Evaluation of the ER algorithm ERAQLES in different ABR environments

Y. Moret [1], S. Fdida [2], A. Fichou [3], C. Galand [3]

[1] École Nationale Supérieure des Télécomunications
F-75263 Paris cedex 13
e-mail : yan.moret@enst.fr

[2] LIP6-CNRS
Université Pierre et Marie Curie
F-75252 Paris Cedex 05
e-mail :serge.fdida@lip6.fr

[3] Centre d'Études et Recherches IBM
F-06610 La Gaude
e-mail : {alinef, galand}@vnet.ibm.com

Abstract

ABR was standardised by the ATM Forum in 1996 . Source, destination and switch behaviours were specified. However, a lot of freedom was left to the switch manufacturers to implement an efficient algorithm compliant with the ABR specifications. There exists three different behaviours. Namely, they are binary switches, Relative Rate (RR) switches and Explicit Rate (ER) switches.

In this paper, a new ER algorithm named ERAQLES is described. It is original because it uses the buffering capacities of the switches as well as a novel

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35353-1_28

D. Kouvatsos (ed.), *Performance Analysis of ATM Networks*
© IFIP International Federation for Information Processing 2000

control function to derive an optimum explicit rate for the connections. The basic mechanisms of ERAQLES are presented and its performance is compared to ERICA and MACR solutions.

The ability of these algorithms to achieve fairness in some interesting situations is investigated. We first consider the case of networking environments where both RR and ER switches are involved. Second, we explore the efficiency of these solutions when facing VBR traffic competing with ABR. Results shows that due to its design properties, ERAQLES outperforms ERICA in most situations.

Keywords

ABR, traffic management, explicit rate marking, relative rate marking, Performance analysis

1 INTRODUCTION

Although it is a controversial solution, ABR can be considered as an interesting service to be provided by carriers to their users. Therefore, it is important to demonstrate that efficient and stable solutions can be designed. The ATM forum ABR specifications (ATM Forum, 1996) include the description of the source, the destination, and the switch behaviour. However, different solutions are compliant with these specifications. A switch can implement three different mechanisms :

- the simpler ABR implementation is called the binary switch behaviour: it sets the EFCI (Explicit Forward Congestion Indication) bit of the data cells when a congestion level is reached in the ABR output queue(s). The ABR destination will then return this congestion indication to the source using the Resource Management (RM) cells. There is no possibility to differentiate low from heavy congestion. Moreover, the indication received by the source is delayed due to the time it takes to carry this signal from the congested node to the destination and back to the source.
- A second solution is the Relative Rate switch behaviour: it sets the *CI* (Congestion Indication) and *NI* (No Increase) bits in the RM (Resource Management) cells according to a switch congestion threshold. The switch can directly update the RM cells and therefore decrease the delay between the time the congestion is detected to the time the source is informed.
- Unlike the other solutions where a blind computation of the source rate is done, the ER switch behaviour aims at providing each connection with its explicit rate. This requires a specific hardware, but is recognized to provide the best performance in terms of cell loss, fairness, and throughput.

The performances of the different solutions are usually measured in terms of fairness among the ABR connections and throughput achieved on the transmission links.

The design of efficient ABR ER algorithms has been extensively studied the last few years. One of the first issue was to decide whether to use credit (Kung 1994) or rate based mechanisms. Although, the former solution exhibits some advantages, the rate-based solution was chosen (Bennet, 1994, Van Boven 1995). Thereafter, ER marking was recognized as being more efficient than RR marking, although it increases the switch complexity that is a major issue in ABR design. EPRCA (Enhanced Proportional Rate Control Algorithm) (Roberts, 1994) was proposed and highly cited in many papers that were either addressing performance studies (Fang, 1994, Ritter, 1996, Ohsaki, 1995, Ohsaki, 1996, Barnhart, 1994) or suggesting improvements (Siu, 1994, Mascolo, 96). The ER is computed according to a congestion threshold. These algorithms are pretty simple but are not always fair and often unable to control situations where sources are not greedy. A second set of algorithms was initiated at Ohio State University: OSU or ERICA (Jain, 1996, Jain, 1997) as well as at MIT (Charny, 1995, Charny, 1996). In both cases, a switch needs to estimate the source rate in order to compute the ER that is mainly a function of the links availability. Efficiency is improved at the expense of an increased complexity (if n is the number of connections flowing through a switch, some parameters are computed in $O(n)$). With these algorithms, the rate allocated to the source is often less than the available rate and fairness is not completely achieved. Most of the subsequent algorithms are a mixed of the two above mentioned ones.

A third generation of algorithms appeared recently with the objective to fix the above mentioned problems. They use the ABR buffer capacity to compute the ER. The first solution presented in this paper is ERAQLES. It is shown that ERAQLES outperforms the other solutions. ERAQLES fairness and convergence have been demonstrated mathematically (Moret, 1997) and evaluated through simulation in various configurations (Moret, 1997). The second solution akin to ERAQLES is ERICA+ (Jain, 1997). It is an extension of ERICA that has not been completely specified and formally proved.

