Computer Communications 25 (2002) 445–463

computer communications

www.elsevier.com/locate/comcom

A simple ER identification with congestion avoidance (SERICA) algorithm to support some TCP differentiated services over the ABR traffic

G.I. Mousadis, T.A. Tsiligirides*

InfoLab, Department of General Science, Agricultural University of Athens, 75 Iera Odos, Athens 11855, Greece

Received 21 December 2000; revised 13 July 2001; accepted 19 July 2001

Abstract

SERICA is a new ER-based with congestion avoidance switch algorithm of the ERICA class schemes, which has been developed in order to regulate the ABR traffic of an ATM network. The prime issues of such an algorithmic scheme are the minimization of the oscillating behavior of the solution path and the buffer overflow avoidance. In a steady-state, these issues are achieved by presenting a stable focus operational point of O(1) complexity with zero cell loss. SERICA algorithm is tested and evaluated on a network model with multiple configuration alternatives allowing the Virtual Source/Virtual Destination (VS/VD) property. High performance is obtained as a result of the throughput maximization under prescribed levels of queuing delays. According to the negotiated traffic descriptor SERICA may also determine the level of the non-conformed cells the user may expect. Thus, it may be easily applied as a TCP algorithm, in order to facilitate the use of some differentiated services over the ABR traffic. Finally, the existing end-to-end TCP protocol may be enhanced, since the proposed algorithm provides a suitable rate-based control mechanism for the TCP management. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: ER-based switch algorithm; ABR traffic; ATM networks; TCP applications; Differentiated Services; Traffic management; SERICA; Real Traffic Cell Flow Processing

1. Introduction

Nowadays, the computer and telecommunication industries are converging on a single communication technology in almost all high-speed wide area networks. In such an environment, a small bandwidth overload may lead to congestion. For example, high queues at a bottlenecked switch may result in cell loss due to queuing overflows and/or to non-conforming cells. In order to reduce congestion, the effective network management of the Asynchronous Transfer Mode (ATM) is required at the Network Layer. ATM has been chosen as the transport technology for the Broadband Integrated Services Digital Network (B-ISDN). It is the most promising communication technology, which supports any kind of traffic with the requested Quality of Service (QoS) through the following five service classes:

- Constant Bit Rate (CBR) is designed for loss and delay of sensitive data with deterministic and constant cell rate (e.g. voice).
- Real Time Variable Bit Rate (rt-VBR) is designed for the statistically predictable variable bit rate data, which is not able to adjust to variations in network loading (e.g. rt-video).
- Non-Real Time Variable Bit Rate (nrt-VBR) is intended for restricted delay data that is not sensitive to jitter (e.g. interactive multimedia, non-compressed nrt-video). It guarantees only the maximum average delay.
- Available Bit Rate (ABR) is intended for loss of sensitive data and for delay of insensitive (disregarded) data with unpredictable but adjustable cell emission rate to the network load (e.g. computer-based applications, long file transfer).
- Unspecified Bit Rate (UBR) is intended for all kind of traffic except for loss or delay of sensitive data (e.g. e-mail).

Depending on the offered service class, typical ATM connections provide a guaranteed QoS specified mainly in terms of delay and cell loss. In particular, CBR and rt-VBR provide delay and cell loss guarantees and therefore they can be used to transfer delay or loss sensitive multimedia information. In contrast, nrt-VBR provides cell loss guarantees, whereas UBR gives no such guarantees.

However, today's challenge is to support integrated Internet based applications known as Differentiated Services

* Corresponding author.
E-mail address: tsili@aua.gr (T.A. Tsiligirides).

0140-3664/02/$ - see front matter
PII: S0140-3664(01)00376-0

Table 1
Some traffic descriptors for typical network applications

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	$1/10^7$	$1/10^7$	$1/10^8$	$1/10^7$	$1/10^7$

over the ABR traffic. ABR service is used in some delay insensitive or burst data applications. It is also used when it is desirable to utilise 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. This results in 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 avoid buffer overflows.

According to their traffic descriptor set, Internet based applications have been classified into different Types of Services (ToS). Some ToS representatives are the typical network applications shown in Table 1 [5]. However, to save bandwidth the so-called Differentiated Services environment has been 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. When accepted the requested QoS of the corresponding ToS of the application will be 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, our focus is 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 [2], 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 $MCR = PCR = SCR$. Furthermore, 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 $\max CTD$ 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 $\max CTD$ are taking some predetermined values.

According to the ATM Forum, the ABR switch may support the Virtual Source/Virtual Destination (VS/VD) property [2]. 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 [2], 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 can be obtained by simulation using some artificial parameters to ensure a target utilization

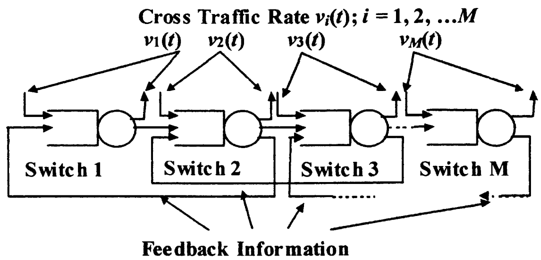

Fig. 1. The hop-by-hop ATM model with M switches in tandem.

with a predetermined bandwidth level [12,13]. ERICA has been suited to provide max–min fairness with MCR criterion. Nevertheless, it is not able to guarantee for a specific *SCR* and thus, it cannot support the extended ABR traffic described above.

Nowadays, the switch technology has been extended in order to support the Real Time Cell Flow Processing™ property (for more, see Ref. [5]). This property allows the switch to re-allocate the available bandwidth according to the traffic descriptor of each connected client and then to divide the buffer in two parts, the dedicated and the shared buffer. The dedicated buffer is a portion of the input queue, reserved for exclusive use by a particular connection. Thus, the Connection Admission Control software instructs all the involved switches to set up a minimum allocated bandwidth and a dedicated buffer per Virtual Channel (VC) in order to serve the *SCR*. The remaining buffer forms the shared switch buffer pool. Connections requiring Best Effort services would primary access to the shared buffer pool only. This property provides an optimum service to the burst traffic, when the use of the dedicated buffers per VC enables the NSP to offer a minimum level of hardware fire-walling per VC. Thus, the switch may serve any kind of the extended ABR traffic specified above, using both dedicated and shared buffers, in any proportion.

The primary focus of this paper is on the ABR service and then on the extended ABR service described above. SERICA algorithm is able to support efficiently a differentiated service environment over the ABR traffic. In a steady state it presents a stable focus operational point of O(1) complexity with zero cell loss. A new queuing control function similar to the one defined for the ERICA+ [12] scheme is defined and a new simulator is developed in order to perform all the appropriate tests. SERICA is defined, evaluated and tested on a network model with multiple configuration alternatives, allowing the VS/VD property. As it appears the proposed scheme provides a high performance in the sense that it maximizes the throughput using some prescribed queuing delay values that when in steady state stabilizes the queuing delay at an appropriate level. Furthermore, it determines a queue threshold at the buffer of the switch per active VC in order to prevent the switch from cell overflows. Finally, it estimates the level of the non-conformed cells that the user may expect according to the negotiated traffic descriptor.

The organization of this paper is as follows. Section 2 analyses the ER model and presents the most important features and parameters required. In the ATM Forum various ER schemes have been proposed so far, but here, interest is given to the congestion avoidance schemes and, particularly, on the ERICA and the ERICA+ switch algorithms. In Section 3, the new approach in the model analysis is presented. The key issue of ERICA+, namely the derivation of a suitable queue control function *f* which controls the queuing delay in steady state conditions, is presented in brief. As a result, the derivation of an alternative queue control function is achieved. The SERICA is presented. The proposed algorithm is numerically solved and tested in Section 4, while all the simulation tests are produced from a new simulator that is also introduced. Finally, the conclusions are presented in Section 5.

2. Presentation of the ER model

In this section the ER scheme is described in some detail. The model is presented in Fig. 1 and consists of M switches and $M-1$ links arranged in a tandem configuration. Each link i is characterized by a transmission capacity $1/ts_i$ (cells/s), a propagation delay τ_i, and a processing capacity $1/tpr_i$ (cell/s), where ts_i is the time the switch i needs to transmit a cell and tpr_i is the time the switch i needs to take a cell from the input and place it on the output queue. Note, that the bandwidth delay product τ_i/ts_i, indicates the number of in flight cells in the transmission link. Since the processing capacity of each node is generally much larger than the total transmission capacity of its incoming link, the only reason causing congestion is the transmission capacity.

It is assumed that source/destination pairs of consecutive switches in the cascade queue contribute to the network traffic. It is also assumed that for such a connection there is a link associated with the path. There are two kinds of traffic. One is the uncontrolled traffic, which is not throttled at the source node since it conforms with the traffic contract. This traffic provides *MCR* guarantee and higher priority classes such as VBR and CBR. The other kind of traffic is the controlled traffic, which is transmitted only when congestion does not exist in the network. This is usually referred to as best-effort traffic and it is calculated from the excess bandwidth capacity after receiving bandwidth for the uncontrolled traffic. Note that, the flow control does not depend on traffic other than that found along the path of the link. This simplifying assumption means that the uncontrolled 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. A link may consist of a number of VCs passing through it. Thus, if a separate queue is maintained for each VC, one way to analyze the dynamic behavior of each queue would be a deterministic fluid approximation. In such cases $q_{i,j}(t)$, denotes the queue length at time t associated with link i and VC j, while $q\max_{i,j}$

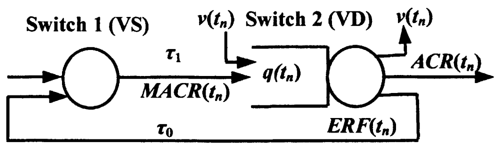

Fig. 2. The single hop single VC model, with background traffic rate $\nu(t_n)$ and FRT given as $\tau = \tau_0 + \tau_1$.

denotes the corresponding queue threshold level. However, this strategy may add some hardware complexity into the network and therefore it should be avoided in practice. An alternative is to use some fairness criteria in the analysis. Thanks to these criteria, the switch buffer is fairly divided among active connections. Clearly, in this case, the index j in the prescribed parameters may be dropped.

