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

Omar Ait-Hellal, Eitan Altman, Driss Elouadghiri, Mohammed Erramdani

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

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

Omar Ait-Hellal, Eitan Altman, Driss Elouadghiri, Mohammed Erramdani

N° 3131

Mars 1997

THÈME 1

ISSN 0249-6399

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

Omar Ait-Hellal, Eitan Altman,[*] Driss Elouadghiri,[**] Mohammed Erramdani[***]

Thème 1 — Réseaux et systèmes
Projet Mistral

Rapport de recherche n˚3131 — Mars 1997 — 30 pages

Abstract: In this paper we investigate the performances of the EFCI-based (Explicit Forward Congestion Indication) and ER-based (Explicit Rate) algorithms for the rate-based flow control of the ABR (Available Bit Rate) traffic in an ATM network. We consider the case of two switches in tandem. We present several definitions of bottleneck, and provide conditions that determine whether the first, the second or both queues are 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.

Key-words: ATM, Available Bit Rate, bottleneck, closed-loop congestion control, differential equations, fluid approximation.

(Résumé : tsvp)

[*] INRIA sophia Antipolis, BP 93, 2004 route des luciolles, 06902 Sophia Antipolis Cedex, France. E-mail: {oaithel, altman}@sophia.inria.fr
[**] Universite My Ismail, Dept de Math & Info, Zitoun, Meknes Marroco.
[***] Universite Mohammed V, Dept de Math & Info, Rabat, Marroco.

Unité de recherche INRIA Sophia Antipolis
2004 route des Lucioles, BP 93, 06902 SOPHIA ANTIPOLIS Cedex (France)
Téléphone : 04 93 65 77 77 – Télécopie : 04 93 65 77 65

Évaluation de Performances des Mécanismes de Contrôle de Flux pour le Service ABR

Résumé : Dans ce présent travail, nous évaluons les performances des algorithmes de contrôle de flux (i) basés sur l'indication de congestion EFCI (Explicit Forward Congestion Indication based algorithm) et (ii) ceux basés sur le débit explicite (Explicit Rate based algorithm) pour le trafic ABR (Available Bit Rate) dans un réseau ATM. Nous considérons le cas de deux commutateurs en tandem. Nous présentons deux définitions du goulot d'étranglement (bottleneck) et les conditions qui déterminent lequel des deux commutateurs est le goulot d'étranglement. Nous montrons que ce n'est pas nécessairement celui dont le débit disponible est le plus faible qui est « responsable » du goulot d'étranglement. Nous obtenons des formules analytiques pour la taille maximale de la file d'attente dans les deux commutateurs. Nous comparons les résultats ainsi obtenus, à ceux obtenus en approchant le réseau par un réseau simple contenant un seul goulot d'étranglement. Nous montrons que la taille maximale de la file d'attente dans ce dernier cas peut largement sous-estimer les tailles obtenues dans un réseau réel.

Mots-clé : ATM, Available Bit Rate, goulot d'étranglement, contrôle de congestion par boucle fermée, équations différentielles, approximations fluides .

1 Introduction

Adaptive 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 [2, 6]. 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, unsuring 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, 10]. 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 [14]; 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 [4, 5, 8, 9, 13, 16] 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 [3, 12, 15, 17]. 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 the 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 two switches along the path of a virtual circuit, where the bandwidth available for the ABR traffic in the first switch is larger than that in the second one.

We consider both EFCI and ER-based switches. The first objective of the paper is to obtain a 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 one queue or the other is the bottleneck. The second objective of the paper is a quantitative one: to obtain bounds for the buffer requirements in the two queues 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.

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

In comparing the lower bounds on the maximum queue size, to the maximum queue length obtained by simulating the simplified single-queue model [12, 15], 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 cost of experimenting 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 2, we describe the rate-based congestion control mechanism, and the model. The main analytic results are presented in Section 3. The exact calculations are delayed to the Appendix, in Section A. In Section 4, numerical results obtained by using our analytic formulas are compared to simulations.

2 Description and the model

In the following section we briefly describe the behavior of the *Source End System* (SES) [10], 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:

- *EFCI-based switch*:
  If the queue length exceeds a certain threshold Q_H (in our case each switch has

a different threshold Q_1 and Q_2, respectively, see Figure 1), the switch sets the EFCI bit in the header of the data cells, until the queue length drops below a threshold (the switch is then considered to be not congested)[13, 15]. 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.

- *ER-based switch*:
  ER-based control has an intelligent marking and a capability to estimate the available bandwidth which permits to reduce, selectively, the rates of the ABR sources by setting the CI bit or by updating the ER field in forward and/or backward RM cells. In this paper, we consider that backward and forward RM cells are updated by the ER-based switch (the case where only forward RM cells are updated for EFCI-based switch as well as for ER-based switch, can be seen to perform worst, since the information about the congestion arrives later. The queue lengths are then larger. The qualitative results, however, remain the same).
  During a congestion, there are different ways to signal ERs (Fair share, load factor, load adjustment factor... [4, 7, 8, 13]). As in [15], the method called *fair share* (FS) is considered in this paper. 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 ($FSi,\ i = 1, 2$) if congestion is detected. If both of the two switches are congested, the value of the ER field is set to the $min(FS1, FS2)$, which we consider equals to $FS2$. The FSi is given by $FSi = LCRi - MCR$ where $LCRi$ is the available bandwidth at the switch i. This FSi multiplied by an *Explicit Reduction Factor* (ERF1=ERF2) smaller than 1 is added to the MCR, and the resulting rate is, then, signaled to the source.

$$ERi = MCR + ERF * FSi \tag{1}$$

  Hence, we define after considering $MCR = 0$

$$ER1 \triangleq ERF * LCR1 \quad \text{and} \quad ER2 \triangleq ERF * LCR2.$$

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 [10] for more details). Such mechanisms are not taken into account in the present paper.

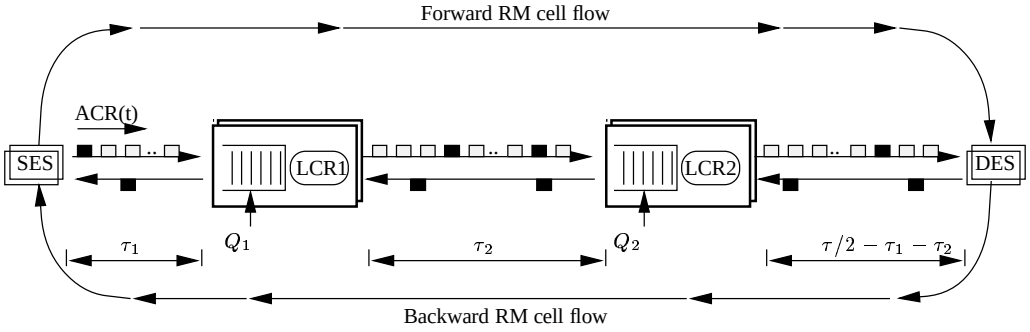


Figure 1: The Model with two bottlenecks

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 two switches, with available bandwidths of $LCR1$ and $LCR2$, respectively. We assume that $PCR > LCR1 > LCR2 > MCR$. We assume that the buffer sizes are sufficiently large so that no losses occur (we compute below sizes of the buffers that guarantee no losses). Data cells and RM cells go through the two switches before they reach the destination (Figure 1). The round trip time is τ (i.e. the time that takes 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 τ_2 between the first and the second switch as depicted in figure 1. The time that spends an RM cell from the second switch to the destination and back to the source is $\tau_3 \stackrel{\Delta}{=} \tau - \tau_1 - \tau_2$.

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 equals to $LCR2$, these moments are refered in what follows by $T_2^n - \tau_1 - \tau_2$ and $T_2^{n+1} - \tau_1 - \tau_2$ (see figure 2 for T_2^n and T_2^{n+1}).

