



ELSEVIER

Computer Communications 19 (1996) 539–552

computer  
communications

# Numerical analysis of some basic fluid communication models via parallel block methods

T. Tsiligrides\*

*Informatics Laboratory, Department of General Sciences, Agricultural University of Athens, Iera Odos 75, Athens 118 55, Greece*

Received 17 February 1995; revised 2 June 1995

## Abstract

In this work we propose a general method for the solution of some basic delayed feedback schemes used in long haul, high speed data transport. In such cases, simple batch Poisson models do not describe packet delays well, while the propagation delay is now becoming a major factor. Two basic virtual circuit networks of balanced form are examined; the single-hop network which aggregates many virtual circuits in parallel, and the multi-hop virtual circuit network having  $M$  nodes in tandem. Using well known adaptive algorithms to dynamically adjust the window size, the above networks are presented as linear systems of some delay differential equations in which the rate of transmission and the queue occupancy are modelled as fluids. Although these systems are locally unstable (in a Liapounov sense), we identify the appropriate scale for the parameters so that the systems will perform near their optimal theoretical values for a wide range of speeds. In addition, we propose a general method for their numerical solution which in reality are large and complex. The approach is based on parallel block methods that are used to solve the systems of the ordinary differential equations in which the original systems of the delay differential equations have been transformed. The basic theory underlying the parallel block methods is developed and numerical stability of low order is deduced.

**Keywords:** Balanced networks; Virtual circuits; Delay differential equations; Parallel block methods; Stability

## 1. Introduction

It is well known that the ATM (Asynchronous Traffic Mode), an internationally standardised basis for B-ISDN (Broadband-Integrated Services Digital Network), is a technique best suited to satisfy the requirements of integrated broadband networks. It is the first switching technique capable of supporting both the circuit switching and packet switching approaches within a single integrated switching mechanism. Obviously, any attempt to introduce ATM for data communication must take into consideration existing communication protocols which have been implemented in a wide variety of existing systems. Furthermore, it must also be ensured that the existing data networks are capable of operating as an access to ATM networks. This applies especially to LANs along with their inter working units, such as bridges, routers and gateways. However, these devices will soon be able to convert LAN protocols directly to ATM in order that ATM-based user-network interfaces can be used.

ATM is a switching and multiplexing technique that

uses constant length packets (cells) as the basic data units. An ATM cell, as defined by ITU (formerly CCITT) recommendation I.361, contains 48 bytes of user information and 5 bytes of control information. The control part, or 'header', consists primarily of VPI/VCI (Virtual Path/Channel Identifiers). An ATM virtual channel, as it is available to a user, is uniquely defined by a combination of VPI and VCI in every transmission system along the path taken by all cells belonging to the same connection. Groups of virtual channels with the same VPI value can be transferred as a virtual path without taking the VCI into consideration. ATM can scale from small multiplexers to very large switches in both an aggregate capacity and in the number of access ports. It can accommodate access ports from low speeds (1.5 Mbps and 6 Mbps) to very high speeds (2.4 Gbps). It is designed to handle multimedia traffic and will be deployed in the public network as the future B-ISDN. Obviously, besides the protocols controlling the data transfer traffic control functions, supporting statistical multiplexing plays a key role in exploiting the potential of ATM networks.

Typical data applications that are currently supported

\* Email: tsili@auadec.aua.ariadne-t.gr

by existing networks, together with a number of future applications that will most likely gain in importance with the introduction of high speed data networks, include real-time video (MPEG), low rate interactive data applications, electronic mail, browsing in graphics databases, animation of graphics, software download, etc. Thus, traffic is often bursty or belongs to multiple classes, such as packetised voice, short data messages, bulk files, etc. However, since the traffic characteristics are unknown, the vast majority of these data applications are incapable of predicting their own bandwidth requirements. Moreover, many of these data applications are relatively tolerant against delays causing quite opposite effects on sojourn time, namely the time a customer spends in the system.

