Performance Evaluation of the Rate-Based Flow Control Mechanism for ABR Service:

Generalization

Omar Ait-Hellal Eitan Altman
INRIA B.P. 93
Projet Mistral
2004 Route des Lucioles
06902 Sophia Antipolis Cedex
FRANCE
E-mail: {oaithel, altman}@sophia.inria.fr

Abstract— **In this paper we investigate the performances of the EFCI-based (Explicit Forward Congestion Indication) and ER-based (Explicit Rate; EPRCA in particular) algorithms for the rate-based flow control of the ABR (Available Bit Rate) traffic in an ATM network. We consider the case of multiple switches in tandem. We present several definitions of a bottleneck, and provide conditions that determine which queue is the bottleneck. We show that it is not necessarily the queue with the slowest transmission rate that is "responsible" for a bottleneck. We derive analytic formulas for the maximum queue length. We compare our results to those obtained by approximating a network by a simpler one, containing only the bottleneck switch. We show that the maximum queue lengths under the approximating approach may largely underestimate the ones obtained in the real network.**

Keywords—**ATM, Available Bit Rate, bottleneck, closed-loop congestion control, differential equations, fluid approximation.**

I. Introduction

AD aptive mechanisms for congestion control have a central role in the efficient sharing of the network resources between many users. These mechanisms have also the role of preventing congestion in the network [4][10]. The fact, however, that the control is performed by the sources (that are not policed by the network), and not by the network, make it hard to protect the network from applications that might not use such mechanisms (e.g. from video conferences that use UDP).

In ATM networks, the ABR (Available Bit Rate) service [1] has been defined for supporting best effort applications, in which the control decisions are taken by the network in the switches, ensuring to achieve fairness among the active connections and controlling the loss cell ratios. This service manages the bandwidth leftover by applications that have guaranteed performance (VBR and CBR), and shares it between the ABR sources by signaling to them their allowable transmission rate. The behavior of the source and destination is specified in [1] as well as the manner in which feedback information should be conveyed back to the source [1][14]. The behavior of the switches, however, is left to the designer of the switch.

Several controllers have been proposed for the switches. They are either based on (i) the EFCI (Explicit Forward Congestion Indication) bit which originates from the approach of the DEC bit [18]; it indicates, whether the congestion is detected or not, or (ii) on ER (Explicit Rate) which informs the source on the bandwidth that is available (see next section and [6][7][12][13][17][21] and references therein). The different control mechanisms include many parameters that influence the performances and the Quality of Services of the ABR connections. Some of these are fixed, such as the buffer size, the round trip times and the available bandwidth. Others may be negotiated at the session establishment. It is therefore crucial to have methods that allow the network to choose those parameters as a function of the fixed parameters and the quality of service which the network wishes to provide (in particular, the Cell Loss Ratio).

In order to propose such methods, simple queueing models have been developed and analyzed in the past years. In particular, models have been developed based on the simplifying assumption that the network can be modeled by a single bottleneck queue [5][16][19][22]. Bounds for the buffer occupancy are then computed for EFCI and ER-based switches as functions of the parameters. This approach enables then to come up with the values of the negotiated parameters so as to avoid queueing overflow.

The main purpose of our paper is to study more accurately the case when the connection actually goes across more than one switch; we examine then validity of the simplified approach of modeling the system by a single bottleneck queue.

It is easy to see that if the first switch on the path of a virtual circuit has a lower bandwidth than the others, then queueing will not occur in downstream switches, and therefore the approach of a single bottleneck can be used. We therefore restrict our study to the opposite case; we consider multiple (N) switches along the path of a virtual circuit, where the bandwidth available for the ABR traffic in the ith switch is larger than in the $(i+1)$st one.

We consider both EFCI and ER-based (EPRCA) switches. The paper has two objectives [2][3]:

- The first objective is to obtain better understanding of the bottleneck phenomenon. We show that it is not necessarily the queue with the slowest transmission rate that is "responsible" for a bottleneck. We present and examine, in particular, several alternative definitions of a bottleneck queue. We obtain conditions under which a given queue is the bottleneck.
- The second objective is a quantitative one: obtain bounds

for the buffer requirements in each queue as a function of the fixed and the negotiated parameters. This permits in particular, to compute the control parameters that are required in order to have no losses in the case that the buffer size is given a priori. These results will also allow us to compare between the maximum queue lengths in our model and the ones obtained in a single bottleneck model.

We restrict our analytical study to the case where the Nth buffer never empties (after some initial time). This regime corresponds to a full utilization of the network, which is a desirable operation condition.

In comparing the lower bounds on the maximum queue size, to the maximum queue length obtained by simulating the simplified single-queue model [16][19], we see there are cases in which the simplified model is not applicable, and large differences appear.

We present both a transient analysis, in order to obtain bounds on the queues which are uniform for all time, as well as a steady state analysis which provides smaller bounds. The steady state bounds allow us to obtain cell loss rates that are negligible in the long run using smaller buffers, at the price of having losses in some initial period.

Another reason for which the transient behavior might be of interest, is that the queue responsible for the bottleneck might change in time, as shown by simulation.

The paper is structured as follows: in section II, we describe the rate-based congestion control mechanism, and the model. We stress on the EPRCA algorithm which we consider as an ER-based algorithm. The main analytic results are presented in Section III. The exact calculations can be found in [2]. In Section IV, numerical results obtained by using our analytic formulas are compared to simulations.

II. Description and the model

In the following section we briefly describe the behavior of the *Source End System* (SES) [14], the *Destination End System* (DES) and the way control information are conveyed back to the source. The source sends data cells at rate no more than its Allowed Cell Rate (ACR) which varies according to the congestion state of the network. At a connection setup, an Initial Cell Rate (ICR), a Minimum Cell Rate (MCR, which we assume 0 in this paper) and a Peak Cell Rate (PCR) are negotiated. The source begins to send with a rate ICR, and its ACR may vary between MCR and PCR.

The ATM forum *Traffic Management Specification,* Version 4.0 [1], has specified the structure of the RM (Resource Management) cells which are sent by the source and make the round trip between the source, destination (forward RM) and back to the source (backward RM). An RM cell is sent every N_{rm} data cells. According to the degree of the network congestion, the switches may alter the content of RM cells in the two directions (forward or backward). At the arrival of a backward RM cell, the source (SES) adjusts its rate according to the *congestion indication* (CI) bit and the *Explicit Rate* field of the RM cell.

