period T_s; and k (k_j) is the delay experienced at the output due to the control input (feedforward).

Feedback of the network performance

The feedback "sensor" provides a measure of performance, using the methodology of Addie and Zukerman, see for example [40], [41] and brief discussion below. (An alternative on-line feedback "sensor" is introduced by [42]).

This feedback sensor offers several features required for on-line control use: it provides a measure of the network QoS; it is accurate enough for the purpose intended, it proves to be robust for deviations from the assumed model; it has predictive abilities; computationally it is not overly complex; and it allows arbitrary choice of interval length, which allows matching to the system dynamics.

Addie and Zukerman in [40] consider a single server queuing system where the net input arrival process to the queue is a stationary Gaussian process $Y_n=m+\underset{k=0}{\overset{\infty}{\varepsilon}} a_k e_{n-k}$ (see Appendix II for a definition of the variables). Their analysis there leads to explicit results for the statistics of the unfinished work distribution in the infinite buffer case (the finite buffer case is considered in [41]). These results are based on a second order approximation and are expressed in terms of three parameters of the net input process: the mean, m, the variance, s^2, and the autocovariance sum, $S = \sum_{k=1}^{\infty} Cov(Y_n, Y_{n+k})$, which are estimated from the net input arrival process. The net input arrival process is measured via its count process Y_n which keeps account of the number of cells arriving in consecutive time periods n (sampling period T_m). The analysis in [40] is based on the fact that, for the case of an infinite buffer, under quite general conditions, the tail of the stationary complementary waiting time distribution of Semi-Markov queues with a finite underlying state space is of the form $ce^{s^* t}$, where s^* is obtainable as the negative real root of a certain functional equation which lies closest to the origin. For the case of Gaussian queues, they derived an exact analytic solution in closed form of the functional equation for s^* and an accurate approximation for the coefficient c, which they designate as $\tilde{c}$. These are defined in [40] as:

$$s^* = \frac{2m}{s^2 + 2S} \text{ and } \tilde{c} = \frac{erfc\left(\frac{-m}{s\sqrt{2}}\right) - e^{\frac{s^{*2} s^2}{2} - ms^*} erfc\left(\frac{s^2 s^* - m}{s\sqrt{2}}\right)}{s^* s\sqrt{2}\left(-u_1 erfc(u_1) + \frac{1}{\sqrt{p}} e^{-u_1^2}\right)}$$

where $u_1 = \frac{s^* s}{2\sqrt{2}}$ and $erfc$ is the complementary error function

The feedback signal $y(t)$ calculates the expected number of queue places required to accommodate $p\%$ of the cell traffic (the pth percentile of the buffer distribution) using $\tilde{c}$ and s^*, calculated as shown above from the triplet (m, s^2, S) estimated from the measured data.

For $\tilde{c} \le 1 - \frac{p}{100}$ $y(t)$=0, otherwise

$$y(t) = \frac{1}{s^*} ln\left(\frac{1 - \frac{p}{100}}{\tilde{c}}\right). \qquad (2)$$

Example: To calculate the expected number of queue places required to accommodate 99.9999999% of the cells passing through the system, we set p=(1-$E_{overflow}$) ? 100% = (1-10^{-9})?100%, i.e. one cell is expected to overflow above $y(t)$ every 10^9 cells passing through. Note that this does not imply that the $E_{overflow}$ cells will be lost. It simply means that cell places above the reference value are expected to be occupied. Also, consider a system whose net input process is a stationary Gaussian process with a utilisation ρ=0.7 and variance s^2=50. For various degrees of correlation of the net input process, as captured by the autocorrelation sum S, the calculated queue places $y(t)$ are shown in Table 1.

Autocorrelation sum, *S*	0	50	100	200
$y(t)$, calculated queue places required to accommodate *p*% of the traffic	54	155	258	465

Table 1. Effect of correlation sum on number of places required to accommodate p% of traffic through buffer

? ? ?

Also using the Addie-Zukerman methodology the stationary utilisation, and for a finite buffer, the expected loss and the probability of loss at steady state can be calculated (see [40], [41]).

3: Problem formulation and solution

We solve the CAC and flow control problem using system identification and predictive adaptive feedback and feedforward techniques. The basic idea of adaptive control is to adapt the control law in response to changes in the system dynamics or environment [14], [29], [31]. The adaptation techniques are based on input/output data and use on line system identification techniques to infer a model of the system. Hence these techniques do not necessitate any prior assumptions with respect to the model behaviour and/or statistical parameters.

The adaptive connection admission and flow control (ACFC) algorithm

The solution of the combined CAC and flow control problem follows readily once the flow control problem is solved. These are presented as phase 1 and phase 2 below.

Phase 1: Flow control

A predictive adaptive control algorithm of the *k-step ahead prediction* (simplest form of the celebrated long range predictive control type algorithm) is used, together with an implicit approach. This enables the derivation of a simple (computationally) control algorithm featuring a Recursive Least Squares (RLS) identification algorithm. Using the system model (1) we derive [33] the integrating k-step-ahead prediction model (in a robustified form)

$$A_m(q^{-1})y(t+k) - A_m(1)y(t) = A\,(q^{-1})\Delta y(t) + B\,(q^{-1})\Delta u(t) + \varepsilon_j G_j(q^{-1})\,\Delta v_j(t) \qquad (3)$$

where: $A_m(q^{-1}) = a_{m,0} + a_{m,1}q^{-1} + ... + a_{m,nm}q^{-m,nm}$ is a given model polynomial which shapes the output in response to the reference $y^{ref}(t)$ as $A_m(q^{-1})y(t) = A_m(1)y^{ref}(t-k)$. The coefficients (parameters) of the polynomials $A\,(q^{-1})$, $B\,(q^{-1})$, and $G_j(q^{-1})$ [$A(q^{-1}) = 1 + a_1q^{-1} + ... + a_{na}q^{-na}$, $B(q^{-1}) = b_0 + b_1q^{-1} + ... + b_{nb}q^{-nb}$, $G_j(q^{-1}) = g_{0,j} + g_{1,j}\,q^{-1} ..., + g_{ngj,j}\,q^{-ngj,j}$) are identified directly from system data using the RLS identification algorithm [31]. Thus the controller coefficients a_1, ..., a_{na}, b_0, ... , b_{nb}, $g_{0,1}$, ... , $g_{ngj,j}$ are identified directly (i.e. we use direct or implicit adaptive control).

We then consider a simple quadratic cost functional

$$\underset{\Delta u}{Min}\left\{\left[A_m(q^{-1})y(t+k) - A_m(1)y^{ref}(t)\right]^2 + l_o\left(\Delta u(t)\right)^2\right\} \tag{4}$$

where $l_o > 0$ is the control penalty factor which trades control effort for output precision. This functional is representative of the so called "extended minimum variance" cost functional [43].

By minimising (4) on the basis of (3), we obtain the control effort [33]

$$\Delta u(t) = \frac{b_o}{b_o^2 + l_o}\{A_m(1)[y^{ref}(t) - y(t)] - A(q^{-1})\Delta y(t) - (B(q^{-1}) - b_o)\Delta u(t) - \varepsilon_j G_j(q^{-1})\Delta v_j(t)\} \tag{5}$$

This control law is clearly a combination of feedback from the output signal $y(t)$ and feedforward from the measured disturbances $v_j(t)$. It also allows the tradeoff between input and output model following variance, by using the control penalty factor l_o. Observe that the output follows a setpoint via the model following polynomial $A_m(q^{-1})$. Furthermore this control law allows us to handle systems in which $A(q^{-1})$ and $B(q^{-1})$ have common roots on the unit circle. This allows us to treat uncontrollable disturbances in the model.

Controller design:

Selection of design variables: The sampling period T_s and the controller order (na, nb, ng_j) are selected using basic knowledge of the dynamic behaviour of the uncontrolled system, which may be obtained from a theoretical understanding of the system or simple experimentation. The sampling period can be selected fairly long (in comparison to digital implementations of control algorithms, such as the PID type) because of the type of algorithm [14, 29, 31]. Short sampling periods may make the controller overambitious (also increasing the control overhead), whereas longer sampling periods will affect the transient response (i.e. settling time and buffer overshoot may be affected due to the initiation of a new connection, or the disconnection of an existing one), and may not be able to handle fast acting disturbances (i.e. rapid unforeseen changes in the uncontrollable connection mix). The choice of the controller order is not very crucial as the controller (na, nb, ng_j) is of the implicit extended minimum variance type. The control horizon k is selected by the user (in multiples of the sampling period) from knowledge of the propagation delay (equivalently distance) between the controllable source and the switch. It must be selected such that k ?T_s is greater than the propagation delay, i.e. our prediction must extend beyond the delay, so that predictive action can be taken in time. For example, considering a source located d km from the switch transmitting at 155 Mbit/sec over a fibre (equivalently

τ=2?5?d μsec round trip propagation delay) then k^{3} ?(2?5?d)/ (T_s ?2.6)? (i.e. it is selected as greater or equal to the next greatest integer toward infinity, given by the ratio of the round trip propagation delay, i.e. 2?5?d μsec, to the sampling rate T_s, given in celltimes, multiplied by the time it takes to process a cell, i.e. 2.6 μsec at 155 Mbit/sec). The rest of the controller design variables [i.e. the model following dynamics $A_m(q^{-1})$ and the penalty factor l_o] influence the behaviour of the controlled system. Their selection is based on easily understood performance values but may require some experimentation. For example $A_m(q^{-1})=1-0.7q^{-1}$ provides a more cautious approach in achieving the reference value than $A_m(q^{-1})=1$, and $\lambda_o=0.75$ provides a tighter response than $\lambda_o=0.25$, i.e. trades off output precision for control effort variation (see [33]). Note that the selection of y^{ref} is based on the desired QoS; see discussion in section 2.

