A SIMPLE ERICA TYPE CORE-ALGORITHM TO SUPPORT TCP OVER ATM.

Georgios I. Mousadis
Theodore A. Tsiligirides
Informatics Laboratory, Dept. of General Sciences,
Agricultural University of Athens,
75 Iera Odos, 11855 Athens, Greece

Abstract
A new ER-based switch algorithm core for the ERICA scheme, namely the SERICA-scheme, used for the ABR traffic flow regulation in an ATM network is derived and discussed. The network switches monitor both their load and buffer occupancy on each link, determining the corresponding factors, the available capacity, and the number of currently active virtual channels. Based on this information the algorithm developed advises the sources about the rates at which they should transmit. It is designed on a hop-by-hop architecture. Its main purpose is to derive analytically the ERICA functional parameters $f(q)$ (dependent on the queue), in order to produce the most throughput in the environment of the Congestion Avoidance strategy.

Introduction
Today’s challenge is to support integrated Internet based applications, known as Differentiated Services over the ABR traffic. ABR service is used to carry the traffic produced from some delay insensitive or burst data applications. It is also used when it is more desirable to use whatever bandwidth is available than to get the connection rejected. The link bandwidth is first allocated to the nondelay, non-predictable traffic of the VBR and CBR classes and the remaining, if there is something left, is first given to the ABR and then to the UBR service, resulting thus to an unpredictable ABR traffic. The above framework requires a suitable flow control mechanism to automatically adjust the service rate fluctuations of the switches. However, it is desirable such an environment to be served from a globally stable feedback mechanism, in order to auto-adjust the source emission rate and to operate safely with almost 100% utilization using a threshold of the queueing delay to prevent from buffer overflows.

However, according to their traffic descriptor set, Internet based applications are classified into different Types of Services (ToS). Some ToS representatives are the typical network applications shown in Table 1 ([29]). However, to save bandwidth the so-called Differentiated Services environment is adopted. The basic characteristic of this environment is the use of the same flow control algorithm in order to serve the different ToS. The Differentiated Service environment is based on a detailed contract between the Network Service Provider (NSP) and a user which offers access on call. Depending on the network resource availability the NSP accepts or drops the connection. In

<table>
<thead>
<tr>
<th>Traffic Descriptor</th>
<th>Voice Telephony</th>
<th>Broadcast Quality Video</th>
<th>Multimedia for PCs</th>
<th>SNA Terminal Session</th>
<th>Client Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>End-to-end Delay</td>
<td>&lt;25 ms</td>
<td>&lt;50 ms</td>
<td>&lt;500 ms</td>
<td>&lt;500 ms</td>
<td>&lt;500 ms</td>
</tr>
<tr>
<td>Delay Variation</td>
<td>&lt;130 ms</td>
<td>&lt;1 ms</td>
<td>&lt;100 ms</td>
<td>&lt;500 ms</td>
<td>&lt;500 ms</td>
</tr>
<tr>
<td>Bandwidth</td>
<td>64 Kbps, constant</td>
<td>50 Mbps</td>
<td>1024 Kbps</td>
<td>16 Kbps, bursty</td>
<td>16-1024 Kbps, bursty</td>
</tr>
</tbody>
</table>

Table 1. Traffic Descriptors for Typical Network Applications

A case of acceptance the requested QoS of the corresponding ToS of the application is provided and the NSP will serve the user properly, using the traffic descriptor set of parameters already negotiated in their contract. The above interface operates either with the TCP (Transmission Control Protocol) or with the UDP (User Datagram Protocol). However, we focus our interest on the TCP since it is a connection-oriented protocol using a feedback window adjusted mechanism in the client-server loop.

Among all the ATM service schemes only the ABR service is able to guarantee for an adjusted to a feedback end-to-end flow control algorithm. For this reason the ABR service is best suited for the TCP environment. As it is well known ([16]), an ABR connection presumes a traffic contract supporting: a Minimum Cell Rate ($MCR$), a Peak Cell Rate ($PCR$) and a Cell Delay Variation Tolerance ($CDVT$), which conforms the data transfer environment. However, in order to serve all kind of traffic, ATM has to guarantee that the maximum Cell Transfer Delay ($maxCTD$) and the deterministic part of the traffic flow, namely the Sustainable Cell Rate ($SCR$), are taking some predefined values. This last property gives rise for the extended ABR service and the extended ABR traffic descriptor set of parameters. Note, that all ToS may be considered as a special case of the VBR (see also Table 1). In this sense, the CBR would be seen as a rt-VBR, with $MCR =$
PCR = SCR. Further, the ABR may be seen as MCR ≥ 0 without a specific SCR value, however, the network will do its best to share the remaining bandwidth according to a fairness criterion, in all active connections. Finally, the UBR may be seen as a special case of the ABR with MCR = 0, providing no guarantees for the parameters maxCTD and CDVT. Thus, in the Differentiated Services environment, all kind of traffic may be supported using a connection of the extended ABR traffic descriptor set of parameters, in which the parameters SCR and maxCTD are taking some predetermined values.