2.1. The ER-based model analysis

In this section the ER-based control algorithm, which regulates the source rate is presented using a well-defined parameter set shown in Appendix A. The state of the controlled connection at the corresponding switch (or link) i is fully captured by three state parameters, namely the $MACR_i$ (Mean Arrival Cell Rate), the ACR_i (Allowed Cell Rate) and the q_i (queue length). Thus, the ERF_i (Explicit Rate field) of the corresponding BRM cell produced by the ith switch, may be written as:

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

In the above, $ERF_i(t_n)$, $ACR_i(t_n)$ and $q_i(t_n)$ denote the corresponding parameters at the time $t_n = nT$, where $n = 0, 1, 2,\ldots$; T is the sampling time and $MACR_i(t_n)$ is the mean arrival cell rate of the n-th time period $[(n-1)T, nT)$, recorded at time t_n. Note, that according to the ATM Forum [2]:

$$\begin{aligned} &ERF_i(t_n) = \min\{ERF_i(t_n), ERF_{i+1}(t_n)\} \qquad \text{and} \\ &ACR_i(t_n) = MACR_{i+1}(t_n) \qquad \text{for all } i \end{aligned} \tag{2}$$

Without loss of generality the analysis may be restricted to a single hop, single VC model (see Fig. 2). Therefore, the index i, in the above parameters may be dropped as well. In the single hop single VC model, the Fixed Round Trip Time (FRTT) is the propagation delay $\tau(=\tau_1+\tau_0)$ experienced by the cells before they reach the bottleneck queue (τ_1) of the VD plus the propagation delay (τ_0) experienced by the BRM cells before returning to the VS. As it appears, $ACR(t_n) = \mu - \nu(t_n)$, where μ denotes the fixed service rate of the switch and $\nu(t_n)$ is the uncontrolled traffic rate at time t_n. Therefore, $ACR(t_n)$ is not a linear function of the parameters $MACR(t_n)$ and $q(t_n)$. The $MACR(t_{n+1})$ of the next period $[nT,(n+1)T)$, recorded at time t_{n+1}, is estimated at the switch at time t_n and its value is passed to the VS through the $ERF(t_n)$ of the BRM cells delivered at least once in every FRTT period. Note that when $T = \tau$, the VS is always able to adjust its transmission rate to the required level through the following function [2]:

$$\begin{aligned} &MACR(t_n + \tau_1) \\ &\quad = \min\{MAR, \min\{PCR, \max\{MCR, ERF(t_n - \tau_0)\}\}\} \end{aligned} \tag{3}$$

In the above, PCR and MCR are the values of the negotiated traffic descriptor parameters PCR and MCR respectively, while MAR is the Maximum Allowed link Rate. Thus, in the Worst Case of Traffic (WCT), namely when the VS has always cells to send in the requested rate, it may be seen that:

$$\begin{aligned} &ERF(t_n - \tau_0) = MACR(t_n + \tau_1) \qquad \text{or} \\ &ERF(t_n) = MACR(t_n + \tau) = MACR(t_{n+1}). \end{aligned} \tag{4}$$

2.2. The ERICA and the ERICA+schemes

Although there is a significant amount of information in the literature [6–10, 12–14] we will briefly discuss these algorithms in this section because we think that it is important to show the connection in our work. ERICA and ERICA+, are two of the most interesting congestion avoidance algorithms introduced by the ATM Forum. Note that Congestion Avoidance using Proportional Control (CAPC) [3], or some other similar schemes like OSU or MIT introduced earlier, will not be discussed here since they have less performance and present several problems [9].

The basic ERICA algorithm achieves a desired steady state operational point, by means of providing a fraction of the link utilization, U. Note that in case the ACR is highly varied, the target queuing delay may never be achieved. Thus, the targets are to keep the average utilization high, the average queuing size small, and the queuing delay bounded. Furthermore, the ERICA algorithm is developed so that it provides max min fairness criterion to all connected sources [4]. This criterion attempts to maximize the allocation of the ACR, namely to provide each contending source with a maximum equal share of the available bandwidth.

The basic ERICA is not able to manage the queuing delay efficiently [6]. The main idea of its modification, namely the ERICA+ algorithm [6], is to combine a queuing delay target with a link utilization U through a queuing control function f, which depends on the queue length. This function is used to calculate the *target_ABR_capacity* of ERICA+ as *f*total_ABR_capacity*, assuming that the *total_ABR_ capacity* is given by the difference *Link_capacity*-(*CBR_capacity* + *VBR_capacity*). Thus, only a selected fraction of the available capacity is allocated to the source, while the remaining capacity is used to drain the actual queue of the switch. Hence, in steady state, a controlled queue length, and subsequently a controlled queuing delay, may be achieved. The properties of such functions may be found in Refs. [12,14]. When a constant function is used, the system utilization is restricted to U maximum, leading the basic ERICA in a steady state. The system cannot achieve a queuing delay target neither provide compensation when measurement and feedback are affected by errors.

To calculate the load factor (z) ERICA+ algorithm periodically monitors the available ABR capacity and the number of currently active VCs. The algorithm also keeps track of the maximum allocation given as feedback during the previous averaging interval. To obtain the unique max–min fairness criterion a fair share estimation of a connection is derived by dividing the available capacity by the number of connections. In case the link is not overloaded, the algorithm gives the explicit feedback rate as the maximum of the current cell rate divided by the load factor and the maximum previous allocation. Otherwise, if the link is overloaded, the algorithm gives the explicit feedback rate as the maximum of the current cell rate divided by the load factor and the fair share. ERICA+ algorithm has been introduced and tested for four families of control functions, namely the step, the linear, the hyperbolic and the inverse hyperbolic. Although the inverse hyperbolic function performs better than the other schemes [12], it presents hardware complexity. Thus, to design such functions is still an open problem. Note that in order to operate effectively, both the ERICA and the ERICA+ algorithms use a number of calibrated parameters. Nevertheless, these parameters are not auto-adjustable in case of an unpredictable traffic variation.

In addition, when designing a new function the problem of the Differentiated Services over ABR should be faced through the following issues.

- The ABR service should provide a guaranteed *MCR* to achieve a minimum QoS. Since most of the current ABR switch schemes assume zero *MCR*, a modification is needed in order to support nonzero *MCR*.
- The queuing delay should be minimized in case the application is supplied with the requested QoS. This is a result of the ABR's service initial design to support delay-insensitive data applications.
- Various fairness criteria should be supported. Providing only the max–min fairness criterion may be not enough, particularly when the switch has to support many ToS over ABR in a unified way.
- ABR multicasting should be supported, by means of providing ABR point-to-multipoint, multipoint-to point and multipoint-to-multipoint connections. This issue will not be addressed here but in a future work.

3. The new scheme

As it has been pointed out, our prime interest is to propose and discuss a new ER based switch algorithm by means of the derivation of a new control function for ERICA+. The analysis will be based on the network configuration presented in Fig. 2. To present the algorithm, some additional parameters to those already defined in the Appendix A are needed:

A, B Parameters to be optimized

$BO(t_n)$ The number of cells overflowed from the switch buffer, up to the time instant t_n; $n = 0, 1, 2,\ldots$

H The switch 'averaging' time interval. In general $h \geq \tau$

$K(t)$ The amount of outstanding unacknowledged cells of the controlled traffic at time t

$K(t_n)$ The amount of outstanding unacknowledged cells of the controlled traffic (window size) at time instant t_n; $n = 0, 1, 2,\ldots$

CTD The Cell Transfer Delay, i.e. the round trip transmission delay, which is the sum of the propagation delay τ, plus any other delay (queuing delay, processing delay, etc.), experienced by the first cell of those already emitted from the source, during an arbitrary period, say $[(n-1)\tau, n\tau)$; $n = 0, 1, 2,\ldots$

maxCTD The maximum Cell Transfer Delay of a real time application, negotiated during the connection setup

$q(t_n)$ The queue length of the switch at time instant t_n; $n = 0, 1, 2,\ldots$

$\hat{q}(t_{n+1})$ An estimation of the queue length of the switch at the time instant t_{n+1}; $n = 0, 1, 2,\ldots$

q_{st} The target queue length of the switch buffer ($= T_0\, ACR(t_n)$) in order to achieve stability

$q$max The maximum queue length allocated on the switch buffer in order to serve the VC

T_0 An acceptable threshold for the average queuing delay of the switch buffer

t_n The time instant $t_n = n\tau$; $n = 0, 1, 2,\ldots$ Generally, $t_n = nT$; $n = 0, 1, 2,\ldots$, but here it is assumed that $T = \tau$

u The number of BRM cells sent by the VD to VS every adaptation period T (here $T = \tau$ and thus T presents the FRTT period)

$[(n-1)\tau, n\tau)$ The n-th time period; $n = 0, 1, 2,\ldots$

It is assumed that in every time period, the VS has always cells to send in the requested rate (WCT environment). The analysis is best suited to the leaky bucket fluid flow model in which the switch takes the place of the leaky bucket. Since $K(t)$ is the amount of the outstanding unacknowledged cells at time t (controlled traffic), it may be expressed as a function of both the updated $MACR(t)$ and $q(t)$, namely:

$$K(t) = F_1\big(MACR(t-\tau_0), q(t-\tau_0)\big) \tag{5}$$

Eq. (5) indicates that the number of the outstanding unacknowledged cells at time t is a function of both the queue length at time $(t-\tau_0)$ and the mean arrival cell rate of the switch at time $(t-\tau_0)$. Note that τ_0 presents the propagation delay from the switch to the source and, thus, a phase lag equal to $-\tau_0$ is introduced. Therefore, an event produced on the switch at a time t, say, is acknowledged at the source with delay τ_0, namely at time $(t-\tau_0)$. In the simplest case one may observe that:

$$K(t) = q(t-\tau_0) + \tau_1 MACR(t-\tau_0) + \tau_0 ACR(t-\tau_0)$$

Unacknowledged cells

= Cells in the queue

+ 'in flight'cells from the VS to the VD

+ 'in flight'cells from the VD to the VS

and since $MACR(t-\tau_0) \approx ACR(t-\tau_0)$ (steady state conditions), one may conclude that:

$$K(t) = q(t-\tau_0) + \tau MACR(t-\tau_0). \tag{6}$$

3.1. Discrete analysis

Using the framework of the discrete model developed in Section 2.1, a version of the leaky bucket model presented above may be derived. This model is more realistic and it will be proved very efficient. We begin by denoting with $K(t_n)$ the amount of the outstanding unacknowledged cells of the controlled traffic at time instant t_n. However, as it may be observed, $K(t_n)$ may also present the optimum window size in case the cell loss is negligible. Taking into account Eq. (6) one may derive:

$$K(t_n) = q(t_n-\tau_0) + \tau MACR(t_n-\tau_0). \tag{7}$$

In case the parameter ACR is varied unpredictably we may assume that the window difference between consecutive time instants is given generally as a linear function of the queue length of the switch. This assumption is realistic because:

$$\Delta q(t_n) = q(t_{n+1}) - q(t_n)$$

$$= \begin{cases} \tau(MACR(t_n) - ACR(t_n)), & \text{if } \begin{cases} 0 < q(t_n) < q\max \text{ or}, \\ q(t_n) = 0 \text{ and } MACR(t_n) \geq ACR(t_n) \text{ or}, \\ q(t_n) = q\max \text{ and } MACR(t_n) \leq ACR(t_n) \end{cases} \\ 0, & \text{otherwise} \end{cases}$$

where $q\max$ denotes the maximum queue length allocated on the switch buffer in order to serve the VC. Note that $q\max$ is one of the most critical parameters and must be determined during the connection setup. If the $q\max$ threshold is not specified, high cell losses may occur either due to queue overflow, or due to the rejection of the non-conformed cells (see also Ref. [2]).

