ATM Rate Based Congestion Control Using a Smith Predictor: an EPRCA Implementation

S. Mascolo

Dipartimento di Elettrotecnica ed Elettronica
Politecnico di Bari, 70125 Bari - Italy
mascolo@poliba.it

D. Cavendish, M. Gerla *

Department of Computer Science
University of California,Los Angeles CA 90024
dirceu@cs.ucla.edu gerla@cs.ucla.edu

Abstract

This paper presents a feedback control algorithm for ATM congestion control in which source rates are adjusted according to VC queue lengths at intermediate nodes along the path. The goal is to "fill in" the residual bandwidth, without exceeding a specified queue threshold. In order to obtain this, we propose a simple and classical proportional controller, plus a Smith Predictor to overcome instabilities due to large propagation delays, as well as to avoid cell loss. We propose an effective EPRCA implementation in which each source computes its input rate based on the maximum VC queue length along the path. Theoretical and experimental results show that high throughput is achieved even with queue sizes independent of the round trip delay.

1 Introduction

In an ATM network, in order to avoid congestion it is necessary to regulate the input traffic rate of the network such that all entering cells can be completely delivered using the existing network resources (i.e. queues, processing power and link trasmission capacity). A classical control approach to deal with this problem consists in monitoring the level of the unused resources and in feeding back the measured levels to a controller which adjusts the input traffic rates so that congestion is avoided.

The input rate control approach is known in literature as rate-based [6] in contrapposition to the credit approach [2] which, instead of the rate, aims at regulating the number of incoming cells. Many rate based algorithms can be found in literature. However none of these is completely satisfactory either for its complexity or for lack of stability properties, as is well reported in the excellent paper by Benmohamed and Meerkov [1]. In fact, due to transmission and propagation delay, most algorithms exhibit persistent oscillations. Furthermore, they have not been analyzed from the stability point of view, and so cannot guarantee the boundedness of the queues. Considering, for example, the well known additive increase/ multiplicative decrease PRCA [4], it is not possible to state its stability nor to guarantee cell loss avoidance. To our best knowledge, the Benmohamed and Meerkov's paper [1] is the first attempt to develop an analytic method for the design of congestion controllers which ensure good dynamic performance along with fairness in bandwidth allocation. However, the control law proposed in that paper requires a complex adjustment of control parameters in order to maintain stability and damp oscillations. Moreover, these parameters must be dynamically tuned to the specific input traffic and network condition. Finally, it is difficult to prove global stability, due to the complexity of the control strategy.

This paper presents a simple and effective rate based congestion control algorithm capable of "filling in" quickly the unused bandwidth with ABR traffic. The main appeal of the proposed congestion control algorithm consists in the use of a simple, first order dynamic model (for the queue levels) in cascade with a delay. This yields the following properties: a) the queue occupancy never exceeds maximum queue capacity (i. e. no cell loss); b) the queue occupancy dynamic is always stable for any positive proportional gain K, thus relaxing the need to dynamically adjust this parameter in order to stabilize queues or damp oscillations; c)the queue capacity strictly required to avoid cell loss is not related to the round trip delay (RTD), rather it is related to the input rate value in stationary condition.

After a description of the proposed control algorithm and its interpretation in terms of credit based end to end flow control, we present an EPRCA implementation for ATM congestion control. Finally we make a comparison with the PRCA scheme and report several simulations results.

2 The Model

In this section we define the notation, and also present the models for the network, queue, and control law used throughout the paper.

2.1 Network Model

We mainly follow the notation reported in [1]. The network consists of $N = \{1, ..n\}$ nodes and $L = \{1, ...l\}$ links. Each link i is characterized by: transmission capacity $c_i = 1/t_i$ (cells/sec); propagation

*This work was partially supported by CNPq under Grant 201597/93-4(NV) and by NSF under Grant NCR-9305376

delay td_i; processing capacity $1/tpr_i$ (cell/sec) where tpr_i is the time the switch i needs to take a packet from the input and place it on the output queue. We assume that the processing capacity of each node is larger than the total transmission capacity of its incoming links so that congestion is caused by transmission capacity only. The network traffic is contributed by source/destination pairs (S, D), where $S, D \in N$. To each (S, D) connection is associated a Virtual Circuit (VC) mapped on the path $p(S, D)$. Each source is characterized by its maximum transmission speed, $c_s = 1/t_s$.

Each link maintains a separate queue for each Virtual Circuit VC passing through it. We indicate with $x_{i,j}(t)$ the occupancy at time t of the queue associated with link i and VC_j, and with $X^o_{i,j}$ the corresponding queue threshold level. The control law computes the source input rate $u(t)$ (cell/sec). The bandwidth delay product td_j/t_j represents the number of cells "in flight" on the transmission link.

