

# SERICA\_ $\omega$ . An extension of the SERICA algorithm

G. I. Mousadis<sup>1</sup>, T. A. Tsiligirides<sup>1</sup> and M. P. Bekakos<sup>2</sup>

<sup>1</sup> Informatics Laboratory, General Sciences Department,  
Agricultural University of Athens, Athens 11855, Greece,

<sup>2</sup> Laboratory of Digital Systems, Department of Electrical and Computer Engineering,  
School of Engineering, Democritus University of Thrace, Xanthi 67100, Greece,

## Abstract

This work presents an extension of a recently developed scheme, namely the Simple Explicit Rate Identification with Congestion Avoidance (SERICA) algorithm, which regulates the ABR traffic flow in an ATM network. Like SERICA, the new scheme, which is called SERICA\_ $\omega$ , takes into account the load and the queue length per VC, calculates the first approximation of the ER field of the BRM cell, and then it informs the switch accordingly. Because of the low buffer availability, the network administrator may not be able to initialize a new session. The solution of the above problem is given by means of deriving a new control function family for the SERICA\_ $\omega$  algorithm. This function has the property to decrease the average available queue length of the switch at about  $O(q^{1/\omega})$ , where  $q$  is the average queue length of the SERICA\_1, i.e., the basic SERICA. The above model is calibrated to use different values of  $\omega$ , in order to provide a guaranteed ABR service with short buffer availability (or, a limited queue length).

**Keywords** - SERICA\_ $\omega$  algorithm, SERICA algorithm, ERICA algorithm, TCP over ATM, extended ABR, ABR, Differentiated Services.

## 1. INTRODUCTION

As it is well known, Transmission Control Protocol (TCP), together with User Datagram Protocol (UDP), carries a significant amount of today's Internet traffic. It serves applications like www (HTTP), file transfer (FTP), E-mail (SMTP), remote access (Telnet), etc., as a means of reliable transport. TCP implements the additive increase multiplicative decrease scheme to achieve a fair division of available bandwidth in a network amongst competing sources. It is a window-based method in which, at any time, window size number of data packets are allowed in the network. This window size is additively increased roughly every Fixed Round Trip Time (FRTT) until congestion is detected. Then, a multiplicatively decreased mechanism is applied. In the current implementations of TCP the detection of congestion is detected implicitly, namely, by the

loss of packets. The diverse and changing nature of service requirements among the emerging Internet applications, desires a network architecture that is both flexible and able to distinguish their needs. A successful deployment of such architecture is offered by the so-called Differentiated Services (*Diff Serv*), which necessitates simple mechanisms inside the network. For example, a basic characteristic of *Diff Serv* architecture is the use of the same flow control algorithm in order to serve the different Type of Services (ToS), namely the Internet based applications, which have been classified according to their traffic description (see also Fujitsu, 2000). The *Diff Serv* architecture is based on a detailed contract between the Network Service Provider (NSP) and a user, which offers access on call. Thus, a connection is accepted or dropped by the NSP depending on the network resource availability. 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, or with the UDP. However, our focus is on the TCP since it is a connection-oriented protocol using a feedback window adjusted mechanism in a client-server loop.

From the above it appears that an interesting problem is to support integrated Internet applications, which are based on *Diff Serv* architecture, over the ABR traffic. Note that in an ATM network only the ABR service is able to guarantee an adjusted to the feedback, end-to-end, flow control traffic. For this reason the ABR service is best suited to support the TCP environment. According to the ATM FORUM (1996) (ATM FORUM Traffic Management Spec. Version 4.0), an ABR connection presumes a traffic contract supporting a *MCR* (Minimum Cell Rate), a *PCR* (Peak Cell Rate) and data transfer conforming with a *CDVT* (Cell Delay Variation Tolerance). In order to serve all kind of traffic, ATM has to guarantee that the maximum Cell Transfer Delay (*maxCTD*) and the deterministic part of the traffic flow, namely the Sustainable Cell Rate (*SCR*) are taking some predefined values. This last property gives rise to support the *Diff Serv* architecture through the extended ABR service and the corresponding traffic descriptor set of parameters (Mousadis (2001) and Mousadis & Tsiligrirides (2002)). In particular, the unpredictable ABR traffic requires a suitable flow control mechanism to automatically adjust the source emission rate according to the service rate fluctuations of the switches. This traffic is desirable to be served from a globally stable feedback mechanism, in order to operate safely with almost 100% utilization. However, to prevent buffer overflow, this mechanism should use a target threshold of the queuing delay ( $T_0$ ). Thus, in a steady state, i.e., when the Allowed Cell Rate (*ACR*) of the switch remains constant for more than one time interval, the target queuing delay, which depends on the current *ACR*, may determine a stable operational point ( $ACR, q_{st}$ ) of the corresponding target queue length ( $q_{st}$ ).

To serve the ABR traffic, the ATM Forum proposed the Explicit Rate (ER) based switch schemes, which are based on the leaky bucket model and offered as an alternative of a window based or a credit based schemes. The ER based closed-loop approach uses feedback acknowledgements via the Resource Management (RM) cells in both directions: in the Forward (from the source to the destination – FRM cells) and in the Backward (from the destination back to the source – BRM cells). To avoid congestion, a hop-by-hop strategy that uses the Virtual Source / Virtual Destination (VS/VD) property

in a tandem configuration may be adopted. The RM cells are used to control the VS emission rate, according to the VD requests. In particular, through the ER Field (*ERF*) of the BRM cells, the VD acknowledges the desirable source emission rate of the next time instant, namely the Worst Case Traffic (WCT) scenario, in which  $MACR(t_{n+1}) = ERF(t_n)$ ;  $t_n = \tau n$ ,  $n = 0, 1, 2, \dots$ .

The ER-based switch algorithms are divided into two major classes. The first class includes the Proportional Control type algorithms, which are locally unstable, whereas the second class includes the Congestion Avoidance algorithms (ERICA and ERICA+), which are globally stable. In particular, these last type of algorithms:

- achieve stability by using some artificial parameters optimized by simulation,
- ensure a target utilization with a predetermined bandwidth level (Kalyanaraman et al. (2001)),
- provide max-min fairness with *MCR* criterion.

However, according to the negotiated traffic descriptor, these algorithms do not guarantee for a specific *SCR*, i.e., they cannot support the extended ABR service.

ERICA type algorithms have been developed into the context of a Continuous State Leaky-Bucket, which has been proposed by the ATM Forum as a reference algorithm, with aim to determine some useful parameters at a connection set-up (Figure 1a). The algorithmic scheme has a close relationship with the single server queue model with deterministic service time (Mignault et al., (1996) ; Mousadis et al. (1996)). An equivalent problem is how to avoid overflow when a fluid tank is refilled (Figure 1b). The floater control valve regulates the input rate in accordance with the volume of the fluid stored into the tank and the input – output fluid flows of the tank. Assuming that the *ERF* of the BRM cell presents the Worst Case of Traffic (WCT) input rate and that the *ACR* is the output rate of the fluid tank (or the  $ACR(t_n)$  of the switch at the current time instant  $t_n = n\tau$ ,  $n = 0, 1, 2, \dots$ ), then a function  $f(q(t_n))$ , which determines the input rate in accordance with the output rate and the fluid stored in the tank may be defined. Obviously, this function presents the up/down movement of the floater mechanical bar and is derived algorithmically in each time instant  $t_n$  through the recursive equation:  $MACR(t_{n+1}) = ERF(t_n) = f(q(t_n)) \cdot ACR(t_n)$ .

An interesting family of such queue control function has been introduced and discussed by Kalyanaraman (1997). This function has the following form:

$$f(q(t_n)) \begin{cases} > 1, & \text{if } 0 \leq q(t_n) < q_0 \\ = 1, & \text{if } q_0 \leq q(t_n) < q_1 \\ < 1, & \text{if } q_1 \leq q(t_n) < q_2 \\ = QDLF, & \text{if } q_2 \leq q(t_n) < \infty \end{cases},$$

In the above,  $q_0$ ,  $q_1$  and  $q_2$ , are the three different levels of the queue length whereas *QDLF* (Queuing Delay Limit Factor) is the lower bound of the function. Four such control functions, namely the step, the linear, the hyperbolic and the inverse hyperbolic have been tested so far. It has been showed that the inverse hyperbolic function is approximated through the use of two rectangular hyperbolic functions in which the target queue length is 1 and performs better than the other schemes. Note that the step scheme is the simplest to implement in hardware since it does not require actual calculation, the

linear scheme is implemented efficiently using shift operations for some of its parameters, whereas the hyperbolic scheme includes a division operation, which adds calculation complexity and needs much more execution time than the other schemes. However, a common disadvantage of all these functions is that they use arbitrary parameters, which need to be optimized by simulation. This is because of the ERICA and ERICA+ schemes are based on an approximation of the window of the source in a steady state condition. Following this, it is assumed that the window of the source depends only on the *ACR* of the source (i.e., it is independent of the queue length of the switches).