According to the ATM Forum, the ABR switch may support the Virtual Source / Virtual Destination (VS/VD) property [16]. Applying this property in a tandem configuration a hop-by-hop strategy may be adopted. In such a configuration a switch acts either as the VD of the previous switch, or as the VS of the next switch. To serve the ABR service, the ATM Forum proposed the Expicit Rate (ER) based switch schemes [16], which are based on the leaky bucket model and offered as an alternative of a window based or a credit based schemes. The appropriate feedback information to manage the network resources properly is transferred from end-to-end using the so-called Resource Management (RM) cells. The FRM and BRM cells travel: from the source to the destination (Forward direction = FRM cells) or from the destination to the source (Backward direction = BRM cells) to acknowledge the emission and congestion information of the source and the destination respectively. Thus, through the ER Field (ERF) of the BRM cells the source is explicitly informed to adjust its emission rate. The ER-based switch schemes are divided into two major classes. The first is the Proportional Control which is locally unstable, while the other is the Congestion Avoidance (ERICA) schemes which are stable. The stability of the ERICA algorithm is obtained by simulation using some artificial parameters to ensure a targeted utilization with a predetermined bandwidth level ((21), (28)). ERICA is suited to provide max-min fairness with MCR criterion. Nevertheless, it is not able to guarantee for a specific SCR and thus, it can not support the extended ABR traffic described above.

On a TCP over ABR service scheme, the router may suggest an “acknowledgement bucket”, incorporated in the TCP acknowledgement header, in order to inform the source for the maximum window size of its next emission period. Thus, in order to propose a new algorithm able to handle effectively the congestion effect, one has to combine the window-based scheme of TCP with the ER-based scheme of the ABR service. In order to study this environment, the main assumptions should be made are as it follows: The Fluid Behavior of the data packets (cells), are processed through the Leaky Bucket model. Further, the Feedback Mechanism is based on the RM cell and the traffic is transmitted through High Speed Channels. Thus, due to a non-trivial Round Trip Time (Propagation Delay), super data packets (cells) may be found in the pipelines. In this paper a new core ER based switch algorithm suited for ERICA, namely the SERICA, is proposed. The model undertaken consists of M switches in tandem (Figure 1), allowing the VS/VD property. This option allows a switch to divide an end-to-end ABR connection into separately controlled ABR segments by acting like a destination on one segment, and like a source on the other. Due to the above property, some hardware complexity may be added. It is assumed that a switch may serve two types of ABR traffic streams: a tagged traffic having the highest priority and a background traffic, which immediately leaves the switch. Every VS in the switch acknowledges the ERF of the BRM cell and adjusts its transmission rate every round trip time. Thus the switch may calculate the Mean Arrival Cell Rate (MACR) for the following time interval. The proposed ER-based switch algorithm utilizes almost 100% of the explicitly ordered bandwidth for the ABR service, when the corresponded basic ERICA targets from 80% to 90% of the available bandwidth [8].

The organization of this paper is as follows. Section 2 presents a brief presentation of the ERICA scheme. The ER based model description and analysis is presented in Section 3. The algorithm is solved and tested in Section 4. Here it is provided a proposal to change the algorithmic core for the switch of the ERICA scheme, with the SERICA scheme, derived by this work. This also includes some produced numerical results and the stability analysis. Finally, the conclusions are presented in Section 5.