2.2 Model of the queue behavior

In this subsection we present a model of the dynamic behavior of each queue in response to input and output rate changes. We assume a deterministic fluid model approximation of cell flow. Each link maintains a separate queue for each Virtual Circuit (VC) passing through it. The reason for this choice is to ensure, through a round robin service discipline, the fair sharing of the link by each VC. Considering the queue associated with the virtual circuit VC_j at link i, the level of occupancy $x_{i,j}(t)$ at time t, starting at $t = 0$ with $x_{i,j}(0) = 0$, is the integral over the time $(0, t)$ of the difference between the rate of packets entering the queue (say $u_{i,j}(t)$) and the rate of packets leaving the queue (say $d_{i,j}(t)$):

$$x_{i,j}(t) = \int_0^t [u_{i,j}(t' - T_d) - d_{i,j}(t')]dt' \qquad (1)$$

where T_d is the transmission delay from the input source to the i, j queue.

2.3 The Rate Control Model

In this subsection, we present the control algorithm to regulate source rates. We propose a closed-loop control based on feeding back the network queue occupancy. In order to control the queue level $x(t)$ [1] for a specific VC, we initially use a simple proportional controller. Letting X^o be a set point for the queue level, we compute the difference between it and the current queue level $x(t)$. This difference, the error $e(t)$, is amplified by a positive constant gain K, so that $Ke(t)$ is the input rate imposed to the VC source. The proposed control implements the reasonable idea of enforcing an input rate proportional to the room available in the queue. This mechanism tends to "fill the queue", thus keeping link utilization high.

The calculated input rate $Ke(t)$ at time t will have effect on rate adjustments only after the round trip delay along the path, i.e. the time that the computed rate needs to reach the source, change the rate value, and finally returns back to the queue as an inflow rate $Ke(t)$. Fig.1 depicts the block diagram of this system, where RTD is the round trip delay. Note that, in wide area networks, the round trip delay is mostly determined by the propagation delay, so we assume that this quantity is fixed and known in advance.

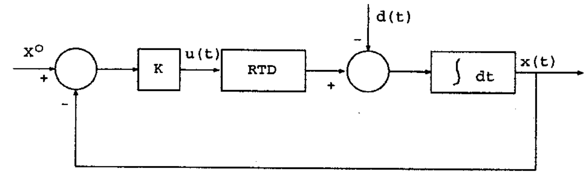


Figure 1: Queue dynamic model with a proportional controller

Due to the large delay, the dynamic behaviour of the queue level might exhibit oscillations, and even become unstable. In order to reduce oscillations it is necessary to reduce the amplification gain K, but this carries the drawback of a very long transient, i.e. the input rate is not able to fill in rapidly the queue, making the outgoing link underutilized [5].

To stabilize this system, still preserving the ability of quickly "filling in" the available queue space, we propose a classical Smith Predictor [3]. Following the Smith's principle, we substitute the constant gain K in Fig. 1 with a controller K^* (see Fig. 3) such that the resultant system dynamic is that of a first order system in cascade with a pure delay (Fig.2).

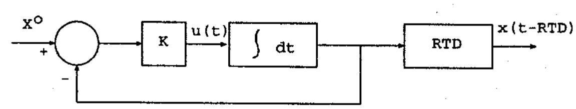


Figure 2: Equivalent model of the queue dynamic in response to the threshold level

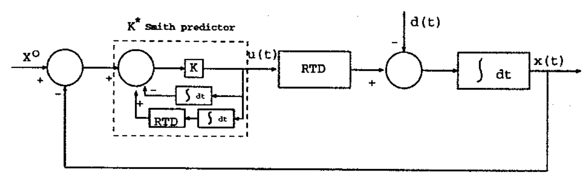


Figure 3: Queue dynamic model using a proportional controller plus a Smith Predictor

Thus, equating the transfer functions of the systems in Figure 3 and in Figure 2 one can verify that the Smith Predictor controller K^* is given by:

$$K^* = \frac{K}{1 + K(\frac{1 - e^{-RTDs}}{s})}$$

The Smith Predictor shown in Fig. 3 (K^*) gives the following input rate control equation:

[1] From now on we drop the i, j subscripts, for sake of simplicity

$$u(t) = K[X^o - x(t) - \int_0^t u(t')dt' + \int_0^t u(t' - RTD)dt']$$
$$= K[X^o - x(t) - \int_{t-RTD}^{t} u(t')dt'] \quad (2)$$

Note that this equation implements a simple proportional control action with the difference that the actual queue level is increased by the number of cells transmitted during the last round trip delay. Thus the physical interpretation is that the controller reacts as if all the "in flight" cells were in the bottleneck queue.

To describe the dynamic of the system it is helpful to look at the equivalent system shown in Fig. 2. In this figure we can observe two parts:

a)The first one, containing the integrator, the constant gain K, and the delay free feedback loop is a first order system, and thus is stable for every positive value of the parameter K. This parameter affects the transient behavior only. Namely $1/K$ is the time constant T of the system (meaning that after 4 T intervals the system reaches stationary condition). Moreover the dynamic response to a step function does not exhibit oscillations in reaching the stationary state. This implies that the queue occupancy never overshoots the set point level X^o, and hence the set point can be set equal to the queue capacity without ever incurring cell loss;

b) The second part consists of a pure delay block that causes a shift in time of the queue level x(t).