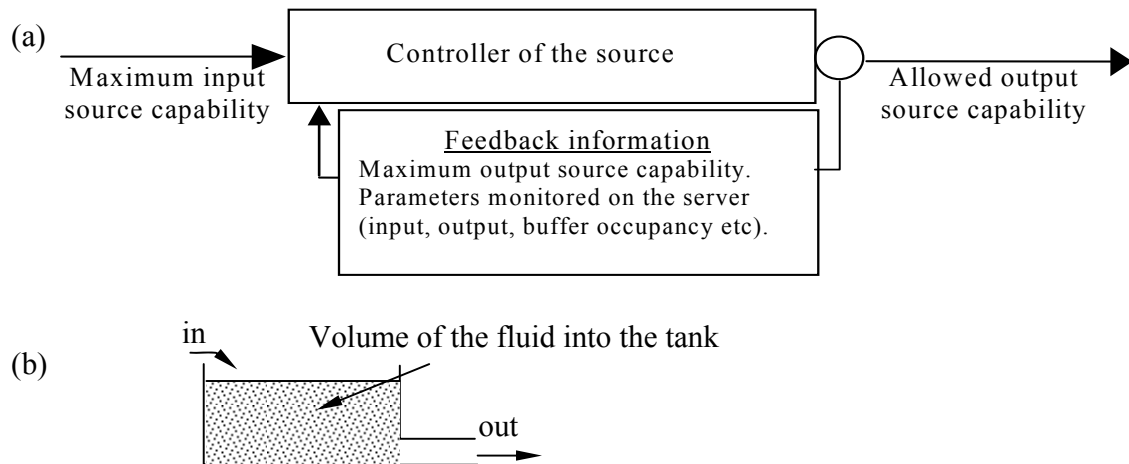


Figure 1: The block diagram of the source controller (a) (Leaky-Bucket problem) and its equivalent fluid tank overflow problem (b).

Into the above framework a recently developed algorithm (Mousadis & Tsiligirides (2002)) called Simple ERICA (SERICA) has shown an interesting superiority over the original ERICA and ERICA+ algorithms, in the following aspects:

- It obtains analytically a stable equilibrium point, using  $O(1)$  calculations.
- It adjusts the optimized parameters in each time slot according with the availability of the network resources.
- It guarantees a specific *SCR*, when a queue length of  $O(PCR)$  is allocated to the VC.
- It uses more than one BRM cells in order to manage in a better way the VS emission rate by taking into account the latest arrived information about a persistent congestion.

The contribution of this work is to extend the SERICA algorithm, in order to minimize further the queuing delay, namely, by giving the possibility to initialize a new session when a queue length less than  $O(PCR)$  is allocated in the VC. The extended algorithm, called  $SERICA_{\omega}$ , is based on both the model set-up and the rate control model of the ER scheme (Mousadis & Tsiligirides (2002)). Note that in order to increase the network operability a new parameter  $\omega$  is introduced. This parameter is determined during the call set-up by the network administrator in order to reduce the average queue length of the switch at about  $O(PCR^{1/\omega})$  in a steady-state. As it appears, the results are comparable with those obtained from the SERICA and ERICA algorithms, in the sense that in a steady-state the throughput of the new algorithm achieves nearly 100% of the available bandwidth, whereas in the non-steady state it does not drop from 79%.

The organization of this paper is as follows. Section 2 presents a review of the SERICA algorithm, Section 3 provides the model, the numerical and the stability analysis of the proposed SERICA<sub>ω</sub> algorithm, whereas Section 4 shows some interesting simulation results. Finally, the conclusions are presented in Section 5.

## 2. A REVIEW OF THE SERICA ALGORITHM

As it is already known ERICA and ERICA+ algorithms do not guarantee a specific *SCR*. This is a drawback since the *SCR* parameter allows the administrator to control the multimedia traffic. To achieve specific *SCR* the SERICA algorithm has been developed (Mousadis & Tsiligirides (2002)), which pays attention in the approximation of the source window. The algorithm is applied on a tandem network model, which consists of  $M$  switches arranged in  $M - 1$  links in a tandem configuration (Figure 2). The switches involved in this network model satisfy the VS/VD property, by means that a switch acts either as the VD of the previous switch, or as the VS of the next switch. Applying this property a hop-by-hop strategy may be adopted.

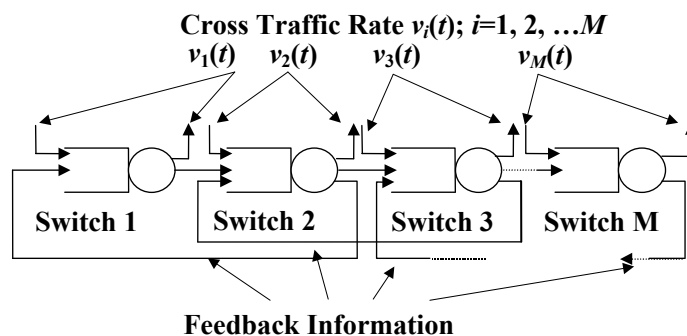


Figure 2: The general hop-by-hop model with  $M$  ATM switches in tandem.

The above model may be simplified into the basic model of Figure 3, in which the switch serves two types of ABR traffic streams: a tagged traffic having the highest priority and a background traffic, which immediately leaves the switch. Based on the ERF of the BRM cell received last, the switch adjusts its transmission rate every round-trip time and calculates the *MACR* for the subsequent time interval.

Let  $K(t_n)$  denotes the amount of the outstanding unacknowledged cells of the controlled traffic in the VS at the time  $t_n = n\tau$ ,  $n=0, 1, 2, \dots$ . As it may be observed,  $K(t_n)$  presents also the optimum window size in case the cell loss is negligible. Then,

- The source window is approximated by the amount of cells of the controlled traffic traveling from the VS to the VD during the  $[(n-1)\tau, n\tau)$  period, plus the cells already waiting in the buffer ( $q(t_n - \tau_0)$ ), plus the cells already accepted from the VD but their acceptance has not been acknowledged so far by the VS.
- The *in flight cells* are approximated by the average input rate of the VD at time  $t_n = n\tau$ ,  $n=0, 1, 2, \dots$ , which in the WCT is  $\tau MACR(t_{n-1} + \tau_1)$ , namely the output rate of the source at time  $(t_n - \tau_0)$ .

Thus, taking into account the notation given in Table 1, it is easy to verify the following

relation:

$$\begin{aligned} K(t_n) &= \tau MACR(t_{n-1} + \tau_1) + q(t_n - \tau_0) \\ \text{Window at the source} &= \text{In flight cells} + \text{Cells in queue} \end{aligned} \quad (1)$$

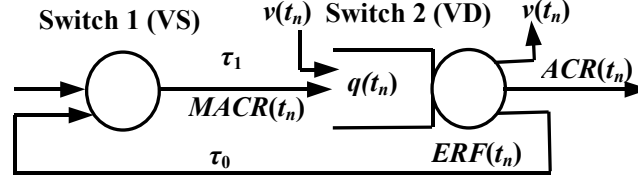


Figure 3: The reduced single hop single VC model, with background traffic rate  $v(t_n)$  and FRTT  $\tau (= \tau_0 + \tau_1)$

To regulate the traffic flow, the values given by the floater control valve mechanism have been approximated through a suitable function. This function has been derived using a discrete linear approximation of the first derivative of the window, which is given by:

$$dK(t_n) / dt_n \approx \Delta K(t_n) / \Delta t_n = [K(t_{n+1}) - K(t_n)] / \Delta t_n$$

The prescribed mechanism increases the system stability as long as the difference  $\Delta K(t_n) = K(t_{n+1}) - K(t_n)$  adjusts its values before the start of the time instant  $t_{n+1}$  and in accordance with the received feedback. However,  $\Delta K(t_n)$  is not known and as a result, different methods of approximation may be proposed. One such method is given by the SERICA algorithm, which assumes that

$$\Delta K(t_n) = a q(t_n - \tau_0) + b \quad (2)$$

and  $\Delta t = u / ACR(t_n - \tau_0)$ ,

where  $u$  is the number of BRM cells sent by the VD to the VS during the Fixed Round Trip Time (FRTT) and  $\Delta t$  is the adaptation period of the source rate.

