

# A New Traffic Control Algorithm for ABR Service

*A. Bak and W. Burakowski*  
*Warsaw University of Technology*  
*Institute of Telecommunication*  
*ul. Nowowiejska 15/19*  
*00-665 Warsaw, Poland*  
*fax: +48 22 660 7564*  
*tel: +48 22 25 21 60*  
*e-mail: bak@tele.pw.edu.pl*

## **Abstract**

In this paper we present an Explicit Rate (ER) ABR flow control algorithm. It directly measures the available link capacity as well as takes into account the ABR buffer occupancy. More specifically, the explicit rate calculated by the switch is proportional to the difference between predefined threshold and the actual state of the queue. Therefore, the algorithm can be regarded as the one based on a proportional control scheme. The effectiveness of the approach was verified by simulation. The paper includes sample results illustrating transient behaviour, queue occupancy, connections throughput and fairness.

## **Keywords**

**ATM Networks, ABR Service, explicit rate algorithm, flow control**

## 1 INTRODUCTION

The ABR (*Available Bit Rate*) service is presently discussed by the ATM Forum (ATM Forum, 1996) as a promising solution to serve, so called, elastic non-real time applications, e.g. computer data traffic. The implementation of this service assumes that the reactive traffic control scheme is used to control the traffic volume offered to the network. For this purpose, the ABR source sends periodically *Forward-RM* cells along the connection. These cells are then sent back by the destination to the sender as *Backward-RM* cells. Initially the source inserts its current ACR (*Actual Cell Rate*) and PCR (*Peak Cell Rate*) parameters into the RM cells (the second parameter corresponds to the desired cell rate of the connection). In the case of *Explicit Rate* algorithms the switch can reduce the value of the desired cell rate (the ER field in RM cells) to match the source sending rates to its current traffic conditions. Additionally the ABR source can use the guaranteed bandwidth allocation by declaring the *minimum cell rate* (MCR) parameter in the call set-up phase.

Numerous proposals for the ER class algorithm were submitted (e.g. (Jain, 1994a), (Jain, 1996), (Ghani, 1997), (Ait-Hallal, 1997), (Hernandez-Valencia., 1997), (Zhao, 1996)). It is worth mentioning *ERICA* (Jain, 1996), *CAPC* (Barnhardt, 1994) and *EPRCA* (Roberts, 1994), which are often considered by the ATM Forum as possible solution for the standardisation. The *ERICA* algorithm assumes that the load corresponding to the high priority traffic (required guaranteed bandwidth) as well as the ABR traffic is measured in the predefined intervals. Additionally, in *ERICA+* (Jain, 1996), the queue occupancy is taken into account. On the basis of these measurements the ER value is calculated with respect to each connection. In the *CAPC* method, only the total traffic load entering the switch is measured. The switch runs an estimate of ER value, which is updated proportionally to the difference between the measured load and assumed target utilisation. Similarly to the *CACP* method, the *EPRCA* algorithm measures also the total traffic load of the switch but the ER value is calculated on the basis of weighted average of the CCR (*Current Cell Rate*) values refereeing to ABR connections.

This paper describes an ABR flow control algorithm which considers both the link utilisation and ABR buffer occupancy. The proposed scheme, called the ER-PR (*Explicit Rate - Proportional Regulator*), belongs to the explicit rate class and assumes that the ER parameter is a function of the current queue size. The ER value, calculated by the switch, is proportional to the difference between predefined queue length threshold and the actual state of the queue. Therefore, the algorithm can be regarded as the one based on a proportional control scheme. This algorithm requires measurements of the load corresponding to both the high priority traffic and to the ABR traffic (only for constrained connections) and the number of non-constrained ABR connections being in progress.

The paper is organised as follows. Section 2 describes the proposed ER-PR algorithm in details. The exemplary numerical results showing the effectiveness of the approach are presented in section 3. Finally the appendix A gives the pseudo-code of the modelled switch.

## 2. THE ER-PR ALGORITHM

It is assumed that an ATM switch handles two types of traffic, with and without bandwidth allocation, denoted as TBA and TNBA, respectively. The TBA traffic is transferred via network with highest priority then the TNBA traffic. The switch stores cells belonging to these traffic classes in separate buffers. The TNBA buffer is served only in case there are no cells in the TBA buffer waiting for transmission (nonpreemptive priority).

