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 $f(q)$ (depended 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 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 queuing 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 dropts the connection. In 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.

Traffic Descriptor	Typical Network Applications				
	Real-Time	**Non Real-Time**			
	Voice Telephony	**Broadcast Quality Video**	**Multimedia for PCs**	**SNA Terminal Session**	**Client Server**
End-to-end Delay	<25 ms	<50 ms	<500 ms	<500 ms	<500 ms
Delay Variation	<130 ms	<1 ms	<100 ms	<500 ms	<500 ms
Bandwidth	64 Kbps, constant	50 Mbps	1024 Kbps	16 Kbps, bursty	16-1024 Kbps, bursty
Data Integrity	$1/10^7$	$1/10^7$	$1/10^8$	$1/10^7$	$1/10^7$

Table 1. Traffic Descriptors for Typical Network Applications

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 (max*CTD*) 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 $MCR \geq 0$ 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 Explicit 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 target 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), superb 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

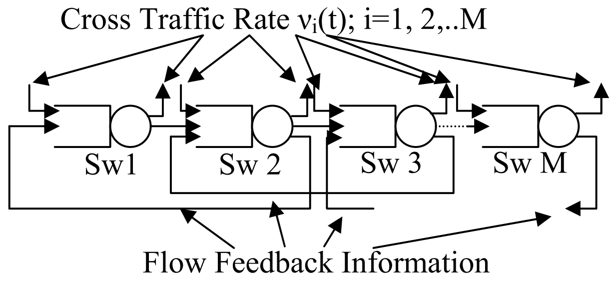


Fig. 1: Model of the Hop-by-Hop strategy (M ATM switches in tandem)

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: $ER(t_n) = f(q(t_n)) * ACR(t_n)$, using two rectangular hyperbolic functions to approximate the function $f\ (q(t_n))$, 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 $c_i = 1/ts_i$ (cells/sec), a propagation delay τ_i , a processing capacity $1/tpr_i$ (cell/sec), where tpr_i is the time the switch i needs to take a cell from the input and place it on the output queue. It is asumed 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 each such connection there is a VC associated on the path. However, the flow control does not depend on traffic other than that found along the path of the VC. This simplifying assumption means that the cross traffic entering a switch does not proceed through the tandem queues but it is renewed at each stage, i.e., it immediately leaves the system.

Each link maintains a separate queue for each VC passing through it. It is assumed that the occupancy at time t of the queue associated with link i and $VC\ j$ is $q_{i,j}(t)$ with $q\max_{i,j}$ the corresponding queue threshold level. Each source is assumed to transmit at its maximum transmission speed $\lambda_s = 1/ts_i$ and the control law computes the source input rate $\lambda(t)$(cells/sec). The bandwidth delay product τ_i / ts_i , represents the number of *in flight* cells on the transmission link. To model the dynamic behavior of each queue it is assumed a deterministic fluid approximation of cell flow. Each link maintains a separate queue for each VC passing through it.

The rate control model

This subsection presents the control algorithm, which regulates the source rates. To simplify the analysis only one VC is considered and therefore the subscript j is dropped. To control the queue length $X_i\ (t)$ for a specific VC, a simple controller is used. The state of the controlled connection at the corresponding switch i is fully captured by three state parameters, namely the Mean Arrival Cell Rate $MACR_i$, the Allowed Cell Rate ACR_i 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

$$ERF_i(t_n) = F\{q_i(t_n), MACR_i(t_n), ACR_i(t_n)\} \tag{1}$$

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

$$ERF_i(t_n) = min\{ERF_i(t_n), ERF_{i+1}(t_n)\}_1 \tag{2}$$

and that in such systems $ACR_i(t_n)$=$MACR_{i+1}(t_n)$ 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 $\tau(= \tau_1 + \tau_0)$ is the propagation delay experienced by the cells before they reach the bottleneck queue (τ_1) of the VD plus the propagation delay experienced by the BRM cells before returning to the VS (τ_0).

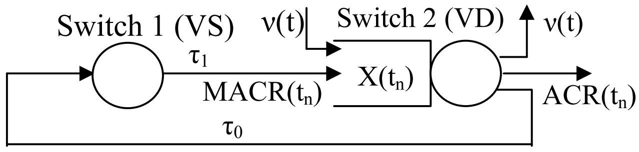


Fig. 2: The single Hop single VC model. Here, the background traffic has a priority rate $\nu(t_n)$. Fixed Round Trip time FRT = $\tau = \tau_0 + \tau_1$.