The queuing control function already derived in the SERICA scheme (Mousadis & Tsiligirides (2002)) is as follows:

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

In the above:

- $h$  denotes the switch averaging time interval, i.e. the adaptation period of the VS.
- $z(t_n) = MACR(t_n) / ACR(t_n)$  denotes the load factor at time instant  $t_n$ .
- $\hat{q}(t_{n+1})$  denotes an estimation of the queue length of the switch at the end of the  $n^{\text{th}}$  time interval  $[t_n, t_n + h]$  i.e., an estimation of the queue length at time instant  $t_{n+1} = t_n + h$ .
- $T \hat{q}(t_{n+1}) = \hat{q}(t_{n+1}) / ACR(t_n)$  denotes an estimation for the queuing delay at the end of the  $n^{\text{th}}$  time instant  $[t_n, t_{n+1}]$  i.e., at time instant  $t_{n+1}$ ;  $t_{n+1} = t_n + h$ .
- $q \text{ max} = PCR (\text{maxCTD} - \tau) = (PCR - MCR) CDVT + PCR \cdot T_0$  denotes the maximum

threshold of the queue length of the switch.

$A, B, C, a,$ $b, \omega$	Parameters to be optimized.
$BO(t_n)$	The number of cells overflowed from the switch buffer, up to the time instant $t_n$ ; $n=0, 1, 2, \dots$
$\tau$	The round trip propagation delay, or the FRT time ( <i>FRTT</i> )
$h$	The switch “averaging” time interval. In general $h \geq \tau$ .
$K(t_n)$	The amount of outstanding unacknowledged cells of the controlled traffic (window size) at time instant $t_n$ ; $n = 0, 1, 2, \dots$
<i>CDVT</i>	The cell delay variation tolerance
<i>CTD</i>	The Cell Transfer Delay, i.e., the round trip transmission delay, which is the sum of the propagation delay $\tau$ , 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 $[(n-1)\tau, n\tau]$ ; $n = 0, 1, 2, \dots$
$\max CTD$	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 $t_n$ ; $n = 0, 1, 2, \dots$
$\hat{q}(t_{n+1})$	An estimation of the queue length of the switch at the end of the $n^{\text{th}}$ time interval $[t_n, t_{n+1})$ i.e., at time instant $t_{n+1}$ ; $n = 0, 1, 2, \dots$
$q_{eq}$	The equivalent target queue length, which is equal to $qst^\omega$
$qst$	The target queue length of the switch buffer, i.e. $T_0 ACR(t_n)$ at time $t_n$ ; $n = 0, 1, 2, \dots$
$q_{max}$	The maximum queue length allocated on the switch buffer in order to serve the VC.
$T$	The sampling time period
$T_0$	An acceptable threshold for the average queuing delay of the switch buffer.
$t_n$	The time instant $t_n = n\tau$ ; $n = 0, 1, 2, \dots$ . Generally, $t_n = nT$ ; $n = 0, 1, 2, \dots$ , 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 $n^{\text{th}}$ time period; $n = 0, 1, 2, \dots$
$f(t_n)$	A discrete form of the control function $f$ .
$z(t_n)$	The load factor given by the ratio $MACR(t_n)/ACR(t_n)$ .
$Tq_{max}(t_n)$	The estimation of the maximum allowed queuing delay, i.e., the ratio $qst / ACR(t_n)$ at time instant $t_n$ ; $n = 0, 1, 2, \dots$
$Tq_{max}$	The maximum queuing delay given by the difference $\max CTD - FRTT$ .
$T\hat{q}(t_{n+1})$	An estimation of the queuing delay at the end of the $n^{\text{th}}$ time interval $[t_n, t_{n+1})$ i.e., at time instant $t_{n+1}$ ; $n = 0, 1, 2, \dots$
$ACR(t_n)$	The Allowed Cell Rate of the $i^{\text{th}}$ switch at time $t_n$ ; $n = 0, 1, 2, \dots$
$ERF(t_n)$	The Explicit Rate Field of the BRM cell of the $i^{\text{th}}$ switch at time $t_n$ ; $n = 0, 1, 2, \dots$
$MACR(t_n)$	The Mean Arrival Cell Rate of the $i^{\text{th}}$ switch of the period $[(n-1)T, nT)$ , recorded at time instant $t_n$ ; $n = 0, 1, 2, \dots$
<i>MCR</i>	The negotiated Minimum Cell Rate of the session
<i>PCR</i>	The negotiated Peak Cell Rate of the session

Table 1: The notation of SERICA and SERICA <sub>$\omega$</sub>



- $Tq_{\max}(t_n) = q_{\max} / ACR(t_n) \leq Tq_{\max} = q_{\max} / MCR$  is the estimation of the maximum allowed queuing delay of the current time instant  $t_n$ .
- $T_0 = h [PCR + MCR - ACR(t_n)] / ACR(t_n)$  presents the target queuing delay of the switch. Note that in case of the ABR service  $T_0 = CDVT$ , whereas in case of the extended ABR service,  $T_0 = \max CTD - \tau$ .

As it has been pointed out the SERICA algorithm, attempts to obtain the most throughput. This has been achieved by adjusting the VS emission rate and the VD buffer occupancy in accordance with the  $ACR$  and the  $qst$  respectively. The algorithm is able to guarantee a specific  $SCR$  according to the traffic descriptor set of the session. Thus, the network is able to support real and/or non-real time traffic over ABR, such as the *Diff Serv* applications. However, SERICA is not able to face a particular problem of a NSP, related with the availability of the buffer resources. For example, a new client would not be able to merit the negotiated service on call due to the low availability of the switch buffer. Thus, this client would prefer to join a session with less guaranteed throughput, than nothing.

From the above it appears that a new ER-based switch algorithm is required, by means of the SERICA $_{\omega}$  scheme, which in general will be able to support a target queuing delay lower than (or at most equal to) the  $T_0$  proposed by the SERICA algorithm. This new scheme will be used as a “gearbox”, to initialize a session according to the buffer availability. Into this context the parameter  $\omega$  plays the role of the “gear ratio” used.

### 3. THE SERICA $_{\omega}$ ALGORITHM

#### 3.1. Model Analysis

As it has been pointed out, the values given by the floater control mechanism are approximated through a suitable control function. This function regulates the traffic flow and keeps the queue length values less than the target queue length  $qst$ , namely the threshold provided by the SERICA algorithm. This is an interesting problem since it corresponds in the case in which the network administrator may not be able to establish a session due to the low buffer availability. Figure 4 depicts graphically various possibilities to regulate the traffic flow, by considering some well-known mechanical examples.

To keep the equations used as simple as possible, the mathematical description of the window  $K(t_n)$  given by equation (1) should not be changed. However to increase the flexibility of the system, by means of establishing a session with target queue length lower than the threshold  $qst$  provided by the SERICA algorithm, the source window variation  $\Delta K(t_n)$  is assumed to scale arbitrarily. From the administrator point of view this approach appears to be simple and efficient, nevertheless, as it will be seen later, it increases the instability of the system.