Assuming the fluid flow model of the ABR connections we can make an analogy between the ABR buffer and a bucket (see Fig .1), where the buffer is filled by a number of input streams (ABR connections). Considering that the ABR traffic is of the TNBA type the capacity available for ABR connections depends on the current load of the TBA traffic. Therefore, the output rate of the ABR buffer, denoted  $C_{ABR}(t)$ , depends upon the time (it can change in the range from 0 to link capacity).

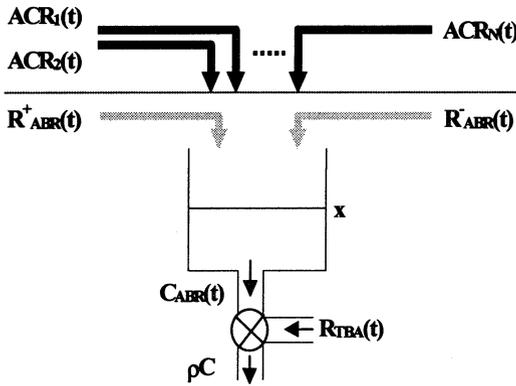


Figure 1. ABR buffer model.

Let's choose the occupancy of the ABR buffer,  $x$ , as the state variable in our system. Denoting the input flows of the  $i$ th ABR connection as  $ACR_i(t)$ ,  $i=1, \dots, N$ , the state equation can be written in the following form:

$$\frac{dx(t)}{dt} = \sum_{i=1}^N ACR_i(t) - C_{ABR}(t) = u(t) \tag{1}$$

Note that the difference between the input and the output flows constitutes the control signal,  $u(t)$ , in the system. The knowledge of this signal enables determination of the future states of the ABR buffer. From (1), we can see that the ABR buffer works as an integrator, i.e. it integrates the signal  $u(t)$  converting it to the buffer occupancy.

Denoting the instant carried TBA traffic by  $R_{TBA}(t)$  the output rate of the buffer can be expressed by:

$$C_{ABR}(t) = \rho C - R_{TBA}(t), \quad (2)$$

where  $\rho$  is the overall link utilisation and  $C$  is the link capacity.

The problem under consideration belongs to the N-input-one-output linear system class. It can be treated as the one-input-one-output system assuming that the cumulative ABR traffic stream is controlled as a whole and then equally distributed among  $N$  ABR connections. This approach is intuitively in accordance with fairness criteria but is not effective in the case when some ABR connections are bottlenecked in other switches. The ABR connections served by a switch can be divided into two groups, i.e. constrained and non-constrained connections, depending on whether they are bottlenecked or not in this switch. Notice that the switch controls only the rate of the non-constrained connections. Therefore, we can rewrite (1) in the following form:

$$\frac{dx(t)}{dt} = R_{ABR}^+(t) + R_{ABR}^-(t) - C_{ABR}(t) = u(t), \quad (3)$$

where  $R_{ABR}^+(t)$  and  $R_{ABR}^-(t)$  denote the aggregate rate of non-constrained and constrained connections, respectively.

In order to control the rate of the non-constrained connections a simple proportional regulator was used. Its objective is to stabilise the ABR buffer occupancy at the predefined threshold and, as a consequence, it gives a chance of achieving high link utilisation.

Assuming that the decision delay is produced on the output of the controller, we can write the control rule in the following form:

$$u(t) = \begin{cases} K_1 \cdot (x_0 - x(t-\tau)), & \text{when } x_0 \geq x(t-\tau) \\ K_2 \cdot (x_0 - x(t-\tau)), & \text{when } x_0 < x(t-\tau) \end{cases}, \quad (4)$$

where  $K_{1,2}$  are the gains of the proportional regulator and  $x_0$  is the buffer threshold value.

The block diagram of the control scheme is shown in the figure 2. The difference between the observed buffer state  $x(t)$  and  $x_0$  is multiplied by  $K_1$  or  $K_2$  and the result constitutes the input signal  $u(t)$ . Therefore, the rate of ABR traffic (non-constrained traffic) is proportional to the difference between the assumed and observed buffer state. With parameters  $K_1$  and  $K_2$  we can adjust the performance of the algorithm for the cases when the queue is below and above the predefined threshold.

