Design and Analysis of an ABR Explicit Rate Algorithm[1]: ERAQLES

Yan Moret, Serge Fdida
Laboratoire MASI
Université Pierre et Marie Curie
4, place Jussieu
75252 Paris Cedex 05
Email: {moret, fdida}@masi.ibp.fr
WWW: http://www-masi-net.ibp.fr/{˜moret, ˜sf}

Abstract

The Available Bit Rate service was initially proposed by the ATM forum and also adopted by UIT. It is an important service class for ATM networks because it provides a rate control mechanism that enable to dynamically adjust the source rate to the amount of resources available in the network. ABR is designed to support data services and will be suited to efficiently carry Internet traffic. We present in this paper an original design of an Explicit Rate based ABR algorithm: ERAQLES. It provides a dynamic evaluation of the level of available resources left unused in the network in presence of both CBR and VBR traffics. The evaluation and distribution functions to obtain the explicit rate allocated to each individual source are computed using a novel distribution that control the filling level of the ABR queue. The switch buffer allocated to the ABR service is used in order to get a better statistical gain. We demonstrate fairness and convergence properties of ERAQLES to the target filling level. The design parameters of ERAQLES as well as various network environments are considered. The influence of parameters such as network delays, signaling rate or convergence factor are analyzed. The efficiency of the algorithm under Generic Fairness Configurations is shown to be excellent.

1. Introduction

The ATM Forum traffic management subworking group has defined a new ATM service class reffered to as the Available Bit Rate (ABR). The idea behind ABR is that there exist

---

[1] A patent application has been filled for the first version of this algorithm (IBM patent, see [Fdid95])

many applications that are sensitive to loss but can tolerate variation in delays. Such applications would expect to use extra bandwidth from the network if any was available. To do so, the network has to inform the user of congestion that appears within the network in order to maintain loss at an acceptable level. We present in this paper, the design and analysis of ERAQLES (Explicit Rate Algorithm using Queue LEngth States) an algorithm developped to provide ABR services. An efficient sizing of the protocol parameters is presented as well as proofs for convergence and fairness. The designed algorithm is shown to be both very efficient and stable in all situations.

2. The ABR service definition

ABR is an end-to-end rate-based service in which the network provides feedback to the sender, in order to adjust its transmission rate to the bandwidth available on the path, and to minimise cell loss ratio without any guarantee on the end-to-end delay.

The ABR service can be provided either on a VCI or VPI basis, depending on which is switched. Fairness among all ABR connections should be achieved following different possible criteria (Min-Max, …). The mechanism used to control the closed-loop should not be dependant on any intrinsic time-scales and must reach some steady state. It could operate end to end, over a segment or link by link. Finally, it has to be robust and easy to implement.

In order to provide an ABR service, several functions have to be performed by the source, the destination and the network switches. These components will use RM (Resource Management) cells in order to carry control information back and forth on the connection under control. It is based on a rate control algorithm where the source rate is decreased or increased as a function of the congestion indications returned either by the destination or the switches. These elements can either advertised congestion through the Congestion Indication (*CI*) and No Increase (*NI*) fields (relative rate marking) of the RM cells or directly compute the Explicit Rate (*ER*) to which the source has to adapt (explicit rate marking). The Explicit Rate solution is the more promising one because it provides a way to compute the maximum rate at which a sender can transmit its traffic while still minimising the cell loss ratio. The following functions have to be performed[2]:

- On the establishment of an ABR connection, the user shall specify different parameters such as an initial (ICR), a minimum (MCR) and a peak (PCR) cell rate.
- The source sends RM cells, every Nrm data cells or Trm ms whichever occurs first, in order to capture congestion information among the path used by the connection under control. The ACR (Allowed Cell Rate) is the rate at which the source will send its traffic to the network according to the level of congestion sense on the connection. When a backward RM cell is received, the source will adapt its rate according to the following conditions:

  ```
  if $CI = 1$ then                              /* congestion occurs */
      $ACR = \min(ER, ACR \times RDF)$          /* $RDF < 1$ */
      $ACR = \max(MCR, ACR)$
  else if $NI = 0$ then                    /* no congestion occurs and increase allowed */
      $ACR = \min(ER, ACR + RIF \times PCR)$    /* $RIF > 0$ */
      $ACR = \min(PCR, ACR)$
  ```

  RDF (Rate Decrease Factor) and RIF (Rate Increase Factor) are constant values negotiated at call set-up, to define the additive increase and multiplicative decrease rates for ACR.
- The feedback congestion indication is achieved by closing the loop and sending congestion control indications back to the source. So the destination returns the RM cells to the source with the CI field set accordingly to the level of congestion sense among the connection.

---

[2] These specifications are conform to the Traffic Management 4.0 [Atmf95] of the ATM Forum.

- When a switch along the path receives an RM cell, it computes its level of available resource as a function of the traffic it has to forward. The parameters of the RM cell are modified according to

  $ER_{\mathrm{RM}} = \min(ER_{\mathrm{RM}}, ER_{\mathrm{Switch}})$

  $CI_{\mathrm{RM}} = CI_{\mathrm{RM}}$ or CI_{Switch}

  $NI_{\mathrm{RM}} = NI_{\mathrm{RM}}$ or NI_{Switch}.

  Where ER_{RM} is the Explicit Rate carried by the RM cell and ER_{Switch} the one locally computed by the switch (same definition for CI_{RM}, NI_{RM}, CI_{Switch}, and NI_{Switch}).

The computation of ER, in the switches, is a major challenge in the definition of the ABR services. Therefore, we present the ER based algorithm ERAQLES, and show that the ABR objectives are all reached rapidly and efficiently.

There has been numerous publications on ABR algorithms the last few years. One of the first issue was to select between credit [Kung95] or rate based. Although, the former solution exhibits some advantages, the rate-based solution was chosen [Benn94], [Vanb95]. Thereafter, explicit rate was recognized as being more efficient than CI, although it increases complexity that is a major issue in ABR design. EPRCA (Enhanced Proportional Rate Control Algorithm) [Robe94] was designed and highly cited in many papers, especially tackling performance studies [Fang94], [Kaly96], [Ritt96], [Ohsa95a], [Ohsa95b] or to suggest improvements [Siu94], [Masc96]. The ER is computed according to a congestion threshold. Other algorithms are based on a congestion threshold, improving efficiency but still holding the same general features. These algorithms are pretty simple but are unable to control situations where sources are not greedy. Moreover, due to the congestion threshold computation, they are not always fair. A second set of algorithms was initiated at Ohio State University: OSU or ERICA [Jain95b], [Jain95b], [Jain96], as well as at MIT [Char95], [Char96]. In both cases, the switch needs to have the knowledge of the sources rate in order to compare the total rate to the global switch output rate and therefore compute the ER. Efficiency is improved at the expense of an increase in complexity (if n is the number of connections flowing through a switch, some parameters are computed in O(n)). The rate allocated to the sources is often less than the available rate and fairness is not completely achieved. Most of the subsequent algorithms are a mixed of the two above mentioned ones [Barn94]. None is able to compute explicitly the total available rate. Therefore, they can not efficiently react to non greedy or bursty sources. VBR trafic is never really considered. Finally, the validation of the ABR service objectives is not fully demonstrated or only addressed through simulation observations. ERAQLES is formally demonstrated, analyzed mathematically, and through simulation.