Beholding that  $\Delta K(t_n)$  depends on the unpredictably varied parameter  $ACR$  of the switch, an unpredictable growth of the queue length may be observed at the switch buffer. Of course, this may result in cell losses, either due to overflow, or due to the non-conformance test. As an example, we may consider the case of processing a long file

transfer, during which it is desired to avoid congestion by keeping cell loss closed to zero. A solution in such problems may be obtained through an Explicit Rate switch algorithm, in which the source window variation  $\Delta K(t_n)$  is assumed to be approximated generally by a polynomial function of order  $\omega$  of the queue length of the switch, acknowledged at the source at time  $t_n$ . Thus, in a steady state, i.e. when  $ACR$  remains constant and in case the observer is located in the VS site,  $\Delta K(t_n)$  may assumed to be of the form:

$$\Delta K(t_n) = A_0 q^\omega (t_{n+1} - \tau_0) + A_1 q^{\omega-1} (t_{n+1} - \tau_0) + \dots + A_{\omega-1} q^1 (t_{n+1} - \tau_0) + A_\omega \quad (3)$$

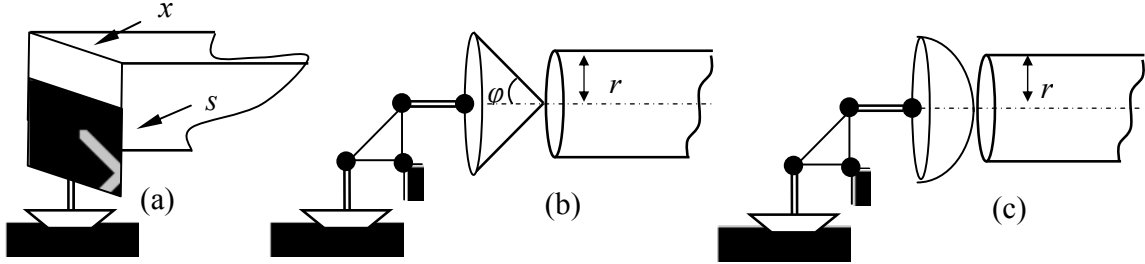


Figure 4: According to the lumen of the pipe and to the faucet used, there are three simple float control valves:

- (a) Orthogonal pipe (lumen area  $LA = s \cdot x$ ), with an orthogonal sliding faucet
- (b) Cylindrical pipe (rad  $r$ ), with a conical sliding faucet (angle  $\phi$ ) and
- (c) Cylindrical pipe (rad  $r$ ), with a hemispherical sliding faucet.

A small change in the volume used of the fluid tank invokes an up/down movement  $\Delta h$  in the float, which produces a change in the corresponding lumen area used namely,  $\Delta LA = A_1 \Delta h$ ,  $\Delta LA = A_2 (\Delta h)^2$  and  $\Delta LA = A_3 (\Delta h)^3$ , respectively.

Generally, this approach is not recommended since it requires  $\omega + 1$  equations in order to evaluate the coefficients  $A_i$ ;  $i = 0, 1, 2, \dots, \omega$ . Alternatively, an interesting case arises by considering the function:

$$\Delta K(t_n) = A(q(t_{n+1} - \tau_0) + B)^\omega + C \quad (4)$$

which requires only three equations to evaluate the parameters  $A$ ,  $B$  and  $C$  and gives a flavour about the contribution of the terms of rank  $\omega$  into the variation of the source window  $\Delta K(t_n)$ . Nevertheless, the above approach leads to a non-linear Delay Differential system of Equations (DDE's). As for the possible values of  $\omega$  one may note the following:

**Remark 1:** The source window variation  $\Delta K(t_n)$  depends on the queue length, raised on the power of order  $\omega$ . There are three cases to be considered:

- In case  $\omega > 1$ , a queue length smaller than the  $qst$  of the SERICA has to be used, in order to request from the source to decrease its transmission rate (Figure 4(b) or 4(c)).
- In case  $0 < \omega < 1$ , the target queue length is greater than the  $qst$  of the SERICA. Therefore SERICA algorithm may be used efficiently.
- In case  $\omega = 1$ , the queue length is equal with the  $qst$  of the SERICA. Thus, SERICA\_1 coincides the basic SERICA (Figure 4(a)) in which equation (4) takes the simple form of (2), with  $a = A$  and  $b = A \cdot B + C$ . The physical interpretation of the

equation (2) is that when  $q(t_{n+1})$  is approximated with  $-b/a$  the window will not change, namely,  $K(t_{n+1}) = K(t_n)$ . In this case  $MACR(t_{n+1}) = MACR(t_n) = ACR(t_n)$ , which verifies that  $(q(t_{n+1}), MACR(t_{n+1})) = (-b/a, ACR(t_n))$  is an equilibrium point. SERICA\_1 has been fully examined in Mousadis & Tsiligirides (2002).

•

### 3.2. Numerical Analysis

The analysis for the case of interest ( $\omega > 1$ ) starts by eliminating the source window parameter  $K(t_n)$ . This may be achieved by using equations (1) and (4) above. Thus, by differentiating the equation (1) above, one may obtain:

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

The first derivative of  $K(t_n)$  is calculated from equation (4) as follows:

$$\frac{d}{dt}K(t_n) \approx \frac{\Delta K(t_n)}{\Delta t} = \left[ A(q(t_{n+1} - \tau_0) + B)^\omega + C \right] \frac{ACR(t_n - \tau_0)}{u}; \Delta t = u / ACR(t_n - \tau_0),$$

Therefore, in case of  $A \neq 0$ :

$$\frac{d}{dt}MACR(t_n - \tau_0) = \left[ (q(t_{n+1} - \tau_0) + B)^\omega + \frac{C}{A} \right] \frac{A}{u\tau} ACR(t_n - \tau_0) - \frac{d}{dt} \frac{q(t_n - \tau_0)}{\tau}.$$

Similarly, assuming the observer is located in the VD site, it follows:

$$\frac{d}{dt}MACR(t_n) = \left[ (q(t_{n+1}) + B)^\omega + \frac{C}{A} \right] \frac{A}{u\tau} ACR(t_n) - \frac{d}{dt} \frac{q(t_n)}{\tau}.$$

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 given by the following system of equations:

$$\frac{d}{dt}MACR(t_n) = \begin{cases} B^\omega + \frac{C}{u\tau} ACR(t_n), & \text{if (a)} \\ \left[ (q(t_{n+1}) + B)^\omega + \frac{C}{A} \right] \frac{A}{u\tau} ACR(t_n) - \frac{d}{dt} \frac{q(t_n)}{\tau}, & \text{if (b)} \\ \left[ (q \max + B)^\omega + \frac{C}{A} \right] \frac{A}{u\tau} ACR(t_n), & \text{if (c)} \end{cases} \quad (5)$$

and

$$\frac{d}{dt}q(t_n) = \begin{cases} MACR(t_n) - ACR(t_n), & \text{if (b)} \\ 0, & \text{otherwise (i.e., (a) \& (c))} \end{cases} \quad (6)$$

where:

$$(a) \quad q(t_n) = 0 \wedge MACR(t_n) < ACR(t_n)$$

$$(b) \quad \begin{cases} 0 < q(t_n) < q \max, \text{ or} \\ q(t_n) = 0 \wedge MACR(t_n) \geq ACR(t_n), \text{ or} \\ q(t_n) = q \max \wedge MACR(t_n) \leq ACR(t_n) \end{cases}$$

$$(c) \quad q(t_n) = q_{\max} \wedge MACR(t_n) \geq ACR(t_n)$$

In the above,

- $q_{\max} = PCR (\maxCTD - \tau) = (PCR - MCR) CDVT + PCR \cdot T_0$  is the maximum threshold of the queue length of the switch
- $q(t_{n+1})$  is estimated through the first derivative of the queue length at time  $t_n$ , using the Euler method. Thus,

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

According to the operational environment determined by the parameters  $MACR$ ,  $ACR$  and queue length at the switch, the system of equations (5) and (6) should lead automatically in a stable focus point, which provides a “safe life” for the session. Provided that the above system has only one equilibrium point, the corresponding root must be real and positive, whereas all the remaining should be conjugate complex, or negative. The desired root is given by  $(MACR, q) = (ACR, q_{st})$ . Note that in the above analysis we exclude the case of  $ACR = 0$  and / or  $MACR = 0$ , which leads to the trivial case of  $(MACR, q) = (0, 0)$ . In addition;

$$\frac{d}{dt} MACR(t_n) \begin{cases} > 0, & \text{if (a)} \\ > 0, & \text{if } q(t_n) \leq q_{st} \\ = 0, & \text{if } q(t_n) = q_{st} \text{ or, } MACR(t_n) \\ < 0, & \text{if } q(t_n) \geq q_{st} \\ < 0, & \text{if (c)} \end{cases} \begin{cases} \uparrow, & \text{if (a)} \\ \uparrow, & \text{if } q(t_n) \leq q_{st} \\ MACR(t_n), & \text{if } q(t_n) = q_{st} \\ \downarrow, & \text{if } q(t_n) \geq q_{st} \\ \downarrow, & \text{if (c)} \end{cases}$$

and the system remains closed to the equilibrium point  $(ACR, q_{st})$ . Note that the parameter values for  $MACR$  and  $ACR$  change at any time instant  $t_n$ ;  $n = 0, 1, 2, \dots$  and therefore different values of  $ACR(t_n)$  determine different equilibria points. However, since  $ACR(t_n)$  varies unpredictably between two consecutive time instants, the system will be considered in a steady state condition, i.e., the  $ACR(t_n)$  remains constant for more than one averaging time interval.