However, while observing the system from the source the above equation takes the form:

$$\Delta q(t_n-\tau_0) = q(t_{n+1}-\tau_0) - q(t_n-\tau_0)$$
$$= \tau(MACR(t_n-\tau_0) - ACR(t_n-\tau_0)),$$

and assuming the desirable stability point is obtained in one step, then:

$$MACR(t_{n+1}) = ACR(t_n), \qquad \text{and thus}$$
$$q(t_{n+1}) = T_0 ACR(t_n), \tag{8}$$

where T_0 is an acceptable threshold of the queuing delay.

Thus, using Eq. (8) we obtain:

$$K(t_{n+1}) = q(t_{n+1}-\tau_0) + \tau MACR(t_{n+1}-\tau_0)$$
$$= T_0 ACR(t_n) + \tau ACR(t_n-\tau_0).$$

Therefore, the window difference is given as:

$$\Delta K(t_n) = K(t_{n+1}) - K(t_n) = T_0 ACR(t_n) + \tau ACR(t_n-\tau_0)$$
$$- q(t_n-\tau_0) - \tau MACR(t_n-\tau_0) \Rightarrow \Delta K(t_n)$$
$$= T_0 ACR(t_n) - q(t_n-\tau_0)$$
$$- \tau(MACR(t_n-\tau_0) - ACR(t_n-\tau_0))$$
$$= T_0 ACR(t_n) - q(t_{n+1}-\tau_0)$$

Based on the above derivation, a simple linear approximation to calculate $\Delta K(t_n)$ is proposed:

$$\Delta K(t_n) = Aq(t_{n+1}-\tau_0) + B, \qquad \text{or}$$
$$\Delta K(t_n) = A(q(t_{n+1}-\tau_0) + B/A). \tag{9}$$

Eq. (9) is familiar because it is referred to the simplest fluid flow regulator, namely the floater. As it is well known, the floater regulates efficiently the fluid flow refill in a leaky bucket. Its procedure is based on a particular level of stability of the fluid into the tank. However, the user may shift the level of stability mainly by changing the geometry of the mechanical bracket of the floater. Thus, the stability level of the fluid may be optimized according to the specific conditions applied. The physical interpretation of the assumption (9) above is that, when $q(t_{n+1}) = -B/A$, the window will not change, namely:

$$K(t_{n+1}) = K(t_n) \qquad \text{and thus}$$
$$MACR(t_{n+1}) = MACR(t_n) = ACR(t_n).$$

What is shown above verifies that the point $(q(t_{n+1}), MACR(t_{n+1})) = (-B/A, ACR(t_n))$ is a stability point.

In the following an optimization procedure for the parameters A and B is presented. Differentiating Eq. (7) above, one may obtain:

$$\frac{d}{dt}K(t_n) = \tau\frac{d}{dt}MACR(t_n-\tau_0) + \frac{d}{dt}q(t_n-\tau_0) \tag{10}$$

Nevertheless, using Eq. (9) another approximation of the first derivative of $K(t_n)$ is derived as:

$$\frac{d}{dt}K(t_n) \approx \frac{\Delta K(t_n)}{\Delta t} = [Aq(t_{n+1}-\tau_0) + B]\frac{ACR(t_n-\tau_0)}{u},$$

where u is the number of BRM cells sent by the VD to the

Table 2
Comparing the regulating algorithms SERICA vs ERICA +

ERICA	SERICA
Is an engineering artificial solution.	Is derived analytically.
It concerns the metaphor of the Best Effort Service (UBR) in to the ABR service environment. I.e. it cannot guarantee for negotiated values of *MCR*, or *PCR*, but only for an equally shared proportion of the *ACR* to all the active users. Finally, it assumes that $0 < ER \leq Link\_Rate$, i.e. that every source emits with $PCR = Link\_Rate$, which in general is not true, because the negotiated *PCR* may be any portion of the *Link_Rate*.	It is designed to support both the ABR and the extended-ABR service, i.e. it supports a negotiated *MCR*, *SCR*, *PCR*, max*CTD* (maximum queuing delay) and *CDVT*, assuming that in every switch a virtual buffer is allocated per active VC.
It is suited to support the max–min fairness criterion only. Its last modification incorporates only the max–min with *MCR* fairness criterion.	It may incorporate all the usual fairness criteria.
The switches send at most 1 BRM cell per active VC and per average time interval (time period). If the switch sends more BRM cells, the source may be confused.	When the switch sends only 1 BRM cell, it is highly likely that it will be dropped. However it may send more than 1 BRM cell per active VC and per time period.
The average time interval of the switch must be at least 2*(max_RTT + max_interRM_cell_time), where: max_FRTT is the maximum FRTT of the switch's active VC; and max_interRM_cell_time is the maximum time required to see at least one RM cell of every active VC.	The average time interval of the switch must be at least the max_FRTT of all the active links.
In order to properly calculate all the appropriate parameters, an internal mechanism used by the algorithm will:	In order to properly calculate all the appropriate parameters, an external mechanism used by the algorithm will:
• measure the actually active users of the switch per averaging interval; otherwise it would result to instability and hence to unbounded queues. To do this, the system spends O(k) calculations. Furthermore, the system delays the calculations all credit to the fact that it must wait the end of the averaging interval. Then it must determine the actual number of active VCs k;	• not measure the actually active users of the switch per averaging interval. Alternatively, whenever it is needed, the system will make use of the active traffic contracts, upgrading the corresponding parameter values every time it is changed. This will reduce the total complexity to O(1), although the parameter set of this algorithm is much more than the ERICA's corresponding set. It is assumed that the system will have all the parameter set buffered in a table;
• monitor the incoming cells at the switch. The switch spends O(k) calculations, when it uses the CCR value of the FRM cell.	• perform the *ERF* value of the just arrived BRM cell, by calculating the appropriate value immediately, using the value CCR from the last known FRM cell of the corresponded VC. This calculation spends O(1) calculations.
• evaluate for every active VC the values Max_Alloc_Previous and Max_Alloc_Current. It requires about O(2*k) calculations.	

VS every adaptation period $T(=\tau)$ and Δt is the adaptation period of the source rate given by $u/ACR(t_n - \tau_0)$. Therefore, in case $A \neq 0$, we obtain:

$$\frac{\mathrm{d}}{\mathrm{d}t}MACR(t_n - \tau_0) = \left[q(t_{n+1} - \tau_0) + \frac{B}{A}\right]\frac{A}{u\tau}ACR(t_n - \tau_0) - \frac{\mathrm{d}}{\mathrm{d}t}\frac{q(t_n - \tau_0)}{\tau}. \tag{11}$$

Similarly, assuming that the observer is located in the VD and using Eq. (10), we obtain:

$$\frac{\mathrm{d}}{\mathrm{d}t}MACR(t_n) = \left[q(t_{n+1}) + \frac{B}{A}\right]\frac{A}{u\tau}ACR(t_n) - \frac{\mathrm{d}}{\mathrm{d}t}\frac{q(t_n)}{\tau}. \tag{12}$$

Thus, the non-linear system of ODEs that combines the rate of the VS with the queue length of the switch at time t_n is

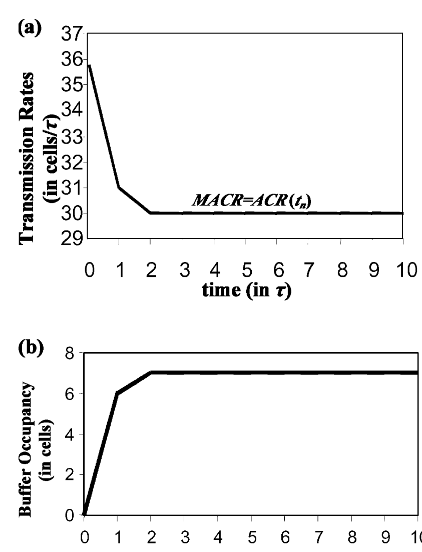

Fig. 3. (a) The Transmission Rate and (b) the queue length over the time, in the single hop single VC model, during steady state and WCT ($\tau = 100$ μs, *Link Rate* = 155.52 Mbps, $ACR(t_n) = 30$ cells/τ, $PCR = 36$ cells/τ, $MCR = 1$ cell/τ). Initial Condition $(MACR(0), q(0)) = (PCR, 0)$.

given by the following equations:

$$\frac{\mathrm{d}}{\mathrm{d}t}MACR(t_n) = \begin{cases} \frac{B}{u\tau}ACR(t_n), & \text{if } q(t_n) = 0 \text{ and } MACR(t_n) < ACR(t_n) \\ \left[q(t_{n+1}) + \frac{B}{A}\right]\frac{A}{u\tau}ACR(t_n) - \frac{d}{dt}\frac{q(t_n)}{\tau}, & \text{if } \begin{cases} 0 < q(t_n) < q\max \text{, or} \\ q(t_n) = 0 \text{ and } MACR(t_n) \geq ACR(t_n), \text{ or} \\ q(t_n) = q\max \text{ and } MACR(t_n) \leq ACR(t_n) \end{cases} \\ \left[q\max + \frac{B}{A}\right]\frac{A}{u\tau}ACR(t_n), & \text{if } q(t_n) = q\max \text{ and } MACR(t_n) \geq ACR(t_n) \end{cases} \tag{13}$$

and

$$\frac{\mathrm{d}}{\mathrm{d}t}q(t_n) = \begin{cases} MACR(t_n) - ACR(t_n), & \text{if } \begin{cases} 0 < q(t_n) < q\max \text{ or,} \\ q(t_n) = 0 \text{ and } MACR(t_n) \geq ACR(t_n) \text{ or,} \\ q(t_n) = q\max \text{ and } MACR(t_n) \leq ACR(t_n) \end{cases} \\ 0, & \text{otherwise} \end{cases}. \tag{14}$$

In the above, $q(t_{n+1})$ is estimated through the first derivative of the queue length at time t_n, using the Euler method. Thus,

$$\hat{q}(t_{n+1}) = \begin{cases} q(t_n) + (MACR(t_n) - ACR(t_n))\tau, & \text{if } \begin{cases} 0 < q(t_n) < q\max \text{ or,} \\ q(t_n) = 0 \text{ and } MACR(t_n) \geq ACR(t_n) \text{ or,} \\ q(t_n) = q\max \text{ and } MACR(t_n) \leq ACR(t_n) \end{cases} \\ q(t_n), & \text{otherwise} \end{cases}$$

Let $A = -u\tau/ACR(t_n)$. Then one may define the target queue length of the switch buffer as:

$q_{st} := -B/A = B^{*}ACR(t_n)/(u\tau) = T_0 ACR(t_n)$.