3. ERAQLES

This algorithm was initially proposed by Fdida and Onvural [Fdid95]. A switch has to carry traffic from different connections having various requirements mapped into:

- CBR : Constant Bit Rate, which receives a peak rate allocation bandwidth.
- VBR : Variable Bit Rate, which requests a sustainable rate bandwidth.
- ABR : Available Bit Rate, which we want to maximise.
- UBR : Unspecified Bit Rate, which we will not consider in the sequel.

Therefore, we assume that, a switch is composed of three queues dedicated to the three types of traffic. A server schedules cells according to priorities indicated in Figure 1 for the CBR, VBR, and ABR queues. The ER algorithm has to compute the available bandwidth left by the VBR and CBR traffic. The parameters and variables of the ABR queue for a given switch are: $e(t)$ (number of ABR cells waiting at time t), b (size), and $n(t)$ (number of ABR connections established through the switch at time t). The capacity of a switch output link is equal to C_{tot}.

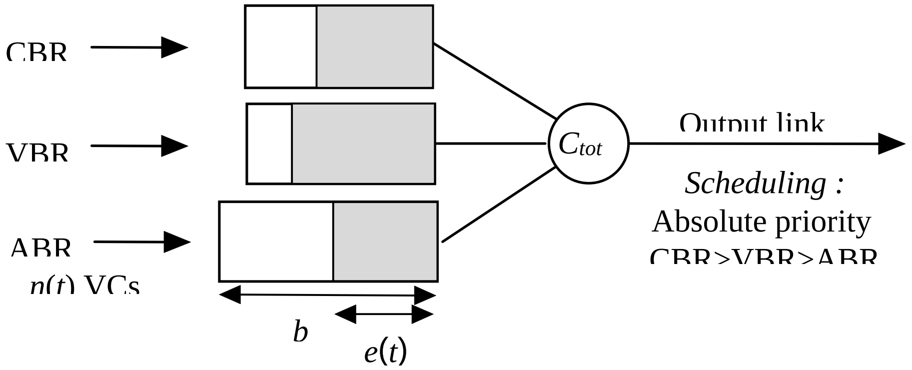


Figure 1: The switch model

3.1 Computation of $\gamma_i(t)$ the Explicit Rate for connection *i*

The Explicit Rate $\gamma_i(t)$ for connection i is computed using

$$\gamma_i(t) = \gamma(t).s_i(t)$$

where $s_i(t)$ is a fair share function and $\gamma(t)$ the total available bandwidth for ABR services. A simple choice for $s_i(t)$ is to record the number $n(t)$ of ABR virtual connections going through the output link and to set $s_i(t)=1/n(t)$. However, some connections might not need to traffic at the allocated rate. To catch the bandwidth left available by other sources that do not use their fair share, we introduce a new function

$$s_i(t) = (1 - s_0).\rho_i(t) + s_0 \qquad (1)$$

where $\rho_i(t)$ is the bandwidth ratio of connection i and s_0 the minimal value of the function. $\rho_i(t)$ can be evaluated proportional to the Current Cell Rate $CCR_i(t)$ of connection i, and the total current cell rate $CCR(t)$, assuming the source and the switch are T seconds apart

$$\rho_i(t) = \frac{CCR_i(t-T)}{CCR(t-T)}$$

If per-VC queuing is done in the switch, the above function can be modified to use the state information available for each individual connection. Then, we can define

$$\rho_i(t) = \frac{e_i(t)}{e(t)} \qquad (2)$$

where $e_i(t)$ is the number of cells in the ABR queue belonging to connection i.

3.2 Computation of $\gamma(t)$ the total available bandwidth for ABR services

We are looking for $\gamma(t)$ that is the available rate at time t, left to the ABR traffic. $\gamma(t)$ is a function of the switch utilisation as well as CBR and VBR traffics. It should provide a good statistical multiplexing. The quantity $\gamma(t)$ is derived as described in [Fdid95]. It states that the available rate is equal to the bandwidth not used by other traffics with higher priority plus the bandwidth able to fill up the buffer capacity during a control (feedback) period of $2T$:

$$\gamma(t) = C^*(t) + \frac{b}{2T}(1-\rho(t)) = C^*(t) + \frac{b-e(t)}{2T}$$

where $C^*(t)$ is the evaluation of the bandwidth left by the VBR and CBR traffics. $b/2T$ can be interpreted as the maximum amount of bandwidth that the switch can distribute at a given instant, while still being able to control the flow of cells in the following $2T$ control period. Such a function was extensively studied in [Roch95] and its properties demonstrated. It was shown that this control function pulls the ABR queue to converge to an utilisation (filling ratio) of 1. Such an objective is too optimistic because the gain in bandwidth will be balanced by an increase in the cell loss, the cell transfer delay, and cell delay variation. One goal of ERAQLES is to make the ABR queue converge to a target threshold r. Such an algorithm reduces oscillations. Therefore, we modify the resource evaluation function as follows:

$$\gamma(t) = C^*(t) + h\frac{r-e(t)}{T} \qquad (3)$$

h is an important parameter of the resource evaluation function. The larger it is, the slower the convergence. The optimum value for h is a function of the network delay (T). Optimal values for h will be analysed below (section 5.1.2). Note that in the case of low ABR buffer utilisation, the sources are allowed to exceed the capacity $C^*(t)$, by a value equal to hr, which allows for statistical multiplexing to take advantage of the available buffers. Using the function $\gamma(t)$ given by (3) to compute the Explicit Rate for all the ABR connections directed to the output link, we can show that the algorithm converges to a filling ratio of r for the ABR queue.

3.3 Computation of $C^*(t)$

$C^*(t)$ is defined as the evaluation of the bandwidth $C(t)$ left over at time t by the CBR and VBR traffics in the switch. This parameter is of utmost importance because it is used to compute the explicit rate as described above. The characterisation of CBR traffic is very simple, since it is exactly equal to the sum of the peak cell rates of all CBR connections, and thus we have $C(t) \leq C_{tot}\text{-}C_{cbr}$.