If $CI = 1$ then the ACR shall be reduced multiplicatively, by $N_{rm} * ACR/RDF$ down to MCR, where RDF is called the *Reduction Decrease Factor*. However if $CI = 0$, the ACR shall be increased by no more than $N_{rm} * AIR$ but not beyond the PCR, where AIR is called *Additive Increase Rate* (we assume that the NI bit is not used, i.e. its value is zero, as is the case in many switches). If the value of the ER field in the received RM cell is lower than the current ACR and higher than MCR, then ACR is set to this value. The CI bit and the ER field are updated by the switches in the following manner depending on the switch architecture:

A. *EFCI-based switch*

If the queue length exceeds a certain threshold Q_H (in our case each switch has a different threshold Q_i $i = 1, \cdots, N$, see Figure 1), the switch sets the EFCI bit in the header of the data cells, until the queue length goes bellow the threshold (the switch is then considered to be not congested) [17][19]. When the DES receives a forward RM cell, it sets the CI bit to the EFCI of the last received data cell. The switch may also set the bit CI of backward RM cells if congestion is detected, which we consider in this paper.

B. *ER-based switch*

The second algorithm considered in the paper is very close to the EPRCA [8][9][20][21] (Enhanced Proportional Rate Control Algorithm):

B.1 EPRCA

EPRCA is an enhancement of an older version namely PRCA (Proportional Rate Control Algorithm) [20], which is based on the congestion indication. EPRCA introduces the estimation of the available bandwidth. We can expect that if the sources send at the rate $MACR$, the fairness among the sources as well as the full utilization without losses will be achieved.

Indeed, EPRCA uses the queue length as a control parameter. More precisely, it uses two congestion thresholds: QT (Queue Threshold) and DQT in order to determine the degree of the congestion. If the congestion is detected, EPRCA has the capability to reduce selectively the rate of all the sources that have an allowed cell rate larger than $MACR$. The latter represents the mean over all the ACRs. To estimate its value, EPRCA reads the CCR field of each forward RM cell and does the following computation:

$$MACR = MACR(1 - AV) + CCR_{cell} AV$$

where AV is a constant equal to $1/16$.

Note that the mean rate (MACR) that we obtain using this method is note a mean over all the ACRs, but a value where the source that has a large ACR will have more effect. The reason is that the number of RM cells coming from a given source is proportional to the rate of that source (one RM cell every N_{rm} data cells).

Each time an RM cell is received by the switch (in preference backward RM cells), EPRCA allocates the rate to the corresponding connection depending on the queue length:

- If the queue length is smaller than QT, the switch detects no congestion and the content of the RM cell is not affected (in this case EPRCA is equivalent to EFCI algorithm)
- If the queue length is between QT and DQT, the switch considers itself as congested and updates the ER field of the RM

cell as follows: if the current cell rate CCR_{cell} of the source is larger than $MACR.DPF$, then the rate of the source shall be reduced to

$$ER = MACR.ERF.$$

DPF (Down Pressure Factor) and ERF (Explicit Reduction Factor) are constant factors and equal to $7/8$. Sources that have a current cell rate smaller than $MACR.DPF$ will not be affected.

- Finally, if the queue length exceeds DQT, then the switch is in heavy congestion and reduces selectively the rate of all the sources to

$$ER = MACR.MRF.$$

Where the value of $MRF = 1/4 < ERF$ (Major Reduction Factor) is recommended.

The ER-based switches that we considered in this work are inspired from those used in [19]. These switches are very close to EPRCA. Indeed, let consider the following parameters: $ERF = MRF$, $QT = DQT = Q_i$ where Q_i is the congestion threshold of switch i. If we consider also, that instead of estimating the MACR, which is the mean over all the ACRs, the switch knows the exact available bandwidth, then the ER-based algorithm analyzed bellow is the EPRCA algorithm.

Thus we obtain an algorithm which works as follows:

When no congestion is detected, the ACR is increased in the same manner as for EFCI-switch. If congestion is detected (queue length exceeds Q_H), the switch computes a FS depending on the available bandwidth (bottleneck rate) and on the MCRs of active connections. In our case, each switch computes its FS ($FS_i,\ i = 1, \cdots, N$) if congestion is detected. If all switches are congested, the value of the ER field is set to the $min(FS_1, \cdots, FS_N)$, which we consider equal to FS_N. The FS_i is given by $FSi = \mu_i - MCR$, where μ_i is the available bandwidth at the switch i. This FS_i multiplied by an *Explicit Reduction Factor* ($ERF_1 = ERF_2 = \cdots = ERF_N$) smaller than 1 is added to the MCR, and the resulting rate is then signaled to the source.

$$ER_i = MCR + ERF * FS_i \tag{1}$$

Hence, we define after considering $MCR_i = 0\ \forall i$

$$ER_i \stackrel{\Delta}{=} ERF * \mu_i$$

There are timers for recovering from *starvation* situations where ACR is zero for a long time (generally 100 ms). The source in this case is allowed to send RM cells called *out-of-rate*, at a rate no more than $TCR = 10\ cells/s$ with lower priority (see [14] for more details). Such mechanisms are not taken into account in the present paper.

C. The model

We consider a saturated ABR source (SES) (i.e. it has always cells to transmit) sending to a sink (DES). The source and the destination are separated by N switches, with available bandwidths of $\mu_1, \cdots, \mu_N$, respectively. We assume that $PCR \stackrel{\Delta}{=} \mu_0 > \mu_1 > \mu_2 > \cdots > \mu_N > MCR$. We assume also that the buffer sizes are sufficiently large so that no losses

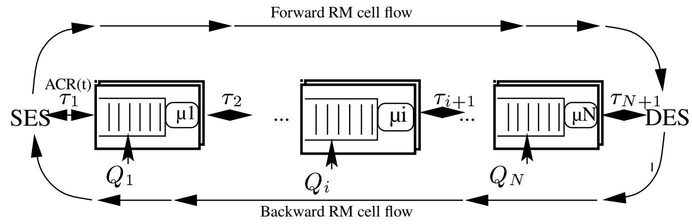


Fig. 1. The model with multiple bottlenecks

occur (we compute below sizes of the buffers that guarantee no losses). Data cells and RM cells go through the N switches before they reach the destination (Figure 1). The round trip time is τ (i.e. the time that is necessary for an RM cell to reach the destination and to return back to the source in an empty system). Cells spend τ_1 from the source to reach the first switch, and generally τ_i between the $(i-1)$st switch and the ith switch as depicted in figure 1. The time that is necessary for an RM cell to go from the last switch to the destination and back to the source in an empty system is $\tau' \stackrel{\Delta}{=} \tau - \sum_{\ell=1}^{N} \tau_\ell$.