THE ERICA SCHEMES

The main reason of introducing ERICA scheme [15] is to use the growth rate of the queue length as overload indicator. The algorithm operates at each output port (link) of a switch. The switch periodically monitors the load on each link and determines the overload factor, the ABR capacity and the number of currently active connections or VCs. Further, the switch measures the ABR input rate as the number of cell arrivals, say \( N \), per time period \( T \). If the available capacity of the link is \( C \) cells per second and the desired target utilization is \( U \), the overload factor \( z \) can be computed as \( z = N/(T*U*C) \). At the end of the measurement time interval, the switch computes the overload factor and informs all the VCs passing through it to adjust their rates according to the overload factor. The scheme also takes fairness into consideration. Fairness is achieved by ensuring that every VC gets at least a fair share of bandwidth, \( FS = (U*C)/k \), where \( k \) is the number of active VCs that where seen transmitting during the last measurement interval of \( N \) cell arrivals. Notable, that ERICA cannot always achieve max-min fairness [19]. Constant functions of \( U \) parameter restrict the system utilization to a maximum of Utilization in the steady state. Thus the system cannot achieve a queuing delay target and it does not
provide compensation when measurement and feedback are affected by errors. The alternative is to calculate the ER field using the Queue Control Functions depended on the queuing delay. Shiv. Kalyanaraman in [21], describes the core switch algorithm of ERICA+ as it follows:

$$ER(t_i) = f(q(t_i)) \cdot ACR(t_i),$$

where, using two rectangular hyperbolic functions to approximate the function $f(q(t_i))$, which both assume a value 1 at the stability buffer queue and are approximated by simulation i.e., their parameters are arbitrary. The function $f$ is lower bounded by the Queue Drain Limit Factor (QDLF), which defines the tolerance limit to a system caused variation (i.e., load, capacity, source activity).

The design of such a kind of functions is an interesting study, presented in the following section.

**MODEL DESCRIPTION AND ANALYSIS**

In this section a new ER scheme is described and analyzed in detail. The model is presented in Fig. 1 and consists of $M$ switches and $L$ links arranged in a tandem configuration. Each link $i$ is characterized by a transmission capacity $c_i = 1/\tau_i$ (cells/sec), a propagation delay $\tau_i$, a processing capacity $1/\tau_{pr}$ (cell/sec), where $\tau_{pr}$ is the time the switch $i$ needs to take a cell from the input and place it on the output queue. It is assumed that the processing capacity of each node is much larger than the total transmission capacity of its incoming links. Thus, the only reason causing congestion is the transmission capacity. The network traffic is contributed by Source/ Destination pairs of consecutive switches in the cascade queue and for such connection there is a VC associated on the path. However, the flow control does not depend on traffic other than that experienced by the cells before they reach the bottleneck queue ($\tau_i$) of the VC plus the propagation delay experienced by the BRM cells before returning to the VS ($\tau_0$).

![Cross Traffic Rate](image)

**Flow Feedback Information**

Fig. 1: Model of the Hop-by-Hop strategy (M ATM switches in tandem)

To model the dynamic behavior of each queue it is assumed that the occupancy at time $t_i$ is fully captured along the path of the VC. This simplifying assumption means that the cross traffic entering a switch does not affect the occupancy at the corresponding switch $i$ is fully captured by three state parameters, namely the Mean Arrival Cell Rate $MACR_{in}$, the Allowed Cell Rate $ACR$, and the buffer occupancy $q_i$. Thus, the $ERF_i$ of the corresponding BRM cell produced by the $i^{th}$ switch, may be written as

$$ERF_i(t_i) = F{q_i(t_i), MACR_i(t_i), ACR_i(t_i)},$$

where, $ERF_i(t_i), ACR_i(t_i)$ and $q_i(t_i)$ denote the corresponding parameters at the time $t_i = nT$, $n = 0, 1, 2, \ldots$ and $T$ is the sampling time, while $MACR_i(t_i)$ denotes the mean arrival cell rate during the period $[(n-1)T, nT]$. The ATM Forum [2] suggested

$$ERF_{in} = \min\{ERF_i(t_i), ERF_{in}\},$$

and that in such systems $ACR_{in} = MACR_{in}$ for all $i$. Without loss of generality the analysis may restricted to the single hop single VC model (see Fig. 2). Thus, the index $i$ may be dropped. The switch is assumed to have a large but finite buffer to use, of volume $q_{max}$. The Fixed Round Trip (FRT) time $\tau = \tau_0 + \tau_1$ is the propagation delay experienced by the cells before they reach the bottleneck queue ($\tau_1$) of the VC plus the propagation delay experienced by the BRM cells before returning to the VS ($\tau_0$).