The case of VBR traffic is more complex due to its statistical behaviour, therefore, evaluation of $C(t)$ directly from the VBR sources is a difficult problem. Thus, we will use the following property: if the real value of $C(t)$ is different from $C^*(t)$, $e(t)$ will converge to $r' \neq r$, and $\gamma(t)$ to $C(t)$. In such a situation, we just need to re-evaluate $C^*(t)$ to have the ABR queue converges to r, which is its equilibrium state when $C^*(t)$ is properly evaluated. In fact, $C^*(t)$ is under-estimated when $r' < r$, and is over-estimated when $r' > r$. The evaluation of r' is obtained on averaging $e(t)$ over an update period equal to $NNrm$ RM cells. Thus, the evaluation of $C^*(t)$ can be obtained by the following formula

$$C^*(t) = C^*(t^-) + a.(r - r') \tag{4}$$

where t^- is the previous estimation time for $C^*(t)$, and a a sizing parameter.

Some operational solutions will not consider VBR service or do service provisioning, therefore the above part of the algorithm will be omitted in such situations.

3.4 Feedback delay considerations

Unfortunately, equation (3) does not account for the case where

$$T < \frac{Nrm}{2C^*(t)} = T_{\min} \tag{5}$$

which corresponds to a lightly loaded connection (less than one RM cell on the link). Therefore, the control period might be kept within the more stringent delay value. In such a case, $\gamma(t)$ might become very large. In the above we assume that all sources are equally distant from the switch, which is an unrealistic assumption. Therefore, to avoid fairness problems due to different distances from a source to the switch, we use $T_{\max}$ as the maximum propagation delay between all sources. Finally (3) becomes

$$\gamma(t) = C^*(t) + h\frac{r - e(t)}{\max(T_{\min}, T_{\max})} \tag{6}$$

4. Convergence and Fairness considerations

Fairness and convergence are two required properties that have to be demonstrated for the proper operation of an ABR algorithm. This section is devoted to that purpose. We will show that ERAQLES achieves both goals. In this section, we consider configurations with n ABR sources and one switch (Figure 2). A source A_i can be:

- a greedy source. The source is always able to use the assigned bandwidth.
- a restricted source. It is restricted to a maximum ratio bandwidth value P_i.
- a privileged source. Its MCR_i is not null, so it can overrun the assigned bandwidth up to $P_i = MCR_i / C$.

We assume that each source and the switch are T seconds apart, and that there is at least one greedy source (i.e. the total bandwidth is always used). The delay to receive $NNrm$ RM cells is called an update period. We keep the same notation otherwise. The analysis can be extended to any configuration.

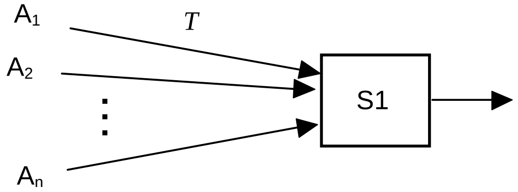


Figure 2: Generic configuration

4.1 Fairness consideration

If we consider that $\rho_i(t)$ is the bandwidth utilisation ratio for connection i, we can write

$$\rho_i(t+2T) \approx \frac{\mu_i(t)}{\mathrm{N}(t)} \tag{7}$$

where

$$\mu_i(t) = s_i(t) \text{ if } \mathrm{A}_i \text{ is a greedy source,} \tag{8}$$

$$\mu_i(t) = \min(s_i(t), \mathrm{N}(t).P_i) \text{ if it is a restricted source,} \tag{9}$$

$$\mu_i(t) = \max(s_i(t), \mathrm{N}(t).P_i) \text{ otherwise,}$$

with

$$\mathrm{N}(t) = \sum_{j=1}^{n} \mu_j(t)$$

If there is only one ABR source A_m, then we have $\rho_m = 1$ and $s_m(t) = 1$: the source receives the total bandwidth as expected. If there exist more than one ABR source, to demonstrate fairness, we compare the assigned bandwidth ratio of two arbitrary ABR sources A_l and A_m. The different cases of interest are developed in the following section.

4.1.1 $\mathbf{A}_m$ and $\mathbf{A}_l$ are greedy

We will show that $s_l(t)$ converges to $s_m(t)$. From (1), we get

$$|s_l(t+2T) - s_m(t+2T)| = |1 - s_0||\rho_l(t+2T) - \rho_m(t+2T)|$$

Thus, if A_m and A_l are greedy sources, from (7) and (8) we obtain

$$|s_l(t+2T) - s_m(t+2T)| = \frac{|1 - s_0|}{\mathrm{N}(t)}|s_l(t) - s_m(t)| \tag{10}$$

To compute $\mathrm{N}(t)$, we must define the type of the remaining n–2 sources. At time t, we assume that there are only k restricted or privileged sources which have reached their limit P, so we can write

$$\mathrm{N}(t) = \sum_{j=1}^{n-k} s_j(t) + \sum_{j=n-k+1}^{n} P_j$$

However, we have

$$\sum_{j=1}^{n}\rho_j(t)=1\Rightarrow \sum_{j=n-k+1}^{n}\rho_j(t)=1-\sum_{j=1}^{n-k}\rho_j(t)\Rightarrow \sum_{j=n-k+1}^{n}P_j=1-\sum_{j=1}^{n-k}\rho_j(t)$$

then

$$\begin{aligned}\mathrm{N}(t)&=\left(\left(1-s_0\right)\sum_{j=1}^{n-k}\rho_j(t)+\left(n-k\right)s_0\right)+\left(1-\sum_{j=1}^{n-k}\rho_j(t)\right)\\&=1+\left(\left(n-k\right)-\sum_{j=1}^{n-k}\rho_j(t)\right)s_0\end{aligned}$$

According to the assumptions and the mathematical derivations, we have

$$\begin{cases}k+1\leq n\\ \sum_{j=1}^{n-k}\rho_j(t)\leq 1\end{cases}$$

Thus we get

$$\mathrm{N}(t)\geq 1 \qquad (11)$$

and, because $0<s_0<1$, if we substitute (11) in (10) we obtain

$$\left|s_l(t+2T.i)-s_m(t+2T.i)\right|\leq B^i\left|s_l(t)-s_m(t)\right|$$

with B<1. Finally, we have

$$\lim_{i\to+\infty}\left|s_l(t+2T.i)-s_m(t+2T.i)\right|=0$$

and the two sources obtain the same bandwidth ratio. For this reason, each greedy source will receive the same share P_s of bandwidth.

4.1.2 A$_m$ is a restricted source

According to (9), if A$_m$ is a source restricted to P_m, its behaviour will be the same as a greedy source, as long as it doesn't reach P_m (i.e. $\min(s_m(t),\mathrm{N}(t).P_m)=s_m(t)$). So $\rho_m(t)$ can only converge to P_m or P_s. If we assume that it converges to:

- P_s, then there exist a fair share between A$_m$ and the greedy sources.
- P_m and $P_m \leq P_s$, then A$_m$ receives its maximum bandwidth, which corresponds to a fair share for the greedy sources and A$_m$.