Further, $ACR(t_n)$=μ-$v(t_n)$, where μ denotes the fixed service rate of the switch and $v(t_n)$ is the exogenous not constant traffic rate at time t_n. Furthermore, $v(t_n)$ is an unpredictable growing / lowered parameter. Thus, in general, $ACR(t_n)$ is not a linear function of the parameters $MACR(t_n)$ and $x(t_n)$. The new rate is calculated at the switch and it is provided to the VS through the ERF 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: $MACR(t_n + \tau_1) =$

$$= \min\{MAR, \min\{PCR, \max\{MCR,\ ERF(t_n - \tau_0)\}\}\}, \tag{3}$$

where MAR is the maximum allowed link rate.

Notable, that in the worst case of traffic $ERF(t_n - \tau_0) = MACR(t_n + \tau_1)$, or

$$ERF(t_n) = MACR(t_n + \tau) \tag{4}$$

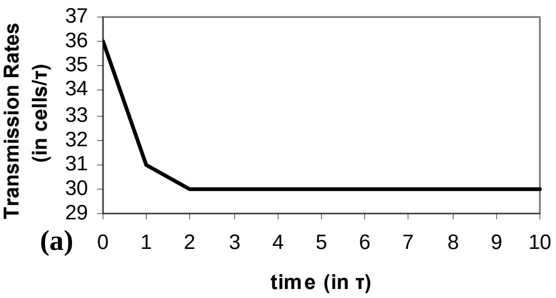
Analysis

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.. . $K(t_n)$ presents the number of cells flowing to the switch from controlled traffic in [(n-1)τ, nτ) 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:

$$K(t_n) = \tau\, MACR(t_{n-1}+\tau_1) + q(t_n-\tau_0) \tag{5}$$

Differentiating the equation above, one may obtain:

$$\frac{d}{dt}K(t_n) = \tau\frac{d}{dt}MACR(t_{n-1}+\tau_1) + \frac{d}{dt}q(t_n-\tau_0) =$$



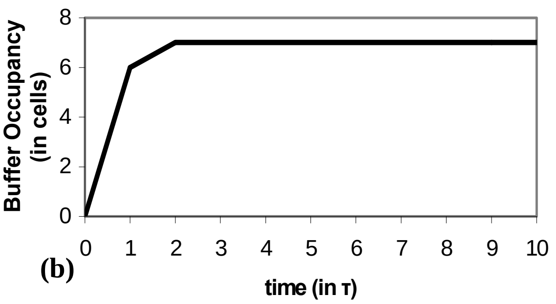


Fig. 3: The model with ACR(t_n) constant (i.e.steady state). Link Rate = 155,52 Mbps. ACR(t_n) =30 cells/τ, PCR = 36 cells/τ, MCR = 1 cell/τ and τ=100 μsec. Initial Condition (MACR(0), X(0))=(PCR, 0).

$$= \tau\frac{d}{dt}MACR(t_n-\tau_0) + \frac{d}{dt}q(t_n-\tau_0)$$

Assuming $\Delta K(t_n) = A\, q(t_n-\tau_0) + B$, and taking into account that the time between adaptation is $\Delta t = \frac{u}{ACR(t_n-\tau_0)}$, it follows:

$$\frac{d}{dt}K(t_n) \approx \frac{\Delta K(t_n)}{\Delta t} = \left[Aq(t_n-\tau_0)+B\right]\frac{ACR(t_n-\tau_0)}{u}.$$

In the above, B represents the internal processing delay, say *tpr,* of the switch and u is the number of BRM cells send by the VD to VS every FRT period. Therefore, in case $A \neq 0$, we obtain:

$$\frac{d}{dt}MACR(t_n-\tau_0) = \left[q(t_n-\tau_0)+\frac{B}{A}\right]\frac{A}{u\tau}ACR(t_n-\tau_0) - \frac{d}{dt}\frac{q(t_n-\tau_0)}{\tau} \tag{6}$$