Note that, T_0 is an acceptable threshold for the average queuing delay, which in our case equals with $B/(u\tau)$ and denotes the target queuing delay of the switch buffer. Regardless the variation of q_{st} with the ACR over the time, we use q_{st} instead of $q_{st}(t_n)$ for simplicity. Thus, the resulting system of ODEs described by Eqs. (13) and (14) may be rewritten as follows:

$$\frac{\mathrm{d}}{\mathrm{d}t}MACR(t_n) = \begin{cases} q_{st}, & \text{if } q(t_n) = 0 \text{ and } MACR(t_n) < ACR(t_n) \\ [q_{st} - \hat{q}(t_{n+1})] - \frac{d}{dt}\frac{q(t_n)}{\tau}, & \text{if } \begin{cases} 0 < q(t_n) < q\max, \text{ or} \\ q(t_n) = 0 \text{ and } MACR(t_n) \geq ACR(t_n), \text{ or} \\ q(t_n) = q\max \text{ and } MACR(t_n) \leq ACR(t_n) \end{cases} \\ [q_{st} - q\max], & \text{if } q(t_n) = q\max \text{ and } MACR(t_n) > ACR(t_n) \end{cases} \tag{15}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}q(t_n) = \begin{cases} MACR(t_n) - ACR(t_n), & \text{if } \begin{cases} 0 < q(t_n) < q\max \text{ or,} \\ q(t_n) = 0 \text{ and } MACR(t_n) \geq ACR(t_n) \text{ or,} \\ q(t_n) = q\max \text{ and } MACR(t_n) \leq ACR(t_n) \end{cases} \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

The above system is solved numerically using the Euler method with step $h = \tau$. Note that the parameter ACR is assumed to remain fixed at least for a period equal to h and that generally $h \geq \tau$.

Under the WCT hypothesis and taking into account that no cell is lost during transmission:

$$MACR(t_n) + h\frac{\mathrm{d}}{\mathrm{d}t}MACR(t_n) = MACR(t_{n+1}) = ERF(t_n) \qquad \text{(see also Eq. (4))}$$

the integration of Eqs. (15) and (16) implies

$$MACR(t_{n+1}) = \begin{cases} MACR(t_n) + q_{st}\tau, & \text{if } q(t_n) = 0 \text{ and } MACR(t_n) < ACR(t_n) \\ [q_{st} - \hat{q}(t_{n+1})]\tau + ACR(t_n), & \text{if } \begin{cases} 0 < q(t_n) < q\max, \text{ or} \\ q(t_n) = 0 \text{ and } MACR(t_n) \geq ACR(t_n), \text{ or} \\ q(t_n) = q\max \text{ and } MACR(t_n) \leq ACR(t_n) \end{cases} \\ MACR(t_n) + [q_{st} - q\max]\tau, & \text{if } q(t_n) = q\max \text{ and } MACR(t_n) > ACR(t_n) \end{cases} \quad (17)$$

$$q(t_{n+1}) = \begin{cases} q(t_n) + (MACR(t_n) - ACR(t_n))\tau, & \text{if } \begin{cases} 0 < q(t_n) < q\max \text{ or,} \\ q(t_n) = 0 \text{ and } MACR(t_n) \geq ACR(t_n) \text{ or,} \\ q(t_n) = q\max \text{ and } MACR(t_n) \leq ACR(t_n) \end{cases} \\ q(t_n), & \text{otherwise} \end{cases} \quad (18)$$

As it appears, starting with initial values of queue length $q(t_0)$, mean arrival cell rate $MACR(t_0)$ and allowed cell rate $ACR(t_0)$, the procedure derives the $MACR(t_1)$ ($= ERF(t_0)$), using the Euler method. This value is used to calculate the new queue length $q(t_1)$ estimated by $\hat{q}(t_1)$, which is then used to produce the new $MACR(t_2)$ ($= ERF(t_1)$), and so on. Note that in case $ACR(t_n)$ remains constant during the n-th period, some computational saving may be achieved, by sending the u BRM cells of this period with the same ERF values.

Clearly, the number of cells exceeding $q$max, say $BO(t_n)$, varies with rate given by:

$$\frac{\mathrm{d}}{\mathrm{d}t}BO(t_n) = \begin{cases} MACR(t_n) - ACR(t_n), & \text{if } q(t_n) = q\max \text{ and } MACR(t_n) \geq ACR(t_n) \\ 0, & \text{otherwise} \end{cases} \quad (19)$$

However, one of the main targets of this work is to derive a suitable function f to control the queuing delay in steady state. This will be achieved in the next section.

3.2. The proposed control function for ERICA+

In this section a new control function suited for ERICA+ is derived and discussed. For this purpose the following additional parameters of those presented so far will be used:

- f — A control function used in the ERICA+ scheme. This function depends on the queue length of the switch. In the single hop single VC model, it is used to obtain the ERF.
- $f(t_n)$ — A discrete form of the control function f.
- $z(t_n)$ — The load factor given by the ratio $MACR(t_n)/ACR(t_n)$.
- Tqmax — The maximum queuing delay given by the difference max$CTD - FRTT$.
- $T\hat{q}(t_{n+1})$ — An estimation of the queuing delay at time instant t_{n+1}; $t_{n+1} = t_n + h$.

It is well known that the ERICA+ switch algorithm uses as a target rate a function of the form [10]:

$$ERF(t_n) = f(q(t_n), q_{st}, MACR(t_n), ACR(t_n)) * ACR(t_n) \quad (20)$$

However the above function is finally reduced to an arbitrary,

Table 3
The single hop, single VC model, in WCT ($\tau = 100$ µs, *Link Rate* = 155.52 Mbps, $PCR = 36$ cells/τ, $MCR = 1$ cell/τ, $h = \tau = 1$ and $q\max = 2*PCR*h$)

p	$PCR*p$	ACR in average	Average queue in buffer	Queuing delay	Slots for the safe region
(1)	(2)	(3)	(4)	(5)	(6)
0.1	3.60000	3.62238	33.342	9.204	1
0.2	7.20000	7.13387	29.830	4.181	1
0.3	10.80000	10.74226	26.222	2.441	1
0.4	14.40000	14.25674	22.707	1.593	1
0.5	18.00000	17.81419	19.150	1.075	1
0.6	21.60000	21.41059	15.553	0.726	2
0.7	25.20000	25.05095	11.913	0.476	2
0.8	28.80000	28.76623	8.198	0.285	3
0.9	32.40000	32.43856	4.525	0.140	7
0.95	34.20000	34.23077	2.731	0.080	13
0.98	35.28000	35.29770	1.661	0.047	25

artificial simple control function f that takes into account only the queue length and the target queue length, or equivalently, the queuing delay and the target queuing delay, respectively [7,12,13].

The analysis developed in Section 3.1 may be extended to numerically derive a suitable control function $f(t_n)$, based, generally, on the relation:

$$ERF(t_n) = f(t_n)^* ACR(t_n) \tag{21}$$

In order to determine the suitable control function $f(t_n)$, Eqs. (15) and (16) will be integrated, using a time step h and taking into account that

$$MACR(t_n) + h\frac{\mathrm{d}}{\mathrm{d}t} = MACR(t_n) = MACR(t_{n+1}) = ERF(t_n).$$

Thus

$$ERF(t_n) = \begin{cases} MACR(t_n) + q_{st}h, & \text{if } q(t_n) = 0 \text{ and } MACR(t_n) < ACR(t_n) \\ MACR(t_n) + [q_{st} - \hat{q}(t_{n+1})]h - \dfrac{h}{\tau}\dfrac{d}{dt}q(t_n), & \text{if } \begin{cases} 0 < q(t_n) < q\max, \text{ or} \\ q(t_n) = 0 \text{ and } MACR(t_n) \geq ACR(t_n), \text{ or} \\ q(t_n) = q\max \text{ and } MACR(t_n) \leq ACR(t_n) \end{cases} \\ MACR(t_n) + [q_{st} - q\max]h, & \text{if } q(t_n) = q\max \text{ and } MACR(t_n) > ACR(t_n) \end{cases}$$

$$q(t_{n+1}) = \begin{cases} q(t_n) + (MACR(t_n) - ACR(t_n))h, & \text{if } \begin{cases} 0 < q(t_n) < q\max, \text{ or} \\ q(t_n) = 0 \text{ and } MACR(t_n) \geq ACR(t_n), \text{ or} \\ q(t_n) = q\max \text{ and } MACR(t_n) \leq ACR(t_n) \end{cases} \\ q(t_n), & \text{otherwise} \end{cases}.$$