The variation of the allowed cell rate at the source ($ACR(t)$) is cyclic. We define a cycle as the time separating the two moments when $ACR(t)$ is increasing and equal to μ_N. Define:

- $Q_i \stackrel{\Delta}{=}$ The queue threshold for the ith switch.
- $Q_i^n \stackrel{\Delta}{=}$ The queue length in the ith switch at the end of the nth cycle. $Q_i^0 = 0\ \forall i$.
- $Q_{max}^{i,n} \stackrel{\Delta}{=}$ The maximum queue length at the ith switch in the nth cycle.

By convention, we consider that the available bandwidth at the source is PCR (that is the source can't send more than PCR), hence we define $\mu_0 = PCR$.

We assume that after the first cycle, queue N never empties (i.e. full utilization. We establish later, the conditions for which this assumption holds: see equation (3) in Theorem III.3 and equation (9) in Theorem III.4) and the congestion is known by the source $\sum_{\ell=1}^{i} \tau_\ell$ after it was detected by the ith switch (i.e. time between backward RM cells is neglected). In practice, it may take longer from the time that the congestion occurs, till the source is informed about it; the extra delay is bounded by the time between two consecutive RM cells, i.e. $(N_{rm} - 1)/\mu_N$. For the transient analysis, we assume also that $ICR = \mu_N$. It is not reasonable to negotiate an ICR larger than the available bandwidth; $ICR = \mu_N$ thus corresponds to the largest initial burst.

III. Main results

In this section we present the results obtained for the transient analysis and the steady state analysis. The transient analysis will provide bounds on the maximum queue lengths Q_i^t in switch i, that will be uniform <u>for all time</u>. Thus, if the actual buffer size is larger than the computed maximum queue length, there will be no losses at any time. The steady state analysis will provide bounds for the case that the queue is initially in its steady state. We thus obtain negligible losses at the long run, for buffer sizes that are larger than the computed maximum queue sizes Q_i^s (but the buffer sizes may be smaller than the ones computed for the transient case).

Supported by simulations, we assume that a steady state periodic regime indeed exists, and is reached in finite time. The maximum length of queue i at steady state is then $Q_i^s \triangleq \overline{\lim}_{n\to\infty}(Q_{max}^{i,n}), i = 1, \cdots, N$.

Lemma III.1: **P1)** If queue i is not empty at time t then queue $i+1$ is not empty at time $t + \tau_{i+1}$.

P2) If the queue length at the ith switch exceeds the threshold Q_i, in the nth cycle, then queue $i-1$ will be empty at the end of that cycle (in fact all queues $l \leq i-1$ will be empty at the end of that cycle).

Proof: The first property is obvious, since if queue i is not empty at time t then the input rate at switch $i+1$ at time $t+\tau_{i+1}$ equals to $\mu_i > \mu_{i+1}$.

For the second property, if queue i exceeds the threshold Q_i and queue $i-1$ is nonempty, then queue i continues to increase. Thus it decreases (at time t) only if queue $i-1$ empties (at time $t-\tau_i$), which should happen (after some delay) since $ACR(t)$ decreases as long as queue i exceeds Q_i. From property (**P1**) it follows that all queues $l < i-1$ will be empty. We shall focus, mainly, on the case when queue N is the bottleneck (see definition III.1). ■

Definition III.1: *i*) Queue i is the strict bottleneck if and only if queue j ($j \neq i$) remains always empty (from the properties above, only queue N can be the strict bottleneck).
ii) Queue i is the essential bottleneck if and only if queue j ($j \neq i$) never reaches the threshold Q_j.
iii) We say that the congestion is always due to the ith switch if and only if queue i reaches its threshold Q_i, not later than $\sum_{\ell=1}^{i}\tau_\ell - \sum_{\ell=1}^{j}\tau_\ell$ before queue j ($\forall j \neq i$) reaches its Q_j in all cycles (the first RM cell indicating congestion is updated by switch i).

Remark III.1: Queue one can not be a strict bottleneck node.

Lemma III.2: If $Q_i^n \geq Q_i^m$ for some n and m then $Q_{max}^{i,n} \leq Q_{max}^{i,m}, i = 1, \cdots, N$.

Theorem III.1: When the congestion is always due to the Nth switch, the maximum queue sizes are obtained in the first cycle:

$$Q_i^t \triangleq \max_{n\geq 0}(Q_{max}^{i,n}) = Q_{max}^{i,0}$$

Proof: We first note that $Q_i^n = 0 \; \forall n$ for $i < N$. This is due to property **P2**. For all n we have $Q_i^n \geq Q_i^0$, since we assume that the queue i is initially empty. We now make use of the following monotone property: Assume that $Q_i^n \geq Q_i^m$ for some n and m. Then the time that queue i takes to reach Q_i in the nth cycle is shorter than the time it takes to reach Q_i in the mth cycle. It now follows that the ACR in cycle n at time $T_{Q_i}^n - \sum_{\ell=1}^{i}\tau_\ell + t$ is smaller than the ACR at cycle m at time $T_{Q_i}^m - \sum_{\ell=1}^{i}\tau_\ell + t$ for $t \geq 0$. It then follows that the size of the ith queue in cycle n after time $T_{Q_i}^n + t$ is smaller than in cycle m at time $T_{Q_i}^m + t$, $t \geq 0$. The proof follows by taking $m = 0$ and arbitrary n. ■

Remark III.2: In the proof of Theorem III.1, we established also the proof of Lemma III.2.

