Evaluation of Required Queue Size to Support ABR Service

1. Introduction

The objective of the ABR service is to provide data transfer via ATM network with the cell loss ratio as small as possible. In order to achieve this goal many traffic control algorithms are discussed, e.g. EFCI, ERICA, CAPC. This paper presents approximation methods to evaluate the upper bound for the queue size in the switch. For this purpose, we assume the worst case of VBR+CBR traffic in the form of the step unit function. The method was applied to evaluate the required buffer size in case of EFCI, ERICA and ER-PR. The ER-PR method (Explicit Rate with Proportional Regulator) is the authors proposition and will be briefly described in the text. The numerical results are also included and compared with simulation.

The organisation of the paper is the following. Section 2. describes the studied algorithms. The approximation methods are outlined in section 3. The exemplary numerical results obtained analytically and by simulation are shown in section 4. Finally, the conclusions are presented.

2. The studied algorithms

Below the main characteristics of the studied algorithms for the ABR traffic control are outlined.

EFCI method

The EFCI method is the simplest way to implement the ABR traffic control. It relies on single bit congestion indication. The congestion is detected on the basis of the ABR queue observation; whenever the queue size exceeds the upper threshold (QH) the congestion is signaled and it is clear down when the queue falls back below the lower bound (QL). During the period of congestion the switch sets the EFCI bit in every data cell belonging to ABR connections. Traffic rate submitted by ARB source is regulated according to the received information about congestion/non-congestion occurrence in the connection route; the rate is increased when non-congestion is indicated and is decreased otherwise. The exact specification of the source and destination behaviour can be found in [1].

ERICA method

The ERICA algorithm (and its extension ERICA+) is now intensively studied by the ATM Forum as a possible solution for the explicit rate method. The details can be found in [2].

The switch measures the offered ABR traffic load (cell rate), the number of active ABR connections as well as the carried background traffic load. These measurements are done independently in each observation interval.

Based on these observations (for each interval) the so-called load factor (LF) is calculated:
\[ LF = \frac{ABR\_Input\_Rate}{ABR\_Capacity} \]  

(1)

where:

\[ ABR\_Capacity = \rho LCR - VBR\_CBR\_Traffic \]  

(2)

LCR – Link capacity  
\( \rho \) – target utilisation of the link (usually 0.95)

The fair share (FS) of each VC is also computed as follows:

\[ FS = \frac{ABR\_Capacity}{\text{Number}_\text{of}_\text{Active}_\text{Sources}} \]  

(3)

The switch allows each source sending at the rate below the FS to rise to FS every time it sends a feedback to the source. If the source does not use all of its FS, then the switch fairly allocates the remaining capacity to the source which can use it. For this purpose, the switch calculates the quality:

\[ VCShare = \frac{CRC}{LF} \]  

(4)

If all VCs changed their rate to their VCShare values then, in the next cycle, the switch would experience unit overload (\( z \) equals one).

The final value of the explicit rate is the max of VCShare and FS.

**The ER-PR method**

To control the rate of the ABR connections a simple proportional regulator was used which aim is to stabilise the ABR buffer occupancy. Such an approach assures high link utilisation while it tries to keep the buffer size close to a given level. Denoting the current load of the background traffic by \( C(t) = C_{CBR+VBR}(t) \) the output rate of the buffer, \( C(t) \), can be expressed by:

\[ C(t) = LCR - C_{CBR+VBR}(t) \]  

(5)

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

\[ \frac{dx(t)}{dt} = \sum_{i=1}^{NVC} ACR_i(t) - C(t) = u(t) \]  

(6)

Note that from the control theory viewpoint, the difference between the input and output flows constitutes the control signal in our system. The knowledge of this signal allows us to determined the future states of the ABR buffer. From the equation (6) we can see that the ABR buffer works as an integrator i.e. it integrates the signal \( u(t) \) converting it to the queue occupancy.

The problem under consideration is N-input-one-output linear system. It can be treated as one-input-one-output system assuming that the total ABR traffic stream is controlled as a whole and then equally distributed among NVC ABR connections. This approach agrees with fairness criteria, which requires that each ABR connection should get equal share of available link capacity. Assuming that we don’t have any knowledge about the background traffic we can treat it as unknown external disturbance in the system. Therefore, we don’t take it into account, while designing the control algorithm assuming that the output link has constant capacity, \( C(t)=LCR \). It is the aim of the feedback to compensate for the disturbance. The block structure of the control system is shown on the Fig. 1. The state equation can be now written in the following form (assuming \( C_{CBR+VBR}(t)=0 \)):

\[ \frac{dx(t)}{dt} = \sum_{i=1}^{NVC} ACR_i(t) - LCR = u(t) \]  

(7)
Figure 1. Block diagram of the control system

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

$$u(t) = K \cdot (x_0 - x(t - \tau))$$

(8)

where $K$ is the gain of the proportional regulator and $x_0$ is the assumed buffer level to be stabilised.