- P_m and $P_m > P_s$. According to (8) and (9), and for a value t sufficiently large, we have

  $$\frac{s_s(t)}{P_s} = \mathrm{N}(t) \text{ and } \frac{s_m(t)}{P_m} > \mathrm{N}(t)$$

  so

  $$\frac{s_m(t)}{P_m} > \frac{s_s(t)}{P_s} \Rightarrow \frac{(1-s_0).\rho_m(t)+s_0}{P_m} > \frac{(1-s_0).\rho_s(t)+s_0}{P_s} \Rightarrow P_s > P_m$$

  which violates our assumption.

Thus, in all situations, the assigned bandwidth between a restricted source and a greedy source is fair.

4.1.3 The privileged sources

Similarly, we can easily show that the assigned bandwidth between a privileged source and a greedy source is fair.

4.1.4 Conclusion

Consequently, the assigned bandwidth is fairly distributed between any source and a greedy source. Therefore, any source will receive a fair share of bandwidth.

4.2 Convergence inside the update period

During an update period, $C^*(t)$ is not re-evaluated and is kept equal to C^*. We have

$$\begin{cases} CCR_i(t) = \mu_i(t-T).\gamma(t-T) \\ \gamma(t) = C^* + h\dfrac{r-e(t)}{T} \end{cases}$$

where $\mu_i(t)$ is defined by (7).

In the previous section, we have shown that ERAQLES is fair. So, if we consider that there are no transmission delays, then for a value t sufficiently large, we have

$$e(t) = e(t-\Delta) + \Delta.\left[\sum_{i=1}^{n} CCR_i(t-T) - C\right]$$

$$= e(t-\Delta) + \Delta.\left[\mathrm{N}.\gamma(t-2T) - C\right]$$

$$= e(t-\Delta) - \frac{\mathrm{N}.h.\Delta}{T} e(t - k.\Delta) + \Delta\left(\frac{\mathrm{N}.h.r}{T} + \mathrm{N}.C^* - C\right) \qquad (12)$$

where Δ is the delay between two processing epochs of RM cells

$$\Delta = \frac{Nrm}{C^*} = \frac{2T}{k} \qquad (13)$$

and

$$\mathrm{N} = \lim_{t \to +\infty} \sum_{i=1}^{n} \mathrm{N}(t)$$

If k is larger than 1, formula (12) can be simplified using the recurrent series $e_i\,(i \in N)$ as

$$e_i = e_{i-1} + \alpha . e_{i-k} + \beta \qquad (14)$$

with

$$e_i = e(i.\Delta),\ \alpha = -\frac{2h.\mathrm{N}}{k} \text{ and } \beta = \frac{2\left[\mathrm{N}.h.r + T.(\mathrm{N}.C^* - C)\right]}{k}$$

In the rest of the section, we will assume that $k \geq 1$.

4.2.1 Convergence

Assume we can write e_i as

$$e_i = f_i - \frac{\beta}{\alpha}$$

then according to (14), we obtain

$$f_i = f_{i-1} + \alpha.\, f_{i-k}$$

which is a Fibonacci serie. So, due to [fibo84], we have

$$f_i = \sum_{n=1}^{k} P_n . z_n^i$$

with z_n root of the equation

$$z^k = z^{k-1} + \alpha \qquad (15)$$

and P_n polynomial. This equation does not have a simple solution, but for some particular cases. However we know that

$$|z_n| < 1$$

and

$$|f_i| = \left|\sum_{n=1}^{k} P_n . z_n^i\right| \le \sum_{n=1}^{k} |P_n| . |z_n|^i$$

which implies that f_i converges to 0, and that the ABR queue converges to r' as expected:

$$\lim_{i \to +\infty} e_i = r + \frac{T.(\mathrm{N}.C^* - C)}{\mathrm{N}.h} = r' \tag{16}$$

4.2.2 Optimum values for *h*

We show that e_i is proportional to f_i . The smaller z^i the faster f_i converges. Therefore, In order to improve the convergence speed of ERAQLES, we have to find a value α_{opt} reducing the larger modulus of (15). Let us study the real function

$$f(x) = x^k - x^{k-1} - \alpha .$$

Its extreme solutions are $x' = 1 - 1/k$ and 0. So, if k is odd, f has three roots x_1, x_2 and x_3. We infer that the optimal min-max modulus real root x_{opt} is obtained when $x_1 = x_2 > 0$, i.e. $x_{opt} = 1 - 1/k$. So, the optimal value for α (α_{opt}) is computed for $f(x_{opt})=0$:

$$f\left(1 - \frac{1}{k}\right) = 0 \Leftrightarrow \alpha_{opt} = -\frac{1}{k}\left(1 - \frac{1}{k}\right)^{k-1}$$

The optimal h is obtained for $\alpha = \alpha_{opt}$

$$h_{\mathrm{opt}} = \frac{1}{2\mathrm{N}}\left(1 - \frac{1}{k}\right)^{k-1}$$

This formula is complicated and long to compute, therefore we use the following approximation instead assuming $\mathrm{N} = 1$ (optimization for one greedy source):

$$h_{\mathrm{opt}} = \frac{1}{2\exp(1)} \approx 0{,}184 . \tag{17}$$

4.3 Convergence outside the update period

We will demonstrate that, during an update period, $e(t)$ converges to r', and not to r as mentioned in section 3.3. Due to our assumptions, the length of these update periods are constant and equal to

$$\Delta' = \frac{NNrm.Nrm}{C}$$

Therefore, we introduce the series

$$\begin{cases} C_i^* = C^*(t + i.\Delta') \\ r_i' = r'(t + i.\Delta') \end{cases}$$

According to (4) and (16) we have

$$\begin{cases} C_{i+1}^* = C_i^* + a.\left(r - r_i'\right) \\ r_i' = r + \dfrac{T.(\mathrm{N}.C_i^* - C)}{\mathrm{N}.h} \end{cases}$$

which implies that

$$C_i^* = \frac{h.r_i'}{T} + \left(\frac{C}{\mathrm{N}} - \frac{h.r}{T}\right) \text{and so } r_{i+1}' = \left(1 - \frac{a.T}{h}\right).r_i' + \frac{a.T.r}{h}$$

We can simplify this formula by

$$r_i' = \left(1 - \frac{a.T}{h}\right)^i .r_0' + r.\left[1 - \left(1 - \frac{a.T}{h}\right)^{i-1}\right]$$

so, if we have $0 < \frac{a.T}{h} \leq 1$, r' converges to the target value r; and the optimal value a_{opt} for a is obtained when

$$\frac{a_{\text{opt}}.T}{h} = 1 \Leftrightarrow a_{\text{opt}} = \frac{h}{T} \qquad (18)$$