Similarly, assuming that the observer is located in the VD site and using the equation (5), the system may be described as follows:

$$\frac{d}{dt}MACR(t_n) = \left[q(t_n)+\frac{B}{A}\right]\frac{A}{u\tau}ACR(t_n) - \frac{d}{dt}\frac{q(t_n)}{\tau} \tag{7}$$

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:

$$\frac{d}{dt}ERF(t_n) = \begin{cases} \frac{B}{u\tau}ACR(t_{n+1}), & \text{if } (a) \\ \frac{B}{u\tau}ACR(t_{n+1}) - \frac{d}{dt}\frac{q(t_{n+1})}{\tau}, & \text{if } (b) \\ \left[qpr+\frac{B}{A}\right]\frac{A}{u\tau}ACR(t_{n+1}) - \frac{d}{dt}\frac{q(t_{n+1})}{\tau}, & \text{if } (c) \\ \left[qpr+\frac{B}{A}\right]\frac{A}{u\tau}ACR(t_{n+1}), & \text{if } (d) \end{cases} \tag{8}$$

where

$$qpr = \begin{cases} q(t_n) + \left(MACR(t_n) - ACR(t_n)\right)\tau, & \text{if } (e) \\ q(t_n), & \text{otherwise} \end{cases}$$

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 $ERF(t_0)$ using the Euler predictor method. This value is used to calculate the new buffer $q(t_1)$ (presented in the algorithm as *qpr*), which is then used to produce the new $ERF(t_1)$ and so on. Note that the switch sends u BRM cells every time period τ having the same ERF value. This is because the corresponding $ACR(t_n)$ is assumed constant during the time period [nτ, (n+1)τ).

$$\frac{d}{dt}q(t_n) = \begin{cases} MACR(t_n) - ACR(t_n), & \text{if(e)} \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

Source **VS/VD 1** **VS/VD 2** **Destination**

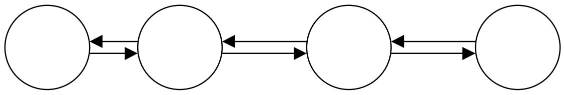


Fig. 4: The 2 hop, 1 VC network model.

Finally, assuming as the worst case of traffic the situation when the buffer overflows, say with rate $BO(t_n)$, we have:

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

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

$$\left.\begin{array}{ll} \text{case } a, & \text{if } qpr=0 \wedge MACR(t_n)<ACR(t_n) \\ \text{case } b, & \text{if } qpr=0 \wedge MACR(t_n)\geq ACR(t_n) \\ \text{case } c, & \text{if } \begin{cases} 0<qpr<q\max, \text{ or} \\ qpr=q\max \wedge MACR(t_n)\leq ACR(t_n) \end{cases} \\ \text{case } d, & \text{if } qpr=q\max \wedge MACR(t_n)>ACR(t_n) \end{array}\right\},$$

$$\text{case } e, \quad \text{if } \left\{\begin{array}{l} 0<q(t_n)<q\max \text{ or,} \\ q(t_n)=0 \wedge MACR(t_n)\geq ACR(t_n) \text{ or,} \\ q(t_n)<q\max \wedge MACR(t_n)\leq ACR(t_n) \end{array}\right\},$$

$$\text{case} f, \quad \text{if } q(t_n)=q\max \wedge MACR(t_n)\geq ACR(t_n)$$

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 τ. The algorithm is as follows:

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

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:

$$ER(t_n)=f(q(t_n))*ACR(t_n)$$

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

$$f(x(t_n))=\begin{cases} z+T0*h, & \text{if } (a) \\ z+T0*h-BORF*\dfrac{h}{\tau}, & \text{if } (b) \\ z+[T0-Tpr]h+BORF*\dfrac{h}{\tau}, & \text{if } (c) \\ z+[T0-Tpr]\tau, & \text{if } (d) \end{cases},$$

where z denotes the load factor (given as the rate of *MACR*/*ACR*), *T*0 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 ← *Max_Alloc_Current* ← *FairShare*

End of Averaging Time Interval:

Total_ABR_Capacity ← *Link_Cap* – *VBR_cap* + *CBR_cap*
ABR_Input_Rate ← Σ(*CCR_of_VC*)
z ← *ABR_Input_Rate* / *Total_ABR_Capacity*
IF ((*e*) is *true*) THEN
    BOR ← *ABR_Input_Rate* – *Total_ABR_Capacity*
ELSE
    BOR ← 0
ENDIF
BORF ← *BOR* / *Total_ABR_Capacity*
IF (case (*a*) is *true*) THEN
    Queue_Funct ← *z* + *T*0 * *h*
ELSEIF (case (*b*) is *true*) THEN
    Queue_Funct ← *z* + (*T*0 - *Tpr*) * *h* + *BORF* * *h* / *τ*
ELSEIF (case (*c*) is *true*) THEN
    Queue_Funct ← *z* + (*T*0 - *Tpr*) * *h* + *BORF* * *h* / *τ*
ELSE
    Queue_Funct ← *z* + (*T*0 - *Tpr*) * *h*
ENDIF
ER ← *Queue_Funct* * *Total_ABR_Capacity*
FairShare ← INT(*Total_ABR_Capacity* / *N*)
MaxAllocPrevious ← *MaxAllocCurrent*
MaxAllocCurrent ← *FairShare*
When a FRM cell is received
CCR_of_VC ← *CCR_in_RM_Cell*

When a BRM cell is received
VCShare ← *CCR_of_VC/z*
IF (z.GT.1+delta) THEN
ER=max(FairShare, VCShare)
ELSE
ER=max(MaxAllocPrevious, VCShare)
ENDIF
MaxAllocCurrent=max(MaxAllocCurrent, ER)
IF (ER.GT.FairShare.AND.CCR_of_VC.LT.FairShare) ER=FairShare
ERin_RM_Cell=min(ERin_RM_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 $ACR(t_n)$ 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:
$(MACR, q) = (ACR(t_n), q_{st}) = (ACR(t_n), -B/A)$,
where the optimal value for the parameter A is given as $A = -(u\ \tau/ACR(t_n))$ and $-B/A=T0*ACR(t_n)$.

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 cells/τ, MCR = 1 cell/τ and $h = \tau = 1$, $q$max = 2 h PCR. To form the simulator, it is assumed that the period τ 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 τ 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 1000τ. 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 2τ 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 $BO(t_n)$ 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 τ. 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 shake of hardware simplicity and low cost implementation, the proposed algorithm share 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 emmitted 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.

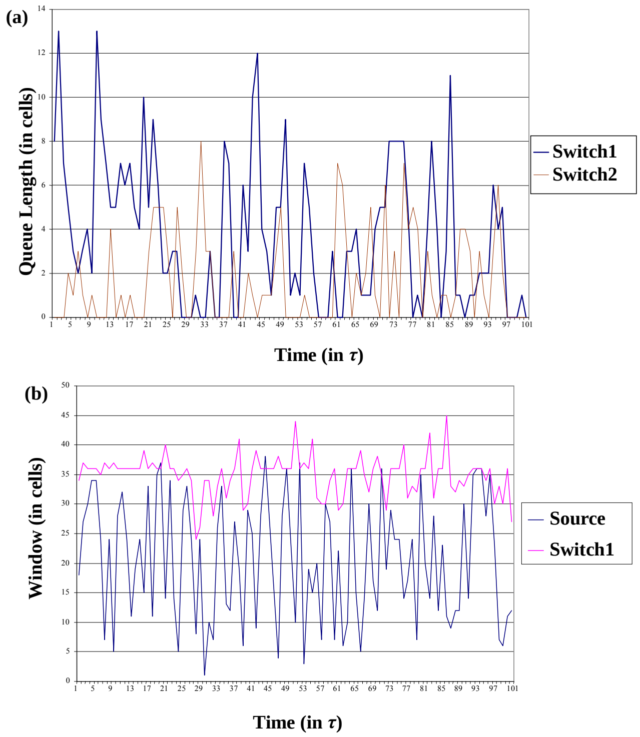


Fig. 6: (a) The queue length of the switches 1 and 2. (b) the proposed from the switch1 and actually implemented windows by the source. (PCR=36 cells/τ, MCR=1 cell/τ and h=τ=1).

REFERENCES

[1] V. Jacobson, Congestion avoidance and control, In *Proceedings of the ACM SIGCOMM,* pp. 314-329, August 1988.

[2] R. Jain, A Timeout Based Congestion Control Scheme for Window Flow Controlled Networks, *IEEE J. Sel. Areas Comm.* Vol.4(7), Oct. 1986, pp. 1162-1167.