ERAQLES and ERICA behaviours are presented in section 2 and will be compared in the remainder of the paper. Section 3 considered the situation when switches provided by different vendors are in the same network. These switches might implement RR or ER algorithms and it is therefore of utmost importance to verify their robustness when both solutions have to interact. Section 4 investigates the case where VBR sources are competing with ABR traffic. Section 5 concludes the paper.

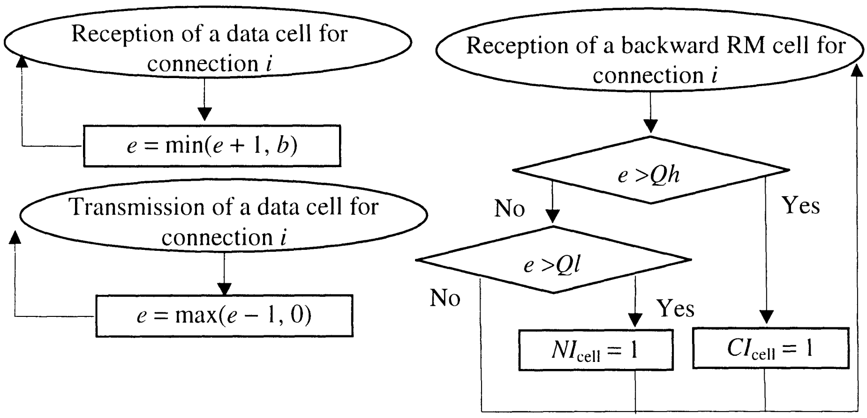

Figure 1. RR algorithm implementation in switches.

2 ERAQLES AND ERICA

2.1 RR switches

A Relative Rate algorithm like [2] is described in figure 1. We define the following notations :

- *e* is the number of cells queued in the ABR queue ;
- *b* is the ABR queue maximum size ;
- *Qh* a high-level threshold. If *e* is larger than *Qh*, the RR algorithm assumes that the switch is heavily congested ;
- *Ql* a low-level threshold. If *e* is larger than *Ql*, the RR algorithm assumes that the switch is lightly congested ;
- NI_{cell} and CI_{cell} the fields of the RM cells.

2.2 ERICA switches

ERICA has been chosen as a comparison to ERAQLES because it is a well-known solution that was extensively studied in the literature, well understood and found to be efficient. After several experiments, the designers of ERICA have mentioned that their solution was not always fair. Therefore, the specification was modified in order to propose a better solution (Jain, 1997). This latter version is not considered in this paper because it still shows many important problems. Indeed, ERICA may become totally unstable when some connections have a non zero MCR (Minimum Cell Rate), and is not able to provide a fast computation of the total available rate nor the explicit rate for each connection. Figure 2 presents the switch behaviour of ERICA with the following definitions :

- *AI* is the timer used to compute all variables of the switch (except *SBRMi* and *ERCi*) ;
- *TU* is the Target Utilisation of the link for ERICA. *TU* is set to a value lower than 100% in order to reduce the ABR queue size and therefore, the end-to-end delays ;
- *Cabr* is the total available bandwidth for the ABR service ;
- *TCR* is the Target Cell Rate according to the *TU* and *Cabr* values ;
- *CLL* is the Cell Load Level of the switch ;
- *RCC* is the Received Cell Counter. It allows to compute *CLL* ;
- $\mathfrak{R}_i$ denotes if connection *i* is considered active. If during the current timer period, a data cell is received from connection *i*, ERICA assumes that the connection *i* is active for the current timer period. This variable is reset for all connections when the timer expires;
- *FS* is the computation of the Fair Share rate. It is equal to *TCR* over the number of active connections ;
- ERC_i is the ER computation for connection *i* during the current timer period. If *CLL* is larger than one, it means that the switch can accommodate a higher traffic and the connection is allowed to increase its Current Cell Rate (according to the CCR_{cell} field of the RM cell). If not, the ER computation results in a rate decrease. In all situations, ER must not exceed *FS* ;

- $SBRM_i$ means that a Backward RM cell was received on connection i before AI has timed-out. It allows to compute a unique ER value for connection i during AI. This parameter is reset when the timer expires ;

For more details, a complete description of ERICA is provided in Jain (1996, 1997).