Define $A_{i,n}, B_{i,n}, C_{i,n}$ as

$$A_{i,n} \triangleq \begin{cases} \sqrt{\dfrac{2(Q_i - Q_i^n)}{AIR * \mu_N}} + \dfrac{\mu_i - \mu_N}{AIR * \mu_N} & \text{if } \dfrac{\mu_{i-1} - \mu_i}{AIR * \mu_N} \geq \sqrt{\dfrac{2(Q_i - Q_i^n)}{AIR * \mu_N}} \\[2ex] \dfrac{Q_i - Q_i^n}{\mu_{i-1} - \mu_i} + \dfrac{\mu_{i-1} + \mu_i - 2\mu_N}{2 * AIR * \mu_N} & \text{otherwise} \end{cases}$$

$B_{i,n} \triangleq \left(2\sum_{\ell=1}^{N}\tau_\ell + A_{N,n} - \frac{\mu_{i-1}-\mu_N}{AIR*\mu_N}\right)^+$, and
$C_{i,n} \triangleq min\left(2\sum_{\ell=1}^{N}\tau_\ell + A_{N,n} - \frac{\mu_i-\mu_N}{AIR*\mu_N}, \frac{\mu_{i-1}-\mu_i}{AIR*\mu_N}\right)$.
where $(x)^+ \triangleq max(0,x)$. Q_i^n is given in Theorem III.3 and Theorem III.4.

A. Results for the Transient case

Theorem III.2: For both EFCI-based switch and ER-based switch, we have
i) Queue i starts to build first, if and only if
$$\frac{\mu_i - \mu_j}{AIR * \mu_N} \leq \sum_{\ell=i+1}^{N}\tau_\ell - \sum_{\ell=j+1}^{N}\tau_\ell \;, \; \forall j \neq i.$$
Particular case: $\tau_i = \tau_j = \tau_1 \; \forall i,j$, we have queue i starts to build first if and only if
$$\frac{\mu_i - \mu_j}{AIR * \mu_N} \leq (j-i)\,\tau_1, \quad \forall j \neq i$$
ii) The following are equivalent:
- (ii.a) The congestion is always due to the ith switch.
- (ii.b)
$$\sum_{\ell=j+1}^{N}\tau_\ell + \sum_{\ell=1}^{i}\tau_\ell - \sum_{\ell=1}^{j}\tau_\ell - \sum_{\ell=i+1}^{N}\tau_\ell \leq A_{j,n} - A_{i,n}, \; \forall n, \quad j < N.$$

Particular case: for switch N, the following are equivalent:
- (ii.a.1) The congestion is always due to the switch N.
- (ii.b.1) $2\sum_{\ell=j+1}^{N}\tau_\ell \leq A_{j,n} - A_{N,n}, \; \forall n, \quad j < N,$.
- (ii.c.1) $2\sum_{\ell=j+1}^{N}\tau_\ell \leq A_{j,0} - A_{N,0}, \quad \forall j < N$

Theorem III.3: If the congestion is always due to the Nth switch, then we have
i) For both EFCI and ER-based switch, the following are equivalent:
- (i.a) Queue N is the strict bottleneck
- (i.b) $2\sum_{i=1}^{N}\tau_\ell \leq \dfrac{\mu_i - \mu_N}{AIR * \mu_N} - A_{N,n}\;; \; \forall n, \; i < N.$
- (i.c) $2\sum_{i=1}^{N}\tau_\ell \leq \dfrac{\mu_{N-1} - \mu_N}{AIR * \mu_N} - A_{N,0}$

i.1) and the following also are equivalent
- (i.1.a) All the queues build up (in the nth cycle)
- (i.1.b) $C_{1,n} > 0$

ii) For the EFCI-based switch, if queue N is not the strict bottleneck, then the maximum queue length at switch i (Q_i^t which is obtained in the first cycle, $Q_i^0 = 0\ \forall i$) is given by

$$Q_1^t = (\mu_0 - \mu_1) B_{1,0} - \frac{RDF * \mu_1}{\mu_N} \log\left(1 + \frac{AIR * \mu_N}{\mu_1} C_{1,0}\right) + \frac{AIR * \mu_N}{2} {C_{1,0}}^2 + RDF * AIR * C_{1,0}$$

and for $i > 1$, we have

$$\begin{aligned} Q_i^t \cong\ & \frac{AIR * \mu_N}{2} {C_{i,0}}^2 \\ & + (\mu_{i-1} - \mu_i) \left[\sum_{j=1}^{i-1} \left(\frac{Q_j^t}{\mu_j} + \frac{RDF(\mu_j - \mu_{j+1})}{\mu_N \mu_j}\right.\right. \\ & \left.\left. + \frac{RDF}{\mu_N} \log\left(1 + \frac{AIR * \mu_N}{\mu_1} C_{1,0}\right) + B_{i,0}\right]\end{aligned} \quad (2)$$

The following bounds hold (for the remaining cells in queue N at the end of a given cycle n):

$$\left(Q_N - \mu_N \left(\tau + \frac{1}{2 * AIR}\right)\right)^+ \le Q_N^n$$

$$\begin{aligned} Q_N^n \le & \left(Q_N + RDF\left(1 - e^{-\frac{\mu_N}{RDF}\tau}\right)\right. \\ & \left. - \frac{\mu_N}{2 * AIR} \left(2 * AIR * \tau + \left(1 - e^{-\frac{\mu_N}{RDF}\tau}\right)^2\right)\right)^+ \end{aligned} \quad (3)$$

iii) For the ER-based switch, the maximum queue length at the switch i (Q_i^t) is given by

$$\begin{aligned} Q_i^t =\ & \frac{AIR * \mu_N}{2} {C_{i,0}}^2 \\ & + (\mu_{i-1} - \mu_i) \left(\sum_{\ell=1}^{i-1} \frac{Q_\ell^t}{\mu_\ell - ER_N} + B_{i,0}\right) \end{aligned} \quad (4)$$

iv) Queue N is the essential bottleneck (ER-based switch) if

$$\sum_{\ell=1}^{i-1} \frac{Q_\ell^t}{\mu_\ell - ER_N} < \sqrt{\frac{2Q_i}{AIR * \mu_N}} - B_{i+1,0} \quad \forall i < N$$

The proofs can be found in [2].