Concluding, the resulting behaviour of the queue occupancy, starting at $t = 0$ with an empty queue, is given by the first order system response to a step function delayed by the round trip RTD, that is: $x(t) = X^o[1-exp(-(t-RDT)/T)]$ (see Fig.4). Finally note that since the calculated input rate cannot be greater than the maximum source transmission speed $1/t_s$, $KX^o = 1/t_s$. Thus the system time constant is $T = X^o t_s$.

The system shown in Fig.3 has a behavior equivalent to the system depicted in Fig.2, in response to the input X^o. Now we consider the behavior of the queue level $x_d(t)$ in response to the output rate $d(t)$, where $d(t)$ can be modelled as a step function $a * 1(t)$, and a is the fraction of bandwidth, normalized to one, given to each connection.

Using Laplace transform method, after some calculations, we find:

$$x_d(t) = -a[t * 1(t) - (t - RTD) * 1(t - RTD)] - \frac{a}{K}[1 - e^{-K(t-RTD)}] * 1(t - RTD) + x(0) * 1(t) - x(0)[1 - e^{-K(t-RTD)}] * 1(t - RTD)$$

where $x(0) \geq 0$ is the queue level at $t = 0$. The overall response to $d(t)$ and X^o , therefore, is given by:

$$x_{tot}(t) = x_d(t) + x(t) \quad (3)$$

In stationary condition ($t \to \infty$), the queue level is:

$$x_{tot}(\infty) = X^o - aRTD - \frac{a}{K} \quad (4)$$

Figure 4 shows the transient behavior $x(t)$ in response to X^o, the transient behavior in response to $d(t) = 1(t) - 0.5 \times 1(t - offset)$ (where 1(t) is the step function), and the overall transient $x_{tot}(t)$. The initial offset is the time when the bandwidth $d(t)$ drops below the input rate of the corresponding VC queue.

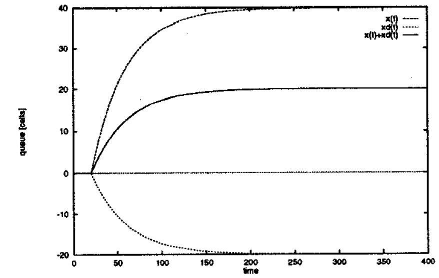


Figure 4: Queue level transient dynamic

3 Discrete time rate based control

So far we have dealt with continuous time models only. However, in ATM, feedback information is relayed in cells, and thus not available in continuous time, but rather in sampled form. Fortunately, the discrete time implemention of the Smith Predictor is simpler than the continuous one [7].

We start with the system model shown in Fig. 5. Here we place the controllers at edge source nodes for implementation simplicity.

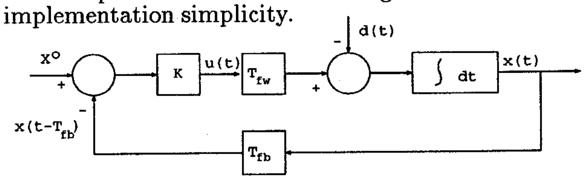


Figure 5: Queue dynamic model from the source point of view

From Fig. 5, we note that the input rate $u(t)$ takes the time T_{fw} (feedforward delay) to reach the queue (an integrator). Likewise, the queue level $x(t)$ takes the time T_{fb} (feedback delay) to reach the source. Following the Smith's principle, we look for the controller K^* so that the transfer function of the system of Fig. 5 is equivalent to that of the system of Fig. 6. This system was chosen so that by equating the transfer functions of Figs. 5 and 6 in the Laplace domain, it is possible to obtain a Smith Predictor, with a delay equal to $RTD = T_{fb} + T_{fw}$.

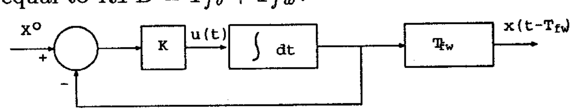


Figure 6: Equivalent model of the system using a Smith Predictor

From Nyquist sampling theorem and from control theory it is known that, in order to have a "continuous like" performance of the system under digitized control, the ratio of the time constant of the system over the sampling time must fall within the interval

(2,4)[7]. Indicating by $\triangle$ the sampling time and recalling that $T = X^o t_s$, it follows:

$$\frac{X^o t_s}{\triangle} = [2,4] \tag{5}$$

To write the discrete time version of the control equation (2) we must consider two cases:

i) $RTD \geq \triangle$: The ratio $RTD/\triangle = m + \epsilon$ where m is an integer and $\epsilon \in [0,1)$. Rewriting the continuous time equation 2 in its discrete version, we obtain the input rate at time $t_k = k\triangle$:[2]

$$\begin{aligned} u(k\triangle) \;=\; & K[X^o - x(k\triangle - T_{fb}) - \\ & u(k\triangle - (m+1)\triangle)\epsilon\triangle - \\ & \sum_{i=1}^{m} u((k-i)\triangle)\triangle] \end{aligned} \tag{6}$$

ii) $RTD < \triangle$:

$$\begin{aligned} u(k\triangle) \;=\; & K[X^o - x(k\triangle - T_{fb}) - \\ & u((k-1)\triangle)RTD] \end{aligned} \tag{7}$$

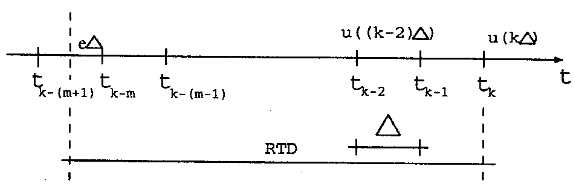


Figure 7: Discrete Time Notation

The notation used in the previous equations is illustrated in Figure 7 and will be followed throughout the paper. It results: $t_k = t_{k-1} + \triangle$.