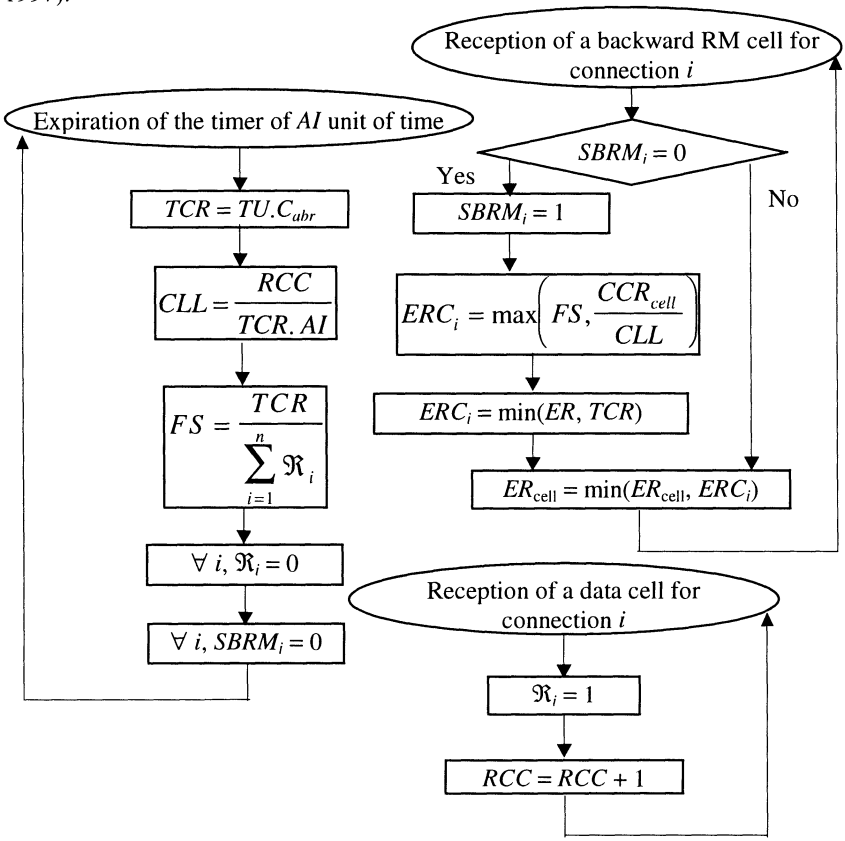

Figure 2. ERICA algorithm implementation in switches.

2.3 ERAQLES switches

ERAQLES behaviour is described in Figure 3 were:

- e, is the number of cells queued in the ABR queue,
- b, is the maximum ABR queue size,
- r, is the target number of cells in the ABR queue,
- r', is an average of e over a period equal to $NNrm$ RM cells,
- T_{max}, is the maximum delay between sources and a switch, adjusted at every connection set-up,
- n, is the number of ABR connections alive at time t.

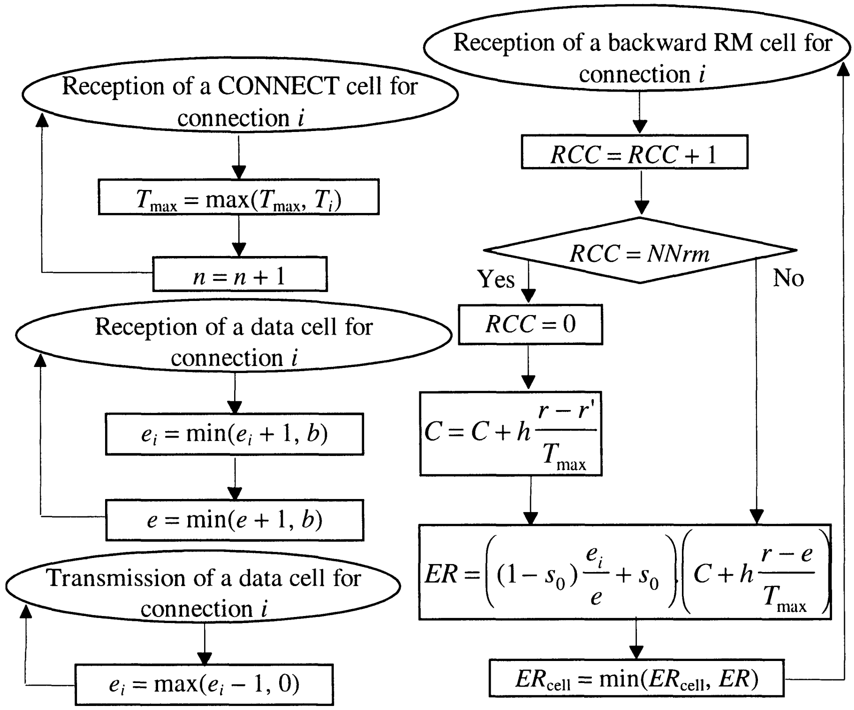

Figure 3. ERAQLES algorithm implementation in switches.