Taking into account (21) one may derive the analytical form of the control function $f(t_n)$, as:

$$f(t_n) = \begin{cases} \dfrac{MACR(t_n)}{ACR(t_n)} + \dfrac{q_{st}}{ACR(t_n)}h, & \text{if } q(t_n) = 0 \text{ and } MACR(t_n) < ACR(t_n) \\ \dfrac{MACR(t_n)}{ACR(t_n)} + \left[\dfrac{q_{st}}{ACR(t_n)} - \dfrac{\hat{q}(t_{n+1})}{ACR(t_n)}\right]h - \dfrac{h}{\tau}\dfrac{\frac{d}{dt}q(t_n)}{ACR(t_n)}, & \text{if } \begin{cases} 0 < q(t_n) < q\max, \text{ or} \\ q(t_n) = 0 \text{ and } MACR(t_n) \geq ACR(t_n), \text{ or} \\ q(t_n) = q\max \text{ and } MACR(t_n) \leq ACR(t_n) \end{cases} \\ \dfrac{MACR(t_n)}{ACR(t_n)} + \left[\dfrac{q_{st}}{ACR(t_n)} - \dfrac{q\max}{ACR(t_n)}\right]h, & \text{if } q(t_n) = q\max \text{ and } MACR(t_n) > ACR(t_n) \end{cases}$$

or,

$$f(t_n) = \begin{cases} z(t_n) + T_0 h, & \text{if } q(t_n) = 0 \text{ and } z(t_n) < 1 \\ z(t_n) + [T_0 - T\hat{q}(t_{n+1})]h - \dfrac{h}{\tau}(z(t_n) - 1), & \text{if } \begin{cases} 0 < q(t_n) < q\max, \text{ or} \\ q(t_n) = 0 \text{ and } z(t_n) \geq 1, \text{ or} \\ q(t_n) = q\max \text{ and } z(t_n) \leq 1 \end{cases}, \\ z(t_n) + [T_0 - Tq\max(t_n)]h, & \text{if } q(t_n) = q\max \text{ and } z(t_n) > 1 \end{cases}$$

In the above, the load factor is $z(t_n) = MACR(t_n)/ACR(t_n)$, the estimation for the queuing delay of the next time instant t_{n+1} is $T\hat{q}(t_{n+1}) = \hat{q}(t_{n+1})/ACR(t_n)$; $t_{n+1} = t_n + h$ and

$$\frac{\mathrm{d}}{\mathrm{d}t}q(t_n) = MACR(t_n) - ACR(t_n).$$

In addition the maximum queuing delay is $Tq\max = q\max/ACR(t_n)$. In the extended ABR this queuing delay may be approximated by the difference $\max CTD$ − FRTT, while in the original ABR by the parameter $CDVT$.

Finally,

$$f(t_n) = \begin{cases} z(t_n) + T_0 h, & \text{if } q(t_n) = 0 \text{ and } z(t_n) < 1 \\ -\dfrac{h-\tau}{\tau} z(t_n) + [T_0 - T\hat{q}(t_{n+1})]h + \dfrac{h}{\tau}, & \text{if } \begin{cases} 0 < q(t_n) < q\text{max}, \text{ or} \\ q(t_n) = 0 \text{ and } z(t_n) \geq 1, \text{ or} \\ q(t_n) = q\text{max and } z(t_n) \leq 1 \end{cases} \\ z(t_n) + [T_0 - Tq\text{max}(t_n)]h, & \text{if } q(t_n) = q\text{max and } z(t_n) > 1 \end{cases}$$

Note that when $h = \tau$, the control function f takes the following final formula:

$$f(t_n) = \begin{cases} z(t_n) + T_0 \tau, & \text{if } q(t_n) = 0 \text{ and } z(t_n) < 1 \\ [T_0 - T\hat{q}(t_{n+1})]\tau + 1, & \text{if } \begin{cases} 0 < q(t_n) < q\text{max}, \text{ or} \\ q(t_n) = 0 \text{ and } z(t_n) \geq 1, \text{ or} \\ q(t_n) = q\text{max and } z(t_n) \leq 1 \end{cases} \\ z(t_n) + [T_0 - Tq\text{max}]\tau, & \text{if } q(t_n) = q\text{max and } z(t_n) > 1 \end{cases}$$

In the sequel the proposed version of ERICA+ algorithm will be presented.

3.3. The algorithm

In this section the SERICA algorithm is developed and discussed. The pseudocode of the algorithm along with the notation of the parameters used is presented in the Appendix B.

The new algorithm operates in a similar fashion with ERICA at a link level. The switch periodically monitors the available ABR capacity, presented with the parameter *total_ABR_capacity*, the number of all active VCs, k, as well as the aggregate ABR input rate of the link, denoted as *link_ABR_input_rate*. Thus, the switch is able to estimate the queue length $\hat{q}(t_{n+1})$ of the next time instant and to determine the Buffer Overflow Rate, say *BOR*, given as:

$$BOR = link\_ABR\_input\_rate - total\_ABR\_capacity.$$

Only the parameter *link_ABR_input_rate* is provided in the switch through the Current Cell Rate (CCR) field of the FRM cells, while the rest needs to be specified in advance. In addition, all the parameters defined above are used to calculate the feedback provided by the BRM cells. For this purpose the 'averaging' time interval h is used. In every period h the switch sends at least one new BRM cell to each source. The computational complexity of the algorithm is caused mainly because of the different FRTTs $\tau^{(j)} = \tau_0^{(j)} + \tau_1^{(j)}$ of all the active VCs of the link $j = 1, 2,\ldots,k$. The ideal averaging time for each VC would be the corresponding FRTT, leading to per VC queue monitoring and per VC calculation of the ERF. However, this approach increases the computational complexity of the switch at least k times of the complexity of the single VC model.

In order to calculate the actual arrived cells over a period h the use of a counter mechanism per VC is suggested. This mechanism would help in the derivation of $MACR_j(t_n)$ of the j-th VC of the link, which subsequently is used in the derivation of the *link_ABR_input_rate*. The approach is more accurate, but results in an increase of the computational complexity of the algorithm in the switch, at least, to the order of O(k**minislots*), where the parameter *minislots* indicates the number of mini-slots the period h is divided [11]. Also, it adds some delay in the final derivation of the ERF, since the computation takes place after the end of the period h, during which the parameter $MACR_j(t_n)$ is determined. Thus, an unpredictable delay in the arrival time of the BRM cells in the switches is introduced. To eliminate this delay and to reduce the computational complexity of O(k**minislots*) two different approaches in counting the number of actually arrived cells have been proposed by Refs. [12] and [11], respectively. In particular, the first approach which achieves complexity, at most, of order O(k) counts the actual arrived cells per VC through the parameter CCR_j of the corresponding j-th VC of the link and indicates their total in the appropriate FRM cell. The second approach which achieves complexity, at most, of order O(1) imposes an external mechanism in order to calculate the ABR input rate using the same parameter CCR_j, immediately after the arrival of the appropriate FRM cell in the switch. SERICA algorithm achieves O(1) complexity using the second approach and hence, the switch copes only with the link and shares the available bandwidth among all active VCs by using some fairness criteria.

To define and evaluate the proposed scheme, one should take into account the real activity of the ABR traffic, which is based either on the worst case assumptions of the CBR/VBR traffic, or on the exploitation of an adaptive prediction scheme. In such schemes, buffer cell overflows should be avoided. Emphasis should be given on the ability to increase the ABR throughput when the CBR/VBR traffic, is reduced. To minimize the oscillating behavior of the solution path, the algorithm determines the equivalence factor $EF_j = h/\tau^{(j)}$

with $h \geq \max\{\tau^{(j)}\}$ for all active VCs $j = 1, 2,\ldots,k$. In the above, the FRTT period of each active session $\tau^{(j)}$ is specified by the Connection Admission Control (CAC) procedure during the establishment of the j-th VC. As it will be seen in Section 4.1, the switch is able to derive the target queuing delays $T_{0,j}$ of all the active connections of the switch, through the equation $T_{0,j} = h^*(PCR_j + MCR_j - ACR_j(t_n))/ACR_j(t_n)$. As one may observe, the maximum queuing delay threshold is usually greater than the target queuing delay, while an approximation of an optimal value for the maximum queue length $q\max_j$, is given by:

$$q\max_j = PCR_j * (\max CTD_j - \tau^{(j)}) = \left(PCR_j - MCR_j\right)*CDVT_j + PCR_j T_{0,j} \quad (22)$$

where, the parameter $CDVT_j$ presents the negotiated value of $CDVT$ of the j-th VC.

However, all the above parameters need to be re-evaluated during a new connection establishment. Note that, the approach requires the original parameters to be re-defined on the corresponding mini-slots of the j-th VC, determined through the EF_j. Thus, the new set of parameters for the j-th VC, is given by:

$$CCReq_j = EF_j*CCR_j$$

$$MCReq_j = EF_j*MCR_j$$

$$PCReq_j = EF_j*PCR_j$$

$$SUM\_MCR = \Sigma EF_j*MCR_j$$

$$SUM\_PCR = \Sigma EF_j*PCR_j$$

$$q\max eq = \Sigma EF_j*q\max_j$$

$$T_0 = \Sigma EF_j*T_0, j/k$$

Using the above set of parameters, the system of Eqs. (17) and (18) may be solved numerically to calculate the desired ERF of the BRM cell. Finally, as one may see in Appendix B, the algorithm may be adapted in the usual fairness criteria.

The algorithm presented above is self-controlled in the sense that during overload it provides a way to adjust the input rate into the required level and therefore to prevent the switches from buffer overflow. Depending on both the queue length and the target queuing delay, the scheme is growing up or drawing down the resulting value of the corresponding ERF of the BRM cell, in O(1) calculations. The main differences between ERICA+ and SERICA are shown in Table 2.

4. Results

In this section, the SERICA scheme, presented above, is tested and the analytical and simulation results obtained are compared and discussed. In addition, both the original ERICA and the SERICA are compared in order to test the efficiency of the proposed scheme.

4.1. Numerical results

Numerical results are presented by means of the Fig. 3(a) and (b), where the $MACR$ and the queue length of the buffer of the switches vs the time are plotted. As it may be pointed out the linear branch of the system of ODEs (13) and (14) has a stable focus point (the relevant theory may be seen in Ref. [10], p. 137), provided the $ACR(t_n)$ remains constant for some time period and the buffer overflow rate given in Eq. (19) is zero (steady-state, WCT conditions). Note, that the same conclusion may be drawn from Fig. 3(a) and (b) verifying the system has a stable focus point at $(MACR,q) = (ACR(t_n),q_{st}) = (ACR(t_n), -B/A)$, where $A = -u\tau/ACR(t_n)$ and $B = u\tau T_0$ specify their optimum values. Thus, at time t_n the q_{st} is given as $T_0*ACR(t_n)$. It is noted here that T_0 must be declared from the switch and that it is not specified during the connection setup. Here, a simple function is used to calculate the target queue length, or the target queuing delay as it follows:

$$q_{st} = (PCR + MCR - ACR(t_n))h, \text{ or}$$
$$T_0 = h\left(PCR_j + MCR_j - ACR_j(t_n)\right)/ACR_j(t_n). \quad (23)$$

We conclude in Eq. (23) assuming that the optimum window size at the source for every different $ACR(t_n)$ remains constant. Then, undertaking that the switch should be always able to serve the j-th VC with a rate at least equal to MCR_j, if the $ACR_j(t_n)$ remains constant at the PCR_j value for some time steps, then the queue length at the buffer should become at least $q_{st} = MCR*h$.