3.1 Control equation and end to end credit interpretation

For sake of simplicity, we consider the case in which the ratio $RTD/\triangle$ is an integer ($\epsilon = 0$)[3]. The summation on the right side of the equation 6 can be rewritten as the sum of two parts:

$$\begin{aligned} I = \quad & u(t - T_{fw} - \triangle)\triangle + \\ & u(t - T_{fw} - 2\triangle)\triangle + \cdots + u(t - RTD)\triangle; \\ II = \quad & u(t - \triangle)\triangle + u(t - 2\triangle)\triangle \cdots + \\ & u(t - T_{fw})\triangle \end{aligned}$$

The first one represents the number of cells that have already arrived at the bottleneck queue but are not yet know at the source due to the feedback propagation delay T_{fb}. The second one represents the number of cells that are travelling from the source to the queue. Therefore the input rate computation at time t can be rewritten as: $u(t) = K[X^o - x(t - T_{fb}) - (I) - (II)]$. We can interpret $x(t - T_{fb}) + (I) + (II)$ as "effective queue level at time t". So the calculation of the input rate $u(t)$ is made as if all "in flight" cells were already at the queue. In this way the dynamic is delay free, which results in stability and lack of oscillations.

Finally we would like to interpret the difference between the queue capacity and the "effective queue level" as a number Q of cells that can be transmitted by the source without causing overflow to the bottleneck queue.

4 From periodic to aperiodic feedback

To implement the proposed discrete time control algorithm we need to supply the controllers located at the sources with periodic feedback information (every $\triangle$ units of time, with $\triangle$ satisfying equation (5)). This can be obtained if the upstream node of a congested link sends the feedback information, at every sampling time, to all the sources in the upstream direction, as in the Backward Congestion Notification (BCN) scheme. This is what is assumed in [1]. We call this type of scheme "Periodic Feedback Control". In systems where Forward Congestion Notification (FCN) scheme is used, like in the PRCA scheme, the source is responsible for transmitting a management cell RM every NRM data cells. The control cell itself has to compete for bottleneck link bandwidth, since it has to reach the destination node before being relayed back to the source through either the same or an alternative reverse path. Clearly, under this scheme, it is not possible to guarantee the periodic feedback information used in the discrete-time control equation. Due to the sharing of the congested link, the rate of the feedback cells that can be received by the sources is $B_{av}/[N_{vc} \times (1 + NRM)]$, where N_{vc} is the number of Virtual Circuits sharing the same bottleneck link and B_{av} is the available bandwidth. Thus the interarrival time of the feedback cells increases as $N_{vc} \times (1 + NRM)$. Because it is necessary to guarantee the sampling time equation (5) in order to achieve a good performance of the feedback control, we have to increase the time constant of the system (i.e. the queue size per VC) as N_{vc} increases. Thus, it is necessary to use VC queue sizes proportional to the number of Virtual Circuits sharing the same bottleneck link. This requirement does not derive from the control algorithm itself but rather from the FCN mechanism used for delivering feedback information. To cope with the somewhat irregular delivery of RM cells, we need a control algorithm which must operate well even if no RM cells are received for a while. If the source receives the feedback information, the control algorithm will adjust the rate accordingly. Otherwise it will compute the rate by estimating the missing feedback information in a conservative way. In other words, this algorithm must perform some kind of "virtual feedback". We call this type of control "Aperiodic Feedback Control".

4.1 A control algorithm for EPRCA

We now propose a version of the previous discrete time control algorithm suitable for the FCN feedback relay scheme. The feedback information is provided by RM cells which collect the maximum buffer level along the path. Note that regardless of the bottleneck location along the VC path, RTD is always the same.

[2] Note that k should not be confused with the gain K

[3] The extension of this subsection to the case $\epsilon \neq 0$ is trivial

The system will still be cell loss free even if we are not able to guarantee the required periodicity of feedback information.