Define, see (Figure 2):
Q_1 *resp* Q_2 is the queue threshold for the first *resp* the second switch.

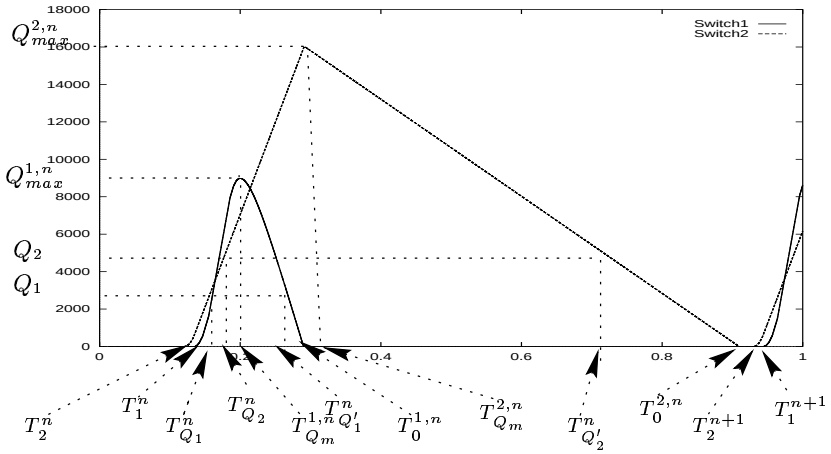


Figure 2: Typical behavior of the queue length

$T_2^n \triangleq$ Time at which queue two starts to build, in the nth cycle.
$T_1^n \triangleq$ Time at which queue one starts to build, in the nth cycle (This definition will only be used for cases where queue one indeed builds up).
$Q_1^n \triangleq$ The queue length in the first switch at time T_1^n, $Q_1^0 = 0$.
$Q_2^n \triangleq$ The queue length in the second switch at time T_2^n, $Q_2^0 = 0$.
$Q_{max}^{1,n} \triangleq$ The maximum queue length at the first switch in the nth cycle.
$Q_{max}^{2,n} \triangleq$ The maximum queue length at the second switch in the nth cycle.
$Q_1(t) \triangleq$ The queue length at the first switch at time t.
$Q_2(t) \triangleq$ The queue length at the second switch at time t.
$T_{Q_1}^n \triangleq$ The time at which queue one reaches Q_1 in the nth cycle, while increasing, if no congestion is detected.
$T_{Q_2}^n \triangleq$ The time at which queue two reaches Q_2 in the nth cycle, while increasing, if no congestion is detected.
$T_{Q'_1}^n \triangleq$ The time at which queue one reaches Q_1 in the nth cycle, while decreasing.
$T_{Q'_2}^n \triangleq$ The time at which queue two reaches Q_2 in the nth cycle, while decreasing.

We assume that after the first cycle, queue two never empties (i.e. full utilization. We establish later, the conditions for which this assumption holds) and the congestion is known by the source τ_1 *resp* $\tau_1 + \tau_2$ after it was detected by the first *resp*

the second 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. $(N_{rm}-1)/LCR2$. For the transient analysis, we assume, also, that $ICR = LCR2$. (It is not reasonable to negotiate an ICR larger than the available bandwidth; $ICR = LCR2$ thus corresponds to the largest initial burst.)

The following properties hold:

- **P1)** If queue one not empty at time t then queue two is not empty at time $t+\tau_2$.
- **P2)** If the queue length at the second switch exceeds the threshold Q_2 in the nth cycle, then queue one will be empty at the end of that cycle.

The first property is obvious, since if queue one is not empty at time t then the input rate at switch two at time $t+\tau_2$ equals to $LCR1 > LCR2$. For the second property, if queue two exceeds the threshold Q_2 and queue one is nonempty, then queue two continues to increase. Thus it decreases (at time t) only if queue one empties (at time $t-\tau_2$), which should happen (after some delay) since $ACR(t)$ decreases as long as queue two exceeds Q_2.

3 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_1^t resp. Q_2^t in switches 1 and 2, resp., that will be uniform for all time. 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_1^s and Q_2^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 $Q_i^s \triangleq \overline{\lim}_{n\to\infty}(Q_{max}^{i,n})$, $i = 1, 2$.

Definition 3.1 i) *Queue i is the strict bottleneck if and only if queue j ($j \neq i$) remains always empty (queue one cannot 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 congestion is always due to the second switch if and only if queue two reaches its threshold Q_2, not later than τ_2 before queue one reaches its Q_1 (the first RM cell indicating congestion is updated by switch two $T^n_{Q_2} + \tau_2 \leq T^n_{Q_1};\ \forall n$), and the congestion is always due to the first switch if and only if queue one reaches its threshold Q_1 not later than τ_2 after queue two reaches its one ($T^n_{Q_2} + \tau_2 > T^n_{Q_1};\ \forall n$).*

Theorem 3.1 *When the congestion is always due to the second switch, the maximum queue sizes are obtained in the first cycle:*

$$Q_1^t \triangleq \max_{n\geq 0}(Q_{max}^{1,n}) = Q_{max}^{1,0} \quad \textit{and} \quad Q_2^t \triangleq \max_{n\geq 0}(Q_{max}^{2,n}) = Q_{max}^{2,0}$$

Proof:
We first note that $Q_1^n = 0\ \forall n$. This is due to property **P2**. For all n we have $Q_2^n \geq Q_2^0$, since we assume that the queue 2 is initially empty. We now make use of the following monotone property: Assume that $Q_2^n \geq Q_2^m$ for some n and m. Then the time that queue two takes to reach Q_2 in the nth cycle is shorter than the time it takes to reach Q_2 in the mth cycle. It now follows that the ACR in cycle n at time $T^n_{Q_2} - \tau_1 - \tau_2 + t$ is smaller than the ACR at cycle m at time $T^m_{Q_2} - \tau_1 - \tau_2 + t$ for $t \geq 0$. It then follows that the size of the second queue in cycle n after time $T^n_{Q_2} + t$ is smaller than in cycle m at time $T^m_{Q_2} + t$, $t \geq 0$. The proof for queue 2 now follows by taking $m = 0$ and arbitrary n. Similar arguments yield the proof for queue 1. ∎

Remark 3.1 *In the proof of Theorem 3.1 we established the following property: If $Q_2^n \geq Q_2^m$ for some n and m then $Q_{max}^{i,n} \leq Q_{max}^{i,m}$, $i = 1, 2$.*

Define $A_{1,n}$, $B_{1,n}$, $C_{1,n}$, $A_{2,n}$, $B_{2,n}$ and $C_{2,n}$ as

$$A_{1,n} \triangleq \begin{cases} \sqrt{\dfrac{2(Q_1 - Q_1^n)}{AIR * LCR2}} & \text{if} \quad \alpha \geq \sqrt{\dfrac{2(Q_1 - Q_1^n)}{AIR * LCR2}} \\ \dfrac{Q_1 - Q_1^n}{PCR - LCR1} + \dfrac{\alpha}{2} & \text{otherwise} \end{cases}$$

$$A_{2,n} \triangleq \begin{cases} \sqrt{\frac{2(Q_2 - Q_2^n)}{AIR * LCR2}} - \beta & \text{if} \quad \beta \geq \sqrt{\frac{2(Q_2 - Q_2^n)}{AIR * LCR2}} \\ \frac{Q_2 - Q_2^n}{LCR1 - LCR2} - \frac{\beta}{2} & \text{otherwise} \end{cases}$$

$$\begin{aligned} B_{2,n} &\triangleq \left(2(\tau_1 + \tau 2) + A_{2,n} - \frac{PCR - LCR2}{AIR * LCR2}\right)^+ \\ \text{and} \quad C_{2,n} &\triangleq min\left(2(\tau_1 + \tau 2) + A_{2,n}, \frac{PCR - LCR2}{AIR * LCR2}\right) \end{aligned}$$

where $(x)^+ \triangleq max(0, x)$, $\alpha = \frac{PCR - LCR1}{AIR * LCR2}$ and $\beta = \frac{LCR1 - LCR2}{AIR * LCR2}$. Q_1^n and Q_2^n are given in Theorem 3.3 and Theorem 3.4.