Taking into account that:

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

(9)

we can write the final formula for the controlled parameter $ACR(t)$:

$$ACR(t) = \sum_{i}^{NVC} ACR_i(t) = LCR + K \cdot (x_0 - x(t - RTT))$$

(10)

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

$$ACR(t) = \sum_{i}^{NVC} ACR_i(t) = NVC \cdot ER(t - RTT)$$

(11)

$$ER(t) = \frac{LCR + K \cdot (x_0 - x(t))}{NVC}$$

(12)

The value of $x_0$ can be fixed at 0.

It should be noted that the value of $NVC$ occurring in (12) corresponds to the number of ABR connections not constrained in other switch in the network. In contrary in the ERICA algorithm this term denotes the number of active sources.

3. Evaluation of maximum queue size

Cell losses in the traffic carried by ABR service can occur only when the ABR buffer is overflowed. Assuming that the switch serves a number of ABR connections not bottleneck in the source (or in another switch) the observed ABR traffic rate follows the changes of the background traffic rate. When the background traffic is of CBR type, the value of the ABR queue depends of the engaged algorithm. For instance, the EFCI method produces queue oscillations with constant amplitude. On the contrary, the algorithms based on the explicit rate mode give constant queue size (usually zero). When the background traffic is of VBR type, the ABR queue occupancy depends on the VBR traffic rate variability (range and duration of subsequent rate changes). In order to avoid cell losses in the ABR traffic the buffer size should be properly dimensioned.

In order to evaluate the required ABR buffer size in the switch we assume that the background traffic takes the form of the step unit function. It means that the link capacity, which is available for the ABR traffic is reduced from the full link capacity to zero in the shortest possible time. Therefore, the required ABR queue size depends on how quickly ABR sources will reduce their sending rates to zero. The buffer has to absorb the whole ABR traffic which is submitted to the switch after the moment the background traffic rate was changed (from zero to link rate). In the optimal case, independently on the algorithm the source can start to reduce its cell rate after the RTT (Round Trip Time). Therefore, we can distinguish two components in the resulting buffer size: the constant factor, $RTT \cdot LCR$ (Link Cell Rate), and variable factor strongly dependent on the algorithm. Illustration of queue occupancy and ACR in the cases of the studied algorithms when the background traffic takes form of the step unit function is depicted in the Fig.2.
2.1 Expressions for $Q_{\text{max}}$

In this point we briefly describe the methods to evaluate the $Q_{\text{max}}$ for the considered algorithms.

**The EFCI method**

Following the analysis presented in [3] we distinguish four basic phases of the ACR$(t)$ evolution, depending whether the system is congested/noncongested and the buffer is empty/non-empty. In this paper we analyse the EFCI method with the priority for RM cells. In this case only two ACR phases are relevant: the phase when the system is non-congested and the buffer is empty and the phase when the system is congested and the buffer is empty (see Fig.2). We recall below the final equations for these phases.
Phase 1: no congestion, buffer empty:

\[ ACR_1(t) = \max(PCR, ACR_{10} e^{\beta t}) \] (13)

\[ \beta = \frac{PCR}{NRM \times RIF} e^{-\frac{RTT}{RTT}} \] (14)

ACR\textsubscript{10} – initial value of the ACR in the phase 1
PCR – Peak Cell Rate
RIF – Rate Increase Factor
NRM – 1: number of data cells between consecutive RM cells
RTT – Round Trip Time

Phase 2: congestion, buffer empty:

\[ ACR_2(t) = \max\{MCR, \frac{RDF \times NRM}{t + \frac{RDF \times NRM}{ACR_{20}}}\} \] (15)

ACR\textsubscript{20} – initial value of the ACR in the phase 1
MCR – Minimum Cell Rate
RDF – Rate Decrease Factor

In the case with priority for RM cells these phases take place one after another. During non-congestion state the ACR evolves according to phase 1 and during congestion according to phase 2. For clarity of the presentation, in the text we assume that the value of ACR denotes the ABR cell rate observed on the input of the switch (not in the source). Therefore, it is not necessary to take into account the propagation delay between source and the switch and, as a consequence, only the RTT is important. The value of queue size at time \( t \) is given by:

\[ Q_1(t) = Q_0 + \int_0^t (NVC \times ACR_i(x) - LCR) dx, \] (16)

where: LCR – Link Cell Rate, NVC – Number of VCC ABR connections, \( i \) (i=1,2) – phase number, \( Q_0 \) - initial queue size.