The basic idea is to update the source rate at least after each $\triangle$ sampling interval, regardless whether the source gets the feedback information or not. Let t_k, t_{k+1} be the instants at which the source receives the last and actual feedback information, respectively. Two cases need to be considered:

i) $t_{k+1}-t_k \leq \triangle$. The source stores the rate $u(t_k)$, as well as its duration $\triangle_k$, so that $u(t_k)\triangle_k$ becomes one of the terms of the summation in the control equation. Thus the rate updating equation is:

$$u(t_k+\triangle_k)=K[X^o - x(t_k+\triangle_k-T_{fb}) - \sum_{i=0}^{m} u(t_{k-i})\triangle_{k-i} - u(t_{k-m-1})(RTD-\sum_{i=0}^{m}\triangle_{k-i})]$$

where $\sum_{i=0}^{m}\triangle_{k-i} \leq RTD < \sum_{i=0}^{m+1}\triangle_{k-i}$, $\triangle_{k-i}\leq\triangle \quad \forall i, \quad t_k = t_{k-1}+\triangle_{k-1}$.

ii) The interval $\triangle$ expires before the source receives its control packet. In this case, the algorithm has to estimate the queue level $x(t_k+\triangle-T_{fb})$. In order to be conservative, and to prevent cell loss, we propose the following "worst case" estimate of the missing queue level. We conservatively assume that in the time interval $[t_k, t_k+\triangle]$ (with $\triangle=\triangle_k$) the queue has zero output rate. Thus the "worst case" queue level is the last value $x(t_k-T_{fb})$ plus what has been received in the interval $[t_k, t_k+\triangle]$. The accrued term corresponds to the number of cells pumped into the network during the interval $[t_k-RTD, t_k-RTD+\triangle]$. Therefore, the "worst case" estimate of the queue level at time $t_k+\triangle_k$ is:

$$x(t_k+\triangle_k-T_{fb})=x(t_k-T_{fb}) + u(t_{k-m-2}))(RTD-\sum_{i=1}^{m+1}\triangle_{k-i}) + u(t_{k-m-1})(\triangle-(RTD-\sum_{i=1}^{m+1}\triangle_{k-i}))$$

We call "virtual feedback" this worst case estimation of the queue level . Note that this is equivalent to storing the last received feedback value, $x(t-T_{fb})$, and adding the new term $u(t_k)\triangle$ to the last sum of "in flight" cells, say $sumF$, i.e.

$$sumF=\sum_{i=1}^{m+1} u(t_{k-i})\triangle_{k-i} + u(t_{k-m-2})(RTD-\sum_{i=1}^{m+1}\triangle_{k-i})$$

and the rate is

$$u(t_k+\triangle_k)=K[X^o - x(t_k-T_{fb}) - u(t_k)\triangle_k - sumF]$$

In this proposed EPRCA algorithm, the sources at the edge nodes of the network update their input rates at least every $\triangle$ units of time. If they do not get information about the occupancy of the congested queue, they decrease their rates based on a "worst case" estimate of the congested queue level. When they get the next feedback information they will increase their rates because the actual queue level cannot be larger than the conservative estimate. In other words, the algorithm behaves as a "positive feedback ", decreasing the rate when feedback is not available and increasing it when feedback information resumes. Note that this is very important aspect to guarantee stability in any feedback congestion control because, due to congestion, it is not possible to guarantee the rate at which feedback cells are received.

4.2 Input rate stationary value

Consider the control equation (6). Under stationary conditions and no congestion (i. e. empty queue) such equation reduces to:

$$u_s = K(X^o - u_s * RTD) \tag{8}$$

where u_s stands for the stationary rate value. Recalling that $K = 1/X^o t_s$, the stationary value of the imput rate becomes:

$$u_s = \frac{1/t_s}{(1+\frac{RTD}{X^o t_s})} \tag{9}$$

Let B_{av} be the available bandwidth for the ABR connections. Due to the fair sharing enforced by the round robin service discipline, we have: $N_{vc}u_s = B_{av}$. Substituting in this equation the stationary value u_s from equation (9), and solving for X^o, we get:

$$X^o = \frac{B_{av} * RTD}{(N_{vc} - B_{av}t_s)} \tag{10}$$

Equation (10) states that, in order to achieve full link utilization, a minimum queue X^o per VC is required. Note that X^o is proportional to $B_{av}RTD$ but decreases with the number of connections N_{vc}.

5 A comparison between the PRCA scheme and the proposed EPRCA

In the ATM Forum PRCA proposal [4], an additive increase/multiplicative decrease rate control is exercised at the sources. Binary feedback information (congested/ not congested) is received at the sources, and rate increase (additive) is performed in case a "not congested" feedback is received. Failure to receive the "not congested" notification causes multiplicative rate decrease at the source after each time interval $\triangle$, thus making the scheme conservative.

Our proposed EPRCA uses the delayed queue occupancy as the feedback information. Like in the PRCA scheme, if no feedback is received, the source calculates the rate at fixed intervals $\triangle$ related to the time constant of the queue. The calculation is performed using a "worst case" estimate of the queue level.