3.1 Results for the transient case

Theorem 3.2 *For both EFCI-based switch and ER-based switch, we have*
i) Queue two begins to build before queue one, if and only if

$$\tau_2 < \frac{LCR1 - LCR2}{AIR * LCR2}$$

ii) The following are equivalent:

- *(ii.a) The congestion is always due to the second switch.*
- *(ii.b)* $2\tau_2 \leq A_{1,n} - A_{2,n}$; $\forall n$.
- *(ii.c)* $2\tau_2 \leq A_{1,0} - A_{2,0}$

Theorem 3.3 *If the congestion is always due to the second switch, then we have*
i) For both EFCI and ER-based switch, the following are equivalent:

- *(i.a) Queue two is the strict bottleneck*
- *(i.b)* $2(\tau_1 + \tau 2) \leq -A_{2,n}$; $\forall n$.
- *(i.c)* $2(\tau_1 + \tau 2) \leq -A_{2,0}$

ii) For the EFCI-based switch, queue two is the essential bottleneck if

$$2(\tau_1+\tau 2) < \sqrt{\frac{2Q_1}{AIR*LCR2}\left(\frac{LCR1}{LCR1+AIR*RDF}\right)} - A_{2,0}$$

iii) For the EFCI-based switch, if queue two is not the strict bottleneck, then the maximum queue length at the first and the second switch (Q_1^t and Q_2^t) are given by

$$\begin{aligned} Q_1^t &= \frac{AIR*LCR2}{2}{C_{2,0}}^2 + (PCR-LCR1)B_{2,0} + RDF*AIR*C_{2,0} \\ &\quad -\frac{RDF*LCR1}{LCR2}\log\left(1+\frac{AIR*LCR2}{LCR1}C_{2,0}\right) \end{aligned} \tag{2}$$

$$\begin{aligned} Q_2^t &\cong \frac{LCR1-LCR2}{LCR1}\Big(Q_1^t + LCR1(2(\tau_1+\tau_2)+A_{2,0}) \\ &\quad +\frac{RDF(LCR1-LCR2)}{LCR2}\Big) + \frac{(LCR1-LCR2)^2}{2*AIR*LCR2} \\ &\quad +\frac{RDF(LCR1-LCR2)}{LCR2}\log\left(1+\frac{AIR*LCR2}{LCR1}C_{2,0}\right) \end{aligned} \tag{3}$$

The following bounds hold:

$$\begin{aligned} Q_2^n &\leq \Big(Q_2 + RDF\left(1-e^{-\frac{LCR2}{RDF}\tau}\right) \\ &\quad -\frac{LCR2}{2*AIR}\left(2*AIR*\tau+\left(1-\ e^{-\frac{LCR2}{RDF}\tau}\right)^2\right)\Big)^+, \\ Q_2^n &\geq \left(Q_2 - LCR2\left(\tau+\frac{1}{2*AIR}\right)\right)^+ \end{aligned} \tag{4}$$

$$Q_1^t \leq \frac{AIR*LCR2}{2}\left(1+\frac{AIR*RDF}{LCR1}\right)(2(\tau_1+\tau 2)+A_{2,0})^2 \tag{5}$$

$$\begin{aligned} Q_2^t &\leq Q_1^t + \frac{RDF(LCR1-LCR2)}{LCR2}\log\left(1+\frac{AIR*LCR2}{LCR1}C_{2,0}\right) \\ &\quad +\frac{(LCR1-LCR2)^2}{2*AIR*LCR2} + (LCR1-LCR2)(2(\tau_1+\tau_2)+A_{2,0}) \\ &\quad +RDF\left(\frac{LCR1-LCR2}{LCR2}-\log\left(\frac{LCR1}{LCR2}\right)\right) \end{aligned} \tag{6}$$

iv) For the ER-based switch, queue two is the essential bottleneck if

$$2(\tau_1 + \tau 2) < \sqrt{\frac{2Q_1}{AIR * LCR2}} - A_{2,0}$$

v) For the ER-based switch, if queue two is not the strict bottleneck, then the maximum queue length at the first and the second switch (Q_1^t and Q_2^t) are given by

$$Q_1^t = \frac{AIR * LCR2}{2} {C_{2,0}}^2 + (PCR - LCR1)\, B_{2,0} \tag{7}$$

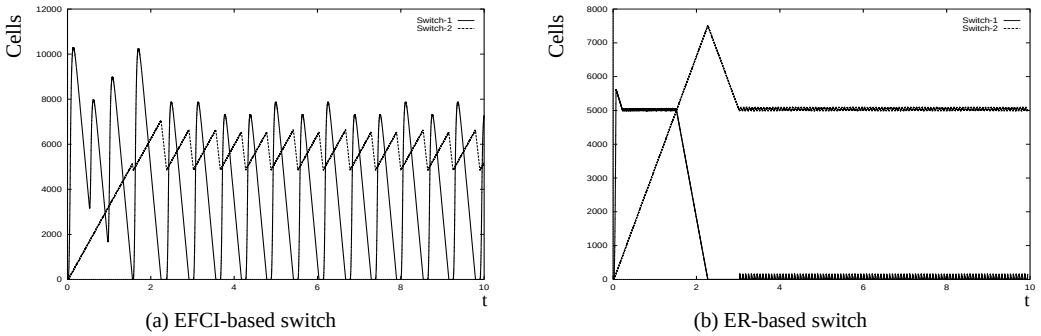
$$\begin{aligned} Q_2^t &= (LCR1 - LCR2)\left(2(\tau_1 + \tau 2) + A_{2,0} + \frac{LCR1 - LCR2}{2 * AIR * LCR2}\right) \\ &\quad + \frac{LCR1 - LCR2}{LCR1 - ER2} Q_1^t \end{aligned} \tag{8}$$

$$Q_1^n = 0 \quad \textit{and} \quad Q_2^n = \left(Q_2 - (LCR2 - ER2)\left(\tau + \frac{LCR2 - ER2}{AIR * LCR2}\right)\right)^+ \tag{9}$$

The following bound holds:

$$Q_1^t \leq \frac{AIR * LCR2}{2} \left(2(\tau_1 + \tau 2) + A_{2,0}\right)^2 \tag{10}$$

3.2 Results for the steady state case



(a) EFCI-based switch (b) ER-based switch

Figure 3: Queue length as a function of the time

In what follows, we shall restrict to the case (in steady state) where congestion is always due to the second switch. Since we assume steady state, this is less restrictive than the corresponding assumption for the transient behavior. Indeed, the steady state behavior is a periodic regime obtained after some transient period, so congestion is always due to the second switch if and only if $T^n_{Q_2} + \tau_2 \leq T^n_{Q_1}$ for some n. The above restriction applies in almost all simulations that we performed, in the case where the second queue never empties. We almost always obtained the following behavior: after a finite transient period, the steady state is reached, and the congestion *in steady state* is always due to the second switch. This can be seen, in particular, in Figure 3 which plots the queue length as a function of the time, for the case where $LCR1$ is close to $LCR2$ ($LCR1 = 8.08\ Mbps$, $LCR2 = 6.73\ Mbps$ $Q_1 = Q_2 = 5000\ cells$ and $\tau = 6.22\ msec$).