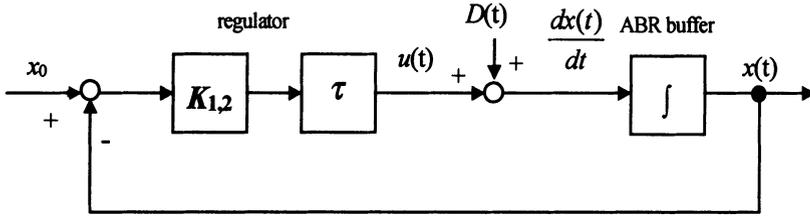


Figure 2. Block diagram of the control system,  $D(t)$  disturbance.

Taking into account that:

$$u(t) = \frac{dx(t)}{dt} \quad (5)$$

we arrive at the following formula:

$$R_{ABR}^+(t) + R_{ABR}^-(t) - C_{ABR}(t) = K_{1,2} \cdot (x_0 - x(t - \tau)) \quad (6)$$

Assuming that the value of explicit rate ( $ER$ ) parameter is calculated at the switch we can write:

$$R_{ABR}^+(t) = \sum_{i \in Z^+} ACR_i(t) = N^+ \cdot ER(t - \tau) \quad (7)$$

where  $N^+$  denotes the number of non-constrained connections and  $Z^+$  is the set of non-constrained connections. So finally the value of the  $ER$  parameter can be expressed by:

$$ER(t) = \frac{\rho C - R_{TBA}(t) - R_{ABR}^-(t) + K_{1,2} \cdot (x_0 - x(t))}{N^+} \quad (8)$$

The implementation of the control scheme given by (8) requires the measurements of the TBA traffic rate ( $R_{TBA}$ ), the constrained ABR connections traffic rate ( $R_{ABR}$ ) and the number of non-constrained ABR connection ( $N^+$ ). This is done in the measurement intervals. The explicit rate in the interval  $k$ th is calculated in the following way:

$$ER = \frac{\rho C - R_{TBA}(k) - R_{ABR}^-(k) + K_{1,2} \cdot (x_0 - x(k))}{N^+} \quad (9)$$

The TBA traffic rate can be calculated by counting the number of TBA cells served by the switch in the measurement interval. In order to reduce the influence

of the TBA traffic variability on the ABR control algorithm, the weighted averaging of consecutive measurements can be used:

$$R_{TBA}(k) = (1 - a)R_{TBA}(k - 1) + a r_{TBA}(k - 1), \quad (10)$$

where  $r_{TBA}(k)$  denotes the TBA traffic rate measured in  $k$ th interval and  $a$  denotes the averaging constant.

In order to measure the constrained connections traffic rate ( $R_{ABR}$ ) and the number of non-constrained connection ( $N^+$ ), the switch has to know which ABR connections are constrained and non-constrained. It can be done in the following way. A switch inserts into the FRM (or BRM cells) cells its unique identifier when it updates the ER value. An ABR connection is marked as non-constrained when the switch identifier equals to with the value read-out from the BRM cell. Unfortunately, it requires a slight modification of the RM cell format. Knowing which ABR connections are constrained and non-constrained the measurements of  $N^+$  and  $R_{ABR}$  can be done in straightforward way. The pseudo-code of the algorithm used in the simulations is presented in Appendix A. Two extensions of the basic scheme described above are implemented. First to avoid transient overloads, when constrained connection becomes unconstrained, the following modification in the explicit rate calculation was introduced:

$$ER = \frac{\rho C - R_{TBA}(k) - \max(0, R_{ABR}^-(k) - CCR) + K_{1,2} \cdot (x_0 - x(k))}{N^+ + 1} \quad (11)$$

where CCR is current cell rate for this connection.

Secondly for the same reason in the case of newly established connection (first seen by the switch) the explicit rate is calculated in the following way:

$$ER = \frac{\rho C - R_{TBA}(k) + K_{1,2} \cdot (x_0 - x(k))}{N} \quad (12)$$

where  $N$  is the total number of ABR connections.

### 3 PERFORMANCE STUDIES

The effectiveness of the algorithm was checked by simulation. The exemplary performance characteristics representing transient behaviour, throughput, queue size and fairness are included in this section. All presented results were obtained with 95% confidence interval.