In the following, we study the dynamic behavior of the rate when the source lacks feedback information. Let $u(0) = \frac{1}{X^o t_s}(X^o - x(t-T_{fb}) - \sum_{RTD} u(t_i)\triangle_i)$ be the rate computed based on the last received feedback cell. If no feedback information is received since then, the rate must be updated every Δ unit of time, using the "worst case" estimate. It follows :

$$u(1) = \frac{1}{X^o t_s}[X^o - x(t - T_{fb}) - u(0)\Delta - \sum_{RTD} u(t_i)\Delta_i]$$
$$= \frac{u(0)}{X^o t_s}(X^o t_s - \Delta)$$
$$u(2) = \frac{u(1)}{X^o t_s}(X^o t_s - \Delta)$$
$$\vdots$$
$$u(k) = \frac{u(k-1)}{X^o t_s}(X^o t_s - \Delta) = u(0)\left[\frac{(X^o t_s - \Delta)}{X^o t_s}\right]^k$$

i.e. the rate decreases exponentially. When the source resumes receiving feedback information, the rate jumps to $(X^o - x(t - T_{fb}) - \sum_{RTD} u(t_i)\Delta_i)$.

Therefore, we note that our EPRCA scheme, developed from a precise and simple mathematical model, operates according to a "positive feedback" mechanism, much like the PRCA scheme. The important difference is that the dynamic behavior of our regulation is related to the network state and parameters. In fact, the rate decreases exponentially with a base related to the sampling time/time constant ratio. More importantly, the increasing jumps are related to the queue level and to the number of cells released from the source during the last round trip interval. As a consequence, our EPRCA scheme does not drop cells, nor does it need a queue size proportional to the RTD to prevent cell loss. In contrast, the conventional PRCA scheme does not use precise information on the queue level and does not take into account the number of cells released during the last round trip delay. Consequently, it cannot perform the correct rate increase so as to prevent congestion and cell loss.

6 Simulation Results

In this section, we present results of a discrete event simulation of our control scheme. We first show the performance of the periodic control algorithm under the same scenario considered by [1]. Then we compare the proposed EPRCA scheme with the conventional PRCA scheme [4].

The network topology, shown in Fig. 8, is the same presented in [1]. Links have uniform speeds, normalized to 1 cell per unit of time [cell/s]. Links to the right of the bottleneck have a bandwidth-delay product of 10 cells, while the links to the left of the bottleneck have zero propagation delay (similar to [1]).

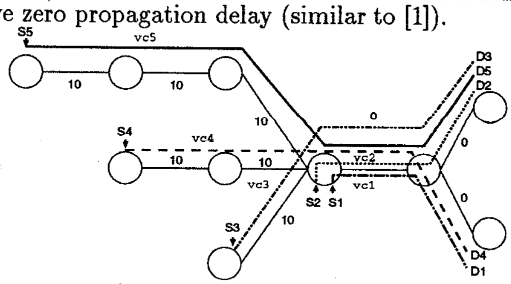


Figure 8: Network Topology

Five VC connections compete for bandwidth resources of a bottleneck link. VC connection activity (i.e. start and end time) is described in the table 1. We assume infinite backlog at each source. We set a queue level $X^o = 40$ for each queue, in order to have a sampling time of the system of $40/4 = 10$, that is, one fifth of the interarrival time of the feedback cells under the FCN scheme, with $NRM = 10$ and $N_{vc} = 5$.

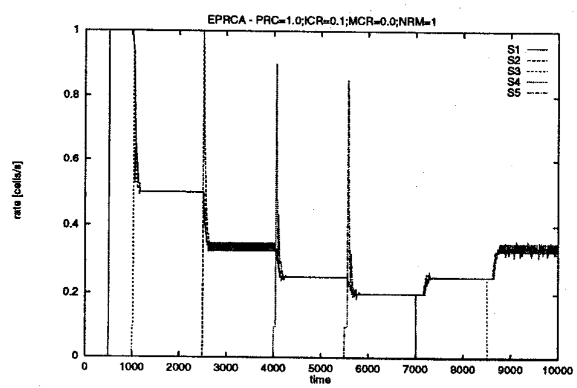
Table 1: VC Connections Activity

Connection #	1	2	3	4	5
Start Time	500	2500	1000	4000	5500
End Time	7000	10000	8500	10000	10000

Periodic Feedback

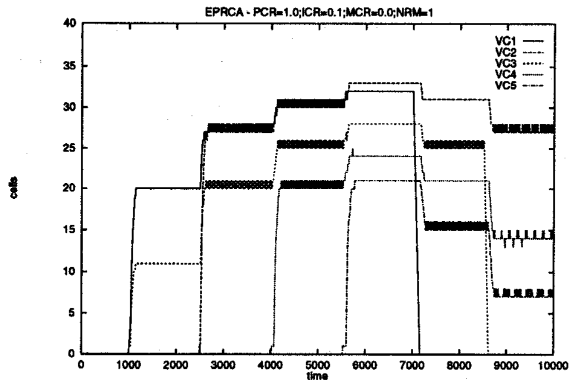
We first show the performance of a periodic sampling version of our control scheme, in conditions similar to [1].

According to equation (5) we choose a sampling time $\Delta = 10$. Figure 9(a) shows the behavior of the five input rates, corresponding to connections $S1 - S5$, at source nodes. For sake of comparison with PRCA, we assume an initial cell rate of 0.1 [cells/s], equal to the PRCA minimum cell rate. After the start/end of a connection, each rate rapidely settles on the new fair stationary value [4]. Figure 9(b) shows the dynamic behavior of the five queues at the bottleneck link, corresponding to $VC1 - VC5$ bottleneck queues. As can be seen, no queue overflow occurs. Moreover, each stationary level is in accordance with equation (9). The overall performance is similar to [1]'s periodic control, without having dynamic tuning of control parameters.