Theorem 3.4 *If queue two is not the strict bottleneck and the congestion is always due to the second switch, then*
i) For the EFCI-based switch, the maximum queue length at the first and the second switch (Q^s_1 and Q^s_2) are given by

$$\begin{aligned} Q^s_1 &= \frac{AIR * LCR2}{2} {C_{2,\infty}}^2 + (PCR - LCR1)\, B_{2,\infty} + RDF * AIR * C_{2,\infty} \\ &\quad - \frac{RDF * LCR1}{LCR2} \log\left(1 + \frac{AIR * LCR2}{LCR1} C_{2,\infty}\right) \end{aligned} \tag{11}$$

$$\begin{aligned} Q^s_2 &= Q^\infty_2 + \frac{LCR1 - LCR2}{LCR1}\left(Q^s_1 + \frac{RDF(LCR1 - LCR2)}{LCR2}\right) \\ &\quad + (LCR1 - LCR2)\left((2(\tau_1 + \tau_2) + A_{2,\infty}\right) + \frac{(LCR1 - LCR2)^2}{2 * AIR * LCR2} \\ &\quad + \frac{RDF(LCR1 - LCR2)}{LCR2} \log\left(1 + \frac{AIR * LCR2}{LCR1} C_{2,\infty}\right) \end{aligned} \tag{12}$$

$$\textit{Where} \qquad Q^\infty_2 = \left(Q_2 - LCR2\left(\tau + \frac{1}{2 * AIR}\right)\right)^+ \tag{13}$$

ii) For the ER-based switch, the maximum queue length at the first and the second switch (Q^s_1 and Q^s_2) are given by

$$Q^s_1 = \frac{AIR * LCR2}{2} {C_{2,\infty}}^2 + (PCR - LCR1)\, B_{2,\infty} \tag{14}$$

$$Q_2^s = Q_2^\infty + (LCR1 - LCR2)\left(2(\tau_1+\tau 2) + A_{2,\infty} + \frac{LCR1-LCR2}{2*AIR*LCR2}\right) + \frac{LCR1-LCR2}{LCR1-ER2}\,Q_1^s \quad (15)$$

Where $$Q_2^\infty = \left(Q_2 - (LCR2-ER2)\left(\tau + \frac{LCR2-ER2}{AIR*LCR2}\right)\right)^+ \quad (16)$$

4 Numerical results and simulations

In this section we present some numerical examples, in order to validate our analytical results with simulation ones, and compare them to the approximating model in which the network is replaced by a single bottleneck node (exactly the same parameters as for our model without the first switch (the first switch is deleted)). Only the transient state (first cycle) is taken into account, since the maximum queue lengths are obtained in this case (see Theorem 3.1). We compute the queue length (Q_1^t, Q_2^t) in each switch as a function of $2(\tau_1+\tau_2)$.

As we have seen (Theorem 3.3), 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 length at switch two is a function of both τ and $2(\tau_1+\tau_2)$ since, this queue can be nonempty at the end of the cycle ($Q_2^n > 0$, see equations (12), (13) and (15), (16)).

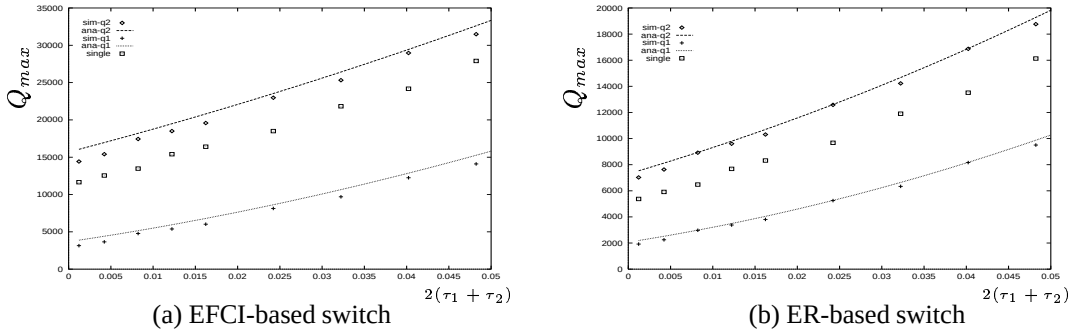
Remark 4.1 *As discussed in section 2, the time between RM cells is neglected in the paper. However, this time will have some effects on the actual results. To take this time into account, we added the quantity* $(N_{rm}-1)/LCR2$ *to* $A_{2,0}$*, when computing analytically the queue lengths (ana-q1 and ana-q2 in the figures), since when the congestion occurs at the first resp. the second switch, it is detected by the source at most* $\tau_1 + (N_{rm}-1)/LCR2$ *resp.* $\tau_1+\tau_2+(N_{rm}-1)/LCR2$ *after.*

In all the following simulations, we considered $PCR = 134.78\ Mbps$ 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 $ERF = 0.8$.

In the first example we considered $LCR1 = 40.435791\ Mbps$, $LCR2 = 4.043\ Mbps$ and $Q_1 = Q_2 = 5000\ cells$, and then from Theorem 3.2 and Theorem 3.3 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 $\tau_1 + \tau_2 < 0.012845$.

Figure 4 shows the maximum queue length in the two switches (Q_1^t and Q_2^t) obtained analytically (ana-q1 and ana-q2 in the figures) and those obtained by simulation (sim-q1 and sim-q2 in the figures) as a function of $2(\tau_1+\tau_2)$ for both EFCI-based and ER-based switch. The results obtained by simulation for the single bottleneck model ($single$ in all the figures) are also plotted. In this example only $\tau_1 + \tau_2$ is varying and 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. As we can see, the analytic results (which are, in fact, bounds, since $(N_{rm} - 1)/LCR2$ is added to $A_{2,0}$) are very close to the simulations ones and the relative error is at most around 10%. The bounds are very close to the simulations results, especially, for the ER-based switch, and queue lengths are always smaller than those obtained for EFCI-based switch.



(a) EFCI-based switch (b) ER-based switch

Figure 4: Queue length as a function of the round trip time

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 Figure5, 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).

From the formulas in section 3, we can see that Q_2^t increases linearly as Q_1^t grows. Q_1^t is, in particular, a function of $LCR1$; the closer $LCR1$ is to $LCR2$, the larger is the queue length at the first switch. Hence, by decreasing $LCR1$, we can obtain cases in which the queue length in the second switch is significantly larger compared to the single bottleneck approximating model, while $LCR2$ 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 $LCR1$ is decreased. 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 $LCR1 = 26.957194\, Mbps$. 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

$$\frac{LCR1 - LCR2}{AIR * LCR2} = 0.0226 \leq \sqrt{\frac{2Q_2}{AIR * LCR2}} = 0.0632 \Longrightarrow A_{2,0} = 0.07693\ sec.$$

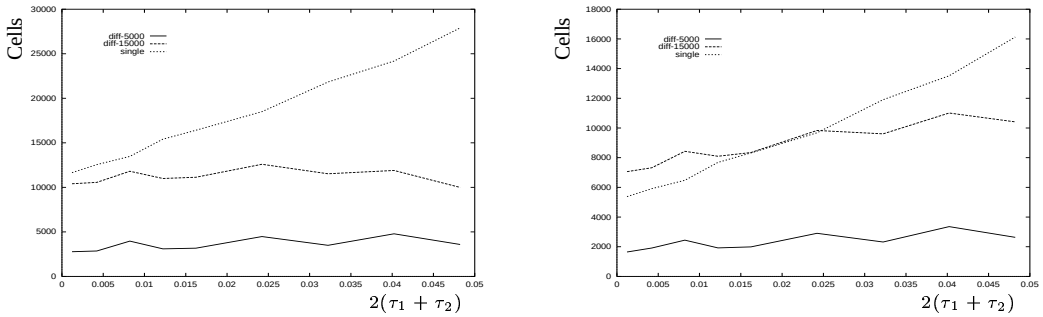
Assume that $\dfrac{PCR - LCR1}{AIR * LCR2} \leq \sqrt{\dfrac{2Q_1}{AIR * LCR2}}$. Then $Q_1 \leq 15308.88$,

and $A_{1,0} = \dfrac{1}{500\sqrt{5}}\sqrt{Q_1}$.