We assume two basic test network topologies, which are usually used to study the effectiveness of ABR service, i.e. the bottleneck (BNT) and simple parking lot (PLT) configurations depicted in figure 3.

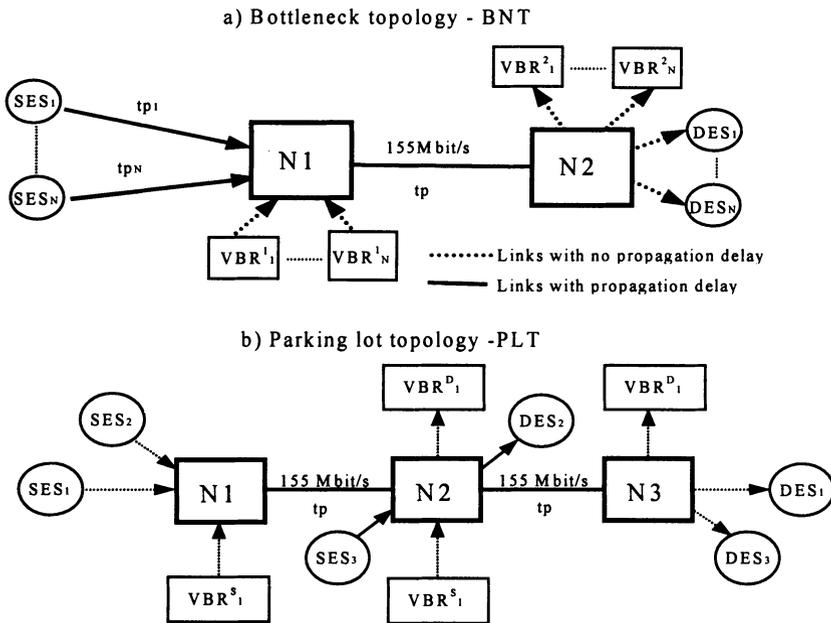


Figure 3. Studied network topologies.

The BNT topology consists of two ATM switches connected with 155 Mbps link. The traffic served by the network is a mixture of TNBA and TBA streams. The TNBA traffic is generated by 5 ABR sources while the TBA traffic is produced by 40 homogenous ON/OFF sources. The ON/OFF sources have geometrically distributed ON and OFF periods and are characterised by peak cell rate (30 Mbps), mean cell rate (3 Mbps) and ON period duration (100 slots). Each ABR source is connected to the switch with access link of propagation delay  $tp_i$  ( $i=1, \dots, 5$ ). The EFCI mode was disabled by setting  $RIF=1$  and  $RDF=\infty$ . Other ABR source parameters were set to default values.

The PLT topology consists of three ATM switches connected with 155 Mbps links. The network serves 3 groups of ABR and 2 groups of TBA sources. The ABR sources belonging to the group no.1 (SES 1 in the figure. 3) generate traffic, which is served by switches N1, N2 and N3. The sources belonging to the group no.2 (3) (SES 2 (3)) generate traffic which is served by switches N1 (N2), N2 (N3).

The parameters of the algorithm were chosen as follows: the controller gain  $K_1=K_2=0.0002$ , the queue thresholds  $x_0=0$  cells and the measurements interval  $T=250$  slots. Setting  $x_0>0$  we can obtain higher link utilisation at the expense of longest queue. In case of nonzero queue threshold the factor  $K_1*x_0$  should be kept constant e.g at 5% of link capacity. In the following experiments weighted averaging of TBA traffic was omitted ( $a=1$ ). The target rate  $\rho$  was fixed at 1.

### 3.1 Transient behaviour

The transient behaviour of the algorithm was studied by using the BNT network. In the following simulations the TBA was not present. In this experiment each ABR source has different access link propagation delay ranging from 1 up to 10000 slots (see table 2). Initially, the value of the ACR parameter for each ABR connection is set to 1 Mbps (ICR). Figure 4 shows the behaviour of ABR connections. One can observe that all connections reach the maximum ACR value (approx. 30 Mbps) after the RTT (Round Trip Time delay with minor oscillations).