(a) Rates

[4]When there are three active connections, the figure shows small oscillations, due to the fact that the control equation tries to regulate the queue occupancy to a value between two integers



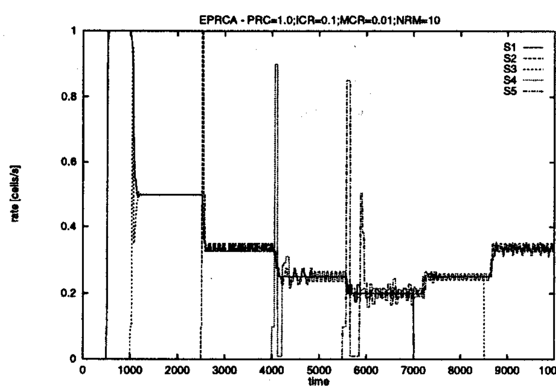
(b) Bottleneck Queues

Figure 9: Periodic Feedback

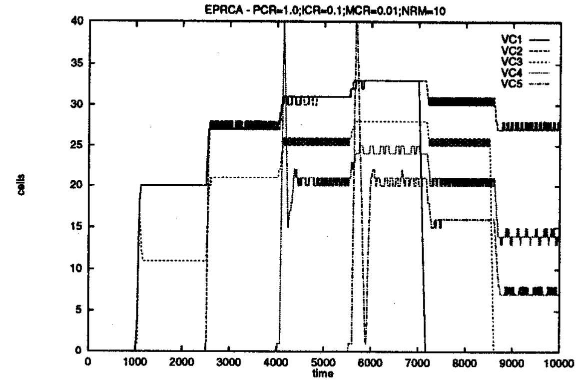
Aperiodic Feedback

The aperiodic control scheme, with $\Delta = 10$, requires $NRM = 1$ (i.e. one control cell every data cell) in order to guarantee the minimum feedback frequency rate. The value $NRM = 1$ derives from the fact that, under the heaviest traffic condition (five connections), the feedback cell interarrival time is $\Delta = N_{vc}(NRM + 1)$. Since the minimum feedback rate is maintained, simulation results are identical to the ones under periodic control, as expected, and hence are omitted.

We next show performance degradation in case equation (5) is not respected. By setting $NRM = 10$, under the heaviest traffic condition, the feedback interarrival time is $55 > 10$. We see from Fig. 10(a) that the rate does not reach rapidely the stationary condition anymore. Moreover, Fig. 10(b) shows that overflow occurs in VC4 and VC5 queues. Other simulation results, not reported here, show that the greater the NRM value, the less controlled the queue levels are.



(a) Rates

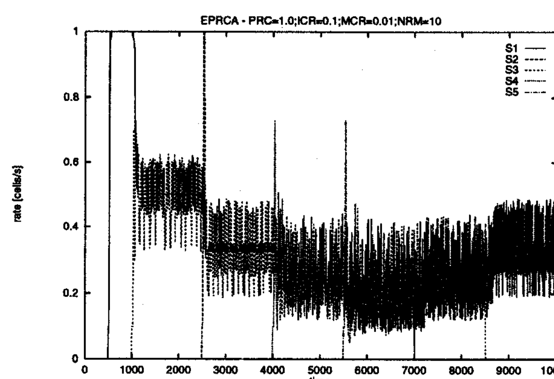


(b) Bottleneck Queues

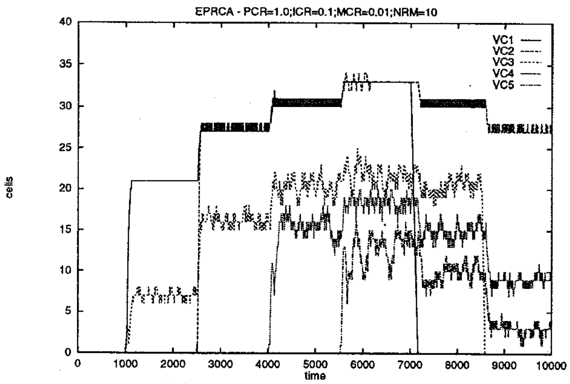
Figure 10: Aperiodic Feedback

Aperiodic + Virtual Feedback (EPRCA)

Next, we study the performance of the proposed EPRCA under the same conditions and feedback frequency ($NRM = 10$), as used above. Figure 11(a) shows the oscillatory behavior of the controlled rates. This is so because the control operates in the "positive feedback" mode, i.e. increasing promptly the rate when a feedback cell is received, and decreasing exponentially otherwise. However, the oscillations are constant in amplitude, and centered at the fair value of the rate, so that the throughput performance is preserved. The frequency of oscillations is high because the virtual feedback period is $\Delta = 10$, while the actual feedback interarrival time is about 50. In fact, the control algorithm decreases the rate every $\Delta = 10$, in a conservative way, increasing it promptly, when a feedback cell is received (approx. every 50 units of time). Figure 11(b) shows that the queue levels are still bounded, guaranteeing no cell loss. Thus, the major advantage of the Virtual Feedback scheme is to prevent cell loss, due to congestion, even if it is not possible to guarantee the frequency of feedback cells.