In order to determine the target queue length of the switch buffer  $q_{st}$ , we assume  $A = -u\tau/ACR(t_n)$ . Note that in the following and for simplicity reasons we will denote by  $q_{st}$ , instead of  $q_{st}(ACR(t_n))$ . Thus,

$$q_{st} := -B + (-C/A)^{1/\omega} = -B + (C \cdot ACR(t_n)/(u\tau))^{1/\omega}.$$

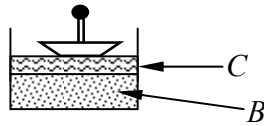


Figure 5: Assuming that the parameter  $B$  presents the fluid quantity ( $\geq 0$ ) already loaded into the tank, then the parameter  $C$  presents the required fluid quantity to be added ( $\geq 0$ ), in order to close the faucet. To keep the buffer occupancy as lower as possible, the optimal value for  $B$  is zero.

Knowing the parameter  $A$ , the remaining parameters  $B$  and  $C$  are obtained easily. In particular, the optimal value for  $B$  is equal to zero, because the system is desired to operate with the lowest buffer occupancy (Figure 5). Note that the optimal value for  $B$

determines the target queue length of the switch buffer as:

$$qst := (-C/A)^{1/\omega} = \left( C \frac{ACR(t_n)}{u\tau} \right)^{1/\omega}.$$

Finally, assuming that the target queuing delay,  $T_0$  say, is specified by the application being in service,  $qst = T_0 ACR(t_n)$  and therefore,  $C = T_0 u \tau (ACR(t_n))^{(\omega-1)/\omega}$ . As it appears, for large values of  $\omega$ , the ratio  $(\omega - 1)/\omega \rightarrow 1$  and hence the parameter  $C \rightarrow T_0 u \tau ACR(t_n)$ , which implies that  $qst \rightarrow (T_0 ACR^2)^{1/\omega} \rightarrow 1$ . However the values of  $\omega > 1$  increase the instability of the system, namely, it causes an oscillating behavior around the equilibrium point (Figure 6(b)). Hence, small values of  $\omega$  are more important (Figure 6(a)). Note that Figure 4 shows some interesting particular cases of the floater control valve, in which a faucet with different geometry may change dramatically the fluid quantity required to close the pipe. Thus, small values of  $\omega$  decrease the queuing delay.

Given the traffic descriptor, the NSP will serve a client, in the worst case, in accordance with the negotiated target queuing delay (here the worst case is presented with the SERICA\_1 algorithm). Hence, parameter  $T_0$  is derived as:

$$T_0 = [h (PCR + MCR - ACR(t_n))]^{1/\omega} / ACR(t_n).$$

and acknowledges that SERICA\_ $\omega$  algorithm always use auto-adjusted parameters, to achieve in the worst case a target queuing delay, related with the selected value of  $\omega$ , as well as, with the negotiated traffic descriptor set for  $PCR$  and  $MCR$ .

Taking into account the values for  $B$  and  $qst$  the system of ODEs described by equations (5) and (6) may be rewritten as follows:

$$\frac{d}{dt} MACR(t_n) = \begin{cases} qst^\omega, & \text{if (a)} \\ -\left[ \left( \hat{q}(t_{n+1}) \right)^\omega - qst^\omega \right] - \frac{d}{dt} \frac{q(t_n)}{\tau}, & \text{if (b)} \\ -\left[ (q \max)^\omega - qst^\omega \right], & \text{if (c)} \end{cases} \quad (7)$$

$$\frac{d}{dt} q(t_n) = \begin{cases} MACR(t_n) - ACR(t_n), & \text{if (b)} \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

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  $h \geq \tau$ . In the WCT, namely when the source is able to transmit at the requested rate:

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

By integrating equation (7), with step  $h$ , one may obtain :

$$ERF(t_n) = MACR(t_n) + \begin{cases} qst^\omega h, & \text{if (a)} \\ -\left[ \left( \hat{q}(t_{n+1}) \right)^\omega - qst^\omega \right] h - \frac{h}{\tau} \frac{d}{dt} q(t_n), & \text{if (b)} \\ -\left[ (q \max)^\omega - qst^\omega \right] h, & \text{if (c)} \end{cases} \quad (9)$$

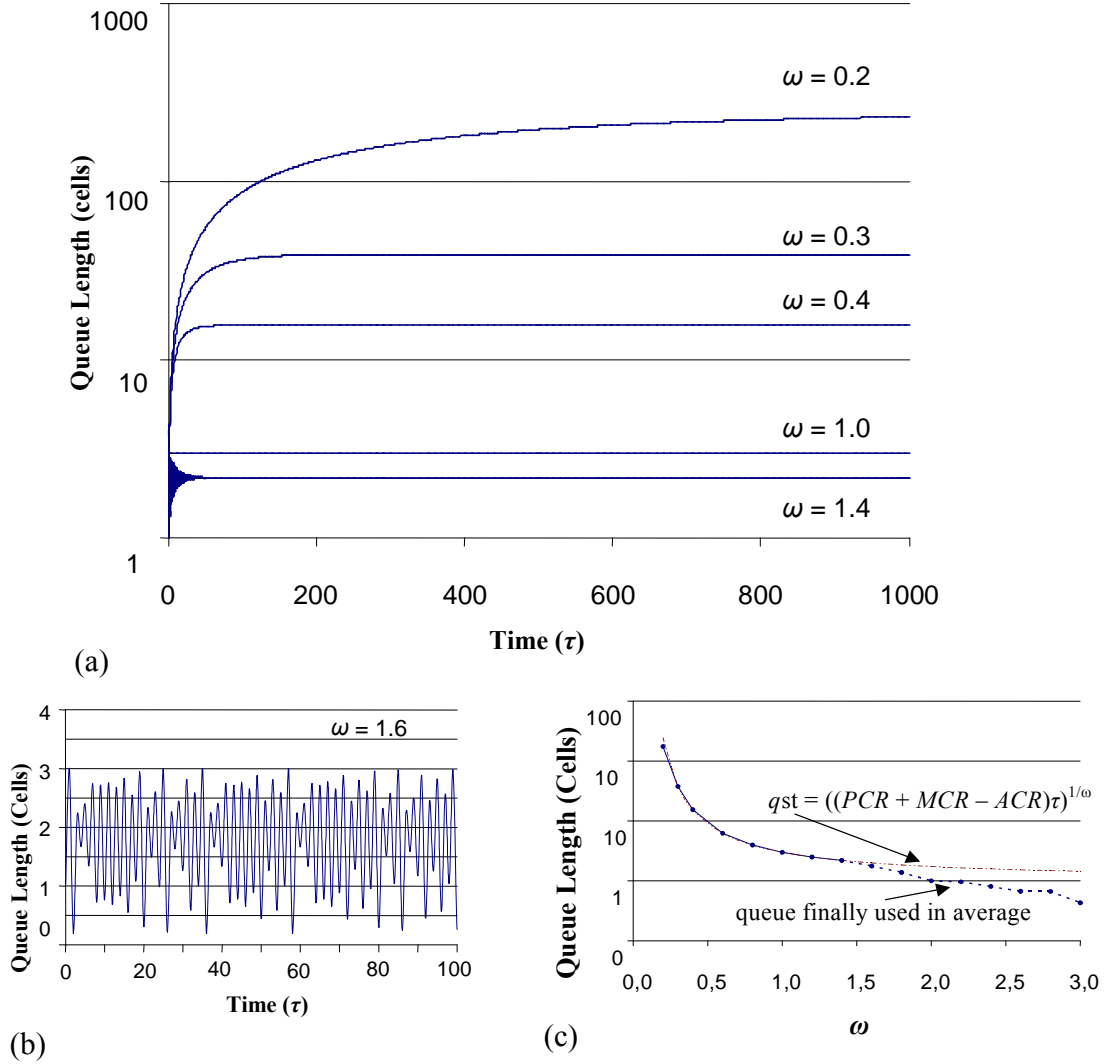


Figure 6: The 1 hop 1 VC network model in a steady state condition ( $PCR = 36 \text{ cells} / \tau$ ,  $MCR = 1 \text{ cell} / \tau$ ,  $ACR = 34 \text{ cells} / \tau$ ). If  $|\omega - 1| \uparrow$ , then the solution path reciprocates around the globally stable equilibrium point  $(ACR, q_{st})$ .