B. *Results for the steady state case*

In what follows, we shall restrict to the case (in steady state) where *the congestion is always due to* the Nth switch. Since we assume steady state, this is less restrictive than the corresponding assumption for the transient behavior.

Theorem III.4: If queue N is not the strict bottleneck and the congestion is always due to the Nth switch, then

i) For the EFCI-based switch, the maximum queue length at switch i (Q_i^s) is given by

$$\begin{aligned} Q_1^s =\ & Q_1^\infty + \frac{AIR * \mu_N}{2} {C_{1,\infty}}^2 + (\mu_0 - \mu_1) B_{1,\infty} \\ & - \frac{RDF * \mu_1}{\mu_N} \log\left(1 + \frac{AIR * \mu_N}{\mu_1} C_{1,\infty}\right) \\ & + RDF * AIR * C_{1,\infty} \end{aligned} \quad (5)$$

and for $i > 1$ we have

$$\begin{aligned} Q_i^s \cong\ & Q_i^\infty + \frac{AIR * \mu_N}{2} {C_{i,\infty}}^2 \\ & + (\mu_{i-1} - \mu_i) \left[\sum_{j=1}^{i-1} \left(\frac{Q_j^s}{\mu_j} + \frac{RDF(\mu_j - \mu_{j+1})}{\mu_N \mu_j}\right.\right. \\ & \left.\left. + \frac{RDF}{\mu_N} \log\left(1 + \frac{AIR * \mu_N}{\mu_1} C_{1,\infty}\right) + B_{i,\infty}\right] \end{aligned} \quad (6)$$

Where $Q_i^\infty = 0\ \ \forall i < N$, and

$$Q_N^\infty = \left(Q_N - \mu_N \left(\tau + \frac{1}{2 * AIR}\right)\right)^+ \quad (7)$$

ii) For the ER-based switch, the maximum queue length at the ith switch (Q_i^s) is given by

$$\begin{aligned} Q_i^s =\ & Q_i^\infty + \frac{AIR * \mu_N}{2} {C_{i,\infty}}^2 \\ & + (\mu_{i-1} - \mu_i) \left(\sum_{\ell=1}^{i-1} \frac{Q_\ell^s}{\mu_\ell - ER_N} + B_{i,\infty}\right) \end{aligned} \quad (8)$$

Where $Q_i^\infty = 0\ \ \forall i < N$, and

$$Q_N^\infty = \left(Q_N - (\mu_N - ER_N)\left(\tau + \frac{\mu_N - ER_N}{2 * AIR * \mu_N}\right)\right)^+ \quad (9)$$

The proof of the theorem above can be found in [2].

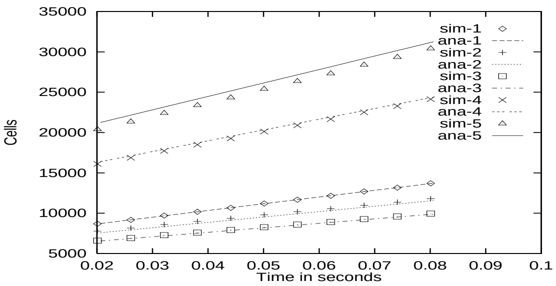
IV. Numerical results and discussions

In this section we present some numerical examples, in order to validate the assumptions considered a long the paper, by using simulations. Then we compare the obtained results in our model to those obtained by using the approximating model in which the network is replaced by a single bottleneck node (exactly the same parameters as for our model without the $N-1$ first switches (switches 1 to $N-1$ are deleted)). Only the transient state (first cycle) is taken into account, since the maximum queue lengths are obtained in this case (see Theorem III.1). We compute the queue length (Q_i^t) in each switch as a function of $2\sum_{\ell=1}^N \tau_\ell$.

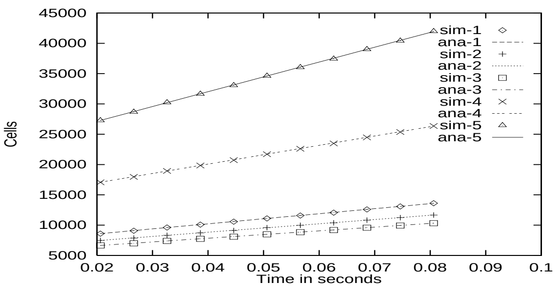
As we have seen (Theorem III.3), the maximum queue length in the switches during the first cycle is not a function of the round trip time τ. This is because switches have the capability to alter the content of backward RM cells (i.e. when congestion is detected, also RM cells in backward direction are updated). However, in the steady state, the maximum queue lengths are function of both τ and $2\sum_{\ell=1}^N \tau_\ell$, since queue N can be nonempty at the end of the cycle ($Q_N^n > 0$ which is function of τ, see equations (7) and (9)). The sufficient conditions that allow a full utilization can be obtained for both ER-based and EFCI-based switches from equations (7) and (9).

Remark IV.1: As discussed in section II, the time between RM cells is neglected in the paper. However, this time will have some effects on the actual results. To take this time into account, we added the quantity $(N_{rm} - 1)/\mu_N$ to $A_{i,0}$, when computing analytically the queue lengths (ana-q_i in the figures), since when the congestion occurs at switch i, it is detected by the source not less than $\sum_{\ell=1}^i \tau_\ell$ after.

In all the following simulations, we considered $PCR = 141.33\ Mbps = 333333.25\ cells/sec$ and hence the transmission time of a cell is about 3 μsec, $MCR = 0.0\ Mbps$ $AIR = 0.1\ Mbps \cong 250\ cells/sec$, $RDF = 512$ and $ERF = 0.8$. We will take different thresholds for each queue, in order to show the effect of nonhomogeneous queues on the maximum queue length at the Nth switch.



(a) EFCI-based switch



(b) ER-based switch

Fig. 2. Queue length (cells) as a function of the round trip time (seconds).