[3] Ramakrisna Satyavolu, Ketan Duveti, Kalyanaraman S., Explicit Rate Control of TCP Applications, *ATM Forum/98-0152*, Feb. 1998.

[4] L. Roberts, Enhanced PRCA (Proportional Rate-Control Algorithm), *ATM Forum/94-0735R1*, July 1994.

[5] H. Hsiaw, S. Roberts, R. Jain, B. Makrucki, T. Tofigh, A.W. Barnhart, B. Holden, G. Fedorlow, J.N. Daigle, D. Kline, H. Suzuki, G. Ramamurthy, N. Giroux, R. Kositpaiboon, D. Hughes, M.G. Hluchyj, G. Garg, K.Y. Siu, H.Y. Tzeng and N. Yin, Closed-Loop Rate-Based Traffic Management, *ATM Forum/94-0438R2*, September 1994.

[6] M. Ritter, Network Buffer Requirements of the Rate-Based Control Mechanism for ABR Services, *Proc. IEEE INFOCOM '96*, March 24-25 1996, session 9, pp. 1190-1197.

[7] R. Jain, Congestion Control and Traffic Management in ATM Networks: Recent Advances and a Survey, *Computer Networks and ISDN Systems*, October 1996.

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

[9] A.W. Barnhart, Explicit Rate Performance Evaluation, *ATM Forum/94-0983R1*, http://www.cis.ohio-state.edu/~jain/atmforum.htm.

[10] R. Jain and K. K. Ramakrishnan, Congestion avoidance in computer networks with a connectionless network layer: Concepts, goals, and methodology, *Proc. IEEE Computer Networking Symposium*, 1988.

[11] H. Tzeng, K. Siu, Intelligent congestion control for ABR service in ATM networks, *Computer Communication Review*, 24(5), pp.81-106, October 1995.

[12] Matthew S. Goldman, Variable Bit Rate MPEG-2 over ATM: Definitions and Recommendations, *ATM Forum/96-1433*, October 1996.

[13] Chien Fang and Arthur Lin, On TCP Performance of UBR with EPD and UBR-EPD with a Fair Buffer Allocation Scheme, *ATM Forum/95-1645*, December 1995.

[14] J. Bennett and G. Tom Des Jardins, Comments on the July PRCA Rate Control Baseline, *ATM Forum/94-0682*, July 1994.

[15] R. Jain, S. Kalyanaraman, R. Viswanathan, R. Goyal, A Sample Switch Algorithm, *ATM Forum/95-0178R1*, February 1995, http://www.cis.ohio-state.edu/~jain/atmforum.htm.

[16] ATM FORUM, The ATM Forum Traffic Management Specification Version 4.0. ftp://ftp.atmforum.com/pub/approved-specs/af-tm-0056.000.ps, April 1996.

[17] A. Charny, D.D. Dark, and R. Jain, Congestion control with explicit rate indication, *Proc. of the IEEE International Communications Conference (ICC)*, June 1995.

[18] R. Jain, S. Kalyanaraman and R. Viswanathan, Rate Based Schemes: Mistakes to Avoid, *ATM Forum/94-0882*, September 1994, http://www.cis.ohio-state.edu/~jain/atmforum.htm.

[19] Danny H. K.Tsang, Wales Kin Fai Wong, Sheng Ming Jiang and Eric Y. S. Liu, A Fast Switch Algorithm for ABR Traffic to Achieve Max-Min Fairness, *International Zurich Seminar on Digital Communications, IEEE*, February 19-23, 1996.

[20] Bobby Vandalore, Sonia Fahmy, Raj Jain, Rohit Goyal, Mukul Goyal, A Definition of General Weighted Fairness and its Support in Explicit Rate Switch Algorithm, *Proc. IEEE ICNP*, Oct. 1998, http://www.cis.ohio-state.edu/~jain/papers.htm.

[21] S. Kalyanaraman, Traffic Management for the Available Bit Rate (ABR) Service in Asynchronous Transfer Mode (ATM) networks, *Ph. D. Dissertation, Dept. of Computer and Information Sciences, The Ohio State University*, August 1997.