(a) When  $0 < \omega \leq 1$ , the time to approach the equilibrium point decreases, as far as  $\omega$  approaches to 1.0. Further, when  $\omega > 1$  and  $\omega$  remains closed to 1, the solution path presents a dumping oscillating behavior.

(b) Increasing further  $\omega$ , the solution path reciprocates around the equilibrium point

(c) In case of low buffer availability in the network, a new session may be initialized with  $\omega = \log_{q_{av}}(qeq)$ ;  $qeq = h(PCR + MCR - ACR)$  and  $q_{av}$  is the queue length available on average for the session.

and

$$q(t_{n+1}) = \begin{cases} q(t_n) + (MACR(t_n) - ACR(t_n))h, & \text{if (b)} \\ q(t_n), & \text{otherwise} \end{cases} \quad (10)$$

As it appears, starting with an initial buffer  $q(t_0)$ , mean arrival rate  $MACR(t_0)$  and

allowed cell rate  $ACR(t_0)$ , the procedure enables the calculation of the  $MACR(t_1)$  ( $=ERF(t_0)$ ) from equation (9), using the Euler method. This value is used to calculate the new buffer  $q(t_1)$  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 the switch sends  $u$  BRM cells every time period  $\tau$  keeping the same ERF value, because the corresponding  $ACR(t_n)$  is assumed constant during the time period  $[n\tau, (n+1)\tau)$ .

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

$$\frac{d}{dt}BO(t_n) = \begin{cases} MACR(t_n) - ACR(t_n), & \text{if } q(t_n) = q_{max} \wedge MACR(t_n) > ACR(t_n) \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

The function  $f(q(t_n))$  is derived in a straight-forward manner as:

$$f_{-\omega}(t_n) = \begin{cases} z(t_n) + hT_0^\omega, & \text{if (a)} \\ z(t_n) \left( \frac{\tau - h}{\tau} \right) + h \left( T_0 - T\hat{q}(t_{n+1}) \right) \varphi \left( \hat{q}(t_{n+1}), qst, \omega \right) + \frac{h}{\tau}, & \text{if (b)} \\ z(t_n) + h(T_0 - T_{max}) \varphi \left( \hat{q}(t_{n+1}), qst, \omega \right), & \text{if (c)} \end{cases}$$

In the above:

- $z(t_n) = MACR(t_n) / ACR(t_n)$  denotes the load factor at time  $t_n$ .
- $T\hat{q}(t_{n+1}) = \hat{q}(t_{n+1}) / ACR(t_n)$  denotes an estimation for the queuing delay at the end of the  $n^{\text{th}}$  time instant  $[t_n, t_{n+1})$  i.e., at time  $t_{n+1}$ ;  $t_{n+1} = t_n + h$ .
- $Tq_{max}(t_n) = q_{max} / ACR(t_n) \leq Tq_{max} = q_{max} / MCR$  is the estimation of the maximum allowed queuing delay of the current time instant  $t_n$ .
- $\varphi \left( \hat{q}(t_{n+1}), qst, \omega \right) = \left( \frac{qst^\omega - \hat{q}(t_{n+1})^\omega}{qst - \hat{q}(t_{n+1})} \right)$

As we shall see, the above function  $f_{-\omega}(t_n)$  will be simplified further, in accordance with the actual value taken of the parameter  $\omega$ .

Thus, the new session will start with the SERICA\_ $\omega$  (e.g., with  $\omega = 2$ ) algorithm. When the buffer availability increases (i.e., when more memory may be allocated to this session), the system toggles the algorithm from the SERICA\_ $\omega$ , to the SERICA\_1. I.e., the new session will start with a “gear ratio”  $\omega > 1$  in the “gearbox” of SERICA. The key point of SERICA\_ $\omega$  algorithm is that it may be used in order to achieve lower values of queue length than the  $qst$  of the SERICA algorithm, which at least resides in the order of  $O(PCR^{1/\omega})$ . Thus, the SERICA\_ $\omega$  algorithm may be used to operate with a small available queue threshold at the switch. Further, one may configure the  $q_{max} = (2 PCR h)^{1/\omega}$  in order to serve most of the users.

### 3.3. Stability analysis

Usually it is very hard to determine the domain of stability of an equilibrium point. Rigorous mathematical definitions are often too prescriptive and it is not always clear which properties of solutions or equations are most important in the context of any particular problem. In practice, different definitions are used according to the problem

being considered. In our case, the dynamic system described in a continuous time domain by a set of first order differential equations and a set of algebraic output equations in the form of:

$$\frac{d}{dt} MACR(t) = \begin{cases} qst^\omega, & \text{if (a)} \\ -\left[(q(t))^\omega - qst^\omega\right] - \frac{d}{dt} \frac{q(t)}{\tau}, & \text{if (b) and} \\ -\left[(q \max)^\omega - qst^\omega\right], & \text{if (c)} \end{cases} \quad (12)$$

$$\frac{d}{dt} q(t) = \begin{cases} MACR(t) - ACR(t), & \text{if (b)} \\ 0, & \text{otherwise} \end{cases} \quad (13)$$

To investigate the Liapounov stability behavior of the system of equations (12) and (13), we define the following functions:

$$L(t) = MACR(t) - ACR(t)$$

$$Q(t) = q(t) - qst$$

or, in a simplified form,

$$L = MACR - ACR$$

$$Q = q - qst$$

Differentiating with respect to the time, one may obtain

$$\frac{dL}{dt} = MACR \text{ or,}$$

$$\frac{dL}{dt} = -\left[q^\omega - qst^\omega\right] - \frac{d}{dt} \frac{q}{\tau} = -\varphi(q, qst, \omega) \cdot Q - \frac{1}{\tau} \frac{dQ}{dt} = -\varphi(q, qst, \omega) \cdot Q - \frac{L}{\tau} \text{ and} \quad (14)$$

$$\frac{dQ}{dt} = \frac{dq}{dt} = MACR - ACR = L \quad (15)$$

where  $\varphi(q, qst, \omega) = \frac{q^\omega - qst^\omega}{q - qst}$ . However,  $q \geq 0$  i.e., the possible values it may take are

0, 1, 2, 3,...,  $q_{max}$  and  $\omega$  depends on the initial conditions. Thus,  $\varphi(q, qst, \omega)$  is a well defined function, always greater than 0, which according to the values of  $\omega$  may be as it follows:

$$0 < \varphi(q, qst, \omega) \begin{cases} < 1, & \text{if } \omega < 1 \\ = 1, & \text{if } \omega = 1 \\ > 1, & \text{if } \omega > 1 \end{cases} \quad (16)$$

The system of equations (14) and (15) is simplified further giving:

$$\frac{dQ}{dt} = L \quad (17)$$

$$\frac{dL}{dt} = -\varphi(q, qst, \omega) \cdot Q - \frac{L}{\tau} \quad (18)$$

The above system of equations is of the form

$$\frac{dx}{dt} = Ax,$$



where

$$\frac{dx}{dt} = \begin{pmatrix} \frac{dL}{dt} \\ \frac{dQ}{dt} \end{pmatrix}, \quad A = \begin{pmatrix} 0 & 1 \\ -\varphi & -\frac{1}{\tau} \end{pmatrix} \text{ and } x = \begin{pmatrix} L \\ Q \end{pmatrix}; \quad \varphi = \varphi(q, qst, \omega).$$

In the above, A is a Jacobian matrix and for simplicity we assume that  $\tau = 1$ . Looking for equilibrium points with  $\frac{dL}{dt} = 0$  and  $\frac{dQ}{dt} = 0$  we see that  $(L, Q) = (0, 0)$ . The

characteristic equation of the above system is

$$|sI-A| = 0, \text{ or, } s(s+1)+\varphi = 0, \text{ or, } s^2 + s + \varphi = 0 \quad (19)$$

Hence, the eigenvalues of A, are:

$$s_{1,2} = \frac{(-1 \pm \sqrt{1-4\varphi})}{2} = \frac{(-1 \pm i\sqrt{4\varphi-1})}{2}$$

Both of them have strictly negative real parts for all  $\varphi > 1/4$  presenting the boundary values between of which the solution path is to oscillate. Provided that  $1-4\varphi \neq 0$ , or,  $\varphi \neq 1/4$ , they are distinct. Therefore, the equilibrium point  $(0, 0)$  is asymptotically stable (Figure 6).