As a result, in a steady-state WCT environment, in which the VS has always cells to send in the requested rate and the $ACR(t_n)$ is constant, the system is shown to be stable focused and self-controlled. In particular, the linear branch of the proposed scheme operates autonomously like the system of a pendulum with friction. However, in order to observe the system behavior in the simple WCT environment, in which the VS has always cells to send in the requested rate and the $ACR(t_n)$ is variable, as well as, in any non-WCT environment, in which the VS has not always cells to send in the requested rate, there is a need for simulation.

4.2. Simulation results

4.2.1. Forming the simulator

The new algorithm is tested on two simple configurations under both environments described below. In the first case, the network configuration consists of a single hop, single

VC model (Fig. 2), with descriptor parameters of the ABR traffic defined as follows: $PCR = 36$ cells/τ, $MCR = 1$ cell/τ and $h = \tau = 1$, $q\max = 2hPCR$. The simulation environment assumes WCT conditions in which the $MACR(t_n)$ is varied during the simulation process throughout the following step equation:

and simulation results obtained. Furthermore, for each p, column (4) shows the average number of cells waiting in the buffer of the switch during the whole simulation period. These values have been found to be almost the same with the corresponding values of q_{st} obtained from the theory, by means of using Eq. (23).

$$MACR(t_{n+1}) = \begin{cases} MACR(t_n) + q_{st}h, & \text{if } \hat{q}(t_{n+1}) = 0 \\ [q_{st} - \hat{q}(t_{n+1})]h + ACR(t_n), & \text{if } 0 < \hat{q}(t_{n+1}) < q\max, \\ MACR(t_n) + [q_{st} - q\max]h, & \text{if } \hat{q}(t_{n+1}) = q\max \text{ and } MACR(t_n) > ACR(t_n) \end{cases} \tag{24}$$

This is a simplifying expression of Eq. (15) in which $q(t_{n+1})$ is estimated by:

$$\hat{q}(t_{n+1}) = \begin{cases} q(t_n) + (MACR(t_n) - ACR(t_n))h, & \text{if } \begin{cases} 0 < q(t_n) < q\max \text{ or,} \\ q(t_n) = 0 \text{ and } MACR(t_n) \geq ACR(t_n) \text{ or,} \\ q(t_n) = q\max \text{ and } MACR(t_n) \leq ACR(t_n) \end{cases} \\ q(t_n), & \text{otherwise} \end{cases} \tag{25}$$

where $q(t_n)$ is known in advance and recalling that q_{st} is given from Eq. (23). Finally, one may deduce that the appropriate value of $Tq\max$ that minimizes the possibility for an arrived cell to find a full buffer, is greater than $2h$. In this case, one may assume that $CDVT \geq Tq\max$. Thus, an initial approximation of $CDVT$ may be given, using Eq. (22):

$$CDVT \geq \lceil h^*PCR/(PCR - MCR) \rceil \tag{26}$$

where $\lceil x \rceil$ is the upper integer part of x. For this reason, in the simulations we used $q\max = 2h^*PCR$, and $CDVT = 2h$. 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.

4.2.2. Testing performance

4.2.2.1. Using a WCT environment Extended simulation results are produced over a period of 1000 τ and presented in Table 3 to compare with their corresponding theoretical values. Thus, for each value of p, appeared in column (1), the average number of cells that have been served by the switch during the whole simulation period, are produced and presented in column (3). These values are referred as average ACR and are compared with the corresponding theoretical values of ACR expressed as PCR^*p and presented in column (2). As one may observe, the comparison shows an excellent agreement between the theoretical

As one may observe, for all the tested values of p, the link utilization achieves 100% of its target value. Moreover, for each value of p, column (5) presents the queuing delay, which helps in the a-priori determination of the parameter T_0, which may not be included in the traffic descriptor parameters. However, as one may observe T_0 may be derived theoretically, using Eq. (23) which is also verified by this simulation, taking into account that $ACR + q_{st}/h = PCR + MCR$. Finally, column (6) shows the number of slots required by the simulation algorithm to leave the transient phase and enter the safe region.

4.2.2.2. Using a non-WCT environment In the second case, the network configuration consists of 2-hops in tandem, traversed by a single VC (Fig. 4), using the same descriptor parameters of the ABR traffic, as defined in the first case. The simulation environment assumes non-WCT conditions in which the VS has not always cells to send in the requested rate and the ACR of a VS may vary during the simulation process. The simulator is formed in a similar way as with the previous case, using the same cell generation process in the source. In addition, any cell waiting in the queue for more than 2τ is removed from the queue. However, as one may observe the ACR of the source equals, on average, with p^*PCR, while the $MACR$ of a VD is given through Eqs. (2) and (3), when the VS is able to respond in the requested rate.

The main question is how SERICA performs vs ERICA+, using an unpredictably variable traffic environment. Note that the ERICA+ cannot be easily compared with other schemes because it uses many artificial parameters. Thus, the control function of the SERICA is compared vs the control function $f = 1$, of the ERICA+ algorithm. The above function, which was first introduced in the basic ERICA scheme [8], is the simplest because it is

Table 4
The 2 hop, single VC model, in non-WCT ($\tau = 100$ µs, *Link Rate* = 155.52 Mbps, $PCR = 36$ cells/τ, $MCR = 1$ cell/τ, $h = \tau = 1$ and $q\max = 2*PCR*h$, $CDVT = 2*h$

p	Cells generated	$\Sigma(ER*h)$		$\Sigma(ACR*h)$ in the source		Cells lost in the source		$\Sigma(ER - ACR)*h$ at source		Cells arrived in the 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	32,427	32,481	29,417	31,533	29,313	894	3114	948	104	31,533	29,313	0	0	1.88	0.12	0	0	1.64	0.1	97	90
0.8	28,830	30,101	25,292	27,962	25,104	868	3726	2139	188	27,962	25,104	0	0	3.59	0.19	0	0	3.26	0.17	97	87
0.7	25,084	27,932	25,292	24,417	21,183	667	3901	3515	259	24,417	21,183	0	0	5.39	0.25	0	0	4.45	0.16	97	84
0.6	21,458	25,584	21,442	20,848	17,342	610	4116	4736	272	20,826	17,342	14	0	6.69	0.24	8	0	6.4	0.21	97	81
0.5	17,786	24,762	13,978	17,551	13,690	235	4096	7211	288	17,305	13,690	143	0	7.67	0.24	103	0	6.59	0.22	97	77
0.4	14,235	25,784	10,444	14,204	10,140	31	4095	11,580	304	13,584	10,140	351	0	7.22	0.25	269	0	5.87	0.2	95	71
0.3	10,736	27,824	6979	10,736	6776	0	3960	17,088	203	9976	6776	408	0	5.42	0.24	352	0	4.77	0.19	93	63
0.2	7118	30,166	3686	7118	3536	0	3582	23,048	150	6348	3534	467	2	3.87	0.15	303	0	2.93	0.11	89	50
0.1	3614	32,525	1340	3614	1282	0	2332	28,911	58	2895	1278	437	3	2.04	0.06	282	1	1.44	0.06	80	35
Arbitrary	18,661	30,550	10,417	17,961	7994	700	10,667	12,589	2423	17,456	7901	359	82	3.58	0.59	146	11	1.9	0.23	94	42

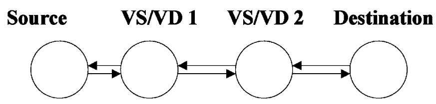

Fig. 4. The 2 hop, 1 VC network model.

constant and because it uses no simulation parameters in order to be determined. Note that the single VC network shown in Fig. 4 has been chosen for the simulation in a non steady-state, non-WCT environment. The produced simulation results over a period of 1000 τ are summarized in Table 4.

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 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, which 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 Eq. (19), is zero for both algorithms. Therefore it is excluded from the table.

The simulation shows some interesting results concerning 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 must also be taken into consideration 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 cause unexpected behavior in the system performance. Thus, the new algorithm performs very well around its operational point during the whole session at 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 neither because of the increase in the average queue of the buffers, nor because of the number of non-conformed cells, but mainly, because of the number of cells rejected in the source.

4.2.2.3. Examining CDVT Simulations made using the new algorithm with $CDVT = 1$ and various values of p $(0 < p < 1)$ verify that the appropriate theoretical value for the $q$max, is $2hPCR$ [see also Eq. (26)] and showed a zero overflow rate. Thus, it is now of interest to evaluate the number of non-conformed cells. This number may be evaluated by a simulation, based on the negotiated $CDVT$ and p. The network used is the one presented in Fig. 4, assuming a non-WCT environment and the same set of

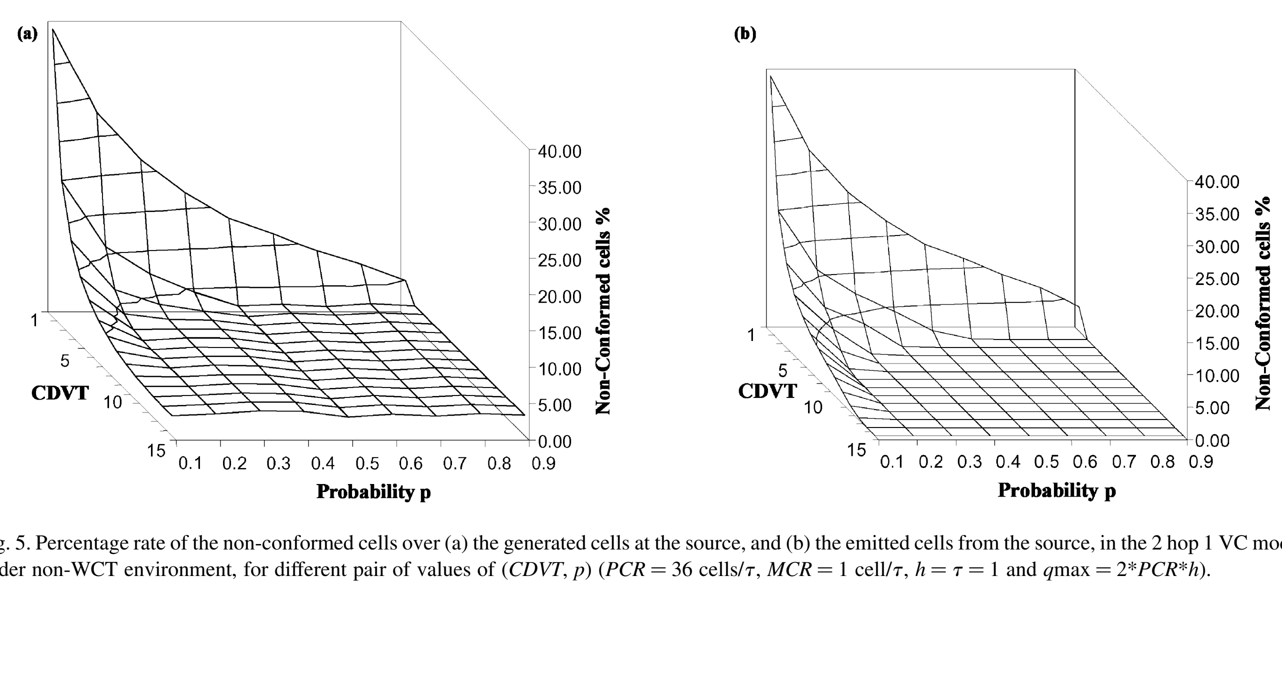

Fig. 5. Percentage rate of the non-conformed cells over (a) the generated cells at the source, and (b) the emitted cells from the source, in the 2 hop 1 VC model, under non-WCT environment, for different pair of values of ($CDVT$, p) ($PCR = 36$ cells/τ, $MCR = 1$ cell/τ, $h = \tau = 1$ and $q\max = 2{*}PCR{*}h$).