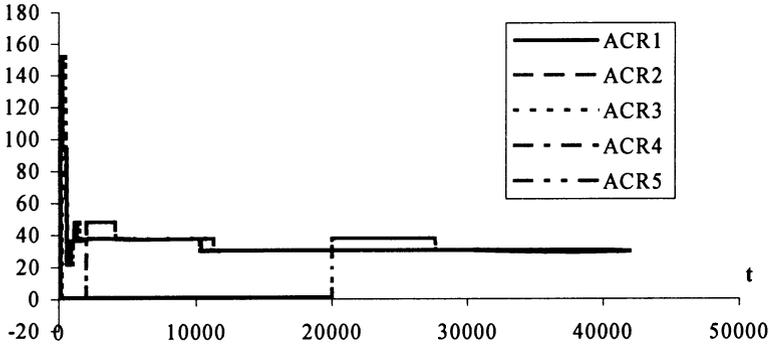


Figure 4. ACR transient characteristics.

### 3.2 Queue size and throughput characteristics

The queue size and throughput characteristics are studied using the BNT network fed by both TNBA and TBA traffic. The mean rate of TBA traffic is 60 Mbps while the ABR traffic is the same as in previous case. Additionally, it is assumed that the propagation delay is the same for each ABR connection.

The queue length characteristics are presented in terms of the required queue size (RQS) and the coefficient of the variation,  $c^2$  ( $c^2 = \text{Var}[x] / \text{Mean}[x]^2$ , where  $x$  is a random variable describing queue length). The RQS\_worst\_case is evaluated assuming the step unit function (changing from zero up to link capacity) as the background traffic (Bak, 1997) while RQS\_sim is obtained by simulation with TBA traffic generated by ONOFF sources. These characteristics vs. RTT are depicted in figure 5 and 6, respectively.

The curve in figure 5 shows that the RQS\_worst\_case mainly depends on the RTT. The resulting RQS value is a function of the signalling delay (RTT \* link capacity) and delay corresponding to the reaction time of the algorithm. The RQS\_sim is always below the RQS\_worst\_case and the difference between these curves strongly depends on the TBA traffic pattern.

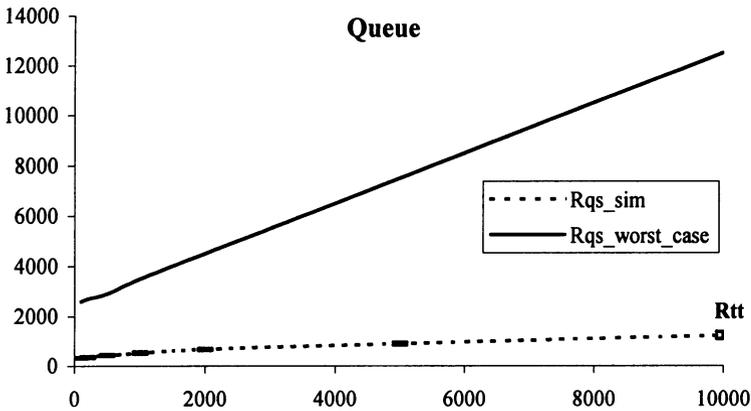


Figure 5. RQS vs. RTT.

The characteristic of the  $c^2$  referring to the queue length as a function of the RTT is depicted in figure 6. It can be noticed that for  $RTT < 1000$  (LAN and MAN networks) the variability of queue length is decreasing while for  $RTT > 1000$  slots (WAN networks) it is slightly increasing (but is close to 1). This result proves that the algorithm is stable at least for this range of RTT.

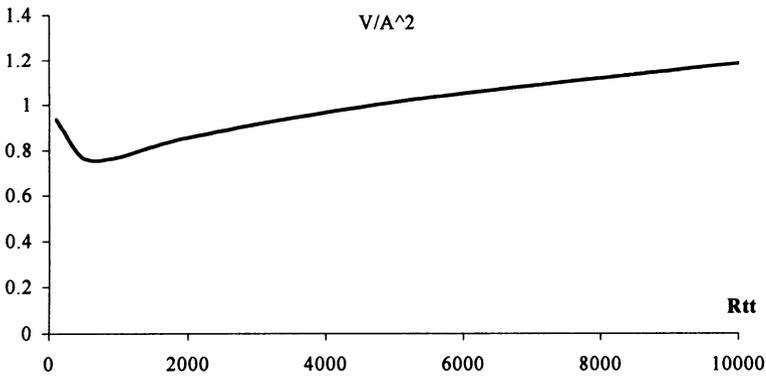


Figure 6. Square coefficient of variation for Queue State vs. RTT.