The ER computed for connection *i* is $\gamma_i = \gamma . s_i$ where s_i is the fairness function and γ the total available bandwidth for ABR services. A simple choice for s_i is $s_i = \frac{1}{n}$, where *n* is the number of active connections going through the switch. While this computation is simple, it does not take into account lightly loaded (referred as « lazy ») sources that do not use their maximum allowed bandwidth. In order to use the bandwidth left by « lazy » sources, a new function is introduced:

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

where $s_0 = \frac{1}{n}$ and ρ_i is the ratio between e_i (the number of cells for connection *i* queued in the ABR queue) and *e* (the total number of cells queued). s_0 is the minimum bandwidth that can be allocated. Some properties of ERAQLES are summarized below. According to the function *s*, if *i* and *j* are not two "lazy" connections, we obtain

$$|s_i - s_j| = |1 - s_0| . |\rho_i - \rho_j|$$

s_i (respectively ρ_i) is the value of the new (respectively current) bandwidth allocation ratio for connection *i*. In addition, we have $|1 - s_0| < 1$. Therefore, the difference between the bandwidth allocated to i and j will decrease, and s_i will converge to s_j. On the other hand, if i is a "lazy" connection, we have $\rho_i < \rho_j$ that implies $s_i < s_j$. Connection j will then get a larger share of the bandwidth than connection i and be allowed to use the bandwidth ratio saved by the "lazy" connection. The total ABR bandwidth available in the switch (γ) is computed as follows:

$$\gamma = C + h\frac{r - e}{T_{\max}}$$

where T_{max} is the control period (maximum delay between sources and the switch), and C the estimation of the bandwidth left unused by reserved (CBR and VBR) traffics. C is re-evaluated every *NNrm* RM cells. In fact, $(h.r/T_{max})$ can be interpreted as the maximum amount of bandwidth that the switch can distribute at a given instant, while still being able to control the flow of cells in the following control period. It was shown (Roche, 1995) that this control function pulls the ABR queue to converge to an utilization (filling ratio) of r. h is an important parameter of the resource evaluation function. The larger it is, the slower the convergence. The optimum value for h (0,1839) was derived mathematically (Moret, 1997).

In order to compute the ER for connection i, ERAQLES needs to know the number of cells for this particular connection stored in the ABR queue. For that purpose, ERAQLES uses a table where this information is stored for each ABR connection. Although this table is of size n, whatever the event being processed, the reading or the modification of this table is always limited to a single element. Then, the complexity is in O(1) and does not depend on the number of connections going through the switch. If we now consider ERICA as a comparison, we found that it also needs a table of size n that accounts for the number of active connections. However, unlike ERAQLES, the complexity to manage this table can be in O(n). This is because the entire table must be reset every *AI* units of time (the designer of ERICA suggest that *AI* must be lower than 1 ms). We can conclude that the complexity of ERAQLES is lower than the one of ERICA.

3 ER ALGORITHMS IN DIFFERENT ENVIRONMENTS

The ability of these algorithms to achieve fairness in some interesting situations is investigated. We first consider the case of networking environments where both RR and ER switches are involved. The homogeneous case assumes that all switches are compliant to an ER algorithm (ERAQLES or ERICA). In the heterogeneous case, the switches that experience the bottleneck conforms to RR switches (section 2.1). This situation was chosen because a Relative Rate switch is the bottleneck one and it appears that ERICA does not behave as expected when RR switches are experiencing/controlling the bottleneck. To our knowledge, this problem was first discovered by Plotkin and Sydir (Plotkin, 1997). The configuration shown in Figure 4 is exactly the same as the one studied by Plotkin and Sydir. It was important to take exactly the same situation to demonstrate that ERAQLES was able to perform properly, unlike ERICA.

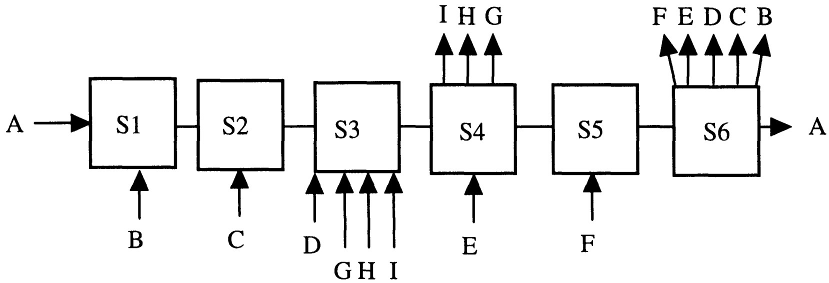

Figure 4. Network topology with multiple bottlenecks.

The network includes 6 switching nodes (S1 to S6) and 9 ABR greedy sources, from A to I. The maximum throughput of the links is C_{tot} = 385 000 cells per second (=155 Mbps). The propagation delay between each consecutive node is 250 ms (= 50 km). Therefore, the first bottleneck is located between nodes S3 and S4, and affects connections A to D and G to I. The second bottleneck is between S5 and S6 and influence connections A to F. Roughly, each of the connections A to D and G to I will be able to transmit 54 000 cells/s (C_{tot} /7), while E and F will send 80 000 cells/s (3/2* C_{tot} /7). The other parameter values are the ABR queue size b = 2000 cells, the decrease factor RDF = 1/16 and the increase factor RIF = 1/256.