(a) Rates

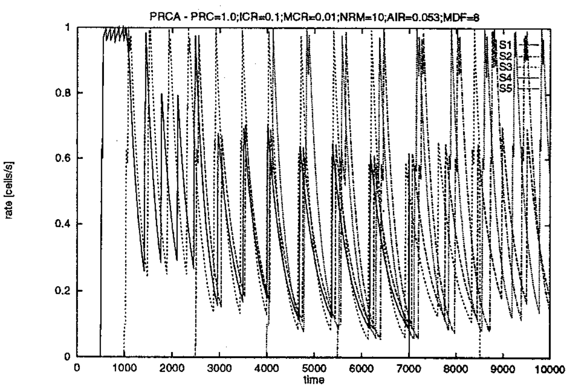


(b) Bottleneck Queues

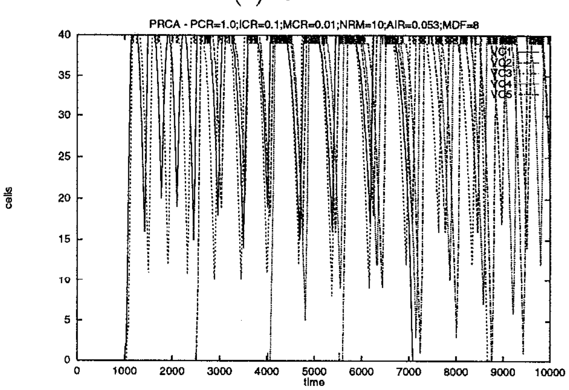
Figure 11: Aperiodic + Virtual Feedback

Conventional PRCA

For sake of comparison, the PRCA scheme has been simulated under the same traffic conditions as before, with parameters: NRM=10; AIR=0.053; MDF=8. The results are shown in Fig. 12. As expected, the PRCA scheme does not prevent cell loss, because it cannot account for the bottleneck queue level and the number of cells "in flight".



(a) Rates



(b) Bottleneck Queues

Figure 12: PRCA Control Scheme

7 Conclusion

Theoretical arguments and simulation results show that the proposed control algorithm performs an effective congestion control in high speed networks, guaranteeing no cell loss, fairness and small queue per VC. The buffer capacity strictly required to guarantee no cell loss, results to be independent of the round trip delay. The control scheme performs very well even under the pratical constraints of the EPRCA implementation in an ATM network. With the proposed EPRCA, every source adjusts its rate based on the delayed queue occupancy value fed back from the congested link and on the number of cells transmitted during the last round trip time. Therefore, as a difference from a "blind" additive increase/multiplicative decrease policy, our scheme implements a feedback regulation which is based on a rigorous control model. Further research is in progress in order to use the proposed control algorithm with a common queue per link. Preliminary results were presented in [8]. For a compreensive version of this work, see [9].

References

[1] L.Benmohamed, S.M.Meerkov, "Feedback Control of Congestion in Packet Switching Networks: The Case of a Single Congested Node", *IEEE/ACM Trans. on Networking*, vol.1, no.6, pp.693-708, December 1993.

[2] H.T.Kung, R.M.Morris, T. Charuhas, D. Lin, "Use of Link-by-Link Flow Control in Maximizing ATM Network Performance: Simulation Results," *Proc. IEEE Hot Interconnects Symposium, '93*, Palo Alto, California,5-6 August 1993.

[3] O.J.Smith, "A controller to overcome dead time," *ISA J., vol.6, no.2*, Feb. 1959,pp.28-33.

[4] A. W. Barnhart, "Baseline Performance Using PRCA Rate-Control," *ATM Forum/94-0597*, July 1994.

[5] D. Cavendish, Y.Oie, M.Murata, H.Miyahara, "Proportional Rate-Based Congestion Control under Long Propagation Delay," *Int. Journal of Communication Systems*, vol. 8, pp. 79-89, 1995.

[6] F.Bonomi, K.W. Fendick, "The Rate-Based Flow Control Framework for The Availble Bit Rate ATM Service," IEEE Network, pp. 25-39, March/April 1995.

[7] K.J.Astrom, B.Wittenmark, Computer Controlled Systems. *Englewood Cliffs, NJ*: Prentice Hall,1990.

[8] D. Cavendish, M. Gerla, S. Mascolo, "ATM Rate Based Congestion Control Using a Smith Predictor: Implementation Issues," *Proceedings of First Workshop on ATM Traffic Management WATM'95*, pp. 289-296, December 1995.

[9] S. Mascolo, D. Cavendish, M. Gerla, "ATM Rate Based Congestion Control Using a Smith Predictor," *UCLA Technical Report Number 960001*, February 1996.