5. Performance Analysis

The aim of this section is to analyse, through simulation, the performance of ERAQLES, and to show that the ABR objectives are achieved. We suppose, for all configurations, that the behaviour of a source, an ATM switch and a destination follow the TM 4.0 specification

[Atmf95]. The Explicit Rates are computed according to the algorithm presented in section 3. Due to its better stability, we use (2) and $s_0 = 1/n$ to evaluate the explicit rate. $e(t)$ is the instantaneous measured value for the number of cells in the ABR queue at time t (not averaged). Unless specified otherwise, the default parameter values are $b = 10\,000$ cells, $r = 5\,000$ cells, $Nrm = 32$ cells, $NNrm = 512$ RM cells, $h = h_{opt}$ (see (17)), and $a = a_{opt}$ (see (18)). $RIF = 384\,000$ cell/s (155.5 Mb/s) so that ACR is not limited by RIF (we have always $ACR = ER$) to capture the ERAQLES properties.

ICR is set at call set-up. We use the same ICR for all connections. If ICR is set too large, the connection will start rapidly and can generate cell losses. We set ICR as follows in the simulations: if C_{link} is the smallest link bandwidth on the path, then $ICR = C_{\text{link}}/100$ (100 is an arbitrary value).

5.1 Inside the update period

In this section we compare the three main mathematical results (14), (16), (17), and the influence of transmission delays with simulation results. To guarantee that simulation results correspond to the mathematical model, we use the configuration shown in Figure 2 with $n = 1$. Therefore, we have $N = 1$.

5.1.1 Transmission delay and computation of γ

The transmission delay was not consider in the computation of the explicit rate. As long as the ratio between the transmission delay Tr and the propagation delay T is small, the model is realistic. We will explore the influence of Tr when it gets large. Tr is equal on average to r/C seconds, so according to (13), we have

$$R = \frac{Tr}{T} = \frac{2r}{k.Nrm}$$

Then, the influence of Tr increases when k gets small. We choose $C = 6\,400$ cell/s so that $k = 4$ and $R = 125$. The initial value for C^* is set to $40\,000$ cell/s. Figure 3 compares $e(t)$ obtained analytically (see (14)) and by simulation for Switch S1. In spite of the large ratio R, the two curves are closed and the influence of the transmission delay is low. In addition, they both converge to $r' = 6\,800$ as expected in formula (16).

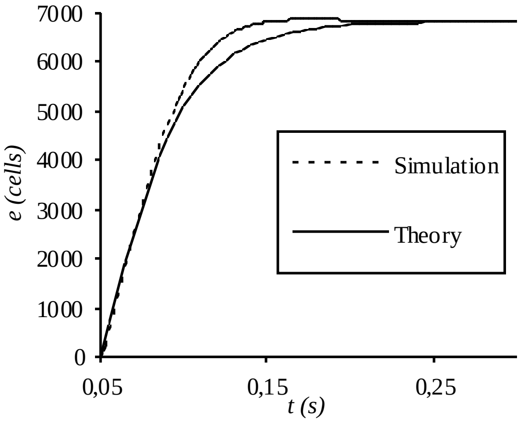


Figure 3: ABR Queue length in theory and simulation

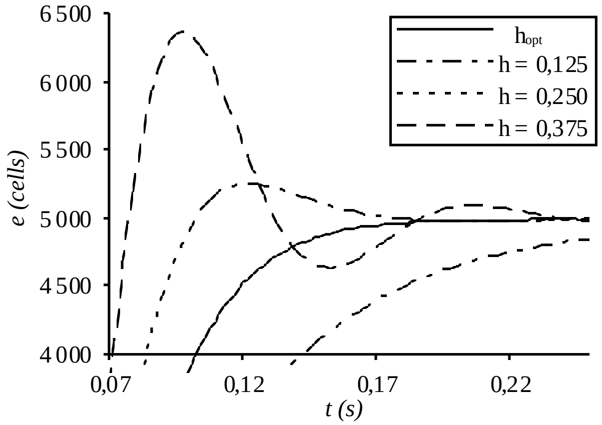


Figure 4: ABR queue length convergence speed as a function of *h*

5.1.2 Optimal value for *h*

Figure 4 shows the convergence speed of the ABR queue for different values of h. We assume C = 50 000 cell/s, $C^*(0) = C$ and different values for h (0.125, 0.250, 0.375 and h_{opt}). It is clear that h_{opt} as computed analytically provides the fastest convergence speed though reducing oscillations.

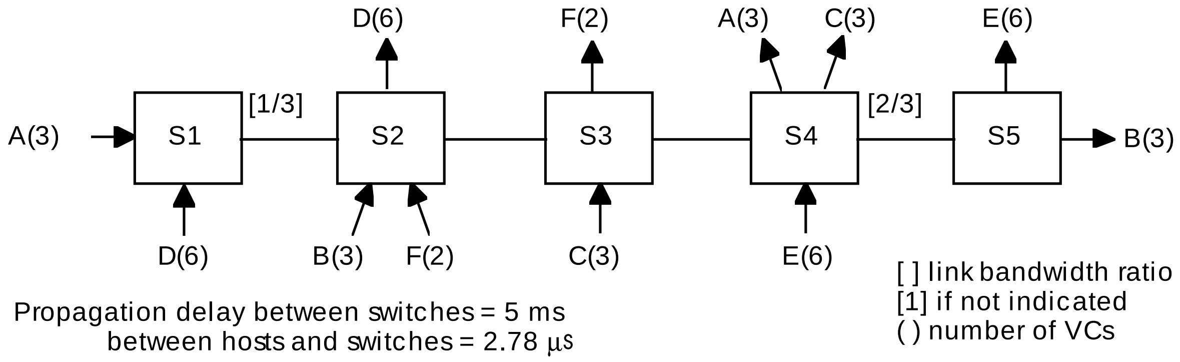
5.2 Numerical results without VBR traffic

5.2.1 Fair share

A set of Generic Fairness Configurations (GFC) has been introduced by the ATM Forum (see [Simc94] and [Bene94] for more details), to test the fairness property of the ABR algorithms with some simple network configurations. We use GFC1 (Figure 5) and GFC3 (Figure 7). All links can carry up to 384 000 cell/s (155.5 Mbps).

5.2.1.1 Analysis for GFC1

For GFC1, there are five ATM switches (S1 - S5), and six groups of greedy ABR connections (A - F). For all sources we have $C^*(0) = C$.