![Switch Model](image)

Switch 1 (VS) \[ v(t) \] Switch 2 (VD) \[ v(t) \]

$MACR(t_a)$ $X(t_a)$ $ACR(t_0)$

$\tau_1$ $\tau_0$

Fig. 2: The single Hop single VC model. Here, the background traffic has a priority rate $v(t_a)$. Fixed Round Trip time $FRT = \tau = \tau_0 + \tau_1$.

Further, $ACR(t_a) = \mu - v(t_a)$, where $\mu$ denotes the fixed service rate of the switch and $v(t_a)$ is the exogenous not constant traffic rate at time $t_a$. Furthermore, $v(t_a)$ is an unpredictable growing / lowered parameter. Thus, in general, $ACR(t_a)$ is not a linear function of the parameters $MACR(t_a)$ and $x(t_a)$.

The new rate is calculated at the switch and it is provided to the VS through the ER of the BRM cells, which are delivered every FRT period. The VS is always able to adjust its transmission rate to the required level through the function:

$$MACR(t_a + \tau_1) =$$

$$= \min\{MACR, \min\{PCR, \max\{MCR, ERF(t_a - \tau_0)\}\}\},$$

where $MAR$ is the maximum allowed link rate.
Notably, in the worst case of traffic
\[ ERF(t_n - \tau_0) = MACR(t_n + \tau) \], or
\[ ERF(t_n) = MACR(t_n + \tau) \] (4)

**Analysis**

Let \( K(t_n) \) be the amount of the outstanding unacknowledged cells in the VS, at the \( t_n = n \tau, n = 0, 1, 2... \) \( \Delta K(t_n) \) presents the number of cells flowing to the switch from controlled traffic in \([n-1] \tau, n \tau) \) period plus the number of cells already waiting in the buffer. Therefore, if the observer were located in the VS site, then the system would be described through the following equation:
\[ K(t_n) = \tau MACR(t_{n-1} + \tau_1) + q(t_n - \tau_0) \] (5)

Differentiating the equation above, one may obtain:
\[ \frac{d}{dt} K(t_n) = \tau \frac{d}{dt} MACR(t_{n-1} + \tau_1) + \frac{d}{dt} q(t_n - \tau_0) \]

\[ \begin{align*}
\frac{d}{dt} K(t_n) & \approx \frac{\Delta K(t_n)}{\Delta t} \\
& = [A q(t_n - \tau_0) + B] \frac{ACR(t_n - \tau_0)}{u}
\end{align*} \] (6)

In the above, \( B \) represents the internal processing delay, say \( \text{pr} \), of the switch and \( u \) is the number of BRM cells send by the VD to VS every FRT period. Therefore, in case \( \Delta K(t_n) \neq 0 \), we obtain:
\[ \frac{d}{dt} MACR(t_n - \tau_0) = \left[q(t_n - \tau_0) + \frac{B}{A} ACR(t_n - \tau_0) \right] \frac{d}{dt} \tau \]

Similarly, assuming that the observer is located in the VD site and using the equation (5), the system may be described as follows:
\[ \frac{d}{dt} MACR(t_n) = \left[q(t_n) + \frac{B}{A} ACR(t_n) \right] \frac{d}{dt} \tau \]

Thus, the non-linear ODEs system controlling the transmission rate of the VS and buffer occupancy at the switch is given by the following equations:
\[ \begin{align*}
\frac{d}{dt} K(t_n) & = \tau MACR(t_{n+1}) - q(t_n) \\
\frac{d}{dt} MACR(t_{n+1}) & = q(t_{n+1}) - \frac{B}{A} ACR(t_{n+1}) \frac{d}{dt} \tau \\
\frac{d}{dt} ERF(t_n) & = \frac{B}{u} ACR(t_{n+1}) \\
\frac{d}{dt} ERF(t_{n+1}) & = \frac{B}{u} ACR(t_{n+1}) - q(t_n) \frac{d}{dt} \tau \\
\end{align*} \] (7)

where
\[ q(t_n) = \begin{cases} ACR(t_n) - ACR(t_{n+1}), & \text{if} \ (a) \\ q(t_n) = \begin{cases} ACR(t_{n+1}), & \text{if} \ (b) \\ \frac{d}{dt} q(t_n), & \text{if} \ (c) \\ \frac{d}{dt} q(t_n), & \text{if} \ (d) \\ \end{cases} \end{cases} \]