[22] Bobby Vandalore, Raj Jain, Rohit Goyal, Sonia Fahmy, Design and Analysis of Queue Control Function for Switch Schemes, *ATM Forum/97-1087R1*, February 1998, http://www.cis.ohio-state.edu/~jain/atmforum.htm.

[23] Lampros Kalampoukas, Anujan Varma, and K.K. Ramakrishnan, An Efficient Rate Allocation Algorithm for ATM Networks Providing Max-Min Fairness, *6th IFIP Int. Conf. on High Performance Networking (HPN)*, September 1995.

[24] Raj Jain, Shivkumar Kalyanaraman, Rohit Goyal, Sonia Fahmy, and Fang Lu, ERICA+: Extensions to the ERICA switch algorithm, *ATM Forum/95/1346*, Oct. 1995.

[25] V. I. Arnold (ed.), *Ordinary Differential Equations*, MIT Press edition, 1978 ISBN 0-262-51018-9.

[26] Xipeng Xiao and Lionel M. Ni, (1999), Internet QoS: A Big Picture, *IEEE Network journal*, V. 13 No 2, pp.8-18.

[27] Raiser M. (1979), A Queuing Network Analysis of Computer Communication Networks with Window Flow Control, *IEEE Trans. Comm.*, 27(8) (1979), pp. 1199-1209.

[28] Sh. Kalyanaraman, et. al.(2000), The ERICA Switch Algorithm For ABR Traffic Management in ATM Networks, *IEEE/ACM Trans. Net.,*8(1) (2000), pp. 87-98.

[29] FUJITSU, (2000), Real Time Cell Flow Processing™, *A White Paper*, http://www.nexen.com/new_web/wprealtime.html.

[30] ATM FORUM, The ATM Forum Performance Testing Specification. 11-30-96 5:42 AM, *ATM Forum/BTD-TEST-TM-PERF.00.00(96-0810R3)*, http://www.cis.ohio-state.edu/~jain/atmforum.htm

p	Cells generated	Σ(*ER***h*)		Σ(*ACR***h*) at source		Cells Lost at Source		Σ(*ER* - *ACR*)**h* at source		Cells arrived to destination		buffer 1 non conformed cells		Buffer 1 Average Queue		buffer 2 non conformed cells		Buffer 2 Average Queue		Throughput %	
(1)	(2)	(3)		(4)		(5)=(2)-(4)		(6)=(3)-4)		(7)		(8)		(9)		(10)		(11)		(12)	
		SERICA	ERICA	SERICA	ERICA	SERICA	ERICA	SERICA	ERICA	SERICA	ERICA	SERICA	ERICA	SERICA	ERICA	SERICA	ERICA	SERICA	ERICA	SERICA	ERICA
0.9	32427	32481	29417	31533	29313	894	3114	948	104	31533	29313	0	0	1.88	0.12	0	0	1.64	0.1	97	90
0.8	28830	30101	25292	27962	25104	868	3726	2139	188	27962	25104	0	0	3.59	0.19	0	0	3.26	0.17	97	87
0.7	25084	27932	25292	24417	21183	667	3901	3515	259	24417	21183	0	0	5.39	0.25	0	0	4.45	0.16	97	84
0.6	21458	25584	21442	20848	17342	610	4116	4736	272	20826	17342	14	0	6.69	0.24	8	0	6.4	0.21	97	81
0.5	17786	24762	13978	17551	13690	235	4096	7211	288	17305	13690	143	0	7.67	0.24	103	0	6.59	0.22	97	77
0.4	14235	25784	10444	14204	10140	31	4095	11580	304	13584	10140	351	0	7.22	0.25	269	0	5.87	0.2	95	71
0.3	10736	27824	6979	10736	6776	0	3960	17088	203	9976	6776	408	0	5.42	0.24	352	0	4.77	0.19	93	63
0.2	7118	30166	3686	7118	3536	0	3582	23048	150	6348	3534	467	2	3.87	0.15	303	0	2.93	0.11	89	50
0.1	3614	32525	1340	3614	1282	0	2332	28911	58	2895	1278	437	3	2.04	0.06	282	1	1.44	0.06	80	35
unpre-dict-able	18661	30550	10417	17961	7994	700	10667	12589	2423	17456	7901	359	82	3.58	0.59	146	11	1.9	0.23	94	42

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