The default parameter values for the ERAQLES switches are $NNrm$ = 512 RM cells and r = 500 cells. For ERICA, the default parameter values are TU = 90% and AI = 1 ms that corresponds to the values recommended by their designers. Finally, for the RR switches, we have Ql = 600 cells and Qh = 1200 cells.

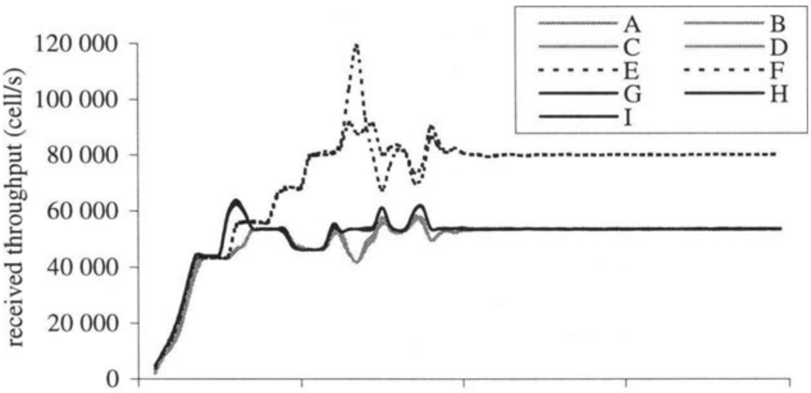

Figure 5: throughput per connection in an homogeneous configuration (ERAQLES)

3.1 ERAQLES performance in an homogeneous environment

The simulation results for ERAQLES are shown in Figure 5. They show that after a certain convergence delay, the transmission rates of the different connections converge to the expected values (54 000 cells/s and 80 000 cells/s). Other algorithms like ERICA would achieve about the same results. As a conclusion, good results are obtained when all switches are implementing an ER algorithm.

3.2 ERAQLES and ERICA in an heterogeneous environment

The congested nodes, S3 and S4, implement the RR behaviour described in section 2.1. All the other nodes implement ERAQLES or ERICA ER algorithms. The simulation results are shown in Figure 6 for ERAQLES and Figure 7 for ERICA.

For ERAQLES, while the convergence delay is larger than in the homogeneous case, the allocated bandwidth converges to 54 000 cells/s and 80 000 cells/s depending on the sources. The fairness property, already demonstrated for ERAQLES in an homogeneous environment is preserved in a mixed environment where RR and ER switches are interconnected. In this case, ERAQLES is said to be compatible with RR switching nodes. This is an important result because it was shown that ERICA is not able to provide a good bandwidth sharing in the same situation. Therefore, ERAQLES is more robust than ERICA.

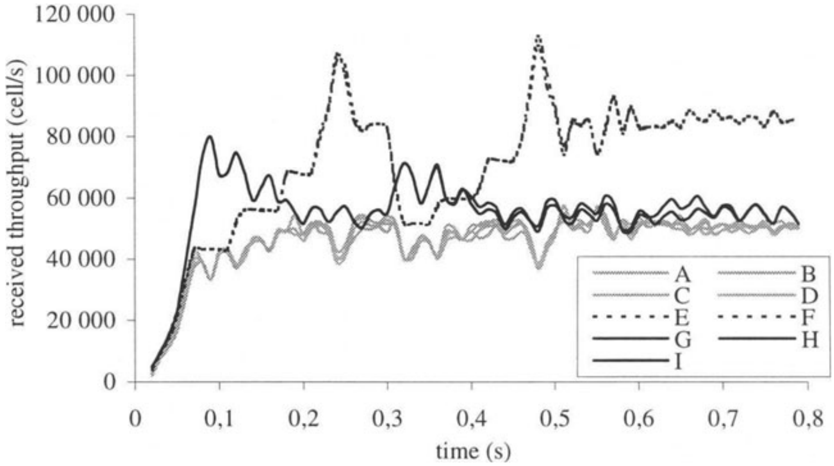

Figure 6: throughput per connection in an heterogeneous configuration (ERAQLES)