Implementation phase: At each sample time t? T_s the adaptive flow control algorithm requires the following steps:

step 1: control coefficient estimation using the prediction model (3), and standard RLS.

step 2: control effort calculation (5) substituting the unknown coefficients of the controller with their estimates from step 1.

Remark: As the controller part (equation 5) features a simple recursive filter (i.e. sum of product terms), it can be easily implemented in real time (e.g. using Digital Signal Processing hardware) or it can even be incorporated in silicon. If required, the RLS identification part of the algorithm can run as a background task at a much slower time scale.

Phase 2: CAC

CAC can be viewed as a background task working with the flow control scheme to bound uncontrollable traffic. Acceptance policy for controllable traffic is a separate matter not considered here. Note that, as long as in the long run the demand is satisfied, within the permissible buffer and time constraints, then there is no restriction in the amount of admitted controllable traffic. For uncontrollable traffic the CAC scheme compares the peak rate of a new uncontrollable connection request with the uncommitted peak rate capacity (the link rate less the sum of the declared peak rates of all existing uncontrollable connections). Note, as discussed earlier, more aggressive CAC strategies (such as admitting uncontrollable traffic using an estimate of their "effective bandwidth" [5]) can be pursued, but then strict controllability is not maintained [33].

4: Performance evaluation

Analysis

Under the assumptions of: convergence of the adaptive regulator (global convergence has been proven under certain conditions for this class of algorithm [33, 44]; an outline of the proof is given in Appendix I); network controllability (ensured by the CAC strategy); saturation of controllable source (only required for the proof of unity utilisation); and consistency of the feedback signal, the ACFC algorithm possesses the following strong properties (see Appendix II):

- The long term network utilisation is equal to unity.
- The long term network stability is guaranteed.
- The long term QoS for uncontrollable sources is guaranteed (with a probability equal to $1 - E_{overflow}$) to lie below bounds determined by the value of $E_{overflow}$ and the reference value. The bounded components are: end-to-end delay; cell-loss; and cell-delay-variation.
- The long term losses for the controllable group of traffic are also bounded, at the same value as for the uncontrollable sources.

Additionally, based on our simulative experience, the algorithm has demonstrated:

- Robustness against traffic uncertainties and connections and disconnections, without sacrificing network throughput
- Tolerance of fairly long propagation delays.

Simulation

The cell-level simulation consists of an ATM switch with multiple input connections, no internal blocking, and one outgoing port with a cell-buffer of 100 cell places (assuming completely partitioned buffers among the output links).

Only one outgoing link is simulated, since there is no interaction between any of the outgoing links. Connections at the input of the switch are multiplexed and routed to the outgoing link. If there are no cells in the cell buffer of the outgoing link an arriving cell is served immediately, otherwise it is placed in the buffer. Note that a cell has fixed duration and it requires a unit of time for transmission. One unassigned (empty) or assigned (non empty) cell is always transmitted every unit of time. Thus we divide the time axis into celltimes. Each connection is simulated at the entry to the switch as a stream of assigned cells spaced apart by an amount of unassigned cells. The spacing between assigned cells depends on the type of connection. Assuming a line rate of 155 Mbit/sec the cell stream generation is as follows:

- voice at 64 Kbits/sec: 171 cells are generated (at a constant rate) every second. Equivalently one cell is assigned for every 2189 unassigned. The starting point of each voice connection is randomised.
- data: The on period of the data connection is randomly selected from an exponential distribution, with the off period chosen to give us the desired burstiness. The on-off period is chosen small to allow multiple on-off transitions during the simulation run. During the on period cells are transmitted at a constant peak rate (e.g. if the peak rate is 10 Mbits/sec, the cell spacing during on period is 14, i.e. out of every 14 cells one is assigned), and during the off period no cells are transmitted. The starting point of each data connection is randomised.
- video: The first order AR model of [45] (appropriately scaled and hard limited) is used to synthesise the variable bit rate video. The cell rate for each new "frame" (for simulative convenience the "frame" size is chosen small, in the region of 100 celltimes; could represent macroblocks) is selected by recursively evaluating the AR model, and then used for the duration of the "frame" to pace the cells into the network. The starting point of each video connection is randomised.

The traffic load comprises speech, data and variable rate video (synthesised as above), with random connections and disconnections (to simulate the CAC acting in the background), and one controlled source, as shown below.

Uncontrollable traffic mix:

170 voice connections	(multiplexed into 6 ATM ports) at 64 Kbit/sec per connection	
3 video connections [#1]	peak rate=25 Mb/s	mean rate=16 Mb/s per connection
2 video connections [#1]	peak rate=15 Mb/s	mean rate=7 Mb/s per connection
2 data connections	peak rate=10 Mb/s	mean rate=1 and 9 Mb/s
3 data connections	peak rate=5 Mb/s	mean rate=0.5, 2.5 and 4.5 Mb/s
2 data connections	peak rate=2 Mb/s	mean rate=0.1 and 1.5 Mb/s

Controllable traffic:

[#1] For simulation case D) these video sources are replaced by a single high variability source, see later.

1 controllable queue	source is saturated, i.e. there is an infinite supply of cells to the queue at all times. It is located in the customer premises, at a distance equivalent to τ celltimes propagation delay.

Strict controllability is enforced by keeping the peak rate of the uncontrollable part of the connection mix at or below the link rate (i.e. the peak rate of the resultant uncontrollable part of the connection mix never exceeds 155 Mbit/sec). The mean rate of the uncontrollable traffic mix is 92 Mbit/sec. It represents a utilisation of about 60%.

The controller sampling period T_s is 620 celltimes, (about six times longer than the duration of the "macroblock" used for the generation of video in this simulation runs) and the sampling period used for the calculation of the feedback T_m (i.e. sampling interval for the estimation of m, s^2, S) is 30 celltimes (about one third of the buffer length). The control horizon k is set at multiples of the sampling period (tolerating a round trip delay of up to k?620 celltimes, i.e. the buffer of the controllable source can be located up to k?310 celltimes away, or k?170 km at 155 Mb/sec).

Four simulation cases are considered here: Case A) and Case B) are designed to exhibit the performance of the proposed scheme in the presence of small (k=1) and very large (k=9) propagation delay; and Case C) and Case D) offer a comparative evaluation of the proposed scheme with four other (representative) control strategies. Note that Case D) considers the case of high variability (burstiness) in the uncontrollable traffic mix.

Case A: The reference is set at y^{ref} =25 cell places to represent a link with a small buffer capacity and $E_{overflow}$ is set to 10^{-9} (even though statistically we cannot claim much, due to the requirements of the run lengths to be in excess of a number of multiples of 10^9, short run lengths still allow us to get a glimpse of the system performance). The control horizon k is set at one (i.e. the buffer of the controllable source can be located up to 310 celltimes away, or 170 km at 155 Mb/sec, or equivalently a round trip delay of up to 340 kms can be tolerated). The control loop then regulates controllable traffic such that only 1 in 10^9 of the served cells is expected to occupy a place above the 25th. The sum of the peak rates of the uncontrollable sources varies during the simulation run, but is always less than or equal to the link server rate of 155Mb/sec. A total of 20 runs is recorded, each run 1/4 second long (approximately 90000 celltimes each run). Figure 3 shows the buffer distribution (in logarithmic scale) for 20 runs. It shows that the controlled buffer distribution for 19 out of the 20 runs is as desired (see Figure 1), and that the violating buffer distribution is still controlled, but it exceeds the reference. The calculated Probability of loss for a finite buffer of 100 places is shown in

Figure 4 (on log scale). It is worth pointing out that over the length of the simulation run of about 1.8 million celltimes the 10^{-9} limit is violated on one occasion, where it reaches $1?10^{-7}$. The time evolution of the uncontrollable and controllable traffic is shown in Figure 6. The mean values of the uncontrollable and controllable traffic are 92 Mb/s and 43 Mb/s respectively. This gives an average utilisation of about 0.87 (the time evolution of the calculated utilisation is shown in Figure 7) compared with a utilisation of 0.61 based on admission on peak rate alone–a 43 % increase. (Note: connection admission based on the peak rate is simulated by setting the controllable traffic to zero and using the identical pseudorandom uncontrollable traffic mix sequence as above–which is admitted on the basis of its peak rate, and is always equal to or less than the line rate). Furthermore, Figure 5 (time evolution of feedback signal) shows that the average behaviour of the system is as predicted by theory, i.e. $\lim_{t\to\infty}\left[y(t)-y^{ref}(t)\right]=0$ (its mean value is equal to 24.97, compared to a reference value of y^{ref}=25). Note that only about 100 cells (from all connections) out of about 1.8 million violate the desired reference, and the worst queuing delay for any one cell is 39?2.6μsec. It is worth pointing out that this performance can be improved even further by supplementing the current strategy with a selective discarding policy of tagged cells–this will aid in ensuring that the buffer distribution is kept within the desired.

Figure 8 to Figure 10 show the time evolution of the identified control coefficients (parameters), where it can be seen that they adapt their values to reflect the changing network conditions. Figure 8 shows the time evolution of the control coefficient α_1 (for feedback signal). Figure 9 displays the coefficient of the "control gain" $\dfrac{b_o}{b_o^2 + l_o}$ (see control law, equation 5), where it adapts to values between 0.1 to 1 (i.e. a tenfold increase or decrease in the "gain") depending on the state of the network. Figure 10 shows the adaptability of $g_{0,1}$, the feedforward signal control coefficient, and thus indirectly displays the influence of the feedforward signal on the calculation of the new allowed cell-rate. The time evolution of the other feedforward coefficients is similar to the one shown in Figure 10, and therefore omitted.

Remarks: (i) The cell delay and cell delay variation for the uncontrollable sources can be kept close to any desired value by appropriate choice of y^{ref}.

(ii) The throughput can be improved further by increasing the value of the reference [33].