As it appears, starting with an initial buffer \( q(t_0) \) and \( MACR(t_0) \) and taking into account the equation (4), the procedure derives the \( ERF(t_n) \) using the Euler predictor method. This value is used to calculate the new buffer \( q(t_{n+1}) \) (presented in the algorithm as \( q_{pr} \)), which is then used to produce the new \( ERF(t_{n+1}) \) and so on. Note that the switch sends \( u \) BRM cells every time period \( \tau \) having the same \( ERF \) value. This is because the corresponding \( ACR(t_{n+1}) \) is assumed constant during the time period \([n\tau, (n+1)\tau)\).
\[ \frac{d}{dt} ERF(t_n) = \begin{cases} MACR(t_n) - ACR(t_n), & \text{if} \ (e) \end{cases} \]

where
\[ q_{pr} = \begin{cases} q(t_n) + MACR(t_n) - ACR(t_n), & \text{if} \ (e) \\ q(t_n), & \text{otherwise} \end{cases} \]

It follows:

\[ d \frac{q(t_n)}{dt} \approx \begin{cases} MACR(t_n) - ACR(t_n), & \text{if} \ (e) \\ 0, & \text{otherwise} \end{cases} \] (9)
Finally, assuming as the worst case of traffic the situation when the buffer overflows, say with rate \(BO(t_n)\), we have:

\[
\frac{d}{dt} BO(t_n) = \begin{cases} 
MACR(t_n) - ACR(t_n), & \text{if } (f) \\
0, & \text{otherwise}
\end{cases}, \tag{10}
\]

where, the cases a, b, c, d, e and f are given as it follows:

- case a, if \(qpr = 0 \land MACR(t_n) < ACR(t_n)\)
- case b, if \(qpr = 0 \land MACR(t_n) \geq ACR(t_n)\)
- case c, if \(qpr = qpr < q \max, \text{or } qpr = q \max \land MACR(t_n) \leq ACR(t_n)\)
- case d, if \(qpr = q \max \land MACR(t_n) > ACR(t_n)\)
- case e, if \(qpr = q \max \land MACR(t_n) \geq ACR(t_n)\), or 
- case f, if \(qpr = q \max \land MACR(t_n) < ACR(t_n)\)

**THE SERICA SCHEME.**

The resulting system of ODEs described by the equations (8) and (9) may be solved numerically using the Euler method. It is supposed here that the parameter ACR is remained steady at least for a period equal to \(\tau\). The algorithm is as follows:

\[
ER(t_n) = MACR(t_{n+1}) = \begin{cases} 
MACR(t_n) + q, \tau, & \text{if } (a) \\
q \tau + ACR(t_n), & \text{if } (b) \\
q \tau + ACR(t_n) + (MACR(t_n) - ACR(t_n)) \tau, & \text{if } (c) \\
MACR(t_n) + [q - qpr] \tau, & \text{if } (d)
\end{cases}
\]

Further, in the steady state the buffer occupancy is given by:

\[
q(t_{n+1}) = \begin{cases} 
q(t_n) + (MACR(t_n) - ACR(t_n)) \tau, & \text{if } (e) \\
q(t_n), & \text{otherwise}
\end{cases}
\]

Shiv. Kalyanaraman in [11], describes the ERICA+ switch algorithm, as it follows:

\[
ER(t_n) = f\left(q(t_n)\right) \times ACR(t_n)
\]

Thus, equalizing the two parts it follows that the proposed by Kalyanaraman function \(f(X(t_n))\) is derived analytically as follows:

\[
f\left(q(t_n)\right) = \begin{cases} 
z + T0 \times h, & \text{if } (a) 
z + T0 \times h - BORF \times h \tau, & \text{if } (b) 
z + \left[T0 - Tpr\right] h + BORF \times h \tau, & \text{if } (c) 
z + \left[T0 - Tpr\right] \tau, & \text{if } (d)
\end{cases}
\]

where \(z\) denotes the load factor (given as the rate of \(MACR/ACR\)), \(T0\) denotes the prescribed queuing delay, and \(Tpr\) denotes the predictable queuing delay for the time \(t_n+1\) (given as \(X(t_n+1)/ACR\)). The new version of ERICA switch algorithm, given as the Simple ERICA version, or the SERICA-scheme, follows:

The SERICA-scheme operates at each output port (or link) of a switch. The switch periodically monitors the load on each link and determines the load factor \(z\) the buffer occupancy rate factor \(BORF\), the Queueing Delay predictable \(Tpr\) the ABR capacity, and the number of the currently active VCs \(N\). A measurement or “averaging” time interval \(h\) is used for this purpose. These quantities are used to calculate the feedback indicated by the BRM cells. It is noticeable here that these measurements are made in the forward direction, when the feedback is given in the reverse direction. Further, the switch gives at most one new feedback per source, per \(h\). The key steps of the SERICA-scheme are as follows:

**Initialization:**

Max Alloc Previous \(\leftarrow\) Max Alloc Current \(\leftarrow\) FairShare

**End of Averaging Time Interval:**

Total ABR Capacity \(\leftarrow\) Link Cap \(-\) VBR_cap \(+\) CBR_cap

ABR Input Rate \(\leftarrow\) \(\Sigma\) (CCR_of_VC)

If \(Z\) is true THEN

BOR \(\leftarrow\) ABR Input Rate \(\leftarrow\) Total ABR Capacity

ELSE

BOR \(\leftarrow\) 0

ENDIF

BORF \(\leftarrow\) BOR \(\leftarrow\) Total ABR Capacity

If \(\text{(case}(a)\text{) is true})\) THEN

Queue Funct \(\leftarrow\) \(z + T0 \times h\)

ELSEIF \(\text{(case}(b)\text{) is true})\) THEN

Queue Funct \(\leftarrow\) \(z + \left(T0 - Tpr\right) \times h + BORF \times h \tau\)

ELSEIF \(\text{(case}(c)\text{) is true})\) THEN

Queue Funct \(\leftarrow\) \(z + \left(T0 - Tpr\right) \times h + BORF \times h \tau\)

ELSE

Queue Funct \(\leftarrow\) \(z + \left(T0 - Tpr\right) \times h\)

ENDIF

ER \(\leftarrow\) Queue Funct \(\times\) Total ABR Capacity

FairShare \(\leftarrow\) INT(Total ABR Capacity \(\div\) N)

MaxAllocPrevious \(\leftarrow\) MaxAllocCurrent

MaxAllocCurrent \(\leftarrow\) FairShare

When a FRM cell is received

CCR_ofVC \(\leftarrow\) CCR_in_RM_Cell
When a BRM cell is received

\[ \text{VCShare} \leftarrow CCR_{o.f. \text{VC}}/z \]

IF \((z \cdot GT 1 + \text{delta})\) THEN

\[ \text{ER} = \max(\text{FairShare}, \text{VCShare}) \]

ELSE

\[ \text{ER} = \max(\text{MaxAllocPrevious}, \text{VCShare}) \]

ENDIF

MaxAllocCurrent = \max(\text{MaxAllocCurrent}, \text{ER})

IF \((\text{ER} \cdot \text{GT} \cdot \text{FairShare.AND.CCR}_{o.f. \text{VC.LT.} \text{FairShare}})\)

\[ \text{ER} = \text{FairShare} \]

ERin_RMC = \min(\text{ERin_RMC.Cell,ER,Total_ABR_Capacity})

**Numerical Solution and Results**

**Numerical Solution**

Numerical results are given in Figs. 3 (a) and (b), where the MACR and the buffer occupancy versus the time are presented. From the above analysis one may point out that the linear branch of the system of ODEs (8) and (9) has a stable focus point (the relevant theory may be seen in [1]), provided the ACR\((t_r)\) remains constant for some time period and the bandwidth overflow rate in equation (10) is zero (steady state conditions). The same results (Fig.3) verify that the system has a stable focus point given by:

\[ \text{(MACR, q)} = (\text{ACR}(t_r), q) = (\text{ACR}(t_r), -B/A), \]

where the optimal value for the parameter \(A\) is given as \(A = -(u \cdot t/\text{ACR}(t_r)) \) and \(-B/A = 70 \cdot \text{ACR}(t_r)\).