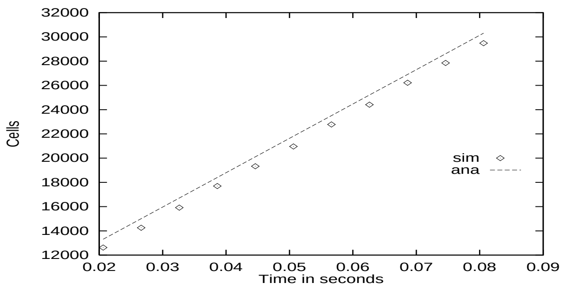
In the first example we consider five switches with the following thresholds: $Q_1 = 20000$, $Q_2 = 15000$, $Q_3 = 10000$, $Q_4 = 10000$ and $Q_5 = 5000$. The available bandwidth in each switch are respectively $\mu_1 = 106\ Mbps = 250000\ cells/sec$, $\mu_2 = 84.8\ Mbps = 200000\ cells/sec$, $\mu_3 = 70.66\ Mbps = 166666.67\ cells$ sec, $\mu_4 = 42.4\ Mbps = 100000\ cells/sec$ and $\mu_5 = 21.2\ Mbps = 50000\ cells/sec$, then we have $A_{1,0} = 0.2593$, $A_{2,0} = 0.314$, $A_{3,0} = 0.3106$, $A_{4,0} = 0.1566$ and $A_{5,0} = 0.102$. From Theorem III.2 we have, *the congestion is always due to* queue N if and only if $2\sum_{\ell=j+1}^{5} \tau_\ell \leq A_{j,0} - A_{N,0}$, $\forall j < 5$, which is equivalent to
$\tau_5 \leq 0.0273\ sec$, $\tau_4 \leq 0.1043 - \tau_5\ sec$,
$\tau_3 \leq 0.106 - \tau_4 - \tau_5\ sec$, $\tau_2 \leq 0.078667 - \tau_3 - \tau_4 - \tau_5\ sec.$

On the other hand, from Theorem III.3 *(i.1)*, we must have $\sum_{\ell=1}^{5} \tau_\ell > -0.086$ which means that all the queues build up. We choose the following initial values for $\tau_i,\ i = 2 \cdots 5 : \tau_5 = 2\,ms$, $\tau_4 = 2\,ms$, $\tau_3 = 2\,ms$ and $\tau_2 = 1ms$. When we compare the analytical results to the simulation ones as function of the round trip time, we increase only τ_1 (from 3 ms to 30 ms by a step of 3 ms, see figure 2) since the queue lengths are function of $\sum_{\ell=1}^{N} \tau_\ell$ (τ_6 is set to 5 ms, it has no effect, for the transient case, since backward RM cells are updated).

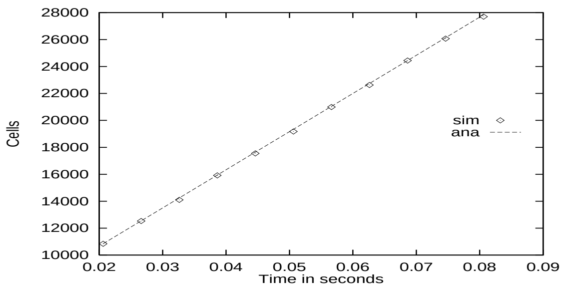
Figure 2 illustrates the maximum queue length in each queue as a function of the round trip delay $2\sum_{i=1}^{N} \tau_i$ for the example above with five switches in tandem, in case of ERFCI-based (a) as well as for ER-based switches (b). We can see that the analytic results (ana) are very close to those obtained by simulation (sim), and the error is almost negligible for ER-based switches.

In general, it is well known that the maximum queue lengths in EFCI-based switches are larger than those obtained for EPRCA-algorithm (ER-based switches). However, in this example we observe the opposite. This is the first wrong conclusion which arises when considering the single node simplified model. The reason is that when a congestion is detected in ER-based switches, the source reduces its rate to a constant rate ($ER_5 = 0.8\mu_5 = 40000 >> 0$). The latter may be very larger than zero, and the difference $\mu_{i-1} - ER_5$ is not so significant, in particular for $i = 4, 5$ in our example. Hence, the queues empty very slowly (see equation 4) in opposite to the case of EFCI-based switches, where the rate of the source reaches a zero rate (or MCR) within a short period after the congestion is detected.

To get the maximum queue length under the simplified model, one can consider the same model studied here (even if we assumed the number of switches larger than one $N > 1$), and use the results from the previous theorems. Indeed, by considering the model described in section II with $\mu_0 = \mu_1 = \cdots = \mu_{N-1} = PCR$ ($\mu_N < PCR$, we get $\forall \tau_i, i = 1, \cdots, N+1$ *the congestion is always due to* switch N), we derive the analytic formulas for computing the maximum queue length in a single bottleneck model.



(a) EFCI-based switch



(b) ER-based switch

Fig. 3. Queue length as a function of the round trip time for the single bottleneck model

Figure 3 illustrates the maximum queue length obtained analytically and by simulation, in the example above after deleting the $N-1$ first switches (single bottleneck model). The maximum queue length is given as a function of the round trip time $2\sum_{i=1}^{N}\tau_i$. We can see that in this example however, the maximum queue length in EFCI-based switch is larger than that obtained using ER-based switch. Thus, the number of switches crossed by a given VC has more negative effects on ER-based switches than EFCI-based ones.

A. *Difference in queue length between our model and the single bottleneck model*

In order to show the difference in queue length in the "real" network and the simplified model which consists in simplifying the network in a single node (the slowest node), we considered another example. In this example, the source goes trough only two switches. The available bandwidth is $\mu_1 = 40.435791\ Mbps$ *resp.* $\mu_2 = 4.043\ Mbps$ and the congestion threshold is $Q_1 = 5000\ cells$ *resp.* $Q_2 = 5000\ cells$, in the first *resp.* the second switch. From Theorem III.2 and Theorem III.3 we get the following results:
For both EFCI and ER-based switch, queue two starts to build before queue one if and only if $\tau_2 < 0.036\ sec$ and *the congestion is always due to* the second switch if and only if $\tau_2 \leq 0.0128\ sec$. Queue two is the *strict bottleneck* if and only if $2(\tau_1+\tau_2) \leq -0.037556\ sec$ which means that queue two cannot be the *strict bottleneck*. However it is the *essential bottleneck* for the EFCI-based resp. ER-based switch if $\tau_1+\tau_2 < 0.002165$ resp. if $\tau_1+\tau_2 < 0.012845$.