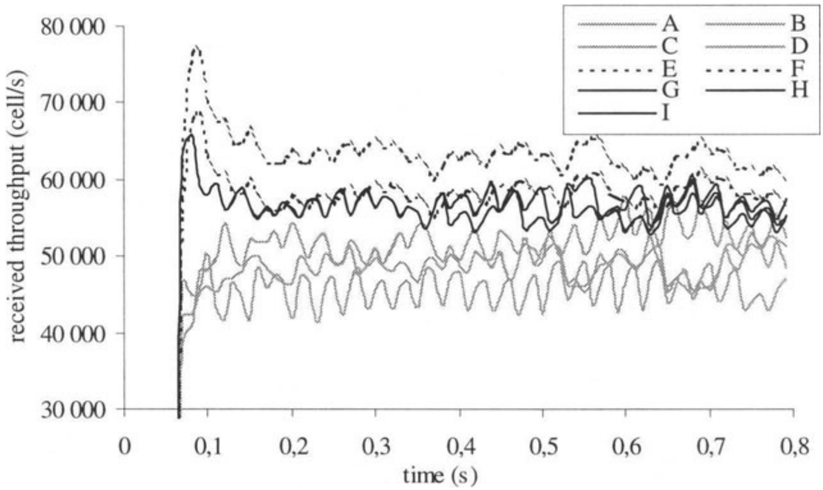

Figure 7: throughput per connection in an heterogeneous configuration (ERICA)

4 ERAQLES, ERICA AND VBR SOURCES

The objective of this section is to explore the behaviour of both solutions when the network is fed by VBR sources that compete with ABR. The simple example considered in this section is sufficient to emphasize the interest to use the ABR queue size capacity (as in ERAQLES) to increase the statistical gain and achieve the target throughput. All the tests are based on the network configuration shown in Figure 8.

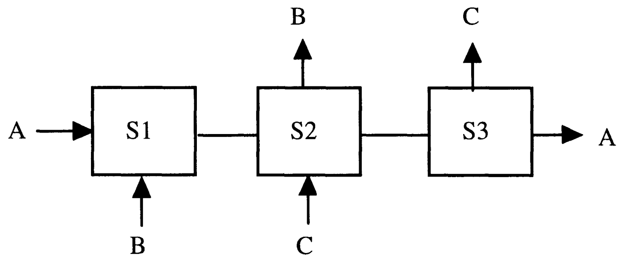

Figure 8. A simple onfiguration with 2 VBR sources.

The distance between 2 consecutive switching nodes (S1,S2 and S3) is 1 kilometer. A is an ABR connection while B and C are VBR connections. B is attached to S1 and terminated in S2. Its traffic will modify the computation of the rate for the ABR connection on the link S1-S2. Similarly, C will influence the computation of the rate on link S2-S3. The VBR sources are modelled by the same ON-OFF processes. Under these conditions, the links between the switches can be congested independently. For an algorithm that does not use the ABR buffer size available in the switch, the control algorithm will consider that the network is congested when a single VBR source is active. Otherwise, it will consider that the network is congested when the two VBR sources are active. Thus, the second algorithm must be more efficient that the first one.

During the activity period, the rate of the VBR sources are 200 000 cells/s and the mean activity/inactivity period length is δ. The other parameters are the maximal link rate C_{tot} = 385 000 cells/s, b = 10 000 cells, r = 5000 cells, $RIF = PCR = C_{tot}$ ($RIF = PCR$ so that the ABR source will always transmit at the ER value carried in the RM cell).

4.1 Simulations results with a δ constant

In this section, we assume that the activity/inactivity period δ is constant. Two synchronization scenarios between the VBR sources are considered as shown in Figure 9.

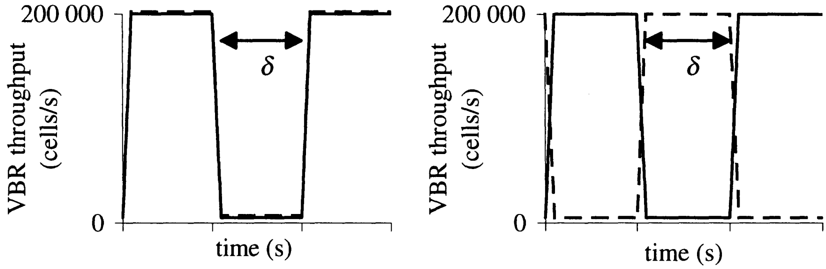

Figure 9. Synchronization schemes for the 2 VBR sources.

When the VBR sources are fully synchronized, the available bandwidth on S1-S2 and S2-S3 links when B and C are active is reduced by 200 000 cells/s. Then, the ABR connection (A) should be able to transmit 185 000 cells/s (including RM cells). The simulations done with any value for δ show that with ERICA, A is able to send 260 000 cells/s and with ERAQLES 284 000 cells/s. In fact ERICA throughput is bounded by a target utilization set to 90% (260 000 = 285 000 * 0.9). In this case, similar results are obtained with ERICA and ERAQLES.