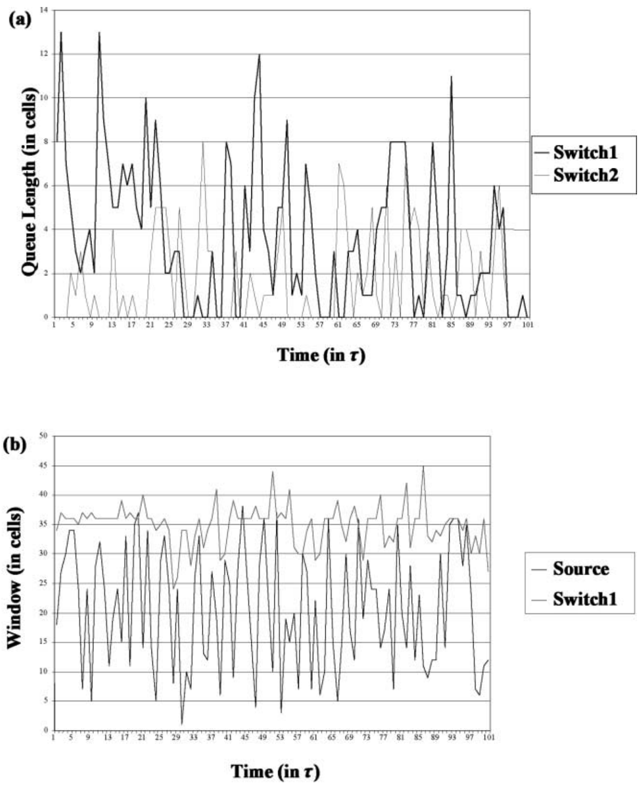

Fig. 6. (a) The queue length of the switches 1 and 2 and (b) the proposed from the switch1 and actually implemented windows by the source ($PCR = 36$ cells/τ, $MCR = 1$ cell/τ and $h = \tau = 1$).

simulation parameters which has been used so far. The only difference stands on the parameter *CVDT*. In particular, for each value of *p*, various values of *CDVT*, have been simulated in the range of 1–15. Thus, this parameter remains fixed during each simulation run of 1000 τ; however it may be varied from run to run. The simulation results, which are showing the number of non-conformed cells in a percentage scale, are presented through the two 3-D graphs, in Fig. 5(a) and (b). Note, that Fig. 5(a) presents the number of non-conformed cells as a percentage of those generated at the source, whereas Fig. 5(b) presents the number of non-conformed cells as a percentage of those emitted from the source. A simple comparison between these two figures shows that there is no any real difference on the way one may count the number of the non-conformed cells. Finally, as one may observe, the analytical evaluation of the number of non-conformed cells based on different pair values of (*CDVT*, *p*) is not an easy task. However, this simulation offers a set of parameter values in order to approximate the number of non-conformed cells for a safe operation. This factor is very important in the cases where the network has to offer guarantees for a specific service scheme, such as the Differentiated Services over ABR.

4.2.2.4. Using a Differentiated Services environment To provide Differentiated Services over an ABR environment, the NSP has to extent the traffic contract of the ABR service (traffic descriptor), with two additional parameters. These parameters are negotiated along with the Sustainable Bit Rate (SBR), which in our case is the probability *p*. These parameters are the maximum queuing delay and the rejected cells due to overflow in the buffers as well as the non-conformed cells, namely, those exceeding their life time, which is taken equally with *CDVT*.

However, the new algorithm may be used to approximate the optimum source window (namely, the number of outstanding cells to be emitted from the source), in a TCP window based scheme in order to support a scheme of Differentiated Services over ABR. The above simulator is used to show the operation of the window mechanism in the configuration of Fig. 4. The results produced are depicted in Fig. 6(a) and (b). Fig. 6(a) shows the queue length (in cells) of each one of the switches 1 and 2 over the time τ. Fig. 6(b) 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.

5. Conclusions

In this work a new ER-based switch algorithm with Congestion Avoidance, by means of the SERICA scheme, used to regulate the flow control of the ABR traffic in an ATM network, has been derived and discussed. The ATM configuration consists of M switches in tandem, with multiple VCs, allowing the VS/VD property. The switches monitor both, the load and the buffer occupancy by determining the available capacity and the number of the active VCs, respectively. Based on this information the algorithm advises the sources about the rates they should transmit through the ERF of the BRM cell received every round trip time.

For the sake of simplicity the analysis has been restricted to a single hop network. The model has been analyzed through a non-linear system of ODEs by assuming that the parameters *ACR*, *MACR* and VD buffer occupancy have been modeled as fluids. A simple numerical method, by means of the Euler predictor corrector technique has been used to allow the calculation of the desired *MACR* of the subsequent time interval.

According to the traffic descriptor, the new algorithm has been developed in order to present a stable focus operational point. The target operating point has been achieved, utilizing 100% of the available bandwidth with a fixed non-zero queuing delay. To guarantee zero cell loss a certain threshold of queue length has been found and applied in all the switches of the network. This threshold is closely related with both, the negotiated maximum queuing delay (or *CDVT*) and the negotiated *PCR*. The queuing control function, which is based on the load factor $z(t_n)$, the target queuing delay (T_0), the max queuing delay and the current queuing delay has been analytically derived using the

Euler procedure. The above modification made over the original version of the well known ERICA+ scheme has shown an interesting effect in the overall performance of the SERICA algorithm. Further tests have shown that the new scheme achieves efficiency and exhibits a very fast transient response towards the desired operating point with computational complexity of O(1). Finally, to minimize the oscillating behavior of the solution path, the algorithm has determined an equivalence factor of all active VCs, which has been proposed in the adaptation of the usual fairness criteria.

The SERICA algorithm has been tested using two network configurations, namely the single hop–single VC network and the two hops–single VC network. A simulator has been developed and various simulation environments of the ABR traffic, such as, the WCT and the non-WCT, for different values of *SCR* and *CDVT* have been tested. As it has been shown the new algorithm maximizes the throughput, in a sense that it does not drop more than 20% of the initially generated cells at the source, irrespectively of the probability p (or *SCR*/*PCR*). In addition, it has been used to determine the level of the non-conformed cells that the user may expect according to the traffic descriptor. However, further research is needed to establish the prescribed level of the non-conformed cells analytically.

The work has a profound interest in the TCP, when it is combined with switching mechanisms at the network layer to perform necessary traffic management functions by the NSPs. It is particularly important in cases where long file transfers and www-servers and clients with persistent data type traffic use the TCP as their transport layer. As it appears, the proposed SERICA algorithm enhances the existing end-to-end TCP protocol and provides a rate-based control mechanism suitable for TCP management.

To conclude, the proposed algorithm may be used for both cases; namely, either as a TCP algorithm or as an ER-based switch algorithm with congestion avoidance to support TCP differentiated services over the ABR traffic. In addition, it may be used as a suitable decision support tool to help the network administrator to establish and subsequently support a new session.

Appendix A

The parameters of the ER-based control algorithm.

Parameters	Description
$ACR_i(t_n)$	The Allowed Cell Rate of the i-th switch at time t_n; $n = 0, 1, 2,\ldots$
$ERF_i(t_n)$	The Explicit Rate Field of the BRM cell of the i-th switch at time t_n; $n = 0, 1, 2,\ldots$
$MACR_i(t_n)$	The Mean Arrival Cell Rate of the i-th switch of the period $[(n-1)T, nT)$, recorded at time t_n; $n = 0, 1, 2,\ldots$
MAR	The Maximum Allowed link Rate
MCR	The negotiated Minimum Cell Rate of the session
PCR	The negotiated Peak Cell Rate of the session
q_i	The queue length of the i-th switch.
$q_i(t_n)$	The value of the queue length of the i-th switch at time t_n; $n = 0, 1, 2,\ldots$
T	The sampling time period
t_n	The time $t_n = nT$; $n = 0, 1, 2,\ldots$
μ	The fixed service rate of the switch
$\nu(t_n)$	The uncontrolled traffic rate at time t_n; $n = 0, 1, 2,\ldots$
τ	The round trip propagation delay, or the FRTT time, $\tau = \tau_{1+}\tau_0$
τ_0	The propagation delay experienced by the cells before returning to the VS
τ_1	The propagation delay experienced by the cells before they reach the bottleneck queue of the VD

Appendix B

The target of the new algorithm is to allow the switch to calculate the value of the parameter ERF of the BRM cell. Before presenting the SERICA algorithm, it will be helpful to declare the appropriate parameter set used by the switch mechanism. It is interesting to note that the same parameter set is always used on any other switch of the same link; however, it is generally, presented with different values, and thus it may result in different ERF.