The differences in queue length between the single bottleneck approximating model and ours is of the order of the threshold value $Q_1 = 5000$ which is relatively high, when considering queue lengths of 18000 cells. This is illustrated in Figure 4 (diff-5000) where the difference between the maximum queue lengths in the two queues case and in the single queue case are given as a function of $2(\tau_1+\tau_2)$, for different values of the threshold Q_1 (5000 and 15000), $Q_2 = 5000\ cells$. The maximum queue length obtained in a single bottleneck model is also plotted (single in figure 4).

In this example only $\tau_1+\tau_2$ is varying. As initial values, we have considered $\tau_1 = 0.263\ ms$, $\tau_2 = 0.35\ ms$. Each time we add the same value to both τ_1 and τ_2 till $\tau_2 = 0.012\ ms$. Note that we obtain the same results if we vary only τ_1 or τ_2.

From the formulas in section III, we can see that Q_2^t increases linearly as Q_1^t grows. Q_1^t is in particular a function of μ_1; the closer $\mu 1$ is to μ_2, the larger is the queue length at the first switch. Hence, by decreasing μ_1, we can obtain cases in which the queue length in the second switch is significantly larger compared to the single bottleneck approximating model, while μ_2 is the same for both models. However, the conditions for *the congestion to be always due to* the second switch also change. Hence an adequate Q_1 should be chosen in order to satisfy them when μ_1 is decreased (we wish to have the same properties as in the previous example, and show how the decrease in μ_1 and increase in $Q1$ affects the maximum buffer size at the second switch). Note that the largest queue lengths at the second switch occur when *the congestion is always due to* this switch.

As an illustration, consider the previous example with $\mu_1 = 26.957194\ Mbps$. Our aim is to determine Q_1 such that *the congestion is always due to* the second switch if and only if $\tau_2 \leq 0.0128\ sec$. We have

$$\frac{\mu 1-\mu_2}{AIR*\mu_2} = 0.0226 \leq \sqrt{\frac{2Q_2}{AIR*\mu_2}} = 0.0632$$

$$\Longrightarrow A_{2,0} = 0.07693\ sec.$$

$$\text{Assume that}\quad \frac{PCR-\mu_1}{AIR*\mu_2} \leq \sqrt{\frac{2Q_1}{AIR*\mu_2}}.$$

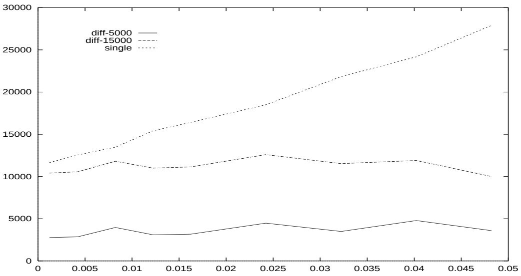
$$\text{Then}\quad Q_1 \leq 15308.88, \quad\text{and}\quad A_{1,0} = \frac{1}{500\sqrt{5}}\sqrt{Q_1}.$$

From Theorem III.2, *the congestion is always due to* the second switch if and only if

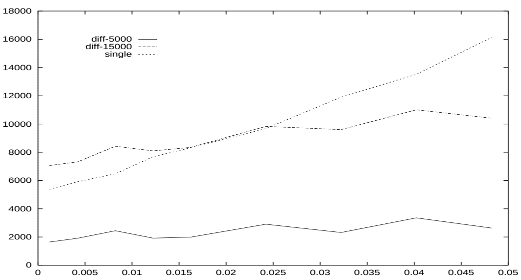
$$0.0296 \leq \frac{1}{500\sqrt{5}}\sqrt{Q_1} - 0.07693 \quad\text{that is}\quad Q_1 \geq 14185.801$$

Figure 4 shows the differences in queue lengths between the model with two queues (diff-5000, diff-15000) and the approximating model which consists of a single bottleneck. The queue differences are much bigger in the second case (diff-15000) when a threshold of $Q_1 = 15000\ cells$ and $\mu_1 = 26.957194\ Mbps$ are considered. In some cases, especially for the ER-based switch, they are equal to the queue length of the approximating model (that is, the queue length at the second switch in the model with two queues is twice bigger than that in the approximating model).

The decrease in the difference of the queue length (diff-15000) in the Figure 4 at $2(\tau_1+\tau_2) = 0.048\ sec$ is due to the fact that ACR reaches PCR in the model with two queues, while in the approximating model it does not.



(a) EFCI-based switch



(b) ER-based switch

Fig. 4. The difference in queue length between our model and the single bottleneck approximating one

The numerical examples given in the present section concern only transient phases, however the same analysis can be done for the steady state and the same qualitative results (buffers in the two models are smaller) can be obtained.

V. CONCLUDING REMARKS

Generally, the thresholds are recommended to be some given fraction of the buffer capacity. The situation where switches have different buffer sizes may be quite common in practice. This will then typically imply the general situation we studied of different thresholds in different switches.

As shown in the example above, it is always possible to find out, Q_1, Q_2, τ μ_1 and μ_2 such that the queue length at the second switch in our model equals to twice, tree times or more the queue length in case of the approximating model that consists of a single bottleneck. Thus, the single bottleneck model doesn't allow to determine the real bounds for the queue length, especially for the EFCI-based switch, which depends on the number of switches that the connection goes across.

This implies in particular, that the buffer requirements of connections that have many hops might be extremely large. This is an important drawback of the *Congestion Indication* (single bit) scheme that is revealed by our analysis.

In the presence of several nodes on a virtual connection in which flow control is performed, the maximum queue lengths obtained when using EPRCA algorithm might be larger than those obtained under EFCI-based switches. The dynamics of the queue length has been shown also to be considerably more complex than in the case of a single node. Non-intuitive behavior was shown to arise, in which congestion actually occurs in a node which is not necessarily the slowest one. This fact, together with analytic bounds that we obtained suggested that one has to be very careful in using approximations based on a single bottleneck link (corresponding to the slowest one).