To present the stability behavior of the system under consideration, some indicative values for  $\omega$  are examined, namely  $\omega = 1$ , (SERICA\_1 – gear ratio 1),  $\omega = 2$  (SERICA\_2 – gear ratio 2) and  $\omega = 3$  (SERICA\_3 – gear ratio 3), which correspond to  $\varphi(q, qst, \omega) = 1$ ,  $\varphi(q, qst, \omega) = q + qst$  and  $\varphi(q, qst, \omega) = q^2 + q \cdot qst + qst^2$  respectively and are all greater than, or equal to 1. Thus, the equilibrium point  $(0, 0)$  is asymptotically stable for each one of the above cases.

#### 4. SIMULATION RESULTS

SERICA\_ $\omega$  introduces a series of similar algorithms according to the different values of  $\omega$ . Nevertheless, the most useful members of this family appears to be:

- SERICA\_1 because it works as the basic SERICA algorithm and
- SERICA\_2, because its target queue length is the square root of the one referred to the SERICA\_1. For example, a switch operating with the SERICA\_2 algorithm will use, in average, less than or equal to 50% of the buffer allocated to the SERICA\_1 algorithm.

In the following the SERICA\_2 is tested by simulation using the simple configuration shown in Figure 7. This network configuration consists of a source, two ABR switches and a destination, connected in tandem using a single VC. The ABR traffic descriptor parameters are:  $PCR = 36 \text{ cells}/\tau$ ,  $MCR = 1 \text{ cell}/\tau$  and  $qmax = 2 h PCR$ , with  $h = \tau = 1$ . The simulator used herein, is the same as the one described by Mousadis & Tsiligrirides (2002) and it is used to produce extended simulation results, over a period of  $1000\tau$ , according to the ATM FORUM (1996), “Performance Testing Spec.”.

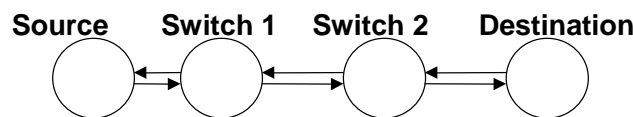


Figure 7: The 2 hop, 1 VC network model.

The SERICA\_2 algorithm is tested versus the SERICA\_1 and the ERICA/ERICA+ scheme, using the same simulation environment of Mousadis & Tsiligrirides (2002). The simulation results are presented in Table 2 (a) and (b). The description of Table 2(a) is beginning by showing in column “p” the corresponding probability values, used to generate a number of cells at the source, presented in the column (1). These cells will be transmitted through the network, using the three ER-based congestion avoidance algorithms, denoted in the columns of Table 2 as ‘SER-2’, ‘SER-1’ and ‘ER’, respectively. Column (2) shows the actual number of cells transmitted in each session, while column (3) presents the number of cells requested by the source during the same session. The average queuing delay in the buffer of the switches 1 and 2 are shown in columns (4) and (5) respectively. Further, in Table 2(b), in column (6) are shown the number of cells waiting in the source to be transmitted, when the time slot expires. These cells will never be transmitted and therefore they are discarded from the queue, namely they are considered as lost. This assumption may help in a theoretical analysis of the model and it is not harmful in real applications. The cells that finally arrived at the destination are shown in column (9). Further, in columns (7) and (8) is presented the number of cells taken away from the buffer queues of the switches 1 and 2 respectively, since their lifetime (CDVT) of  $2\tau$  has been expired. Finally, in column (10) the throughput of each session is presented in a per % scale. As it appears from the simulation, the overflow rate  $BO(t_n)$  given by the equation (11) is zero for all of the tested algorithms and thus it is excluded from this table.

The simulation shows some interesting results concerning the overall performance of the SERICA\_ω algorithms versus the ERICA/ERICA+ algorithm, by means of the higher throughput achieved. In particular, as one may observe, SERICA\_1 gives always better throughput than SERICA\_2 and ERICA. Note that:

- SERICA\_2 achieves better throughput than the ERICA/ERICA+ in cases where  $p$  takes moderate values. Overall, using SERICA\_ω the throughput achieved is always more than 79%.
- In case the probability  $p$  is “arbitrary”, (namely,  $p$  varies from slot to slot during a session) SERICA\_1 performs better than SERICA\_2, whereas SERICA\_2 performs better than ERICA/ERICA+. This case may correspond to the unpredictable ABR traffic environment, in which sudden changes of the input rate in the source causes unexpected behavior in the system performance.
- In terms of the queue length, or of the queuing delay, SERICA\_2 seems to achieve very low values for all  $p$ .
- Although SERICA\_1 is always better than SERICA\_2, in terms of throughput, there are cases where SERICA\_2 should be preferred, particularly when the buffer availability is low and a new session is to be established

(a)

p	Cells Generated at source	$\Sigma(ACR \cdot h)$ in the source			$\Sigma(ER \cdot h)$			Average queuing delay in the buffer of					
								switch_1			switch_2		
		(1)	(2)		(3)			(4)			(5)		
	SER-2	SER-1	ER	SER-2	SER-1	ER	SER-2	SER-1	ER	SER-2	SER-1	ER	
0.9	32427	28026	31533	29313	28825	32481	29417	.00002	.00006	.11788	.00001	.00005	.00312
0.8	28830	23751	27962	25104	25709	30101	25292	.00003	.00013	.18781	.00002	.00011	.00593
0.7	25084	20416	24417	21183	23932	27932	25292	.00005	.00022	.25275	.00003	.00018	.00658
0.6	21458	17618	20848	17342	23281	25584	21442	.00007	.00032	.23776	.00004	.00030	.00967
0.5	17786	14815	17551	13690	22631	24762	13978	.00009	.00043	.24376	.00006	.00037	.01251
0.4	14235	11841	14204	10140	22436	25784	10444	.00013	.00051	.24575	.00008	.00041	.01443
0.3	10736	8922	10736	6776	22332	27824	6979	.00018	.00050	.24176	.00012	.00045	.01756
0.2	7118	6125	7118	3536	23123	30166	3686	.00029	.00054	.14785	.00020	.00042	.01600
0.1	3614	3455	3614	1282	26172	32525	1340	.00049	.00055	.06094	.00036	.00041	.01693
arbitrary	18661	14825	17961	7994	24436	30550	10417	.00008	.00020	.58541	.00003	.00010	.01239

(b)

p	Cells Lost at source			non-conformed cells counted on buffer of						Cells arrived to destination			Throughput (%)		
				switch_1			switch_2								
	(6)=(1)-(2)			(7)			(8)			(9)			(10)		
	SER-2	SER-1	ER	SER-2	SER-1	ER	SER-2	SER-1	ER	SER-2	SER-1	ER	SER-2	SER-1	ER
0.9	4401	894	3114	2	0	0	0	0	0	28024	31533	29313	86	97	90
0.8	5079	868	3726	1	0	0	0	0	0	23750	27962	25104	82	97	87
0.7	4668	667	3901	13	0	0	0	0	0	20403	24417	21183	81	97	84
0.6	3840	610	4116	15	14	0	0	8	0	17603	20826	17342	82	97	81
0.5	2971	235	4096	14	143	0	0	103	0	14801	17305	13690	83	97	77
0.4	2394	31	4095	68	351	0	2	269	0	11771	13584	10140	83	95	71
0.3	1814	0	3960	76	408	0	21	352	0	8825	9976	6776	82	93	63
0.2	993	0	3582	185	467	2	86	303	0	5854	6348	3534	82	89	50
0.1	159	0	2332	370	437	3	233	282	1	2852	2895	1278	79	80	35
arbitrary	3836	700	10667	111	359	82	0	146	11	14687	17456	7901	79	94	42

Table 3: (a), (b) The 2 hop, single VC model, in non-WCT ( $\tau=100 \mu\text{sec}$ , Link Rate = 155.52 Mbps).