**Simulation Results**

The new algorithm is tested on a simple configuration (figure 4) in the environment described below. The network configuration consists of a source, two ABR switches in tandem and a destination, connected with a single VC. The ABR traffic descriptor parameters are defined as follows: \(PCR = 36 \text{ cells/}\tau\), \(MCR = 1 \text{ cell/}\tau\) and \(h = \tau = 1\), \(q_{\max} = 2 \cdot h \cdot PCR\). To form the simulator, it is assumed that the period \(\tau\) is divided into 36 mini-slots and that a mini-slot may accommodate at most one cell. The source may generate (or do not generate) at most one cell in a mini-slot, with some probability \(p\) (or \(1-p\)) following an on-off process. During a period \(\tau\) this probability is always considered fixed (0<\(p<1\)), however it may be changed for different periods of the simulation process according to the required level of ACR and the service rate of the switch. When the buffer is full any newly arrived cell is rejected. The test bench specification [30] has been followed. Thus, extended simulation results are produced, using on-off processes, over a period of 1000\(\tau\). The SERICA scheme is tested versus the ERICA scheme. The simulation results are presented in Table 2. We begin our description of Table 4 by showing in column (2) the number of cells generated at the source, using the corresponding probability values presented in column (1). These cells will be transmitted through the specified network configuration, using the two ER-based congestion avoidance algorithms, namely, the proposed scheme, and the basic ERICA scheme, denoted in the Table 4 as ‘SERICA’ and ‘ERICA’, respectively. Column (3) presents the number of cells requested to be sent by the source in each session, while column (4) shows the actually transmitted cells during the same session. When a time slot expires, the number of cells waiting in the source to be transmitted is shown in column (5). These cells will never be transmitted and therefore they are discarded from the queue, namely, are considered as lost. This assumption may help in a theoretical analysis of the model and it is not harmful in real applications. Further, for each value of \(p\), column (6) presents the sum over the whole simulation period of the differences between the requested and the actually transmitted cells by the source. The cells finally arrived at the destination are shown in column (7). In addition, columns (8) and (10) present the number of cells taken away from the buffer queues of the switches 1 and 2, respectively, since their life-time of \(2\tau\) has been expired. Moreover, columns (9) and (11) present the average queues of the buffers of the switches 1 and 2, respectively. Finally, in column (12) the throughput of each session is presented. As it appears from the simulation, the overflow rate \(BO(t_0)\) given by the equation (19), is zero for both algorithms and thus it is excluded from the table.

The simulation shows some interesting results concerning on the overall performance of the link of the new algorithm, by means of the higher throughput achieved, which is not lower than 0.8 in all the examined cases. In addition, the average queue is higher in the new algorithm for both buffers. This is also true for the queuing delay, as one may easily deduce. However, the most interesting result is shown through the variable case of \(p\), in which the value of \(p\) does not remain constant during the session as in all the previous cases, but it varies from slot to slot, taking random values between 0 and 1. This is a more realistic situation since it corresponds to a non-WCT environment, in which sudden changes of the input rate in the source causes unexpected behavior in the system performance. Thus, the new algorithm performs very well around its operational point during the whole session in the expense of an increase in the average queue of the buffers of the switches and the number of non-conformed cells. In contrast, the basic algorithm has an unpredictable behavior caused nor because of the increase in the average queue of the buffers, neither because of the number of non-conformed cells, but mainly, because of the number of cells rejected in the source.

The above simulator is used to show the operation of the window mechanism in the configuration of Figure 4. The results produced are depicted in Figures 5a and 5b. Figure 6a shows the queue length (in cells) of each of the switches 1 and 2 over the time \(\tau\). Figure 6b shows the window (in cells), as it is proposed from the switch 1 to be implemented by the source, as well as the window actually implemented by the source.
CONCLUSIONS

In this work a new ER-based switch algorithm with congestion avoidance, namely the SERICA algorithm, derived, presented and discussed. It is based on a hop-by-hop strategy for the flow regulation of the ABR traffic in an ATM network. The network configuration consisted of M switches in tandem, with multiple VCs, allowing the VS/VD property. The analysis restricted to a single hop, in which the background traffic immediately leaves the switch and has a priority over the tagged stream cells. For the sake of hardware simplicity and low cost implementation, the proposed algorithm shares a common buffer, with FCFS service discipline. It is noticeable that it uses only the available on the switch parameters.

The proposed algorithm may dynamically adjust the adaptive VS window size, by regulating its emission rate, through the ERF of the BRM cells. The model is analysed to a non-linear system of ODEs, assuming that the ACR, MACR, and the VD buffer occupancy are modeled as fluids. It is also known that in steady-state conditions those systems are stable foci. A simple numerical method for solving such a system derived. This method is based on the Euler predictor numerical method. Using this method as an ER-based switch algorithm, the oscillating behavior of the Hop-by-Hop system is eliminated. The numerical results agree with the theory. Extended simulation results on a 2-hop, 1 VC network versus the ERICA scheme, show that SERICA is a robust algorithm, since its throughput does not drop more than 80% of the initially emitted cells from the source, especially in the unpredictable ABR environment. Thus, it may be used as a powerful tool for achieving the best possible performance. Extended simulation results will be presented in a future work.