B.1. Notation

k	The last connection established. It also presents the number of all active VCs
τ_j	The FRT time of the j-th active session of the switch
EF_j	The Equivalence Factor $EF_j = h/\tau_j$, with $h \geq \max\{\tau_j\}$of the j-th active incoming VC of the switch; $j = 1, 2,\ldots,k$
$T_{0,j}$	The target queuing delay, of the j-th active session (VCs) of the switch
T_0	The target queuing delay of the link, defined as $T_0 = \Sigma\ (EF_j{*}\ T_{0,j}/k)$
CCR_j	The *CCR* of the j-th active incoming VC of the switch, acknowledged from the last arrived FRM cell
$CCReq_j$	Defined as $CCReq_j = EF_j{*}CCR_j$. It represents the equivalent value of the CCR_j, over the time period h,for every active VC; $j = 1, 2,\ldots,k$
MCR_j	The *MCR* of the j-th active incoming VC of the switch negotiated during the connection setup
$MCReq_j$	Defined as $MCReq_j = EF_j{*}MCR_j$. It represents the

equivalent value of the MCR_j, over the time period h, for every active VC; $j = 1, 2, \ldots, k$

SUM_MCR Defined as $SUM\_MCR = \Sigma\ (EF_j{*}MCR_j)$. The switch accesses this value by using the last arrived FRM cell

PCR_j The PCR of the j-th active incoming VC negotiated during the connection setup

$PCReq_j$ Defined as $PCReq_j = EF_j{*}PCR_j$. It represents the equivalent value of the PCR_j, over the time period h, for every active VC; $j = 1, 2, \ldots, k$

SUM_PCR Defined as $SUM\_PCR = \Sigma\ (EF_j{*}PCR_j)$. The switch accesses this value using the last arrived FRM cell

Link_Rate The available total capacity of the switch

CBR The capacity of the switch allocated for the CBR traffic of the link at the end of the previous time interval

VBR The capacity of the switch allocated for the VBR traffic of the link at the end of the previous time interval

Total_ABR The capacity of the switch allocated to the ABR traffic of the link at the end of the previous time interval, defined as: $Total\_ABR = Link\_Rate - (CBR + VBR)$

link_ABR_input_rate An approximation of the Mean Arrival Cell Rate of the ABR traffic over the time period h. This parameter changes immediately when a new FRM cell arrives at the switch

BOR The buffer overflow rate

q The queue length of the switch, recorded at the end of the previous time interval h. This value is available during the subsequent time interval h

qpr The prediction $\hat{q}(t_{n+1})$ of the queue buffer of the switch during the next time instant

$q\max_j$ The maximum queue capacity allocated to the switch buffer and the j-th active session of the switch. This may be defined, during negotiation setup as $\mathrm{INT}\{(PCR_j - MCR_j){*}CDVT_j$, when there are enough resources, or less, otherwise

qst The stable point $T_0{*}Total\_ABR$ of the queue length, when all the active VCs of the switch are matched

qmaxeq Defined as $q\max eq = \Sigma(EF_j){*}q\max_j$. It represents the equivalent value of the $q\max_j$, over the time period h

$CDVT_j$ The Cell Transfer Delay Variation Tolerance of the j-th VC, negotiated during the connection setup

ER_in_RM The recorded value in the Backward Resource Management (BRM) cell of the Explicit Rate Field (ERF) received from each VS

ERF The parameter determining the estimated value of the *link_ABR_input_rate* of the subsequent time interval h. This value will be fairly shared among all the active VCs, to produce the *ER_in_RM* value

FairShare The value produced using a specific criterion according to which the *ERF* of every active VC is fairly shared

Theta Defined as *ERF/Total_ABR*

B.2. The key steps of the SERICA algorithm

From the Connection Admission Control algorithm
When a new connection is accepted:
$k \leftarrow k + 1$
set τ_k, $T_{0,k}$, $q\max_k$, MCR_k, PCR_k
if $(h < \tau_k)$ **then** $h \leftarrow \tau_k$
for $j = 1$ **to** k **do**
$EF_j = h/\tau_j$
$MCReq_j \leftarrow EF_j{*}MCR_j$; $SCReq_j \leftarrow EF_j{*}SCR_j$;
$PCReq_j \leftarrow EF_j{*}PCR_j$
$SUM\_MCR \leftarrow \Sigma(EF_j{*}SCR_j)$; $SUM\_SCR \leftarrow \Sigma(EF_j{*}SCR_j)$; $SUM\_PCR \leftarrow \Sigma(EF_j{*}PCR_j)$
$q\max eq \leftarrow \Sigma(EF_j{*}q\max_j)$; $T_0 = \Sigma(EF_j{*}T_{0,j}/k)$
end do
else
$EF_k = h/\tau_k$
$MCReq_k \leftarrow EF_k{*}MCR_k$; $SCReq_k \leftarrow EF_k{*}SCR_k$;
$PCReq_k \leftarrow EF_k{*}PCR_k$
$SUM\_MCR \leftarrow SUM\_PCR + EF_k{*}MCR_k$
$SUM\_SCR \leftarrow SUM\_SCR + EF_k{*}SCR_k$
$SUM\_PCR \leftarrow SUM\_PCR + EF_k{*}PCR_k$
$q\max eq \leftarrow q\max eq + EF_k{*}q\max_k$
$T_0 = (T_0{*}(k-1) + (EF_k{*}T_{0,k})/k$
end if
When the connection *i* is disconnected:
$k \leftarrow k - 1$
if $(h = T_{0,i})$ **then** $h \leftarrow \mathrm{MAX}(T_{0,i})$
$SUM\_MCR \leftarrow SUM\_PCR - EF_i{*}MCR_i$
$SUM\_SCR \leftarrow SUM\_SCR - EF_k{*}SCR_k$
$SUM\_PCR \leftarrow SUM\_PCR - EF_i{*}PCR_i$
$q\max eq \leftarrow q\max eq + EF_i{*}q\max_i$
$T_0 = (T_0{*}(k+1) - (EF_i{*}T_{0,i})/k$
for $j = i$ **to** k **do**
$\tau_j \leftarrow \tau_{j+1}$; $T_{0,j} \leftarrow T_{0,j+1}$; $q\max_j \leftarrow q\max_{j+1}$
$MCR_j \leftarrow MCR_{j+1}$; $SCR_j \leftarrow SCR_{j+1}$; $PCR_j \leftarrow PCR_{j+1}$
end do
The core of the SERICA
Initialization/at the end of an Averaging Time Interval:
set *q, Link_Rate, VBR, CBR*
$Total\_ABR \leftarrow Link\_Rate - (VBR + CBR)$
$qst = T_0{*}Total\_ABR$; $qpr = qu(q,\ Link\_ABR\_input\_rate,\ Total\_ABR,\ qmaxeq,\ h)$
$ERF \leftarrow ER(Link\_ABR\_input\_rate,\ Total\_ABR,\ qst,\ qmaxeq,\ q,\ qpr,\ h)$
$Theta \leftarrow ERF/Total\_ABR$
When a FRM cell is received via the *j*-th VC

$CCRold \leftarrow CCR_j$; $CCR_j \leftarrow CCR\_in\_RM\_Cell$
$Link\_ABR\_input\_rate \leftarrow Link\_ABR\_input\_rate + (CCR_j - CCRold)*EF_j$
When a BRM cell is received via the *j*-th VC
$ER\_in\_RM = min(ER\_in\_RM, FairShare)$
where, the *FairShare* is calculated according to:
When the Max–Min Fairness Criterion is used,
$FairShare \leftarrow INT(Theta*Total\_ABR/k)$
When the Max–Min Fairness Criterion, with MCR is used,
$FairShare \leftarrow MCR_j + INT((Theta*Total\_ABR - SUM\_MCR)/k)$
When the equally weighted with MCR Fairness Criterion is used,
$\Theta_j = (PCR_j - MCR_j)/(SUM\_PCR - SUM\_MCR)$
$FairShare \leftarrow MCR_j + INT(\Theta_j*(Theta*Total\_ABR))$
When the equally weighted with SCR Fairness Criterion is used,
$\Theta_j = (PCR_j - SCR_j)/(SUM\_PCR - SUM\_SCR)$
$FairShare \leftarrow SCR_j + INT(\Theta_j*(Theta*Total\_ABR))$
Procedures contained:
REAL FUNCTION $ER(Link\_ABR\_input\_rate, Total\_ABR, qst, qmaxeq, q, qpr, h)$
$BOR \leftarrow Link\_ABR\_input\_rate - Total\_ABR$
if $(q = 0)$ **then**
if $(BOR \leq 0)$ **then**
$ER \leftarrow Link\_ABR\_input\_rate + qst*h$
else
$ER = (qst + Total\_ABR)*h$
end if
else
if $(q = q\mathrm{max}eq.$ **AND**. $BOR > 0)$ **then**
$ER = (qst - qpr)*h + Total\_ABR$
else
$ER = Link\_ABR\_input\_rate + (qst - qpr)*h$
end if
end if
End of Function *ER*
REAL FUNCTION $qu(q, Link\_ABR\_input\_rate, Total\_ABR, qmaxeq, h)$
$BOR \leftarrow Link\_ABR\_input\_rate - Total\_ABR$
if $((q = 0.$ **AND**. $BOR < 0)$. **OR**. $(q = q\mathrm{max}eq.$ **AND**. $BOR > 0))$ **then** $BOR \leftarrow 0$
$qu \leftarrow q + BOR*h$
if $(qu > q\mathrm{max}eq)$ ***then*** $qu \leftarrow qmaxeq$
if $(qu < 0)$ **then** $qu \leftarrow 0$
End of Function *qu*

References

[2] 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.

[3] A.W. Barnhart, Explicit rate performance evaluation, ATM Forum/94-0983R1,http://www.cis.ohio-state.edu/~jain/atmforum.htm.

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

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

[6] R. Jain, S. Kalyanaraman, S. Fahmy, R. Goyal, Source behaviour for ATM ABR traffic management: an explanation, IEEE Communications Magazine, November 1996.

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

[8] 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.

[9] R. Jain, S. Kalyanaraman, R. Viswanathan, Rate based schemes: mistakes to avoid, ATM Forum/94-0882, September 1994, http://www.cis.ohio-state.edu/~jain/atmforum.htm.

[10] R. Jain, S. Kalyanaraman, R. Goyal, S. Fahmy, F. Lu, ERICA+: Extensions to the ERICA switch algorithm, ATM Forum/95/1346, October 1995.

[11] L. Kalampoukas, A. Varma, 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.

[12] S. Kalyanaraman, Traffic management for the Available Bit Rate (ABR) service in Asynchronous Transfer Mode (ATM) networks, PhD Dissertation, Dept. of Computer and Information Sciences, The Ohio State University, August 1997.

[13] S. Kalyanaraman, et al., The ERICA switch algorithm for ABR traffic management in ATM networks, IEEE/ACM Trans. Net. 8 (1) (2000) 87–98.

[14] B. Vandalore, R. Jain, R. Goyal, S. 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.

Further reading

V.I. Arnold (Ed.), Ordinary differential equations, MIT Press edition, 1978, ISBN 0-262-51018-9.