The time duration of particular phases are determined by the values of queue thresholds \( Q_L \) and \( Q_H \). Based on (16) we can calculate the queue size for non-congestion (\( Q_1 \)) and congestion phase (\( Q_2 \)), respectively:

\[ Q_1(t) = Q_{10} + \frac{ACR \times NVC}{\beta}(e^{\beta t} - 1) - LCR t \] (17)

\[ Q_2(t) = Q_{20} + \frac{NVC \times NRM}{RDF} \left( \ln\left( \frac{1}{NRM} \right) - \frac{1}{RDF \times ACR_{20}} \right) - \ln\left( \frac{RDF \times ACR_{20}}{NRM} \right) - LCR t \] (18)

where \( Q_{10} (Q_{20}) \) is the initial queue size value at the beginning of the phase 1 (2).

The maximum queue size which is reached after background traffic increase, depends on the current phase of ACR evolution. As stated in [COST242], the queue has the highest value when the step of the background traffic occurs when the upper threshold is exceeded.

Therefore, during the congestion period the queue size value is calculated by:
\[
Q_2(t) = Q_{20} + \frac{NVC \cdot NRM}{RDF} \left( \ln \left( \frac{1}{t + \frac{NRM}{RDF \cdot ACR_{20}}} \right) - \ln \left( \frac{RDF \cdot ACR_{20}}{NRM} \right) \right) - (LCR - C_{CBR+VBR})t
\]

where \( Q_{20} = Q_1(t_{H}+RTT) \).

Time \( t_H \) denoting the time when the queue reaches the upper threshold is calculated by solving the following equation:

\[
Q_{H} = Q_1(t_H)
\]

The maximum queue is at the moment the ACR value is equal to LCR/NVC; it takes place at the time \( t_{max} \) conforming the equation:

\[
\frac{LCR}{NVC} = ACR_2(t_{max})
\]

Finally, the maximum queue size is calculated by:

\[
Q_{max} = Q_2(t_{max})
\]

**The ER-PR Method**

For the ER algorithms, the illustration of the buffer occupancy and the ACR in the case the step unit is assumed for the background traffic is depicted in the Fig.2. One can observe that \( ACR = LCR \) and \( Q = 0 \) when \( C_{CBR+VBR} = 0 \).

In the ER-PR method the value of explicit rate signalled to the source is directly related to the ABR buffer size. This relation is given by equation (12). Assuming that we know the rate of high priority traffic at time \( t \), the following equation can be written:

\[
Q(t) = Q_0 + \int_{0}^{t} C_{VBR+CBB}(x) + NVC \cdot Er(x-RTT) - LCR \cdot dx
\]

Using relation (12) we get:

\[
Q(t) = Q_0 + \int_{0}^{t} C_{VBR+CBB}(x) - KQ(x-RTT)dx
\]

Taking into account that the background traffic is zero for \( t < 0 \) and is constant for \( t \geq 0 \) we can write:

\[
Q(t) = Q_0 + C_{VBR+CBB}t - K \int_{0}^{t} Q(x-RTT)dx
\]

If we assume that we know the queue size in the interval \((-RTT,0)\) denoted \( Q_0(t-RTT) \) we can write for the interval \((0,RTT)\) the following equation:

\[
Q_1(t) = Q_{10} + C_{VBR+CBB}t - K \int_{0}^{t} Q_0(x_1)dx_1
\]

Analogously for the interval \((RTT,2RTT)\) we obtain:

\[
Q_2(t) = Q_{20} + C_{VBR+CBB}t - K \int_{0}^{t} Q_1(x_2)dx_2
\]

We can write the following general formula for \( k \)-th interval:
\[ Q_k(t) = Q_{k0} + C_{VBR+CBR} t - K \int_0^t Q_{k-1}(x_k) dx_k \]  

(28)

Taking into account the relations for subsequent intervals \((k-1, k-2, \ldots, 1)\) we can derive the following equation for the \(k\)-th interval:

\[
Q_k(t) = \sum_{i=1}^{k-1} \left[ \frac{(-1)^{k-1-i}}{(k-1-i)!} Q_{k-i}^0 (Kt)^{k-1-i} \right] 
\frac{(-1)^i}{(k-i)!} C_{CBR+VBR} K^{k-i} t^{k-i}
\]

(29)

\[ + (-1)^k K^k \prod_{0}^{k-1} \int_0^t Q_0(x_i) dx_i dx_2 \cdots dx_k \]

where \(Q_{k0}\) is the buffer state at the beginning of the \(k\)-th interval.