Propagation delay between switches = 5 ms
between hosts and switches = 2.78 µs

[ ] link bandwidth ratio
[1] if not indicated
( ) number of VCs

Figure 5: GFC1

Table 1 shows the expected and received bandwidth, as well as the bottleneck link. We see that the expected and received bandwidth for each connection are very close, and that the behaviour of ERAQLES is compliant with the ABR objectives. The ABR queue length for the four switches are presented in Figure 6. Compared with an algorithm which used EPRCA (see [Bene94]) the computation of the available bandwidth is very stable. In addition we have an efficient control on the total buffer utilisation that converges towards the target value r.

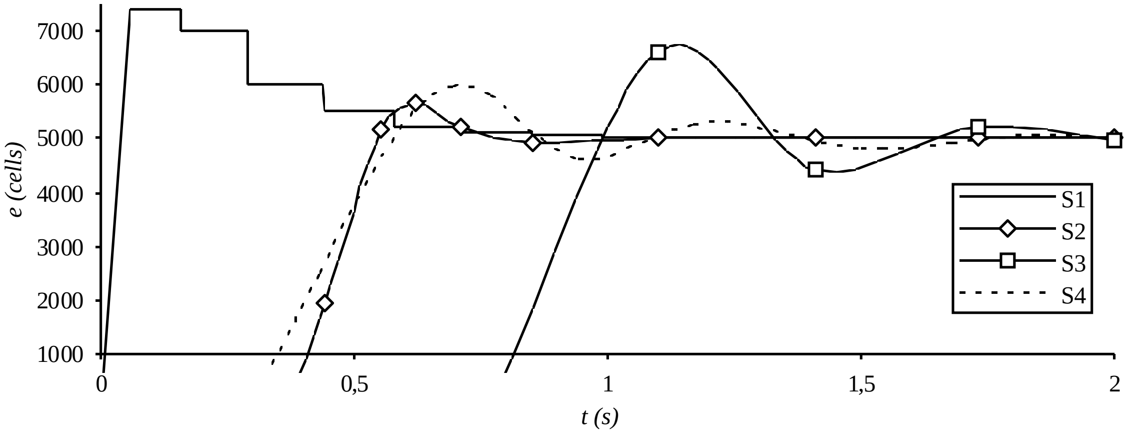


Figure 6: ABR Queue length for the 4 switches (S1-S4)

Connection	Expected bandwidth (cell/s)	Received bandwidth (cell/s)	Bottleneck link
A	$1/27C \approx 13\,315$	13 286	S1-S2
B	$2/27C \approx 26\,630$	26 618	S4-S5
C	$2/9C \approx 79\,970$	79 966	S3-S4
D	$1/27C \approx 13\,315$	13 741	S1-S2
E	$2/27C \approx 26\,657$	26 617	S4-S5
F	$1/3C \approx 119\,957$	119 925	S2-S3

Table 1: Expected and received bandwidth

5.2.1.2 Analysis for GFC3

GFC3 is composed of ten greedy ABR connections (A - J) and a single ATM switch (S1).This configuration tests the influence of the propagation delay on the connections with different values of the link bandwidth. The computation of C^* is analyzed with $C^*(0)$ = 200 000 cell/s, which corresponds to an under-estimation of the real bandwidth.

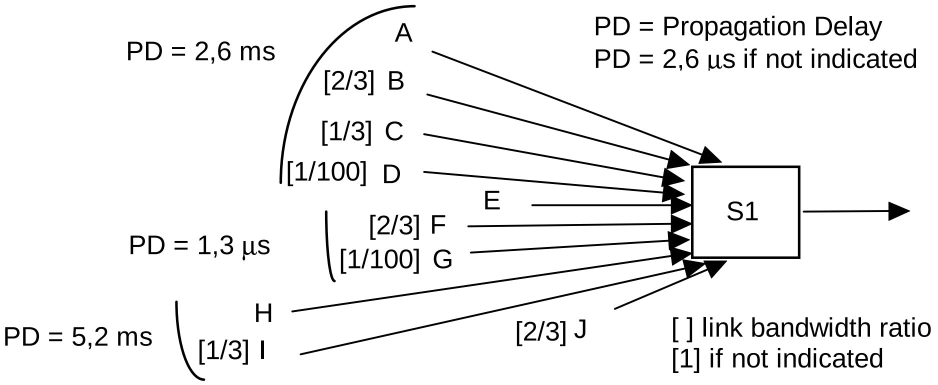
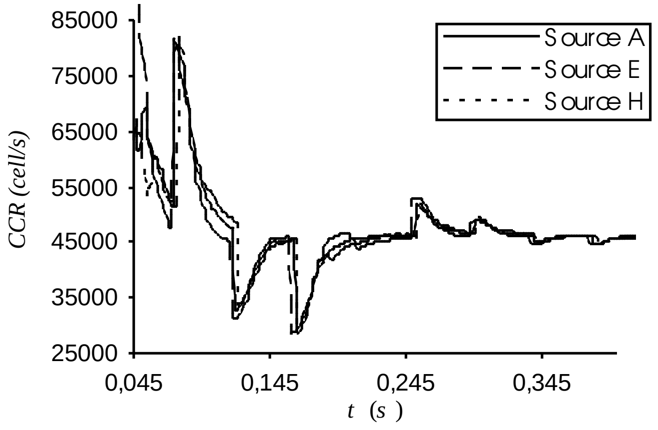


Figure 7: GFC3

With this configuration, source D and G expect 3 725 cell/s while the others will receive 45 636 cell/s. Table 2 shows the received bandwidth. We see that in spite of the variety of propagation delays and maximum link bandwiths, the expected and received bandwidth matched very well. In addition, all current cell rates of the sources are almost identical (Figure 8), therefore, sources with a small propagation delay are not priviliged and ERAQLES is found to be fair.

A	B	C	D	E	F	G	H	I	J
45 865	46 500	45 271	3 776	46 400	45 987	45 499	3 776	44 938	45 862

Table 2: Received bandwidth (cell/s)



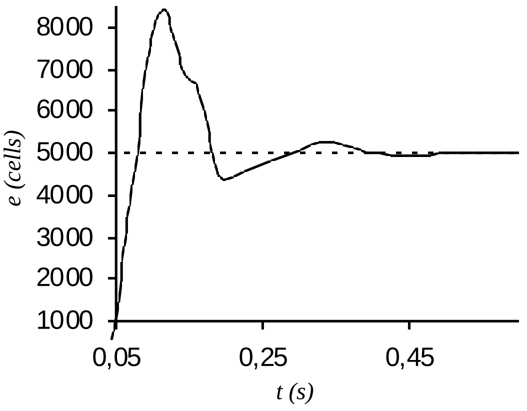


Figure 8: Current Cell Rate for sources A, E, Figure 9: Queue length and target value *r*

and H

Figure 9 shows that the ABR queue length converges to the target value r, which confirms the expected results as computed analytically.