Figure 7 shows the utilisation of the inter-node link vs. RTT. One can observe that (for the considered range of RTT) the obtained link utilisation is about 97% link capacity. For small value of RTT the throughput slightly increases ( $RTT < 1000$ ) while for greater values of RTT the throughput decreases but it is still above 96% of link capacity.

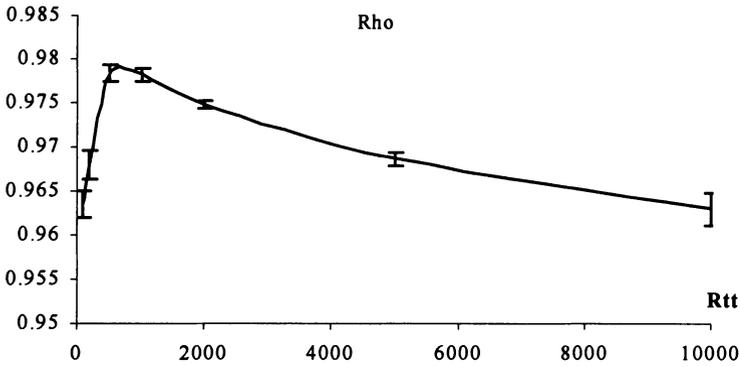


Figure 7. Throughput vs. RTT.

### 3.3 Fairness

The ABR algorithm should fairly allocate the available bandwidth between active ABR connections. The fairness of the ABR algorithms is usually defined in terms of max-min-fairness criteria (Jain, 1994).

In order to show the fairness of the ER-PR algorithm the results from two simple experiments are presented. In the first experiment the BNT network was used with 60 Mbps TBA traffic in inter-node link. Five ABR connections (unconstrained) with different propagation delay were simulated. The throughput obtained by these connections is shown in table 1. The difference between the lowest and highest allocated rate is about 0.6 Mbps. It is not too much considering the differences in RTT delays. This result can be improved by choosing greater measurements intervals.

Table 1. Simulation results for BNT network

Connection No.	1	2	3	4	5
RTT [slots]	1	10	100	1000	10000
Throughput [Mbps]	17.6	17.6	17.8	18.2	18.2
Expected throughput	18	18	18	18	18

In the second experiment the case with constrained and unconstrained ABR connections was considered. For this purpose a simple paring lot (PLT) network was used. The cell rate of TBA traffic was equal to 60 Mbps on each inter-node link leaving about 60% of link capacity for TNBA (ABR) traffic. The RTT delay for each link is 1000 slots. Two cases were simulated. In the first case the ABR connection group 2 has 10 connections while remaining groups have 5 connections. The connections belonging to group 1 and 2 are bottlenecked in node

N1 while the group 3 is bottlenecked in node N2. In the second case the group 3 has 10 connections while the remaining group have 5 connections. Now the connections of group 1 are bottlenecked in node N2. The average throughput obtained by connections of group 1-3 is shown in table 2. It can be noticed that in the first case the connections of group 3 get twice as much bandwidth as connections of group 1 and 2. Similar results were obtained for second case. As one can see the available bandwidth is fairly allocated.

Table 2. Simulation results for PLT network

<i>Group No.</i>	<i>1</i>	<i>2</i>	<i>3</i>
Number of Conn.	5	10	5
Throughput [Mbps]	5.77	5.80	11.4
Expected throughput	6	6	12
Number of Conn.	5	5	10
Throughput [Mbps]	5.7	11.5	5.73
Expected throughput	6	12	6

#### 4. CONCLUSIONS

The distributed explicit rate ABR algorithm based on queue observation and traffic measurements has been proposed. The ABR buffer state in the form of proportional control flow is used in the calculation of explicit rate parameter. This assures the stability of the algorithm. The scheme can effectively operate in the presence of TBA traffic. The included exemplary numerical result shows that the algorithm gives high link utilisation and satisfies fairness criteria.