On the basis of (29) we can calculate queue size for arbitrary \(t\). In this paper we are interested in the maximum queue size. After the increase in background traffic the ABR queue oscillates before new steady state is reached. For the intervals \(k=1\) and \(k=2\) we have:

\[ Q_1(t) = C_{CBR+VBR} t \]

(30)

\[ Q_2(t) = Q_{20} + C_{CBR+VBR} t - \frac{1}{2} KC_{CBR+VBR} t^2 \]

(31)

where \(Q_{20} = Q_0(\text{RTT})\).

Assuming that \(1/K < \text{RTT}\) (practical case) the maximum queue size is reached in the interval 2 and is given by:

\[ Q_{\text{max}} = C_{CBR+VBR} \text{RTT} + \frac{1}{2} \frac{C_{CBR+VBR}}{K} \]

(32)

**The ERICA method**

In the case of the ERICA the maximum queue size is determined by average interval of the algorithm (AI), RM cell spacing \((NVC \times NRM/C_{ABR})\) and RTT. Therefore the following formula is valid:

\[ Q_{\text{max}} = (C_{CBR+VBR} + C_{ABR} - LCR) (\text{RTT} + AI + \frac{NVC \times NRM}{C_{ABR}}) \]

(33)

**4. Numerical results**

The correctness of the formulas from the section 2 is validated assuming the network structure from the Fig. 3. A number of ABR connections are established between two ATM switches. Additionally, the inter-node link serves the background traffic of the high priority.
For this system we calculate the values of the maximum queue size assuming three different ABR control schemes, e.g. EFCI, ERICA and ER-PR. These results are compared with simulation. The cell rate of background traffic was changed from 0 to 1 of the total link capacity. In the experiments number of ABR connections was 2, 5 and 10. The obtained results for different values of the RTT are presented in the Table 1.

The parameters of the studied algorithms were the following:

The EFCI method: RIF=32, RDF=16, NRM=32, QH=QL=100 (rest parameters are set to default values)
The ERICA method: AI = 100, target utilisation = 1
The ER-PR method: \( X_c = 0, K = 0.005 \)

Table 1.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>VCCs</th>
<th>RTT</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERICA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>377</td>
<td>599</td>
</tr>
<tr>
<td>5</td>
<td>402</td>
<td>634</td>
</tr>
<tr>
<td>10</td>
<td>547</td>
<td>798</td>
</tr>
<tr>
<td>ER-PR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>356</td>
<td>533</td>
</tr>
<tr>
<td>5</td>
<td>349</td>
<td>531</td>
</tr>
<tr>
<td>10</td>
<td>455</td>
<td>593</td>
</tr>
<tr>
<td>EFCI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2112</td>
<td>2242</td>
</tr>
<tr>
<td>5</td>
<td>4945</td>
<td>5177</td>
</tr>
<tr>
<td>10</td>
<td>9230</td>
<td>10214</td>
</tr>
</tbody>
</table>

Table 2.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>VCCs</th>
<th>RTT</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERICA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>312</td>
<td>567</td>
</tr>
<tr>
<td>5</td>
<td>398</td>
<td>653</td>
</tr>
<tr>
<td>10</td>
<td>541</td>
<td>796</td>
</tr>
<tr>
<td>ER-PR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>270</td>
<td>450</td>
</tr>
<tr>
<td>5</td>
<td>2351</td>
<td>5859</td>
</tr>
<tr>
<td>10</td>
<td>10162</td>
<td>10994</td>
</tr>
</tbody>
</table>

The table 1 says that the proposed approximations give accurate results. The conclusions are:
- The ERICA and the ER-PR give similar values of \( Q_{\text{max}} \) while the EFCI requires buffers of greater size
- Analytical evaluation of \( Q_{\text{max}} \) gives similar results to simulation for the ERICA and the ER-PR
- The approximation for the EFCI method gives the upper bound.

4. Conclusions

In the paper the approximate formulas to evaluate the maximum queue size for different ABR control schemes in the presence of background traffic were proposed. For this purpose the worst case corresponding to the background traffic (carried with high priority) was assumed and it took the form of the step unit function. Comparison with simulation shows that the approach is correct for all studied algorithms, i.e. ER-PR, ERICA and EFCI. The method can be applied to evaluate the required buffer size (upper bound) for the ABR traffic in network nodes to provide no cell losses. Having information about the ABR connections (RTT
values) and the value of buffer sizes assigned to absorb ABR traffic in the nodes we can find out whether the cell losses are expected or not.

References

[6] A.Bak, ABR Congestion Control Algorithm Based on the Proportional Regulator, to be published