From Theorem 3.2, 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 5 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 $LCR1 = 26.957194\ Mbps$ 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).



(a) EFCI-based switch (b) ER-based switch

Figure 5: The difference in queue length between our model and the single bottleneck approximating one

The decrease in the difference of the queue length (diff-15000) in the Figure 5 (EFCI-based switch) 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.

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.

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 $LCR1$ and $LCR2$ 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 uses, only, a single bit congestion indication), which depend 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.

References

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

[2] L.S. Brakmo, L.L. Peterson, TCP Vegas : End to End Congestion Avoidance on a Global Internet," *IEEE Journal on selected Areas in communications,* vol. 13, pp. 1465-1480, 1995.

[3] 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, University of Illinois at Urbana-Champaign, May, 1996.

[4] Y. Chang, N. Golmie, D. Siu, "A Rate-Based Flow Control Switch Design for ABR Service in an ATM Network," *Twelfth International Conference on Computer Communication ICCC'95,* August 1995.

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

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

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

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

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

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

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

[12] H. Ohsaki, G. Hasegawa, M. Murata and A.Miyahara, "Parameter Tuning of Rate-Based Congestion Control Algorithms and its Application to TCP over ABR,” *Proceedings WATM’95 First Workshop on ATM Traffic Management,* Paris, December 1995.

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

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

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

[16] 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.

[17] 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.

A Appendix: Analysis and proof of Theorems

A.1 Evolution of the ACR

The evolution of the $ACR(t)$ follows two phases in our case ($ICR = LCR2$, queue two never empties and RM cells have no priority), depending on whether the congestion is detected or not.

Phase 1: No congestion and queue two nonempty

This phase concerns (i) the first cycle ($n = 0$) which begins after the first RM cell returns back; we consider, then, $ACR(T_2^0 - \tau_1 - \tau_2) = ACR(\tau) = LCR2$, and (ii) all other cycles while no congestion is detected. The evolution of the $ACR(t)$ in this phase is given by

$$\frac{dACR(t)}{dt} = AIR * LCR2$$

from which we get

$$ACR(t) - ACR(t_0) = AIR * LCR2(t - t_0) \tag{17}$$

t_0 is the beginning instant of this phase which ends once an RM cell with bit CI set is received.

Phase 2: Congestion is detected
When congestion is detected, ACR(t) is reduced multiplicatively down to MCR and its evolution is given by

$$\frac{dACR(t)}{dt} = -\frac{LCR2}{RDF}ACR(t)$$

from which we get

$$ACR(t) = ACR(t_1)e^{-\frac{LCR2}{RDF}(t-t_1)} \tag{18}$$

t_1 is the beginning instant of this phase, which corresponds to the ending time of phase 1.

A.2 Transient analysis

First we compute T_1^n and T_2^n. We have by definition

$$ACR(T_1^n - \tau_1) = LCR1 \qquad \text{and} \qquad ACR(T_2^n - \tau_1 - \tau_2) = LCR2.$$

Since queue two is assumed always nonempty, the evolution of the $ACR(t)$ follows equation (17). Hence

$$ACR(T_1^n - \tau_1) = LCR2 + AIR * LCR2(T_1^n - T_2^n + \tau_2) = LCR1,$$

from which we get

$$T_1^n - T_2^n = \frac{LCR1 - LCR2}{AIR * LCR2} - \tau_2 \tag{19}$$

Queue two starts to build up before queue one iff $T_1^n - T_2^n < 0$, hence from (19) we get Theorem 3.2 (*i*).

Define T_{PCR}^n such that

$$ACR(T_{PCR}^n) = PCR = LCR1 + AIR * LCR2(T_{PCR}^n - T1 + \tau_1)$$

$$\iff T_{PCR}^n = T_1^n - \tau_1 + \frac{PCR - LCR1}{AIR * LCR2} = T_2^n - \tau_1 - \tau_2 + \frac{PCR - LCR2}{AIR * LCR2} \tag{20}$$

RM cells with a CI bit cleared are sent by the switches till queue one (*or* queue two) reaches the threshold Q_1 (*or* Q_2) at time $T_{Q_1}^n$ (*or* $T_{Q_2}^n$) (figure 2) which we compute next as if there is no congestion (only equation (17) is used). For $T_{Q_1}^n$, we distinguish two cases:

1) $T_{Q_1}^n - \tau_1 \leq T_{PCR}^n$

$$Q_1 - Q_1^n = \int_0^{T_{Q_1}^n - T_1^n} AIR * LCR2 * x \; dx = \frac{1}{2} AIR * LCR2 (T_{Q_1}^n - T_1^n)^2$$

from which we get

$$T_{Q_1}^n = \sqrt{\frac{2(Q_1 - Q_1^n)}{AIR * LCR2}} + T_1^n \tag{21}$$

Case (1) is equivalent to $\frac{PCR - LCR1}{AIR * LCR2} \geq \sqrt{\frac{2(Q_1 - Q_1^n)}{AIR * LCR2}}$ as can be seen by combining (20) and (21).

2) In other cases, we have

$$Q_1 - Q_1^n = \int_0^{T_{PCR}^n - T_1^n + \tau_1} AIR * LCR2 * x \; dx + (PCR - LCR1)(T_{Q_1}^n - \tau_1 - T_{PCR}^n)$$

from which we get, after substituting (20)

$$T_{Q_1}^n = \frac{Q_1 - Q_2^n}{PCR - LCR1} + \frac{PCR - LCR1}{2 * AIR * LCR2} + T_1^n \tag{22}$$

Since the input rate at queue two cannot exceed $LCR1$, then we have to distinguish two cases:

1) for $T_{Q_2}^n - \tau_2 \leq T_1^n$ we have

$$Q_2 - Q_2^n = \int_0^{T_{Q_2}^n - T_2^n} AIR * LCR2 * x \; dx = \frac{1}{2} AIR * LCR2 (T_{Q_2}^n - T_2^n)^2$$

from which we get

$$T^n_{Q_2} = \sqrt{\frac{2(Q_2 - Q^n_2)}{AIR * LCR2}} + T^n_2 \tag{23}$$

Case (1) is equivalent to $\frac{LCR1-LCR2}{AIR*LCR2} \geq \sqrt{\frac{2(Q_2-Q^n_2)}{AIR*LCR2}}$ as can be seen by combining (19) and (23).

2) In other cases, we have

$$\begin{aligned} Q_2 - Q^n_2 &= \int_0^{T^n_1 - T^n_2 + \tau_2} AIR * LCR2 * x \; dx \\ &\quad + (LCR1 - LCR2)(T^n_{Q_2} - T^n_1 - \tau_2) \\ &= \frac{(LCR1 - LCR2)^2}{2 * AIR * LCR2} + (LCR1 - LCR2)(T^n_{Q_2} - T^n_1 - \tau_2) \end{aligned}$$

from which we get, after substituting (19)

$$T^n_{Q_2} = \frac{Q_2 - Q^n_2}{LCR1 - LCR2} + \frac{LCR1 - LCR2}{2 * AIR * LCR2} + T^n_2 \tag{24}$$

The congestion is always due to the second switch if and only if $T^n_{Q_2} + \tau_2 \leq T^n_{Q_1} \; \forall n$, from which the equivalence of (ii.a) and (ii.b) in part (*ii*) of Theorem 3.2 follows. This is obtained after substituting (19) and remarking that $T^n_{Q_2} + \tau_2 - T^n_{Q_1} = A_{2,n} - A_{1,n} + 2\tau_2$.