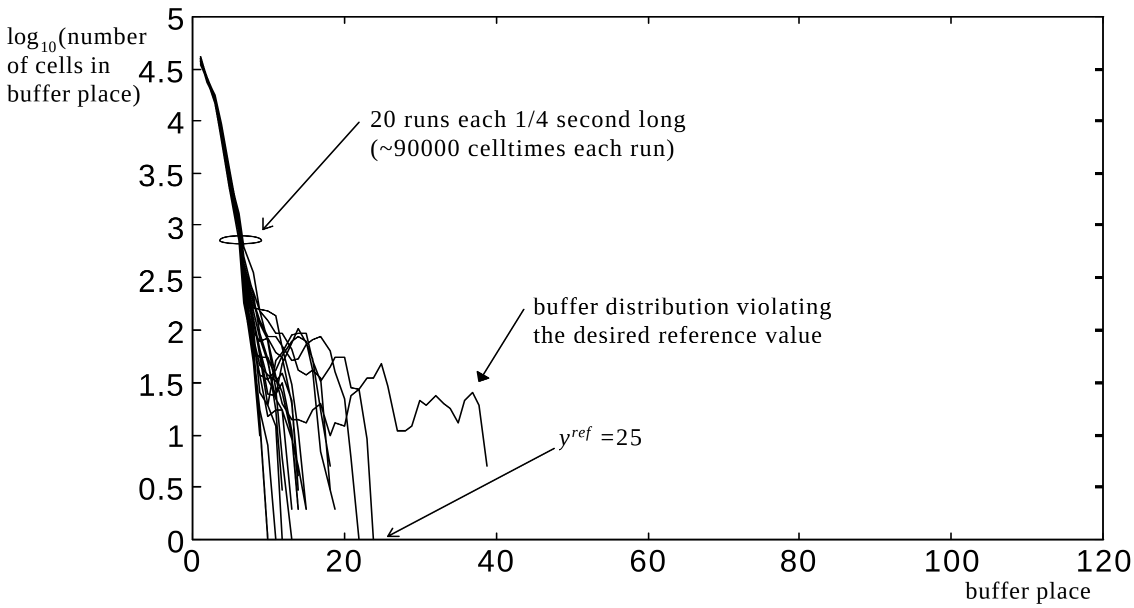

Figure 3. The buffer distribution (in logarithmic scale) for *k*=1 (round trip delay ~ 320 kms) for 20 runs

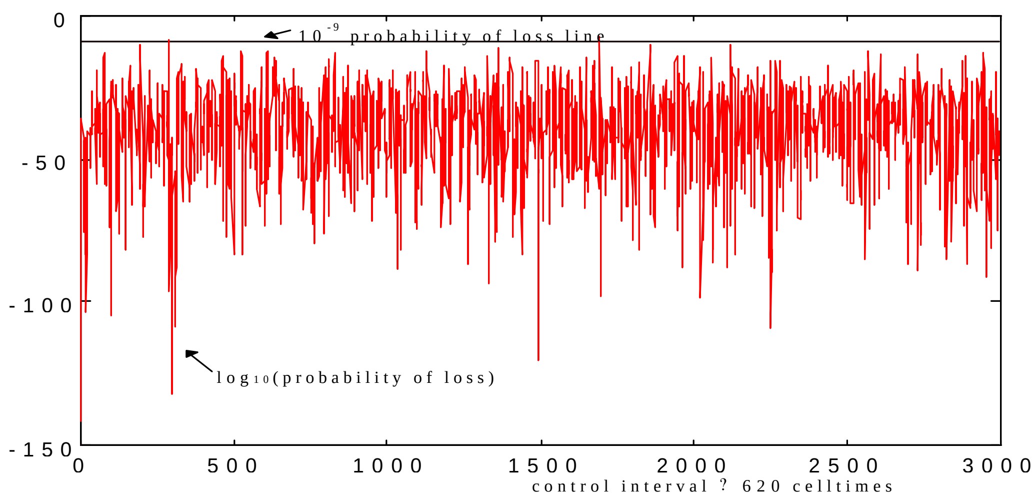

Figure 4 Calculated Probability of loss: typical time evolution for a finite 100 cell places buffer

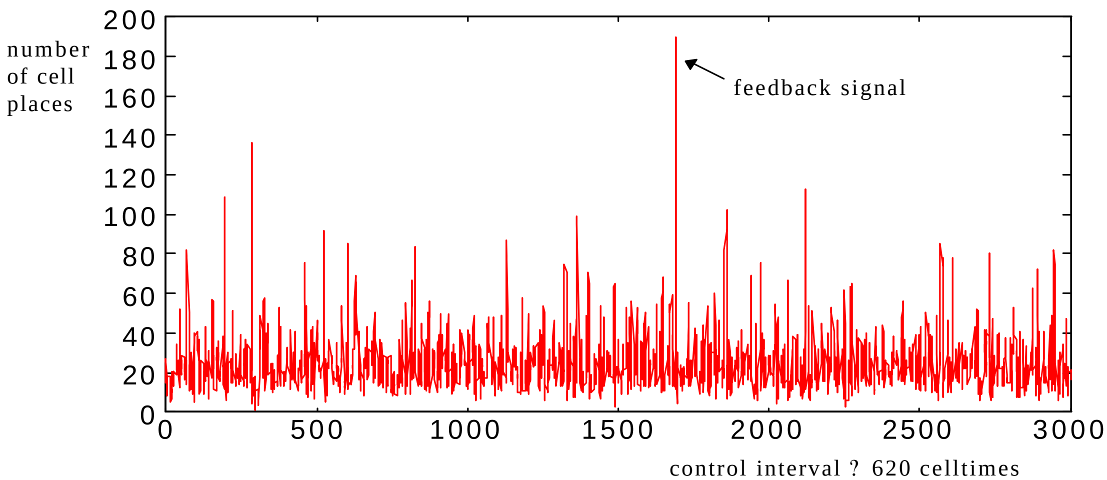

Figure 5. Feedback signal: time evolution for typical simulation run of ~1.8 million celltimes

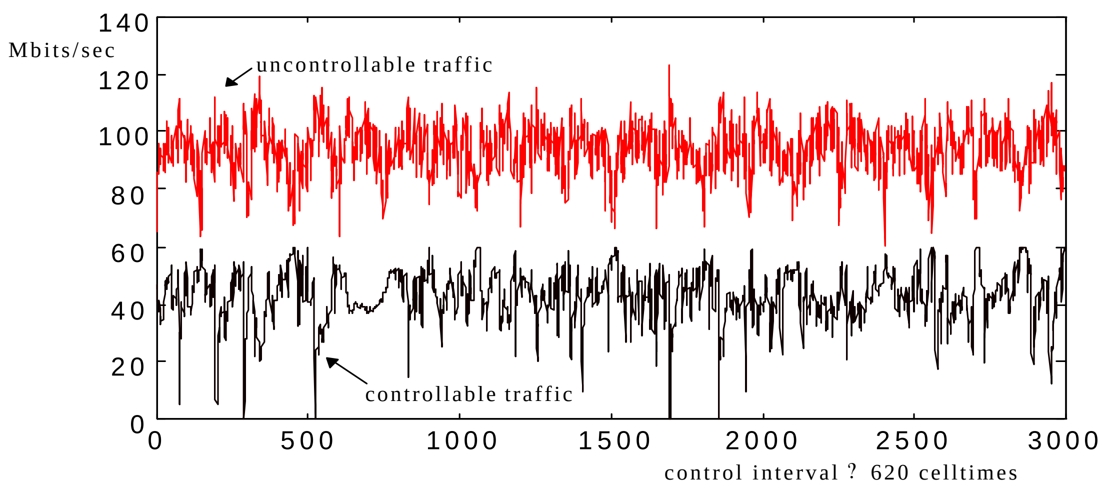

Figure 6. Controllable and uncontrollable traffic: time evolution for typical simulation run of ~1.8 million celltimes

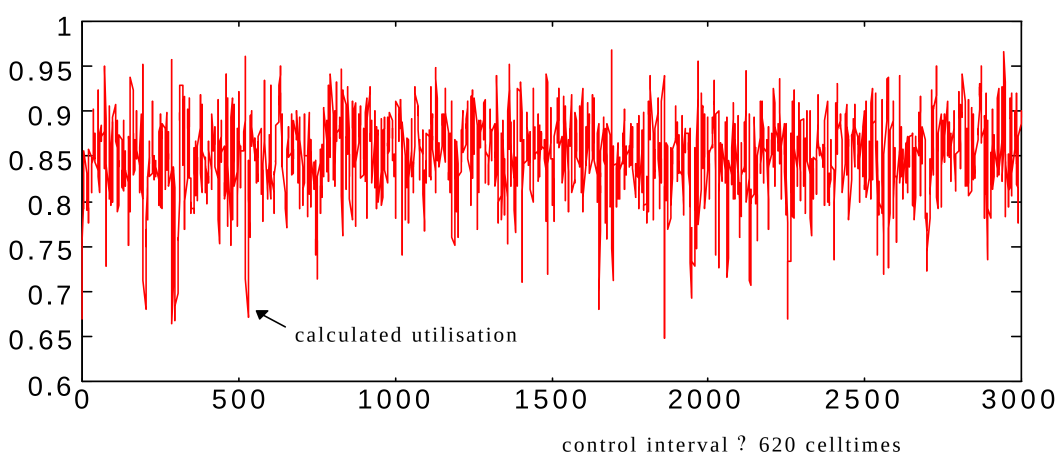

Figure 7 Calculated utilisation: time evolution for typical simulation run of ~1.8 million celltimes