5.2.2 Influence of the *Nrm* parameter

The ATM Forum recommends to set Nrm to 32 in order to limit the RM cells overhead to 3%. This value is not negligible because the processing of an RM cell requires computations that reduces performance, and increases with the number of connections going through the switch. We use the configuration in Figure 10 with different Nrm values (multiple of 32), and propagation delays T to explore the influence of Nrm. Sources A and B start together with $C^*(0) = C = 100\ 000$ cell/s.

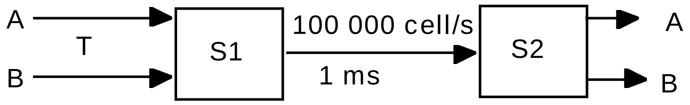


Figure 10: Configuration to explore the influence of *Nrm*

Figure 11 shows the maximum ABR queue length for S1. For $T = 10$ ms, the curve is always linear. For $T = 1$ ms, the curve is linear between $Nrm = 32$ and 256 cells, and between 512 and 4 096 cells. The knee is due to condition (5) that does not hold for $Nrm > 256$. Therefore, the behaviour of the maximum queue length can be predicted, and the Nrm value can be increased with a limited performance degradation.

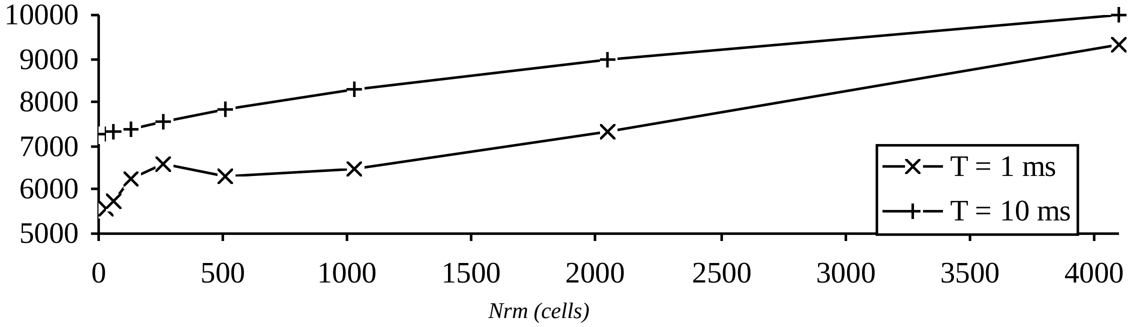


Figure 11: Influence of *Nrm* on the maximum ABR queue length

5.3 Results in presence of VBR traffic

We show that ERAQLES does converge to the expected target value as predicted analytically in presence of VBR sources. Moreover, the ABR sources are able to capture the bandwidth temporarily left unused by the variable sources.

We consider a simple configuration composed by three ABR connections A, B, C, a VBR connection D, and two ATM switches S1 and S2 (Figure 12). A and C are greedy, B is privileged with $MCR_c = 30\ 000$ cell/s, and D is ON/OFF (ON period is 1 s with a constant rate of $10\ 000$ cells/s, OFF period is set to 1 s). All sources get active at $t = 0$ but C that is activated at $t = 2$ s. S1 and S2 have a capacity of $C = 50\ 000$ cell/s, and $C^*(0)=C$. The delay between sources and a switch is 10 ms.

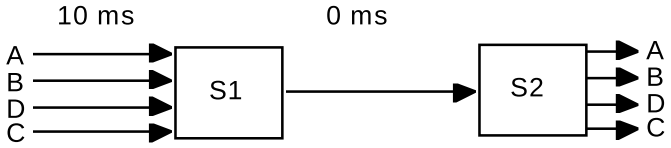


Figure 12: Mixed connection configuration

Four different periods are identified:

- T1=[0 s, 1 s[. Source D is ON. The ABR sources can only receive $40\ 000$ cells/s. Source A (resp. B) shall be allocated $10\ 000$ cells/s (resp. $30\ 000$ cells/s).
- T2=[1 s, 2 s[. D is OFF, therefore the ABR sources can receive $50\ 000$ cell/s. Source A (resp. B) will be allocated $20\ 000$ cells/s (resp. $30\ 000$ cells/s).
- T3=[2 s, 3 s[. D is ON, C is activated. Source A and C (resp. B) will receive $5\ 000$ cells/s each (resp. $30\ 000$ cells/s).
- T4=[3 s, 4 s]. D is OFF. Source A and C (resp. B) will receive $10\ 000$ cells/s each (resp. $30\ 000$ cells/s)

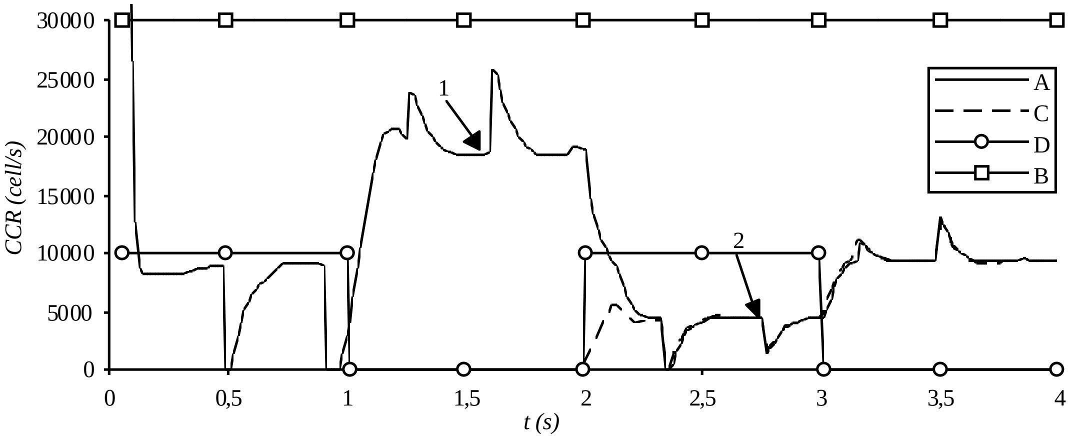


Figure 13: Transmitted cell rate for the four sources (ABR and VBR traffics)