(ii.b) implies (ii.c) follows trivially.

Now, we prove that $2\tau_2 \leq A_{1,0} - A_{2,0} \Longrightarrow 2\tau_2 \leq A_{1,n} - A_{2,n} \; \forall n$. Since $Q^n_2 \geq 0 = Q^0_2$ for all integers n, we have $A_{2,0} \geq A_{2,n} \; \forall n$. We have $2\tau_2 \leq A_{1,0} - A_{2,0}$. Hence the congestion is due to the second switch in the first cycle, i.e. queue two reaches its threshold Q_2, not later than τ_2 before queue one reaches its Q_1. Hence $Q^1_1 = 0$ (property **P2**). Hence $A_{1,1} = A_{1,0}$, so that $2\tau_2 \leq A_{1,1} - A_{2,0}$. Combining this with the fact that $A_{2,0} \geq A_{2,n} \; \forall n$, we have $2\tau_2 \leq A_{1,1} - A_{2,1}$. Repeating this argument we obtain (ii.b).

For the maximum queue length in each switch, we consider both ER-based and EFCI-based switches. Let $T^n_P \triangleq min(T^n_{PCR}, T^n_{Q_2} + \tau_1 + \tau_2)$, and $ACR^n_{max} \triangleq ACR(T^n_P)$, we have $ACR(T^n_P) = LCR1 + AIR * LCR2(T^n_P - T^n_1 + \tau 1)$. We shall assume

- (**H1**) The congestion is always due to queue two ($T^n_{Q_2} + \tau_2 \leq T^n_{Q_1}$)

A.2.1 EFCI-based switch

The first RM cell with CI bit set to one returns back to the source at time $T^n_{Q_2} + \tau_1 + \tau_2$ (backward RM cells are updated when congestion is detected), from this time the evolution of the $ACR(t)$ corresponds to phase 2, and then, the instant when $ACR(t) = LCR1$ *resp* $ACR(t) = LCR2$, in the nth cycle, while decreasing, is given by

$$\begin{aligned} & T^n_{LCR1} &=& \frac{RDF}{LCR2}\log\left(\frac{ACR^n_{max}}{LCR1}\right) + T^n_{Q_2} + \tau_1 + \tau_2 \\ resp \quad & T^n_{LCR2} &=& \frac{RDF}{LCR2}\log\left(\frac{ACR^n_{max}}{LCR2}\right) + T^n_{Q_2} + \tau_1 + \tau_2 \end{aligned}$$

Since the congestion is always due to queue two (**H1**), then from property **P2**, we have $Q^n_1 = 0\ \forall n$ and hence

$$\begin{aligned} Q^{1,n}_{max} &= \int_{T^n_1-\tau_1}^{T^n_P} (ACR(x) - LCR1)\,dx + \int_{T^n_P}^{T^n_{Q_2}+\tau_1+\tau_2} (PCR - LCR1)\,dx \\ &\quad + \int_{T^n_{Q_2}+\tau_1+\tau_2}^{T^n_{LCR1}} (ACR(x) - LCR1)\,dx \\ &= \int_0^{T^n_P - T^n_1 + \tau_1} AIR * LCR2 * x\,dx \\ &\quad + (PCR - LCR1)(T^n_{Q_2} + \tau_1 + \tau_2 - T^n_P) \\ &\quad + \int_0^{T^n_{LCR1} - T^n_{Q_2} - \tau_1 - \tau_2} (ACR^n_{max} e^{-\frac{LCR2}{RDF}x} - LCR1)\,dx \end{aligned}$$

Finally,

$$\begin{aligned} Q^{1,n}_{max} &= \frac{AIR * LCR2}{2}\left(T^n_P - T^n_1 + \tau_1\right)^2 \\ &\quad + (PCR - LCR1)(T^n_{Q_2} + \tau_1 + \tau_2 - T^n_P) \\ &\quad + \frac{LCR1 * RDF}{LCR2}\left(\frac{ACR^n_{max}}{LCR1} - 1 - \log\left(\frac{ACR^n_{max}}{LCR1}\right)\right) \end{aligned} \tag{25}$$

Equation (2) of Theorem 3.3 is obtained by setting $C_{2,n} \triangleq T^n_P - T^n_1 + \tau_1$ and $B_{2,n} \triangleq T^n_{Q_2} + \tau_1 + \tau_2 - T^n_P$ which are given in section 3 after substituting (23), (24) and (19). Recall that $Q^t_1 = Q^{1,0}_{max}$ and $Q^t_2 = Q^{2,0}_{max}$

For both EFCI and ER-based switch, queue two is the strict bottleneck iff

$$\begin{aligned} & Q_{max}^{1,n} = 0 \Longleftrightarrow ACR_{max}^{n} \leq LCR1 \\ \Longleftrightarrow \quad & AIR * LCR2 * (T_{Q_2}^{n} + \tau_1 + \tau_2 - T_1^n + \tau_1) + LCR1 \leq LCR1 \end{aligned}$$

from which the equivalence between (i.a) and (i.b) in Theorem 3.3 (*i*) follows. This is obtained by (i) noting that $A_{2,n}$, defined in Section 3, satisfies $A_{2,n} = T_{Q_2}^{n} - T_1^n - \tau_2$, and (ii) by using (19), (23) and (24).

It remains to show that (i.c) implies (i.b) (the opposite relation is trivial). This follows since $2(\tau_1+\tau_2) \leq -A_{2,n}\ \forall n \Longrightarrow 2(\tau_1+\tau_2) \leq -A_{2,0}$ and on the other hand we have $A_{2,n} \leq A_{2,0}\ \forall n$ which implies that $2(\tau_1+\tau_2) \leq -A_{2,0} \leq -A_{2,n}\ \forall n$.

Queue two is the essential bottleneck iff $Q_{max}^{1,n} < Q_1$. Let $z_n \stackrel{\Delta}{=} 2(\tau_1+\tau_2) + A_{2,n} \geq 0$, from (25), since $ACR_{max}^{n} \leq ACR(z_n)$ and $C_{2,n} \leq z_n$, then we have

$$\begin{aligned} Q_{max}^{1,n} \quad \leq \quad & \frac{AIR * LCR2}{2} {z_n}^2 + \frac{RDF}{LCR2}\left(ACR(z_n) - LCR1\right) \\ & - \frac{LCR1 * RDF}{LCR2} \log\left(\frac{ACR(z_n)}{LCR1}\right) \\ \leq \quad & \frac{AIR * LCR2}{2} {z_n}^2 + AIR * RDF * z_n \\ & - \frac{LCR1 * RDF}{LCR2} \log\left(1 + \frac{AIR * LCR2}{LCR1} z_n\right) \end{aligned}$$

On the other hand we have

$$\log\left(1 + \frac{AIR * LCR2}{LCR1} z_n\right) \geq \frac{AIR * LCR2}{LCR1} z_n - \frac{1}{2}\left(\frac{AIR * LCR2}{LCR1} z_n\right)^2$$

then the sufficient condition for that $Q_{max}^{1,n} < Q_1\ \forall n$ is

$$Q_{max}^{1,n} \leq \frac{AIR * LCR2}{2}\left(1 + \frac{AIR * RDF}{LCR1}\right)\left(2(\tau_1+\tau_2) + A_{2,n}\right)^2 < Q_1$$

from which we get (*ii*) and inequality (5) of Theorem 3.3, since $A_{2,n} \leq A_{2,0}$.