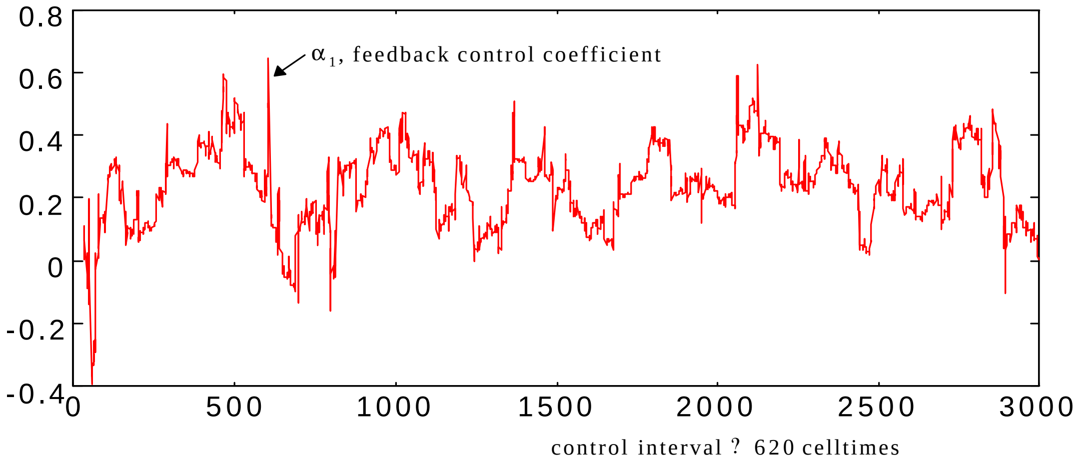

Figure 8. Control coefficient a $_1$ (for feedback): time evolution for typical simulation run of ~1.8 million celltimes

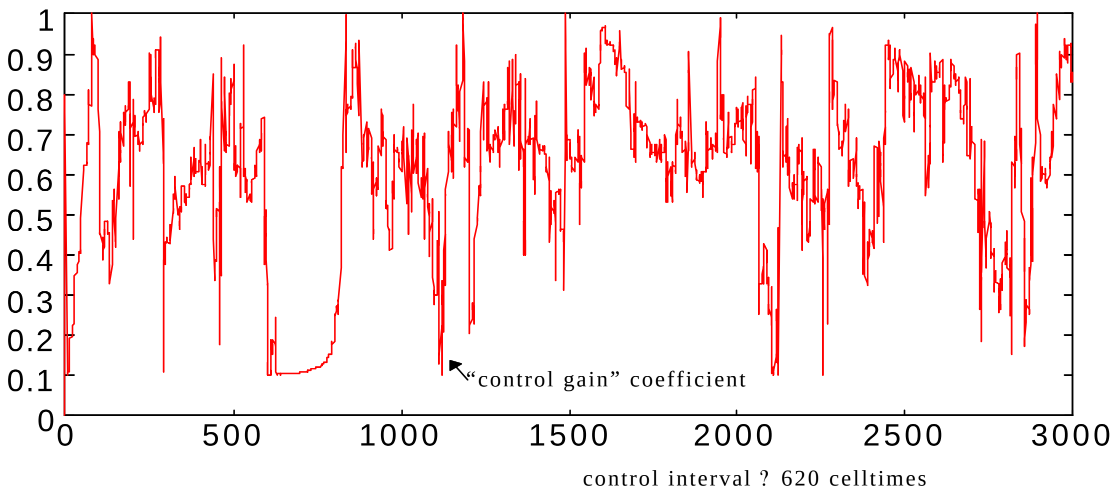

Figure 9. "Control gain" coefficient (see equation 5): time evolution for typical simulation run of ~1.8 million celltimes

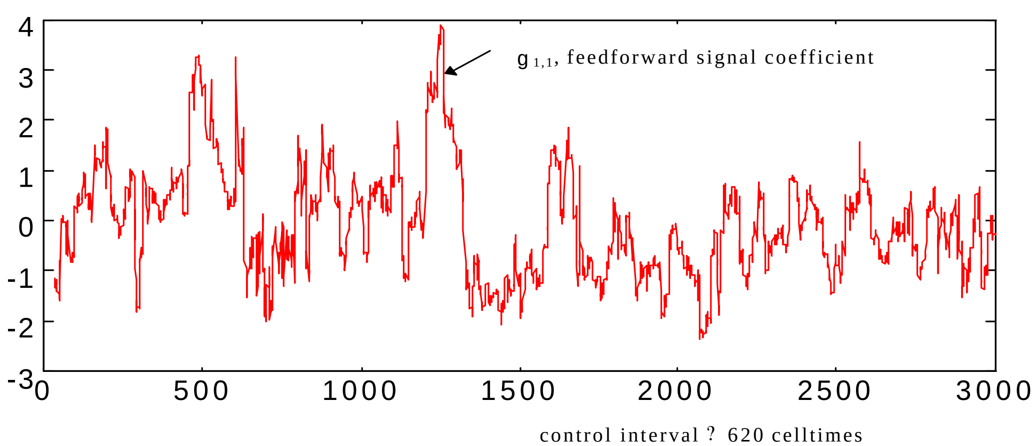

Figure 10. Coefficient $g_{1,1}$ (for feedforward) : time evolution for typical simulation run of ~1.8 million celltimes

Case B): uses same simulation parameters and pseudorandom sequences for uncontrollable traffic mix as Case A), however controllable buffer is placed at about 1600 km from ATM switch. The control horizon k is set at nine (accommodating a round trip delay of up to 9?620 celltimes, i.e. 3500km @155Mbit/sec). Figure 11 shows the tolerance of the control scheme to large increases in the propagation delay. A nine fold increase in the propagation delay for Case B), in comparison to Case A), is accompanied by only a twofold increase in the control ineptness (the maximum buffer place occupied in this case is the 79th, as compared to the 39th for Case B)–probably an acceptable tradeoff. Once again an appropriate cell discarding mechanism may be used to supplement the adaptive policy that will maintain the buffer distribution closer to the desired levels. Note that the utilisation remains at about 0.85. Figure 12 shows the calculated Probability of loss (on logarithmic scale) together with the 10^{-9} Probability of loss line. It is worth pointing out, that even with such a large increase in the propagation delay, there are only a very small number (tens) of momentary violations of the 10^{-9} line. At the very worst the calculated Probability of loss reaches 0.3, as compared to 10^{-7} for Case A).

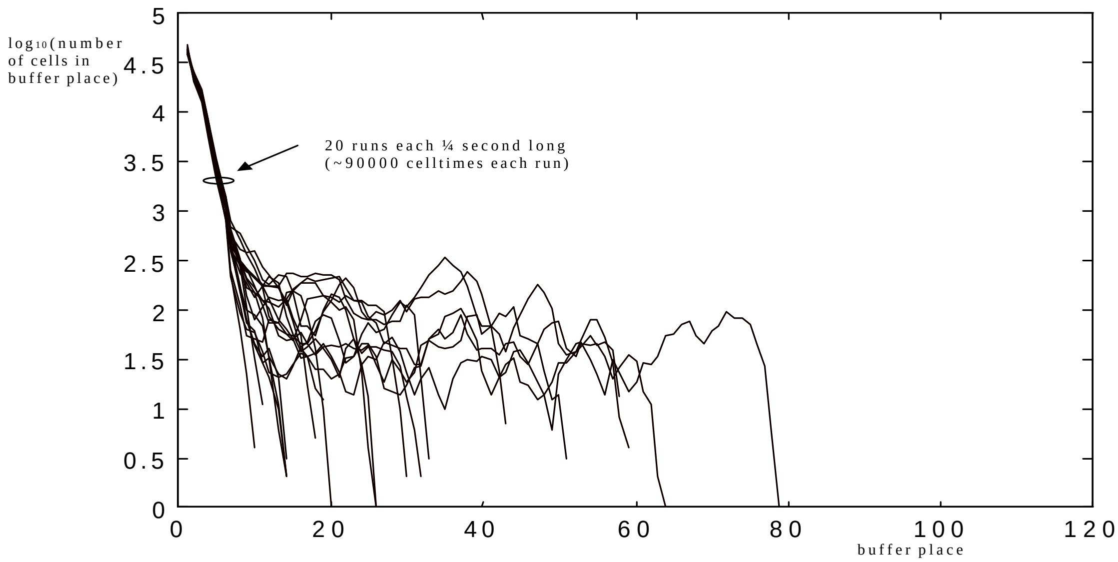

Figure 11. The buffer distribution (in logarithmic scale) for k=9 (round trip delay ~ 3600 kms) for 20 runs

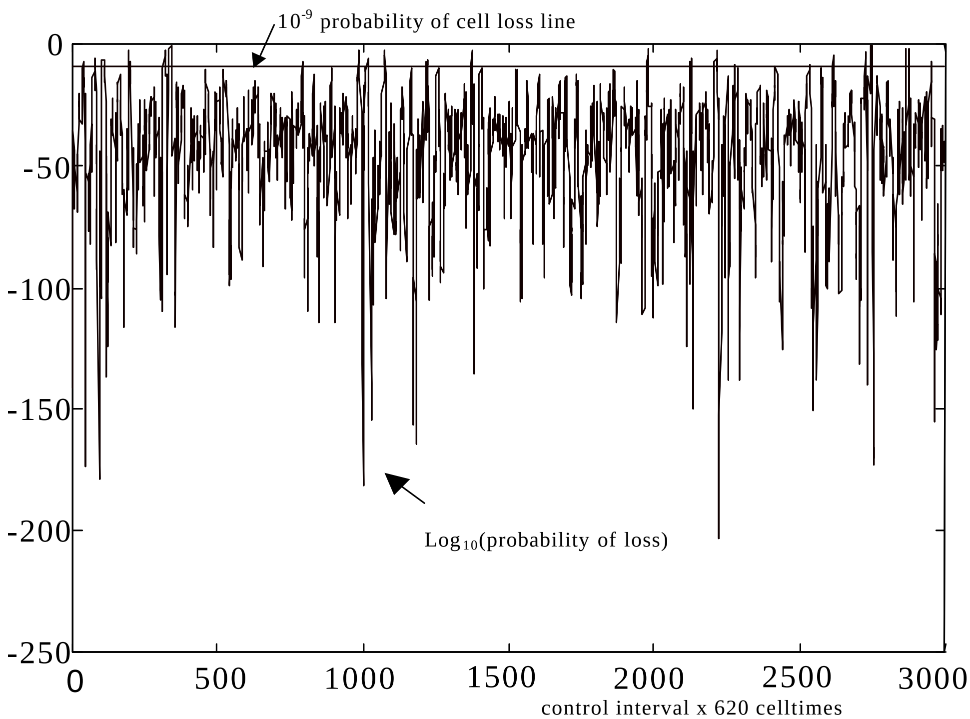

Figure 12. Calculated Probability of loss (in logarithmic scale) for a finite 100 cell places buffer for *k*=9 (round trip delay ~ 3600 kms) for 20 runs

Case C): uses same simulation parameters and pseudorandom sequences for uncontrollable traffic mix as Case A). A comparative evaluation of the proposed scheme, with four other types, is performed. (Note that since this paper was written, many new schemes, as well as variants of existing ones, have appeared in the literature, e.g. see [23]; not many deal with the presence of real time traffic competing for the resources[#1]). This simulation case is designed to show the superiority of the proposed adaptive scheme (which uses feedback calculated from measurements on the net input traffic stream, and feedforwards from queue length and mean uncontrollable traffic) to other schemes (three are based on obtaining the feedback from the queue length, and the last one from measurements of the net input traffic rate). For the five schemes:

- scheme ***i***: proposed adaptive scheme; feedback based on measurements of the net input traffic stream.