We focused on the case where the ith link is faster than the $i+1$st, since in the opposite case, the faster link can be ignored: it will have an input rate smaller than its output rate, and will thus always be empty. In our case however, the following may occur

- The queue in the faster node may start to build before the one in the slower node (Theorem III.2 (i)),
- When queueing occurs in two successive nodes, the rate at which the queue size increases might be *larger in the faster queue than in the slower one*! This behavior is seen in Fig. 2 (switch 3 with switch 2 or 1). The reason is that the rate of transmission may increase rapidly at the input of the faster queue, whereas the rate into the slower queue might be considerably lower, as this input stems from the output of the previous (faster) queue. The input rate into the slower queue is thus bounded by the output rate of the faster node, whereas the input rate into the faster node can be indirectly bounded by the control mechanism (for switch 1 for instance), and might be much larger. We obtain a phenomenon in which the faster node "protects" the slower node, as it does not allow that queue to build up fast.
- An important phenomenon seen, is that congestion in the slowest queue (queue N) might last and even become worse much longer after the source starts to receive indication of congestion. In fact, the queue in the slowest node continues to *grow up* after the congestion has been indicated, until the queues in the predecessor switches empty.

This type of huge overshoot does typically not occur in the case of a single node.

We presented an analysis for both the transient as well as the steady-state regime. In our transient analysis we provided upper bounds on the queue lengths that are uniform in time. By using these formulas to design the control parameters, we may guarantee that no losses occur. This analysis may be too conservative if losses in the transient phase are acceptable, and therefore the less conservative bounds for the steady state can be used instead.

REFERENCES

[1] The ATM forum Technical Committee, *Traffic Management Specification,* Version 4.0, April 1996.

[2] O. Ait-Hellal, E. Altman, *Performance Evaluation of the Rate-Based Flow Control Mechanism for ABR Service: Generalization*, manuscrip. http://www-sop.inria.fr/mistral/personnel/Omar.Ait-Hellal/pub_list.html.

[3] O. Ait-Hellal, E. Altman, D. Elwadghiri and M. Erramdani, *Performance Evaluation of the Rate-Based Flow Control Mechanism for ABR Service*, INRIA Research Report No. RR-3131. To appear in Journal of telecommunication systems.

[4] L.S. Brakmo, L.L. Peterson, *TCP Vegas : End to End Congestion Avoidance on a Global Internet*, IEEE JSAC, vol. 13, pp. 1465-1480, 1995.

[5] M. O. Brouillet and U. Madhow, '*Rate control for adaptive bit rate sources on ATM networks using one bit congestion notification*, Research report UILU-ENG-96-2214, Univ. of Illinois at Urbana-Champaign, May, 1996.

[6] Y. Chang, N. Golmie, D. Siu, *A Rate-Based Flow Control Switch Design for ABR Service in an ATM Network*, Proc. ICCC'95, August 1995.

[7] A. Charny, D.D. Clark, R.Jain, *Congestion Control With Explicit Rate Indication*, Proc. ICC'95, June 1995, 10 pp.

[8] H. Hsiaw, L. Roberts, A. Jain, B. Makrucki, T. Tofigh, A. W. Barnhart, B. Holden, G. Fedorkow, J. N. Daigle, D. Kline, H. Suzuki, G. Ramamurthy, N. Giroux, R. Kositpaiboon, D. Hughes, M. G. Hluchyj, G. Garg, K. Y. Siu, H. Y Tzeng, N. Yin, *Closed-Loop Rate-Based Traffic Management*, ATM Forum Contribution 94-0438R2, september 1994.

[9] C. Ikeda, H. Suzuki, M. Murata, *Transient State Analysis and Maximum Buffer Requirements for Enhanced PRCA*, ATM Forum Contribution 94-0910, september 1994.

[10] Van Jacobson, *Congestion avoidance and control*, ACM SIGCOMM 88, pages 273-288, 1988. ftp.ee.lbl.gov/papers/congavoid.ps.Z

[11] R. Jain, S. Kalyanaraman, R. Goyal, S. Fahmy, R. Viswanathan, *ERICA switch algorithm: A Complete Description*, ATM Forum/96-1172.

[12] R. Jain, S. Kalyanaraman, R. Viswanathan, *The OSU Scheme for Congestion Avoidance in ATM Networks using Explicit Rate Indication*, Proc. WATM'95. First Workshop on ATM Traffic Management, Paris, Dec. 1995.

[13] R. Jain, *Congestion Control and Traffic Management in ATM Networks: Recent Advances and A Survey*, invited submission to Computer Networks and ISDN Systems.

[14] R. Jain, S. Kalyanaraman, R. Goyal, S. Fahmy, *Source Behavior for ATM ABR Traffic Management: An Explanation*, IEEE Communications Magazine, November, 1996.

[15] S. Keshav, *REAL: A network simulator*, Department of Computer Science, UC Berkeley, Technical Report 88/472, 1988.

[16] H. Ohsaki, G. Hasegawa, M. Murata and A.Miyahara, *Parameter Tuning of Rate-Based Congestion Control Algorithms and its Application to TCP over ABR*, Proc. WATM'95, Paris, december. 1995.

[17] H. Ohsaki, M. Murata, H. Suzuki, C. Ikeda and H. Miyahara, *Rate-based control for ATM Networks*, Comp. Comm. Review, acm-sigcomm, spec. issue in ATM, ed. R. Jain and K.Y. Siu, Vol 25, No. 2, pp. 60-71, 1995.

[18] K. Ramakrishnan and R. Jain, *A Binary Feedback Scheme for Congestion Avoidance in Computer Networks with Connectionless Network Layer*, Proc. SIGCOMM'88, August 1988.

[19] Michael Ritter, *Network Buffer Requirements of the Rate-Based Control Mechanism for ABR services* IEEE INFOCOM'96, san francisco, USA, March 1996.

[20] L. Roberts, *Enhanced PRCA (Proportional Control Algorithm)*, ATM Forum contribution 94-0735R1, aot 1994.

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

[22] N. Yin and M. G. Hluchyj, *On closed-loop rate control for ATM cell relay networks*, Proc. IEEE Infocom'94, pp. 99-108, June 1994.