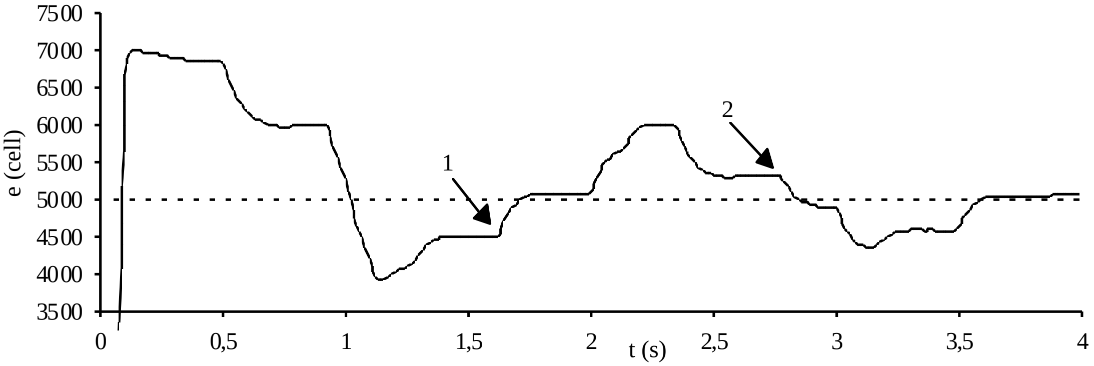


Figure 14: Queue length *e*(*t*) for the switch S1

Figure 14and Figure 13 present the results obtained by simulation. The bandwidth used by each source is fully compliant with the theoretical values. Figure 14 shows that when the rate of the VBR source is constant, e doesn't converge directly to r but in an iterative way. There exist several stages that corresponds to the update periods and successive evaluations of C^*. Arrows 1 and 2 in the figures are illustrations of this phenomena. When the average of e is equal to r the stages become invisible.

6. Conclusion

We have presented an original algorithm able to provide an efficient ABR service. The advantage of ERAQLES is to allow for a statistical gain, to control and use the buffers available in the switches and work in presence of VBR sources. We demonstrate analytically the convergence and fairness properties of the algorithm. The performance analysis, carried out by simulation, shows that ERAQLES exhibits the behaviour expected for ABR services and identify the influence of the many parameters involved. The algorithm was shown to be robust and not sensitive to a large range of parameter values. Complimentary results, presented in a companion paper, shows that it outperforms other algorithms such as OSU or ERICA, due to the control function used to compute the explicit rate. Extension to multipoint operation is also provided through the aggregation of RM cells in the switches.

7. References

[Atmf96] "Traffic Management Specification Version 4.0*", ATM Forum af-tm-0056.000*, 1996.

[Barn94] A. Barnhart, "Example of Switch Algorithm for TM Specification"*, ATM Forum contribution 95-0195*, february 1994.

[Bene94] J. C. R. Bennett, K. Chang, H. T. Kung and D. Lin, "A Comparison of EPRCA September 1994 version and FCVC Control Schemes: Simulation results", *ATM Forum 94-0929*, Ottawa, Ontario, Canada, September 26-29 1994.

[Char95] A. Charny, D.D. Clark and R. Jain, "Congestion Control With Explicit Rate Indication", *IEEE Infocom'95*, pp 1954-1963, 1995.

[Char96] A. Charny, K. K. Ramakrishnan and A. G. Lauck, "Scability Issues for Distributed Explicit Rate Allocation in ATM Networks", *IEEE Infocom'96*, pp 1198-1205, March 1996.

[Fang94] C. Fang and H. Chen, "TCP Performance of Enhanced PRCA scheme", *ATM Forum 94-0932*, September 1994.

[Fdid95] S. Fdida and R.O. Onvural, "An Explicit Rate Algorithm for ABR Services", *IBM RTP Internal report*, Research Triangle Park, NC, USA, July 1995.

[Fibo84] "Fibonacci numbers and their applications", *Proceeding of the first International Conference Numbers and their Applications*, University of Patras, 1984.

[Jain95] R. Jain, S. Kalyanaraman and R. Viswanathan, "The OSU Scheme for Congestion Avoidance in ATM Networks Using Explicit Rate Indication", *First Workshop on ATM Traffic Management WATM'95*, pp 273-280, Paris, France, December 6-8, 1995.

[Jain95] R. Jain, S. Kalyanaraman, R. Viswanathan and R. Goyal, "A Sample Switch Algorithm", *ATM Forum 95-0178R1*, February 6-10, 1995.

[Jain96] R. Jain, S. Kalyanaraman, R. Goyal, S. Fahmy and R. Viswanathan, "ERICA Switch Algorithm: a Complete Description", *ATM Forum 96-1172*, august 1996.

[Kaly96] S. Kalyanaram, R. Jain, S. Fahmy, R. Goyal, F. Lu and S. Srinidhi, "Performance of TCP/IP over ABR", *IEEE Globecom'96*, London, United Kingdom, November 1996.

[Kung95] H.T. Kung and R. Morris, "Credit-Based Flow Control for ATM Networks", *ACM Sigcomm'94*, pp 101-114, September 1994.

[Masc96] S. Mascolo, D. Cavendish and M. Gerla, "ATM Rate-Based Congestion Control Using a Smith Predictor: an EPRCA Implementation", *IEEE Infocom'96*, pp.569-576, March 1996.

[Mudd95] S. Muddu, F.M. Chiussi, C. Tryfonas and V.P. Kumar, "Max-Min Rate Control Algorithm for Available Bit Rate Service in ATM Networks", *First Workshop on ATM Traffic Management WATM'95*, pp. 297-304, 1995.

[Ohsa95] H. Ohsaki, M. Murata, H. Suzuki, C. Ikeda and H. Miyahara, "Rate-Based Congestion Control for ATM Networks", *Computer Communication Review*, vol.25, n°2, pp. 60-72, April 1995.

[Ohsa96] H. Ohsaki, G. Hasegawa, M. Murata and H. Miyahara, "Performance Evaluation and Parameter Tuning of TCP over ABR Service in ATM Networks", *IEICE Transactions on Communications*, pp. 668-83, May 1996.

[Ritt96] M. Ritter, “Network Buffer Requirements of the Rate-Based Control Mechanism for ABR Services”, *IEEE Infocom’96*, pp. 1190-1197, March 1996.

[Robe94] L. Roberts, “Enhanced PRCA (Proportional Rate-Control Algorithm)”, *ATM Forum 94-0735R1*, August 11, 1994.

[Roch95] C. Roche and S. Fdida, “A Dynamic Resource Management Mechanism for LAN Interconnection across High-Speed Networks”, *IEEE Infocom’94*, pp.884-891, Toronto, Canada, june 1994.

[Simc94] R.J. Simcoe, “Test Configurations for Fairness and Other Tests”, *ATM Forum 94-0557*, Irvine, California, USA, July 18-21, 1994.

[Syu94] KY. Siu and HY Tzeng, “Intelligent Congestion Control for ABR Service in ATM Networks”, *Computer Communication Rerview*, vol. 24, n° 5, pp. 81-106, October 1994.

[Vanb95] C. Van Boven, A. Polak and M. Gagnaire, “ABR Traffic Management in ATM Networks: a comparison of the Rate-based and of the Credit-based approaches”, *First IFIP Workshop of ATM Traffic Management WATM’95*, pp 205-220, Paris, France, 1995.