In particular, SERICA\_2 algorithm uses in average lower queue length than the SERICA\_1, because the parameter  $\omega$  introduced in order to minimize the buffer occupancy at the switches. Thus, whilst the SERICA\_2 versus SERICA\_1 achieves very low queuing delays, it increases the non conformed cells at the source. The overall drawback of this disadvantage is small, since the throughput is almost the same for both of these cases. Finally, the most interesting result is shown through the variable case of  $p$  (in the Table 2 is depicted as varied). 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 situation is the most realistic 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. As it appears, during the whole session the SERICA\_2 algorithm performs better than ERICA, but SERICA\_1 performs better around its operational point, 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 ERICA algorithm shows an unpredictable behavior did not cause of the increase in the average queues of the buffers, neither because of the number of the non-conformed cells, but mainly, because of the number of cells rejected in the source.

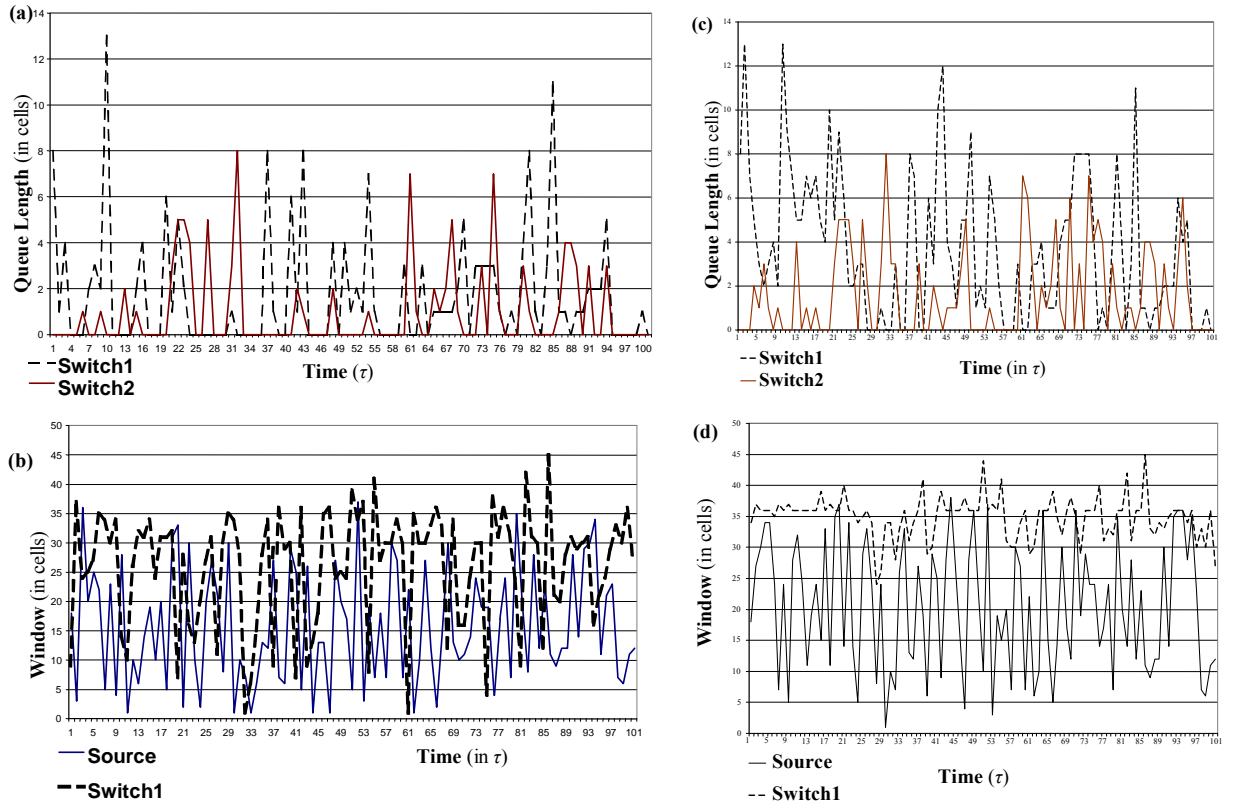


Figure 8: (a),(c) The queue length of the switches 1 and 2 and (b), (d) the proposed from the switch1 and actually implemented windows by the source, using SERICA\_2 and SERICA\_1 correspondingly. ( $PCR=36$  cells/ $\tau$ ,  $MCR=1$  cell/ $\tau$  and  $h = \tau = 1$ ).

The simulation above is used also to show the operation of the window mechanism in the

network configuration of Figure 7. The results produced using the SERICA\_2 and the SERICA\_1, are depicted in Figure 8 (cases a, c and b, d respectively). Figures 8a and 8c show that using the SERICA\_2, the queue length (in cells) of each of the switches 1 and 2 over the time is lower in average than that using the SERICA\_1. Further, Figures 8b and 8d show 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. As one may see, the proposed window of the switch 1 via the SERICA\_2 presents the expected bigger amplitude (corresponded to the  $\varphi(qpr_n, qst)$  for  $\omega = 2$ ) in the window, than that produced using the SERICA\_1 .

## 5. CONCLUSIONS

In this work we extended the SERICA and derived the SERICA\_ $\omega$  scheme. This new scheme examined through simulation to regulate the flow of the ABR traffic in an ATM network. SERICA\_ $\omega$  derivation is based on a network configuration consisted of  $M$  switches in tandem, with multiple VCs, allowing the VS/VD property. The analysis used a hop-by-hop strategy in which the cross traffic immediately leaves the switch.

SERICA\_ $\omega$  was developed to dynamically adjust the VS window size, by regulating its emission rate, through the ERF of the BRM cells sent from the VD. The  $ACR$ ,  $MACR$ , and  $q$  were modeled as fluids and as a result the discrete model was presented as a non-linear system of ODEs. Thus, for every case of  $\omega$  and in steady state it is totally stable. The above system of ODEs was solved using a simple numerical method. Derivation of the appropriate family  $f(q)$  for the SERICA\_ $\omega$  was left for the future.

Extended simulation results on a simple network configuration proved the robustness of the new scheme i.e., that the achieved throughput has always been greater than 79% of the initially generated cells at the source and that the queuing delay has been minimized, in the unpredictably varied ABR environment, as well as, in the predictable VBR environment. The most important is that the new scheme may be used from the network administrator as a “gear box” to initialize a new session with a low buffer availability and then to serve the session in a “safe life” operating procedure.

## REFERENCES

1. Arnold V. I. (1978), “*Ordinary Differential Equations*”, in Arnold V. I. (ed.) MIT Press edition, ISBN 0-262-51018-9.
2. ATM FORUM (1996), “Traffic Management Spec. V. 4.0”. <ftp://ftp.atmforum.com/pub/approved-specs/af-tm-0056.000.ps>, April 1996.
3. ATM FORUM (1996), “Performance Testing Spec.”. 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>
4. FUJITSU (2000), “Real Time Cell Flow Processing TM”, *A White Paper*, [http://www.nexen.com/new\\_web/wprealtime.html](http://www.nexen.com/new_web/wprealtime.html).
5. Jain R., Kalyanaraman S., Goyal R., Fahmy S. and Fang Lu (1995), “ERICA+: Extensions to the ERICA switch algorithm”, *ATM Forum/95/1346*, Oct. 1995.
6. Kalyanaraman S. (1997), “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.

7. Kalyanaraman S., Jain R., Fahmy S., Goyal R., and Vandalore B. (2000), "The ERICA Switch Algorithm For ABR Traffic Management in ATM Networks", *IEEE/ACM Trans. Net.*,8(1) (2000), pp. 87-98.
8. J. Mignault, A. Gravey and C. Rosenberg, (1996), "A survey of straightforward statistical multiplexing models for ATM networks", *Telecomm.Sys.* 5 (1996) pp.177-208.
9. Mousadis G., Tsiligirides T. and Bekakos M. (1996), Traffic Considerations for the Performance Evaluation of the ATM Networks, *Proc. HERMIS- '96*, pp. 289 – 299.
10. Mousadis G. (2001), "Flow Control Algorithms for the High Speed Computer Networks", *Ph. D. Dissertation, Dept. of General Sciences, Agricultural University of Athens*, September 2001.
11. Mousadis G. and Tsiligirides T. (2002), "A Simple Explicit Rate Identification Congestion Avoidance (SERICA) algorithm to support some TCP Differentiated Services over the ABR traffic", *Com Com Journal*, 2041 paper accept.