[#1] We recommend the setting up of a Common Simulation Framework (CSF), so that all proposed algorithms can be tested (with regard to a number of given attributes, e.g. robustness, efficiency, transient behaviour, complexity, fairness, scalability, interoperability, etc...). A fairer comparison can then be made, e.g. see 802.14 Modelling: Advantages of a Common Simulation Framework", IEEE working group, January 1996.

- scheme ***ii***: feedback based on threshold from the buffer length (e.g. Backward Explicit Congestion Notification). The buffer threshold is selected as 25, the Additive Increase Rate, AIR, is set at 2 Mbits/sec, and the Multiplicative Decrease Factor, MDF, is set at 0.5. Note that the selection of appropriate values for these parameters is a difficult task, dependant on many criteria [46];
- scheme ***iii***: as in scheme *ii*, but with a deadzone (high limit=40, low limit=20). The rest of the parameters are selected as in scheme ***ii***;
- scheme ***iv***: same adaptive controller structure as for scheme ***i*** but with feedback from the length of the buffer (reference value is set at 5; mean value over the interval T_s of 620 celltimes). A fixed parameter controller, with the same structure as the adaptive one, has also been simulated. The control coefficients (could be likened to a, ß, and d, equations (1) to (4)[#], in [26], or a_i and $ß_i$, equation (15), in [28]) were manually tuned from observations of the behaviour of the (controlled) system. No set of selected control coefficients has shown better behaviour than the adaptive case for a set of random initial conditions (e.g. different Initial Cell Rate), hence no further reference to a fixed controller structure will be made.
- scheme ***v***: Explicit Rate Indication for Congestion Avoidance (ERICA) [22]. For every averaging interval this algorithm calculates two quantities: fair share and this connection`s (VC`s) share. The VC share is calculated as the ratio of the current cell rate (CCR) and the load factor *z* (*z* is computed as the ratio of the input rate with the target rate), and the fair share as the ratio of the target rate with the number of active connections. The target rate is set to a fraction of the link rate (for this set of simulations it is set at 90%). Note that the selection of the averaging interval is a difficult task, dependant on the propagation delay and the input stream time dynamics—the larger the propagation delay, the greater the averaging interval, but for responsiveness (especially for high variability inputs) the opposite is true. In this study, the averaging interval is selected as 390 celltimes (about 1 msec) for short propagation delay and 5000 celltimes (about 13 msec) for large propagation delays (shorter averaging periods were also tested, but the losses, for the

[#] The control policy, as shown in the paper, cannot be implemented because in equation (4) there is reference to future values of the actual service rate for predictions beyond one step ahead [as required in equation (1)]. To implement one can replace all references to future values, by either the most currently available one or a prediction of its value.

case of large propagation delay, were greater). Note that ERICA could be seen as a simplistic adaptive scheme, adapting one system parameter (z).

Figure 13 and Figure 14 show the buffer distribution (logarithmic scale) for the five schemes. As shown, the proposed scheme (scheme ***i***) outperforms the rest of the schemes (schemes ***ii*** to ***v***) in both cases of none and large propagation delay. The maximum buffer place occupied by the adaptive scheme for the case of no propagation delay is 21. In contrast, all other schemes occupy higher maximum buffer places, with the scheme based on the double threshold unable to avoid losses. For the case with large propagation delay, the maximum buffer place occupied by the proposed adaptive scheme is 64 (considering the large propagation delay, it exhibits well controlled behaviour, with zero losses). All other schemes reported losses (i.e. the physical buffer limit was often occupied). The adaptive scheme based on feedback from the network queues lost about 0.5%, ERICA about 1.5%, and the simpler schemes based on the queue threshold, with or without deadzone, lost about 3%.

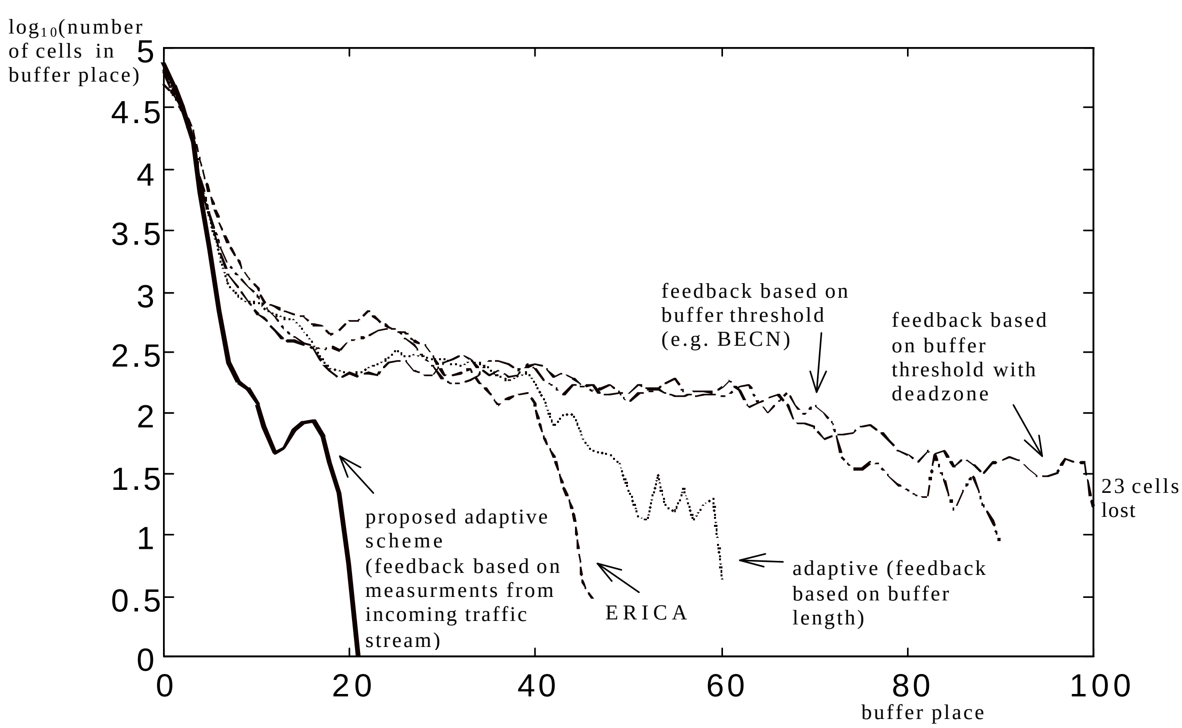

Figure 13. Comparative evaluation of schemes i)-v): Buffer distribution (in logarithmic scale) for no propagation delay, for about 1 second simulation time

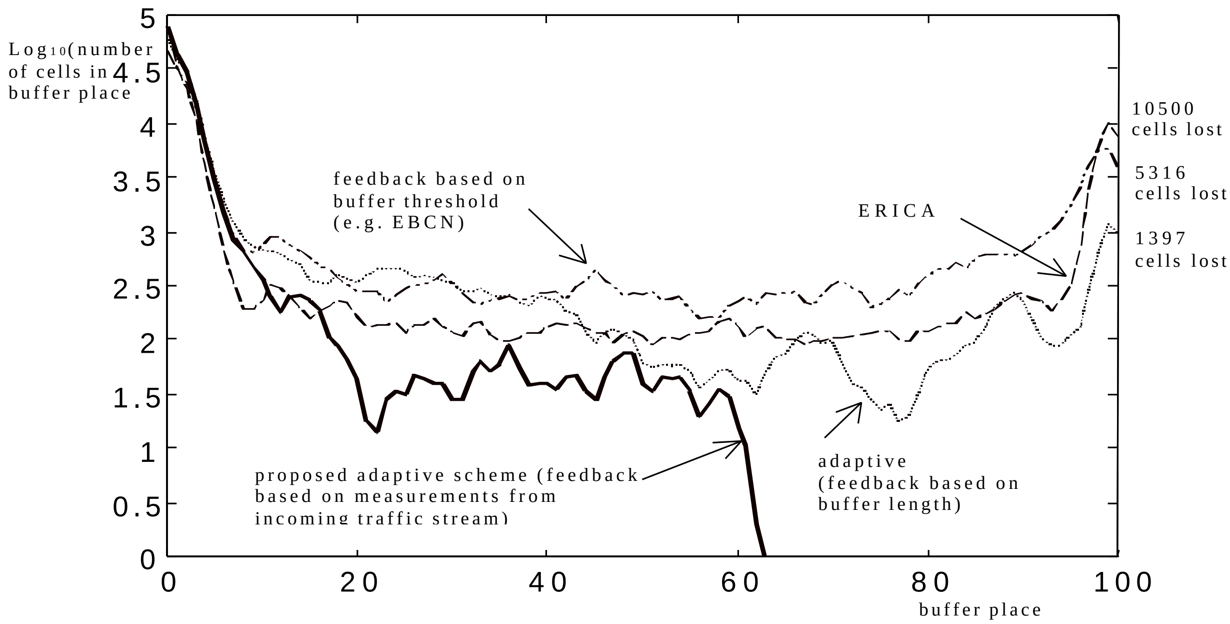

Figure 14. Comparative evaluation of schemes i)-v): Buffer distribution (in logarithmic scale) for *k*=9 (round trip delay ~ 3600 kms) for about 1 second simulation time

Figure 15 and Figure 16 show typical time evolution for the case of none and large propagation delay of the assigned cell-rate (controllable part of traffic) together with the uncontrollable part of the traffic stream (for clarity of presentation only the proposed adaptive scheme, ERICA and EBCN schemes are shown). The expected oscillatory behaviour of BECN, scheme ***ii*,** (scheme ***iii*** exhibited similar behaviour) is shown in both figures, with the cycle period increasing as the propagation delay increases. The time evolution of ERICA, on average, appears to track the available capacity, but in the case of the large propagation delay it does not scale up as well as the proposed adaptive scheme.