To compute $Q_{max}^{2,n}$, we need first to compute $T_0^{1,n}$, the instant when queue one empties. We have

$$\begin{aligned} Q_1(T_0^{1,n}) - Q_{max}^{1,n} \quad = \quad & \int_{T_{LCR1}^{n}}^{T_0^{1,n} - \tau_1} \left(ACR(x) - LCR1\right) dx \\ = \quad & \int_{0}^{T_0^{1,n} - \tau_1 - T_{LCR1}^{n}} \left(LCR1 e^{-\frac{LCR2}{RDF}x} - LCR1\right) dx \end{aligned}$$

Let $y \triangleq T_0^{1,n} - \tau_1 - T_{LCR1}^n$, y is then a solution of

$$Q_{max}^{1,n} + \frac{LCR1 * RDF}{LCR2} = LCR1 * y + \frac{LCR1 * RDF}{LCR2}\, e^{-\frac{LCR2}{RDF}y} \qquad (26)$$

Let, also, $T_M \triangleq max(T_0^{1,n} - \tau_1, T_{LCR2}^n)$, $Q_{max}^{2,n}$ is then given by

$$\begin{aligned} Q_{max}^{2,n} &= Q_2^n + \int_{T_2^n-\tau_1-\tau_2}^{T_1^n-\tau_1} (ACR(x) - LCR2)\, dx \\ &\quad + \int_{T_1^n-\tau_1}^{T_0^{1,n}-\tau_1} (LCR1 - LCR2)\, dx + \int_{T_0^{1,n}-\tau_1}^{T_M} (ACR(x) - LCR2)\, dx \\ &= Q_2^n + \frac{(LCR1 - LCR2)^2}{2 * AIR * LCR2} + (LCR1 - LCR2)(T_0^{1,n} - T_1^n) \\ &\quad + \frac{RDF}{LCR2} ACR(T_0^{1,n} - \tau_1) \left(1 - e^{-\frac{LCR2}{RDF}(T_M - T_0^{1,n} + \tau_1)}\right) \\ &\quad - LCR2(T_M - T_0^{1,n} + \tau_1) \end{aligned}$$

After some calculations and the introduction of T_{LCR1}^n, we get

$$\begin{aligned} Q_{max}^{2,n} &= Q_2^n + \frac{(LCR1 - LCR2)^2}{2 * AIR * LCR2} + (LCR1 - LCR2)(T_{LCR1}^n + \tau_1 - T_1^n) \\ &\quad - \frac{RDF * LCR1}{LCR2} + LCR1(T_0^{1,n} - \tau_1 - T_{LCR1}^n) \\ &\quad + \frac{LCR1 * RDF}{LCR2}\, e^{-\frac{LCR2}{RDF}(T_0^{1,n} - \tau_1 - T_{LCR1}^n)} + \frac{RDF * LCR1}{LCR2} \\ &\quad - LCR2(T_M - T_{LCR1}^n) - \frac{LCR1 * RDF}{LCR2}\, e^{-\frac{LCR2}{RDF}(T_M - T_{LCR1}^n)} \end{aligned} \qquad (27)$$

Since the quantity in the last line of the equation (27) admits its maximum for $T_M = T_{LCR2}^n$ then, after substituting (26), we get

$$\begin{aligned} Q_{max}^{2,n} &\leq Q_2^n + Q_{max}^{1,n} - LCR2(T_{LCR2}^n - T_{LCR1}^n) \\ &\quad + \frac{(LCR1 - LCR2)^2}{2 * AIR * LCR2} + (LCR1 - LCR2)(T_{LCR1}^n + \tau_1 - T_1^n) \\ &\quad + \frac{LCR1 * RDF}{LCR2} \left(1 - e^{-\frac{LCR2}{RDF}(T_{LCR2}^n - T_{LCR1}^n)}\right) \end{aligned}$$

from which we get inequality (6) of Theorem 3.3, after substituting $A_{2,n},\ T_{LCR1}^n$.

On the other hand, equation (26) implies $T^n_{LCR2} \geq T^{1,n}_0 - \tau_1 \Longleftrightarrow Q^{1,n}_{max} \leq 0$, hence $T_M = T^{1,n}_0$, and then

$$\begin{aligned} Q^{2,n}_{max} &= Q^n_2 + \frac{(LCR1 - LCR2)^2}{2 * AIR * LCR2} + (LCR1 - LCR2)(T^{1,n}_0 - T^n_1) \\ &= Q^n_2 + \frac{(LCR1 - LCR2)^2}{2 * AIR * LCR2} + (LCR1 - LCR2)\left(T^{1,n}_0 - \tau_1 - T^n_{LCR1}\right) \\ &\quad + (T^n_{LCR1} + \tau_1 - T^n_1) \\ &= Q^n_2 + \frac{(LCR1 - LCR2)^2}{2 * AIR * LCR2} + (LCR1 - LCR2)\,(T^n_{LCR1} + \tau_1 - T^n_1) \\ &\quad + \frac{LCR1 - LCR2}{LCR1}\left(Q^{1,n}_{max} + \int_{T^n_{LCR1}}^{T^{1,n}_0 - \tau_1} ACR(x)\,dx\right) \end{aligned} \tag{28}$$

from which the approximation (3) follows, after substituting $A_{2,n}$ and T^n_{LCR1} and assuming that

$$\int_0^{T^{1,n}_0 - \tau_1 - T^n_{LCR1}} LCR1\; e^{-\frac{LCR2}{RDF}x}\,dx \cong \int_0^{T^n_{LCR2} - T^n_{LCR1}} LCR1\; e^{-\frac{LCR2}{RDF}x}\,dx. \tag{29}$$

The time at which the queue length at switch two equals Q_2 while decreasing ($T^n_{Q'_2}$) is the solution of

$$Q_2 = Q^{2,n}_{max} + \int_0^{T^n_{Q'_2} - \tau_2 + T^{1,n}_0} \left(ACR(T^{1,n}_0 - \tau_1)e^{-\frac{LCR2}{RDF}x} - LCR2\right)\,dx$$

and the number of remaining cells in queue two at the end of the cycle,is then given by

$$\begin{aligned} Q^{n+1}_2 &= Q_2 + \int_{T^n_{Q'_2} - \tau_1 - \tau_2}^{T^n_{Q'_2} + \tau_3} (ACR(x) - LCR2)\,dx \\ &\quad + \int_{T^n_{Q'_2} + \tau_3}^{T^{n+1} - \tau_1 - \tau_2} (ACR(x) - LCR2)\,dx \\ &= Q_2 + ACR(T^n_{Q'_2} - \tau_1 - \tau_2)\frac{RDF}{LCR2}\left(1 - e^{-\frac{LCR2}{RDF}\tau}\right) - LCR2 * \tau \\ &\quad - \frac{\left(LCR2 - ACR(T^n_{Q'_2} - \tau_1 - \tau_2)\,e^{-\frac{LCR2}{RDF}\tau}\right)^2}{2 * AIR * LCR2} \end{aligned} \tag{30}$$

Inequalities (4) in Theorem 3.3 are derived since $0 \leq ACR(T^n_{Q'_2} - \tau_1 - \tau_2) \leq LCR2$.