## LITERATURE

- Ait-Hellal, O. and Altman, E. (1997) Rate Based Flow Control with Bandwidth Information", European Transactions on Telecommunications, Vol.8, No.2.
- ATM Forum, (1996) ATM Forum Traffic Management Specification Version 4.0. ATM Forum, 95-0013R11, 1996.
- Bak, A. Burakowski, W. and Kopertowski, Z. (1997) Evaluation of Required Queue Size to Support ABR Service", Document of the COST 257 project, Helsinki, September 1997
- Barnhart, A.W. (1994) Explicit Rate Performance Evaluation, ATM Forum 94-0983R1.
- Ghani, N. and Mark, J.W. (1997), „Measurement Based Flow Control for ABR Services in ATM Networks", European Transactions on Telecommunications, Vol.8, No.1, 1997.
- Hernandez-Valencia, E.J. Benmohamed, L. Nagarajan, R. and Chong, S. (1997) Rate Control Algorithms for the ATM ABR Service, European Transactions on Telecommunications, Vol.8, No.2.
- Huges, D. and Daley, P. (1994) More ABR simulation Results, ATM Forum 94-0777.
- Jain, R. Kalyanaraman, S. and Viswanathan, R. (1994a) The OSU Scheme for Congestion Avoidance using Explicit Rate Indication, ATM Forum 94-0883.
- Jain, R. (1994b) Fairness: How to measure quantitatively?, ATM Forum 94-08981.
- R. Jain et al, (1996) ERICA Switch algorithm: A Complete Description, ATM Forum 96-1172.
- Roberts, L. (1994) Enhanced PRCA (Proportional Rate-control Algorithm, ATM Forum 94-0735R1.
- Siu, K.Y. and Tzeng, H.Y. (1995) Intelligent Congestion Control for ABR Service in ATM Network, Internal Report of the Dept. of Electrical & Computer Engineering University of California, Irvine.
- Zhao, Y. Li, S.A. and Sigarto, S. (1996) A Linear Dynamic Model for Design of Stable Explicit Rate ABR Control, ATM Forum 96-0606.

```

if ( End of Measurement Interval)
  R_TBA = (1 - AVF)*R_TBA +
    AVF*(TBA_Count / MI);
  R_ABR = ABR_Count / MI;
  TBA_Count = 0;
  ABR_Count = 0;
  QS      = current queue length;
  C_ABR = RHO*LCR-R_TBA;

```

```

if (Queue not empty)
  if (TBA connection) TBA_Count++;
  else if (Cell is RM cell)
    if (Cell->DIR=0) Process FRM cell
    if (Cell->DIR=1) Process BRM cell
  else // Data cell
    if (Constrained connection)
      ABR_Count++;

```

```

// Forward RM cell
if (First Seen Connection)
  Mark this connection as transient;
  NUP++;
  if ( Cell->Switch_ID==null)
    Cell->Switch_ID=SWITCH_ID;
switch( Connection type)
case Transient:
  ER = (C_ABR +
    QueueFunction( QS) ) / (NUP+NDN);
case Non-constrained:
  ER = (C_ABR - R_ABR +
    QueueFunction( QS) ) / NUP;
case Constrained:
  ER = MAX( 0, R_ABR -Cell->CCR);
  ER = (C_ABR - ER +
    QueueFunction( QS) ) / (NUP+1);
ER = MIN(LCR,ER);
ER = MAX(0,ER);
if ( Cell->ER) > ER)
  Cell->ER      = ER;
  Cell->Switch_ID = SWITCH_ID;

```

```

// Variables
MI      - length of measurements interval
AVF     - averaging factor
K1,2  - gain
X0    - queue threshold
RHO     - target utilisation
TBA_Count - TBA cells counter
ABR_Count - Constrained connections cells
          counter
R_TBA   - measured TBA traffic rate
R_ABR   - constrained connections rate
QS      - queue state at the end of
          measurement interval
C_ABR   - ABR capacity
ER      - Explicit rate
NDN     - number of constrained connections
NUP     - number of non-constrained and
          transient connections

```

```

// Backward RM cell
if (Cell->Switch_ID==SWITCH_ID)
  if (Constrained Connection)
    NUP++;
    NDN--;
    mark this connection as non-constrained;
  else
    if (Non-constrained or transient connection)
      NUP--;
      NDN++;
    mark this connection as constrained;
  if (Constrained connection) ABR_Count++;

```

```

// Queue function - proportional control
QueueFunction( QS)
  if (QS < X0)
    return((X0 - QS)*K1);
  else
    return(( X0 - QS)*K2);

```