Ignoring cell retransmissions (which may involve the retransmission of whole packets), the calculated utilisation, for the case of no propagation delay, over the simulation period is: scheme ***i*** equals 0.88; scheme ***ii*** equals 0.94; scheme ***iii*** equals 0.92; scheme ***iv*** equals 0.91; and scheme ***v*** equals 0.9 (as chosen by design). For the case with significant propagation delay, the calculated utilisation for: scheme ***i*** equals 0.85; scheme ***ii*** equals 0.89; scheme ***iii*** equals 0.86; scheme ***iv*** equals 0.87, and scheme ***v*** equals 0.91.

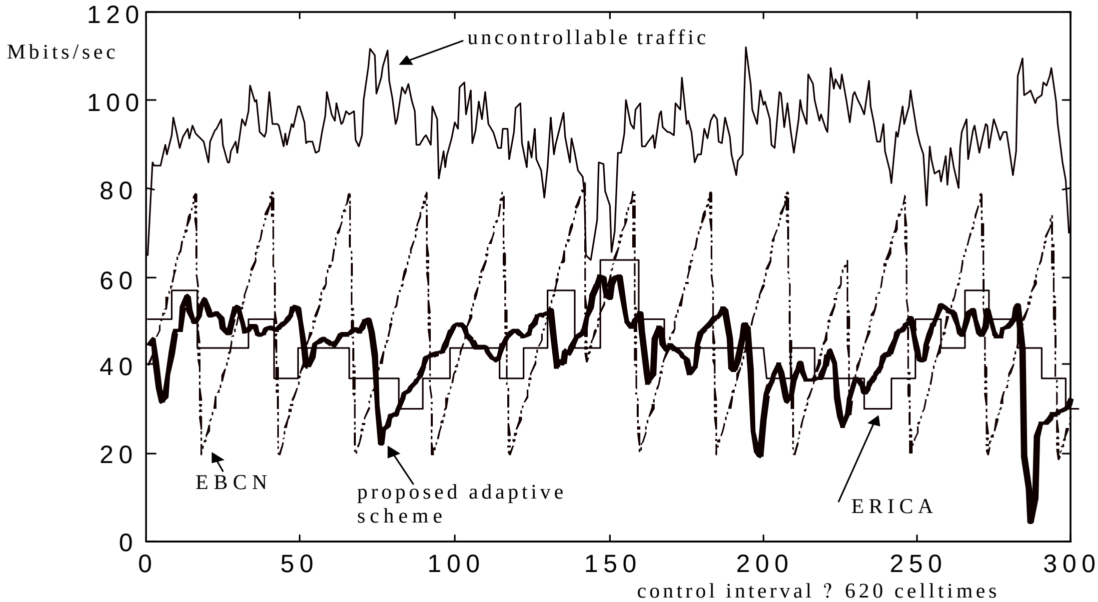

Figure 15. Comparative evaluation of schemes *i-v*: time evolution of controllable (assigned cell-rate) and uncontrollable traffic for no propagation delay, for about 1 second simulation time. For clarity of presentation we only show the proposed adaptive scheme, ERICA, and EBCN.

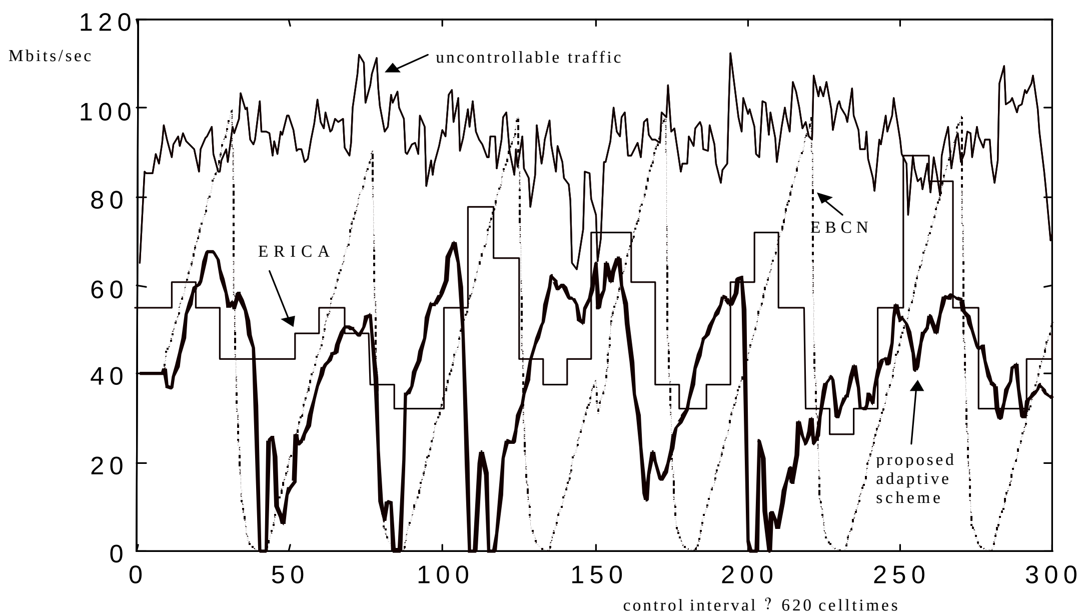

Figure 16. Comparative evaluation of schemes *i-v*: time evolution of controllable (assigned cell-rate) and uncontrollable traffic for *k*=9 (round trip delay ~ 3600 kms), for about 1 second simulation time. For clarity of presentation we only show the proposed adaptive scheme, ERICA, and EBCN.

Case D): uses same simulation parameters and scenarios as in Case C), however a number of uncontrollable (guaranteed) traffic sources (video) are replaced with a single high variability video traffic source (mean rate of uncontrollable traffic stream is about 43 Mbits/sec). The burstiness (ratio of peak to mean) of the video traffic source is about 7. We only show the case with a large propagation delay (as in Case B), as the differences between the schemes are more pronounced.

Figure 17 shows the buffer distribution, and Figure 18 the time evolution. The proposed adaptive scheme, once again, has outperformed all other schemes considered here (again considering the presence of the long propagation delay, it displays a controlled behaviour without any losses, whereas all other schemes loose a considerable amount of cells). Note that the adaptive scheme based on feedback from the queue length does not converge to any useful parameters, and ERICA performs very poorly, possibly due to its inability to track the high variability of the uncontrollable traffic (large averaging interval, and lack of prediction).

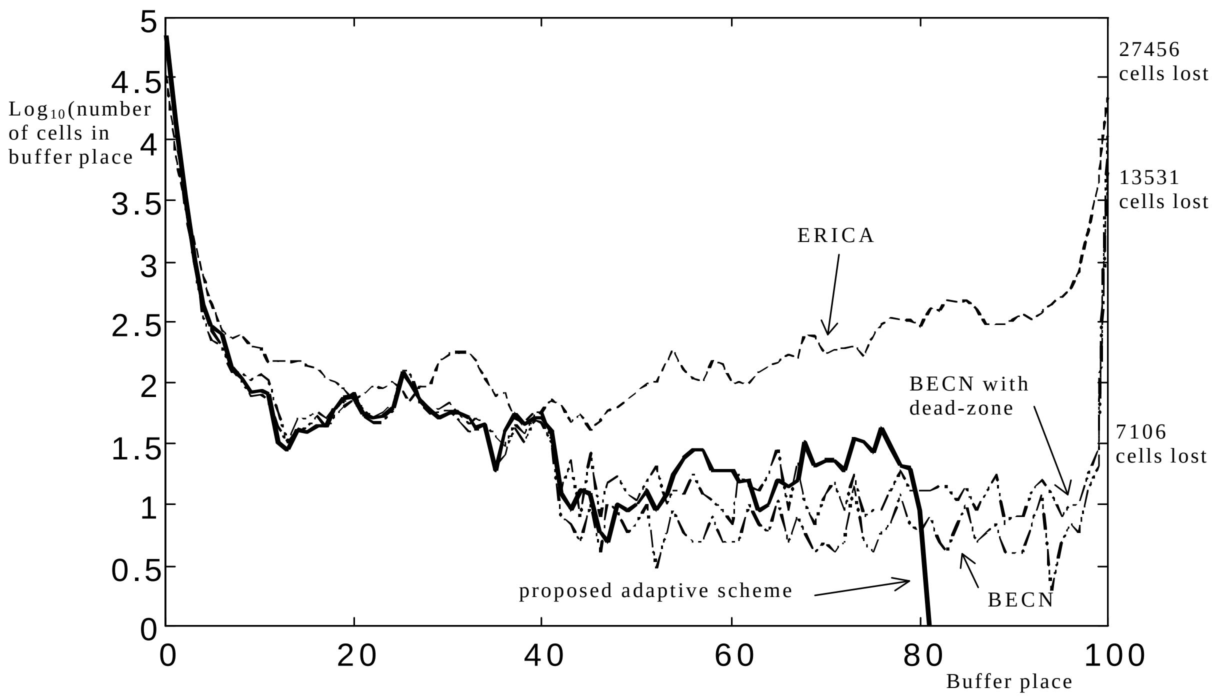

Figure 17. Comparative evaluation of schemes i)-v): Buffer distribution (in logarithmic scale) for *k*=9 (round trip delay ~ 3600 kms) for about 1 second simulation time, for high variability traffic.