We now assume that the VBR sources are in opposition: they are never active simultaneously. The throughput achieved as a function of δ is presented in Figure 10. It shows that the throughput allowed by ERAQLES is larger than with ERICA:

- The throughput derived from ERICA is constant and equal to 167 000 cells/s; In such a situation, ERICA measures a constant VBR load of 200 000 cells/s, and then allows the ABR source to transmit at 167 000 cells/s ((385 000 - 200 000)*0.9).
- The throughput derived from ERAQLES varies from 285 000 cells/s (small δ) to 185 000 cell/s (large δ). In fact, S1 absorbs the traffic issued by A, which implies that the ABR queue in S2 is often lower than r. When B is OFF, the queue in S1 decreases ; its occupancy will also be lower than r and ERAQLES will compute an ER larger than 185 000 cells/s. Moreover, when B

gets ON, there is a certain delay necessary to fill the ABR queue in S1 up to *r*; during this delay, the rate is larger than 185 000 cells/s. Then when δ is large, the transition periods are not frequent and ERAQLES has the same behaviour than ERICA. On the other hand, when the transitions are frequent, ERAQLES behaviour is improving, reaching the same throughput than when sources are synchronized.

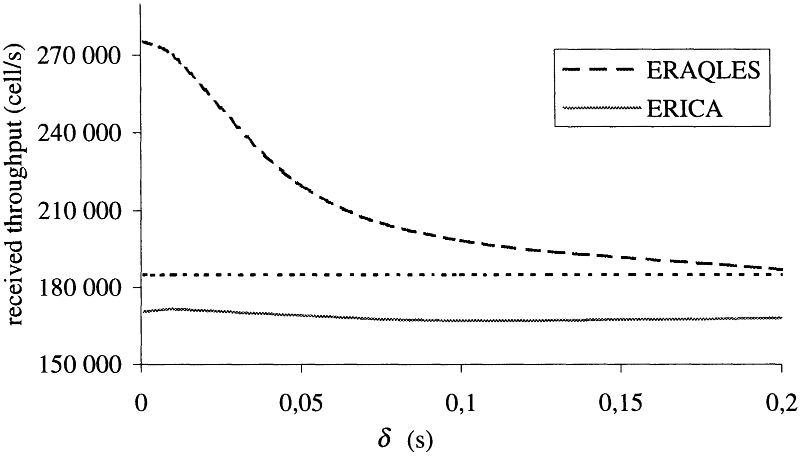

Figure 10. simulation results with 2 VBR sources in opposition - constant period.

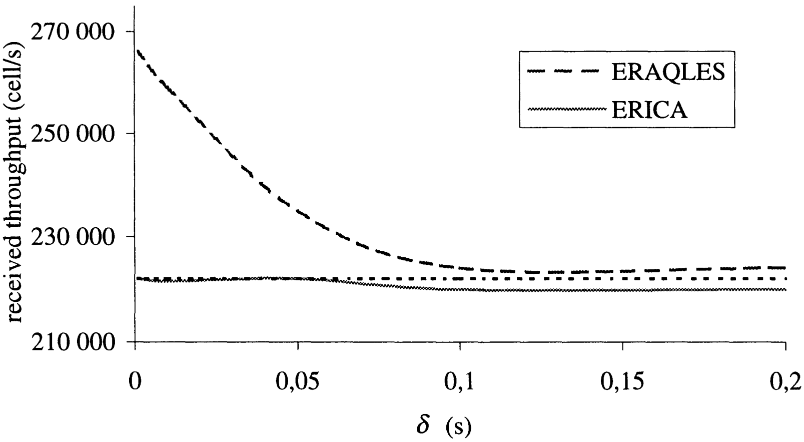

Figure 11. Simulation results with exponential ON/OFF period duration.

4.2 Simulations results with an exponential δ

Additional tests where carried out when the active period length follows an exponential distribution. Results are presented in Figure 11. They show that ERICA behaviour is improved compared to the previous situation; the maximum throughput value is 222 000 cells/s instead of 185 000 cells/s, which implies a total of 322 000 cells/s on the links (83.6%) for a target load of 90%. However, the difference is still in favour of ERAQLES that can allow up to 268 000 cells/s, that corresponds to a total throughput of 95.6% on the links.

5 CONCLUSION

ERAQLES is a novel Explicit Rate ABR algorithm. It is original because it uses the buffers available for the ABR traffic in the switches. This feature allows to smooth the rate variations due to the VBR traffic and therefore limit the throughput degradation during the congestion periods. Moreover, it uses a novel control function that makes the ABR queue converges to a target value *r*. It was important to show that ERAQLES is robust in various environments where ERICA was found to have problems. We have chosen ERICA as a comparison « metric » because it is a well-know solution that was extensively studied in the literature and found to be efficient.

The simulation results presented in this paper have shown that our solution provides a fair and efficient bandwidth allocation among ABR connections, whatever the environments being considered. The allowed rate and link throughput achieved by ERAQLES are about 25% larger than with ERICA.

6 REFERENCES

ATM Forum (1996) Traffic Management Specifications, Version 4.0, *ATM Forum contribution*, af-tm-0056.

Barnhart, A. (1994) Example of Switch Algorithm for TM Specification, *ATM Forum contribution,* 95-0195.