REFERENCES


<table>
<thead>
<tr>
<th>p</th>
<th>Cells generated</th>
<th>$\Sigma(ER^*h)$ at source</th>
<th>Cells Lost at Source</th>
<th>$\Sigma(ACR^*h)$ at source</th>
<th>Cells arrived to destination</th>
<th>buffer 1 non conform ed cells</th>
<th>Buffer 1 Average Queue</th>
<th>buffer 2 non conform ed cells</th>
<th>Buffer 2 Average Queue</th>
<th>Throughput %</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9</td>
<td>32427</td>
<td>32481</td>
<td>31533</td>
<td>29313</td>
<td>894 3114</td>
<td>948 104</td>
<td>31533 29313</td>
<td>0 0</td>
<td>1.88 0.12</td>
<td>0 0 1.64 0.1 97 90</td>
</tr>
<tr>
<td>0.8</td>
<td>28830</td>
<td>30101</td>
<td>25292</td>
<td>27962</td>
<td>25104</td>
<td>868 3726</td>
<td>2139 188</td>
<td>27962 25104</td>
<td>0 0 3.59 0.19</td>
<td>0 0 3.26 0.17 97 87</td>
</tr>
<tr>
<td>0.7</td>
<td>25084</td>
<td>27932</td>
<td>25292</td>
<td>24417</td>
<td>21183</td>
<td>667 3901</td>
<td>3515 259</td>
<td>24417 21183</td>
<td>0 0 5.39 0.25</td>
<td>0 0 4.45 0.16 97 84</td>
</tr>
<tr>
<td>0.6</td>
<td>21458</td>
<td>25584</td>
<td>21442</td>
<td>20848</td>
<td>17342</td>
<td>610 4116</td>
<td>4736 272</td>
<td>20826 17342</td>
<td>14 0 6.69 0.24</td>
<td>8 0 6.4 0.21 97 81</td>
</tr>
<tr>
<td>0.5</td>
<td>17786</td>
<td>24762</td>
<td>13978</td>
<td>17551</td>
<td>13690</td>
<td>235 4096</td>
<td>7211 288</td>
<td>17305 13690</td>
<td>143 0 7.67 0.24</td>
<td>103 0 6.59 0.22 97 77</td>
</tr>
<tr>
<td>0.4</td>
<td>14235</td>
<td>25784</td>
<td>10444</td>
<td>14204</td>
<td>10140</td>
<td>31 4095</td>
<td>11580 304</td>
<td>13584 10140</td>
<td>351 0 7.22 0.25</td>
<td>269 0 5.87 0.2 95 71</td>
</tr>
<tr>
<td>0.3</td>
<td>10736</td>
<td>27824</td>
<td>6979</td>
<td>10736</td>
<td>6776</td>
<td>0 3960</td>
<td>17088 203</td>
<td>9976 6776</td>
<td>408 0 5.42 0.24</td>
<td>352 0 4.77 0.19 93 63</td>
</tr>
<tr>
<td>0.2</td>
<td>7118</td>
<td>30166</td>
<td>3686</td>
<td>7118</td>
<td>3536</td>
<td>0 3582</td>
<td>23048 150</td>
<td>6348 3534</td>
<td>467 2 3.87 0.15</td>
<td>303 0 2.93 0.11 89 50</td>
</tr>
<tr>
<td>0.1</td>
<td>3614</td>
<td>32525</td>
<td>1340</td>
<td>3614</td>
<td>1282</td>
<td>0 2332</td>
<td>28911 58</td>
<td>2895 1278</td>
<td>437 3 2.04 0.06</td>
<td>282 1 1.44 0.06 80 35</td>
</tr>
<tr>
<td>0.0</td>
<td>18661</td>
<td>30550</td>
<td>10417</td>
<td>17961</td>
<td>7994</td>
<td>700 10667</td>
<td>12589 2423</td>
<td>17456 7901</td>
<td>359 82 3.58 0.59</td>
<td>146 11 1.9 0.23 94 42</td>
</tr>
</tbody>
</table>

Table 2. The 2-hop, 1 VC network, with PCR=36 cells/τ, MCR= 1 cell/τ, and τ=1