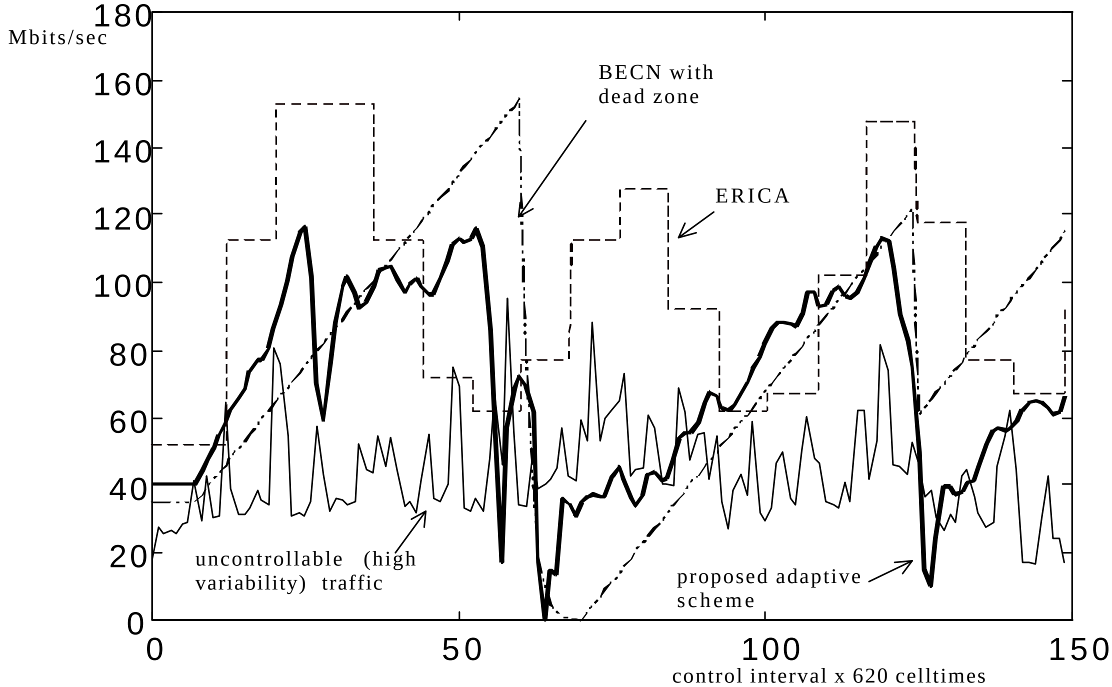

Figure 18. Comparative evaluation of schemes *i-v*: time evolution of controllable (assigned cell-rate) and uncontrollable traffic for *k*=9 (round trip delay ~ 3600 kms), for about 1 second simulation time for high variability traffic. For clarity of presentation we only show the proposed adaptive scheme, ERICA, and EBCN.

5: Conclusion

We propose a congestion control strategy which integrates CAC and flow control and uses predictive adaptive (feedback and feedforward) control techniques. It provides robust, effective and efficient congestion control with guaranteed QoS and high utilisation of link capacity. It also tolerates (fairly) long propagation delays because it uses traffic prediction that, due to the (high level) correlation present in some VBR traffic, can forecast beyond the propagation delay. The sensitivity to traffic modelling is also addressed by using adaptive predictive control.

Analysis and simulation show that high utilisation is achievable together with guaranteed quality of service (probability of loss only for the controllable group of traffic), as well as being adaptable, robust and tolerant to fairly long propagation delays, and random connections and disconnections. The proposed scheme uses feedback calculated from the net input process whose time dynamics are much greater than the buffer time dynamics, and can be comparable to the propagation delays. A comparative evaluation confirms the superiority of this scheme over other similar ones based on heuristics. The scheme makes use of only two broad traffic groups: uncontrollable (such as delay intolerant Real-Time Variable Bit Rate, RT-VBR) and controllable (e.g. delay tolerant, such as Available Bit Rate) traffic. It does not require any user declared parameters other than the peak rate for connections identified as uncontrollable.

Further work may include the integration of the above scheme with other control functions, such as the selective discarding of cells, and the dynamic bandwidth allocation. Also an investigation of the most suitable control option, as discussed in section 2, could prove useful toward a real time implementation. A "formal" measure of the effect of propagation delay on control quality could also be sought.

Appendix I

Proof of global convergence of the adaptive ACFC algorithm [33]

The proof of global convergence is a complex one. The basic approach we take is to exploit the properties of the parameter estimation algorithm via the Key Technical Lemma [44] together with the equations for the closed loop to prove convergence in the case of a deterministic system. It is given in terms of theorem 1.

Theorem 1 Subject to assumption 1 below, the control algorithm (5) when applied to the deterministic system [i.e. equation (1) with $e(t) = 0$] is globally convergent, that is the following properties are satisfied:

i) $\{y(t)\}$, $\{u(t)\}$, $\{v_1(t)\}$ and $\{v_2(t)\}$ are bounded sequences for all t.

ii) $\lim_{t\to\infty}\left[y(t) - y^{ref}(t)\right] = 0$.

Remark: The boundedness of the sequences also implies global stability.

Assumption 1

i) the time delay k is known

ii) An upper bound for the orders of the polynomials in the system model (1) is known.

iii) The following conditions apply (these are the same conditions as for the nonadaptive deterministic case). a) All modes of the "inverse" models [relating y^{ref} to $u(t)$ and y^{ref} to $y(t)$] lie inside or on the unit circle. Additionally any modes of the "inverse" model on the unit circle have a Jordan block size of 1. b) All controllable modes of the "inverse" models relating y^{ref} to $u(t)$ and y^{ref} to $y(t)$ lie strictly inside the unit circle

iv) sequences $\{v_1(t)\}$ and $\{v_2(t)\}$ are bounded.

Note that: assumption i) is required due to the look ahead nature of the controller; assumption ii) is of importance since it allows the system order to be overestimated thus ensuring that the controller has adequate degrees of freedom; assumption iii) is necessary in order to achieve perfect tracking and closed loop stability (same assumptions as for the deterministic nonadaptive case; we cannot expect to do better than the deterministic case); assumption iv) is always satisfied by the design of the CAC part of ACFC algorithm.

Using assumption 1 the proof of global convergence of the regulator follows readily [33].

Discussion

The key conclusions are that:

- Closed-loop stability is achieved.
- The output tracking error asymptotically goes to zero.

The extension of the proof to the stochastic case follows along the same lines as above proof, however the analysis tools are derived from the probabilistic framework (e.g. see [44], section 11.3.4).

Appendix II

Proof of strong properties of adaptive regulator [33]

Define the following continuous random variables:

$u(t)$ the controllable input entering the system (or equivalently the controller output)

$v_1(t)$ the uncontrollable traffic entering the system (i.e. the disturbance)

$y(t)$ the system output, given in terms of the pth percentile of the stationary distribution of V_∞

y^{ref} the reference value, given in the same terms as $y(t)$

T sampling period

A_n amount of work entering the system during the nth sampling interval, i.e. $A_n = \int_{nT}^{(n+1)T} \{u(t) + v_1(t)\} dt$

B_n amount of work that can be processed by the server during the nth sampling interval, i.e. $B_n = C^{link}$ the link cell server rate in cells/time unit T

Y_n net input process, given by $Y_n = A_n - B_n$, $n \geq 0$

V_n unfinished work at the beginning of the nth sampling interval. For the case of infinite buffer, V_n satisfies the following recurrent equation $V_{n+1} = (V_n + Y_n)^+$, $n \geq 0$, where $V_o = 0$

m mean value of Y_n, i.e. $m = E\{Y_n\} = E\{A_n\} - E\{B_n\}$

s^2 variance of Y_n, i.e. $s^2 = Var\{Y_n\}$

e_n mutually independent Gaussian random variables with zero mean and unity variance

r network utilisation (unity indicates 100% utilisation)

k_i a constant

The following proofs are based on the assumptions **i - iv**, given below:

assumption **i**) The adaptive controller has converged (i.e. $\lim_{t \to \infty}\left(y - y^{ref}\right) = 0$, and system variables $\{u(t)\}$ and $\{y(t)\}$ remain bounded; see Appendix I). Note that boundedness of the system variables implies global stability of the system.

assumption **ii**) the uncontrollable source variable $\{v_1(t)\}$ remains bounded (using the CAC algorithm the total peak rate of the uncontrollable sources does not exceed the link rate)

assumption **iii**) controlled source is saturated, i.e. there is an infinite supply of cells to the controllable queue at all times.

assumption **iv**) the feedback measurement is consistent and unbiased.

Remarks: The third assumption is only required for the proof of unity utilisation, and the unbiasedness of the feedback measurement introduced in the last assumption can be removed, at the expense of a more complicated proof. The output will still be bounded, however the bias must be added to it.

a) The long term network utilisation is equal to unity.

Proof: Using assumptions **i ii** and **iv** $E\{y(t)\} = E\{y^{ref}\}$? $E\{V_{n+1}\} = E\{V_n\} = k_1$? $E\{y^{ref}\} = k_2$, and therefore since $E\{V_{n+1}\} = E\{V_n\} + E\{Y_n\}$? $E\{Y_n\} = 0$. Now, using assumption **iii** (controlled source is saturated) and since the excess work in the system is equal to zero the system utilisation is unity, i.e. $r = 1$? ? ?

b) The long term network stability is guaranteed.