Bennett, J. C. R. Chang, K. Kung H. T. and Lin, D. (1994) A Comparison of EPRCA September 1994 version and FCVC Control Schemes: Simulation results, *ATM Forum contribution*, 94-0929.

Charny, A. Clark, D.D. and Jain, R. (1995) Congestion Control With Explicit Rate Indication, *IEEE INFOCOM'95*, 1954-63.

Charny, A. Ramakrishnan, K.K. and Lauck, A.G. (1996) Scability Issues for Distributed Explicit Rate Allocation in ATM Networks, IEEE INFOCOM'96, 1198-205.

Fang, C. and Chen, H. (1994) TCP Performance of Enhanced PRCA scheme, *ATM Forum contribution*, 94-0932.

Jain, R. Kalyanaraman, S. Goyal, R. Fahmy, S and Viswanathan, R. (1996) ERICA switch algorithm: a complete description, *ATM forum contribution* 96-1172.

Jain, R. Kalyanaraman, S. Goyal, R. Fahmy, S and Viswanathan, R. (1997) ERICA switch algorithm for ABR traffic management in ATM networks, part I: descrpition, *submitted to IEEE/ACM Transactions on Networking*.
Kung, H.T. and Morris, R. (1994) Credit-Based Flow Control for ATM Networks, *ACM Sigcomm'94*, 101-14.
Mascolo, S. Cavendish, D. and Gerla, M. (1996) ATM Rate-Based Congestion Control Using a Smith Predictor: an EPRCA Implementation, *IEEE INFOCOM'96*, 569-76.
Moret, Y. and Fdida, S. (1997) Design and analysis of an ABR Explicit Rate algorithm: ERAQLES, *submitted.* Available at http://www-rp.lip6.fr/~moret.
Moret, Y. and Fdida, S. (1997) ERAQLES an Efficient Explicit Rate Algorithm for ABR, *IEEE GLOBECOM'97*, Phœnix, USA, **2**, 801-5.
Ohsaki, H. Hasegawa, G. Murata M. and Miyahara, H. (1996) Performance Evaluation and Parameter Tuning of TCP over ABR Service in ATM Networks, *IEICE Transactions on Communications*, 668-83.
Ohsaki, H. Murata, M. Suzuki, H. Ikeda, C. and Miyahara, (1995) H. Rate-Based Congestion Control for ATM Networks, *Computer Communication Review*, **25**, 60-72.
Plotkin, N.T. and Sydir, J.J. (1997) Behavior of multiple ABR flow control algorithms operating concurently within an ATM network, *IEEE INFOCOM'97*.
Ritter, M. (1996) Network Buffer Requirements of the Rate-Based Control Mechanism for ABR Services, *IEEE INFOCOM'96*, 1190-97.
Roberts, L. (1994) Enhanced PRCA (Proportional Rate-Control Algorithm) *ATM Forum* contribution, 94-0735R1.
Roberts, L. (1994) Enhanced-PRCA (Proportional Rate Control Algorithm), *ATM forum contribution* 94-0735.
Roche, C. and Fdida, S. (1995), A Dynamic Resource Management Mechanism for LAN Interconnection across High-Speed Networks, *IEEE Infocom'94*, Toronto, Canada.
Siu, K.Y. and Tzeng, H.Y. (1994) Intelligent Congestion Control for ABR Service in ATM Networks, *Computer Communication Review*, **24**, 81-106.
Van Boven, C. Polak A. and Gagnaire, M. (1995) ABR Traffic Management in ATM Networks: a comparison of the Rate-based and of the Credit-based approaches", *First IFIP Workshop of ATM Traffic Management WATM'95*, Paris, France, 205-20.

7 BIOGRAPHY

Yan Moret received his Ph.D in computer science from the University Pierre et Marie Curie (Paris) in 1997. From 1994 to 1997, he worked on the design and evaluation of an explicit rate algorithm for ABR, named ERAQLES. He is currently an associate professor in the École Nationale Supérieure des Télécommunications and is involved in the design of differential services for the Internet and voice over ATM.

Serge Fdida received a Ph.D degree in 1984 and an Habilitation in 1989, both from the university Pierre and Marie Curie (Paris, France). He is currently a full professor at the University P. & M. Curie (Paris 6). He has been an assistant professor from 1983 to 1989 and a professor with the university René Descartes (Paris) from 1989 to 1995. He also spend a sabbatical year in 1995 with IBM RTP (Raleigh, USA). Serge Fdida is heading the Network and Performance Research group of the Laboratoire d'Informatique de Paris 6 (LIP6-CNRS). His areas of interest are multicast algorithms, IP over ATM techniques, congestion control and resource management in high-speed networks. He is a member of IEEE, ACM, ATM Forum and IFIP TC6 (WG6.3 & WG6.4). He is currently serving in the COST237 project as a French representative, running the working group on « Communications Support for Multimedia Applications ».