A.2.2 ER-based switch

For ER-based switch once the congestion is detected, equation (1) is used. $ACR(t)$ is then set to $ER2 < LCR2$. The maximum queue length at switch one is reached τ_1 after an RM cell with a CI bit set to one has arrived to the source. Hence we have

$$\begin{aligned} Q_{max}^{1,n} &= \int_{T_1^n-\tau_1}^{T_P^n} (ACR(x) - LCR1)\, dx + \int_{T_P^n}^{T_{Q_2}^n+\tau_1+\tau_2} (PCR - LCR1)\, dx \\ &= \int_0^{T_P^n - T_1^n + \tau_1} AIR * LCR2 * x\, dx \\ &\quad +(PCR - LCR1)(T_{Q_2}^n + \tau_1 + \tau_2 - T_P^n) \\ &= \frac{AIR * LCR2}{2} \left(T_P^n - T_1^n + \tau_1\right)^2 \\ &\quad + (PCR - LCR1)\left(T_{Q_2}^n + \tau_1 + \tau_2 - T_P^n\right) \end{aligned} \tag{31}$$

from which equation (7) in Theorem 3.3 follows after substituting $C_{2,n}$ and $B_{2,n}$ already defined.

Queue two is the essential bottleneck iff $Q_{max}^{1,n} < Q_1$. From (31), since $ACR_{max}^n \leq ACR(z_n)$, and $C_{2,n} \leq z_n \leq z_0$ we have

$$Q_{max}^{1,n} \leq \frac{AIR * LCR2}{2} \left(T_{Q_2}^n + \tau_1 + \tau_2 - T_1^n + \tau_1\right)^2 = \frac{AIR * LCR2}{2} {z_n}^2$$

then the sufficient condition for that $Q_{max}^{1,n} < Q_1\ \forall n$ is

$$Q_{max}^{1,n} \leq \frac{AIR * LCR2}{2}\ {z_0}^2 = \frac{AIR * LCR2}{2} (2(\tau_1 + \tau_2) + A_{2,0})^2 < Q_1$$

From which we get (*iv*) and inequality (10) in Theorem 3.3.

For the maximum queue length at switch two, we need to compute the instant $T_0^{1,n}$ when queue one becomes empty. We have

$$\begin{aligned} Q_1(T_0^{1,n}) - Q_{max}^{1,n} &= \int_{T_{Q_2}^n+\tau_1+\tau_2}^{T_0^{1,n}-\tau_1} (ER2 - LCR1)\, dx \\ &= -Q_{max}^{1,n} = (ER2 - LCR1)\left(T_0^{1,n} - T_{Q_2}^n - 2\tau_1 - \tau_2\right) \end{aligned}$$

from which we get

$$T_0^{1,n} = \frac{Q_{max}^{1,n}}{LCR1 - ER2} + T_{Q_2}^n + 2\tau_1 + \tau_2 \tag{32}$$

Since $ACR(T_0^{1,n} - \tau_1) = ER2 < LCR2$, then $Q_{max}^{2,n}$ is given by

$$\begin{aligned} Q_{max}^{2,n} - Q_2^n &= \int_0^{T_1^n - T_2^n + \tau_2} AIR * LCR2 * x \; dx \\ &\quad + \int_{T_1^n - \tau_1}^{(T_0^{1,n} + \tau_2) - \tau_1 - \tau_2} (LCR1 - LCR2) \; dx \\ &= \frac{AIR * LCR2}{2} \left(T_1^n - T_2^n + \tau_2\right)^2 + (LCR1 - LCR2) \left(T_0^{1,n} - T_1^n\right) \end{aligned}$$

Finally, we get after substituting (19) and (32)

$$\begin{aligned} Q_{max}^{2,n} - Q_2^n &= (LCR1 - LCR2) \left(T_{Q_2}^n + 2\tau_1 + \tau_2 - T_1^n\right) \\ &\quad \frac{(LCR1 - LCR2)^2}{2 * AIR * LCR2} + \frac{LCR1 - LCR2}{LCR1 - ER2} \, Q_{max}^{1,n} \end{aligned} \tag{33}$$

from which we get (8) in Theorem 3.3, after substituting $A_{2,n}$.

In case of ER-based switch, $Q_2^{n+1} = Q_2^n \;\; \forall n \geq 1$; as we can see bellow, this is due to the fact that $ACR(t) = ER2$ is constant, once the congestion is detected and it doesn't depend on its maximum value during a cycle. When queue two falls bellow Q_2, the source is informed τ_3 after, by receiving an RM cell with a bit CI cleared, and increases its rate according to phase one. Hence the time elapsed since the first RM cell with bit CI cleared by the switch is sent till queue two starts to build again, is given by $T_2^{n+1} - T_{Q_2'}^n = \tau + \frac{LCR2 - ER2}{AIR * LCR2}$ (see eq. (17). Hence

$$\begin{aligned} Q_2^{n+1} &= Q_2 + (ER2 - LCR2) \, \tau \\ &\quad + \int_0^{\frac{LCR2 - ER2}{AIR * LCR2}} ((ER2 - LCR2) + AIR * LCR2 * x) \; dx \end{aligned} \tag{34}$$

from which (9) in Theorem 3.3 follows.

A.3 Steady state analysis

The conclusions that we drew for the transient analysis held for $Q_i^n = 0$, but the formulas that we developed hold for any initial queue lengths, and in particular, to the ones obtained in steady state, in the case of a periodic behavior. (A periodic behavior is indeed obtained for the ER switch, as the queue lengths become constant in steady state. For EFCI, we have observed a periodic behavior in most of the simulations).

In what follows, we do not assume that the steady state has a periodic behavior. For EFCI switches, we use the formulas developed for the transient behavior in order to compute bounds on the maximum of the queue lengths. We obtain below an asymptotic upper bound Q_i^s on the maximum length of queue i by using equations (25) and (28), in which we substitute Q_2^n by an asymptotic *lower bound* of Q_2^n:

$$Q_2^\infty \stackrel{\Delta}{=} \underline{\lim}_{n\to\infty}(Q_2^n)$$

and taking $Q_1^n = 0$. The fact that a lower bounds on Q_2^n yields an upper bound on the maximum queue length follows from the arguments in the proof of Theorem 3.1 and from Remark 3.1. Similarly, one may obtain an asymptotic lower bound on the maximum length of the queues by substituting in (25) and (28) an asymptotic upper bound for Q_2^n.

A.3.1 EFCI-based switch

From (30), we have

$$\begin{aligned} Q_2^\infty &= \underline{\lim}_{n\to\infty}\left(Q_2 + ACR(T^n_{Q_2'} - \tau_1 - \tau_2)\frac{RDF}{LCR2}\left(1 - e^{-\frac{LCR2}{RDF}\tau}\right)\right. \\ &\quad \left. -LCR2 * \tau - \frac{\left(LCR2 - ACR(T^n_{Q_2'} - \tau_1 - \tau_2)\, e^{-\frac{LCR2}{RDF}\tau}\right)^2}{2 * AIR * LCR2}\right) \end{aligned}$$

Since $\underline{\lim}_{n\to\infty} ACR(T^n_{Q_2'} - \tau_1 - \tau_2) = \underline{\lim}_{n\to\infty} LCR2\, e^{-\frac{LCR2}{RDF}(T^n_{LCR2} - T^n_{Q_2'} + \tau_1 + \tau_2)} = 0$

and since $Q_2^\infty \geq 0$, we obtain equation (11) from equation (25) and equation (12) from (28) by considering the approximation (29). The proof for Theorem 3.4 *(i)* is then established.

A.3.2 ER-based switch

For an ER-based switch we have $Q_2^n = Q_2^{n+1}\ \forall n \geq 1$. The steady state regime is, then, reached just after the first cycle. Hence, since $Q_2^\infty \geq 0$, we have from (34)

$$Q_2^\infty = \underline{\lim}_{n\to\infty}(Q_2^n) = \left(Q_2 - (LCR2 - ER2)\left(\tau + \frac{LCR2 - ER2}{AIR * LCR2}\right)\right)^+.$$

The proof of Theorem 3.4 *(ii)* follows after substituting Q_2^∞ in (31) and (33).

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
ISSN 0249-6399