Proof: From assumptions **i** and **ii,** the system variables (i.e. $\{u(t)\}$ and $\{y(t)\}$ and $\{v_1(t)\}$) are bounded. Hence, the system is globally stable. ? ? ?

c) The long term buffer occupancy is bounded (with a probability of 1 - $E_{overflow}$).

Proof: Using assumptions **i ii** and **vi**, $E\{y(t)\} = E\{y^{ref}\}$ therefore the buffer occupancy is bounded below y^{ref} with a probability of $\left(1 - E_{overflow}\right)$? ? ?

d) The long term QoS for uncontrollable sources is guaranteed.

i) The total long term end-to-end delay is bounded.

Proof: Let the total end-to-end delay equal

$$D_T = \sum_{i=1}^{i=M} t_i^q + \sum_{i=1}^{i=M-1} t_i^p + t_{pd}$$

where: t_i^q is the delay term caused by the queueing at each node spanned by the connection; t_i^p is the propagation at each link i along the connection path; and t_{pd} is all the other fixed delay terms, including the packetisation and depacketisation delays. Using assumptions **i** and **vi** $E\{y\} = E\{y^{ref}\}$, therefore the long term queueing delay is upper bounded (with a probability of $1 - E_{overflow}$) to $\lim_{t\to\infty}(t_i^q) = \frac{y^{ref}}{C^{link}}$ (where C^{link} is in units of cells/sec) and $\lim_{t\to\infty}\left(D_T\right) = \sum_{i=1}^{i=M} \frac{y_i^{ref}}{C_i^{link}} + \sum_{i=1}^{i=M-1} t_i^p + t_{pd} = t_{total}$, i.e. a constant. Note: this is an upper bound on expected value of the end-to-end delay (with a probability equal to $1 - E_{overflow}$). ? ? ?

ii) the long term losses are bounded, at least with a probability of $1 - E_{overflow}$ (assuming that buffer size$\geq y^{ref}$).

Proof: Follows readily from c). ? ? ?

iii) the long term CDV is bounded.

Proof: Follows readily from c) ? ? ?

References:

1 ITU-T Recommendation I.371 (previously CCITT Recommendation), "Traffic Control and Congestion Control in B-ISDN", March 1993.

2 G. Gallassi, G. Rigolio, L. Fratta, "ATM: Bandwidth assignment and bandwidth enforcement policies", IEEE GLOBECOM'89, 49.6, pp. 1788-1793, 1989.

3 G. M. Woodruff, et al "A congestion control framework for high speed integrated packetized transport", GLOBECOM'88.

4 H. Suzuki, T. Murase, S. Sato, T. Takeuchi, "A simple and burst-variation independent measure of service quality for ATM traffic control", 7th ITC Specialist Seminar, 13.1, New Jersey, October 1990.

5 R. Guerin, H. Ahmadi, M. Naghshineh, "Equivalent capacity and its application to bandwidth allocation in high-speed networks", IEEE Journal of Selected Areas in Communications, Vol. 9, No. 7, September 1991.

6 E. Rathgeb, "Modelling and performance comparison of policing mechanisms for ATM networks", IEEE Journal of Selected Areas in Communications, April, 1991.

7 J. W. Roberts, "Traffic control in the B-ISDN" Computer Networks and ISDN Systems, Vol. 25, 1991.

8 F. Guillemin, P. Boyer, A. Dupuis, L. Romoeuf, "Peak Rate Enforcement in ATM", IEEE INFOCOM'92, pp. 753-758, Florence, May 1992.

9 A. DeSimone, "Generating burstiness in networks: A simulation study of correlation effects in networks of queues", Computer Communication Review, Vol. 21, January 1991.

10 H. Suzuki et al "A burst traffic control scheme for ATM networks using a simple quality estimate", GLOBECOM'90.

11 A. Pitsillides, "Connection admission control (CAC)" Report LTR0492, Laboratory for Telecommunication Research, Swinburne University of Technology, July 1992.

12 G. Ramamurthy, R. S. Dighe, "A Multidimensional framework for congestion control in B-ISDN", IEEE Journal of Selected Areas in Communications, Dec. 1991.

13 P.Boyer, "A congestion control for ATM", 7th ITC Seminar, Morristown, NJ, USA, 1990.

14 K. J. Astrom, B. Wittenmark, "Adaptive control", Addison- Wesley, 1989.

15 D. Mitra, J. B. Seery, "Dynamic adaptive windows for high speed networks with multiple paths and propagation delays", INFOCOM' 91.

16 K. K. Ramakrishnan, R. Jain, "A binary feedback scheme for congestion avoidance in computer networks with a connectionless network layer", ACM SIGCOMM 1988.

17 N. Yin, M. G. Hluchyj, "A dynamic rate control mechanism for source coded traffic in a fast packet network", IEEE Journal of Selected Areas in Communications, Sept. 1991.

18 H. T, Kung, "The FCVC (Flow-Controlled Virtual-Channels) proposal for ATM networks", International Conference Network Protocols, San Francisco, October, 1993.

19 Osama Aboul-Magd and Henry Gilbert, "Incorporating congestion feedback in B-ISDN traffic management strategy" ISS'92 International Switching Symposium, Osaka, Japan, October 1992

20 Peter Newman, "Backward explicit congestion notification for ATM local area networks", GLOBECOM'93, Houston, USA, December, 1993.

21 L. Roberts, "Enhanced PRCA (Proportional rate control algorithm)", Tech. Rep. AF-TM 94-0735R1, August 1994.

22 R. Jain, S. Kalyanaraman, R. Goyal, S. Fahmy, R. Viswanathan, "ERICA switch algorithm: a complete description", ATM Forum, AF/96-1172, August 1996.

23 A. Arulambalam, X. Chen, N. Ansari, "Allocating fair rates for available bit rate services in ATM networks", IEEE Communications magazine, November 1996.

24 S. Chan. Distributed congestion control for ATM networks. Australian Telecommunications Research, 28(1): 45-62, January 1994.

25 G. Ramamurthy and B. Sengupta, “A predictive hop-by-hop congestion control policy for high speed networks”, INFOCOM'93, May 1993.

26 P. P. Mishra, H. Kanakia, “A hop by hop rate-based congestion control scheme”, ACM SIGCOM, Baltimore, Maryland, 1992.

27 K-T Ko, P. P. Mishra, S. K. Tripathi, “Predictive congestion control in high-speed wide-area networks”, 2nd IFIP WG6.1/WG6.4, International Workshop on Protocols for high-speed networks, Palo Alto, November 1991.

28 L. Benmohamed, 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, No. 6, December 1993.

29 D.W. Clarke, "Introduction to self-tuning control", Implementation of self-tuning controllers, K. Warwick (Ed), Peter Peregrinus, 1988.

30 D. Tipper, S. Pappu, A. Collins, J. George, “Space Priority Buffer Management for ATM networks”, Asynchronous Transfer Mode Networks, Y. Viniotis and R. O. Onvural (Editors), pp 157-166, Plenum Press, New York, 1993.

31 G.C.Goodwin, K.S.Sin, "Adaptive Filtering, prediction, and Control, Prentice Hall, Englewood Cliffs, NJ, 1984.

32 CCITT: Recommendation I.121, "Broadband Aspects of ISDN", Geneva, 1991.

33 A. Pitsillides, "Control structures and techniques for Broadband-ISDN communication systems", Ph.D. thesis, Swinburne University of Technology, 1993.

34 G.L. Choundhury, D. M. Lucantoni, W. Whitt, “On the effectiveness of effective bandwidths for admission control in ATM networks”, 14th International Teletraffic Congress (ITC), Antibes, France, June 1994.

35 F. Bonomi, K.W. Fendick, “The rate-based flow control framework for the available bit rate service”, IEEE Network, March/April 1995.

36 R. Jain, “Congestion control and traffic management in ATM networks: recent advances and a survey”, Computer Networks and ISDN Systems, Oct. 1996.

37 "The ATM Forum Traffic Management Specification, V4.0, ATM Forum, April 1996.

38 D. Tipper, M. K. Sundareshan, "Numerical Methods for modelling Computer Networks Under Nonstationary Conditions", IEEE Journal of Selected Areas in Communications, Dec. 1990.

39 J-C. Bolot, A. Udaya Shankar, "Analysis of a fluid approximation to flow control dynamics", IEEE INFOCOM'92.

40 R. Addie, M. Zukerman, "An approximation for performance evaluation of stationary single server queues" IEEE Transactions on Communications, Vol. 42, no. 12, pp. 3150-3160, December 1994. Also in INFOCOM '93, San Francisco, April, 1993.

41 R. Addie, M. Zukerman, "Performance evaluation of Gaussian queue with a finite buffer", 7th ATRS, Nov. 1992.

42 B. Warfield, S. Chan, " Real-time traffic estimation in B-ISDN", 7th ATRS, Australia, Nov. 1992.

43 D. W. Clarke, P. J. Gawthrop, "Self-Tuning Controller", Proc. IEE Pt. D, vol. 122, no. 9, pp. 929-934, 1975.

44 G. C. Goodwin, D. J. Hill, M. Palaniswami, "A perspective of convergence of adaptive control algorithm", Automatica, Vol. 20, No. 5, pp. 519-531, 1984.

45 B. Maglaris, D. Anastassiou, P. Sen, G. Karlsson, J. Robbins, "Performance models of statistical multiplexing in packet video communications", IEEE Transactions on Communications, Vol. 36, No. 7, pp. 834-844, July 1988.

46 L. Benmohamed, D. Su, "Analysis of the rate-based traffic management proposal for ATM networks", 3rd Int. Conf. On Telecommunication Systems Modelling and Analysis, March 1995.