To prevent or reduce overload situations in ATM networks, and to optimise network resources, it has been proposed to use a closed loop feedback control mechanism that allows the network to control the cell emission process at each source. Each virtual connection must have an independent control loop, since each connection may follow a different path through the network. Using feedback control, sending terminals contributing to congestions can be forced to reduce the speed at which they are sending. Two classes of feedback schemes have been proposed. The first is a credit based, link-by-link, per connection window flow control mechanism, while the second is a rate based, end-to-end load control mechanism. Credit based mechanisms are similar to existing data networks (e.g. the sliding window procedures used in X.25 networks). With rate based mechanisms, the maximum rate at which a source may send can be dynamically adapted, depending on network load conditions.

ITU recommendation I.371 provided the possibility to support a rate based reactive load control mechanism by means of EFCI (Explicit Forward Congestion Indication). Congested nodes may set a congestion indication in the cell header of cells that pass congested nodes. The receiving terminal will then relay the congestion information to the sending terminal using a higher layer protocol. It has also been proposed that the receiving terminal could use resource management cells for this purpose. Resource management cells indicating congestion could also be sent directly from congested nodes to the terminal, thus providing BECI (Backward Explicit Congestion Indication).

The major problem of reactive close loop control in high speed wide area networks is the large amount of data (equal to the delay-bandwidth-product) that may be under way in the network. In wide area high-speed networks, the fibres used for transmission become large buffers that cannot be easily controlled. Note that a multiplexer may be fed by a number of high speed links, each of which can store a large amount of data. For example, 0.28 Mbytes of data can be on its way in a

fibre of 3000 km in length operating at 150 Mbps. Under unfavourable conditions, a congested multiplexer indicating congestion may have to buffer many Mbytes before the congestion indication will show any effect. On the other hand, the load situation may have changed by the time the control mechanism has taken effect. In particular, this is a problem when round trip delays are large. Thus, with the ever increasing speeds in data transmission the round trip propagation delay is now becoming a major factor. A further example in long haul, high speed data transport is the case of a virtual circuit network in which the service is always available, and in which sequenced delivery of packets in either direction between the two end users is ensured. Assuming the propagation delay is neglected, the sliding window mechanism to control the congestion may be applied [1,2]. However, if the propagation delay cannot be neglected, new design procedures have to be established. Much work to date on flow control in high-speed networks may be found elsewhere [3–8].

Propagation delays do exist in geographically dispersed long haul data networks. For example, in US the round trip propagation delay through a fibre link is approximately 47-ms (one round trip propagation delay is approximately the distance across the continental US divided by the speed of light). With a transmission rate of 45 Mbits/s and packet size of 1 Kbyte a representative service rate  $\mu$  of 264 for the various nodes is obtained. At the other end, with the optical transmission rate of 1.7 Gbits/s, and keeping the other two parameters constant,  $\mu$  is about 9989. Clearly, high transmission rates and non-scalable propagation delays give rise to large  $\mu$ . An interesting asymptotic analysis has been developed [9] to face the problem of scaling. As it appears, keeping delay and packet loss within acceptable levels, high efficiency cannot be provided using only static (non adaptive to the variations of traffic) control schemes. Thus, dynamic schemes that adaptively conform to the traffic environment are always used. In such schemes the large propagation delays and the high speeds must be included in the analysis. Several dynamic algorithms for congestion control/avoidance already exist in the literature [10–16], but the following is of interest here.

Fendic et al. [10] studied the stability properties of a large class of dynamic control algorithms with delayed feedback, as well as how parameters should scale with the speed. The analysis was restricted to a single session and a single bottleneck node to avoid complexities (see Fig. 1). The network has been presented as a system of DDEs (Delay Differential Equations) in which the rate of transmission and the queue occupancy are modelled as fluids. It is assumed that the acknowledgements carry information about the state of the queue at the bottleneck node, and that this information is used to control the transmission rate through a control function which may not be linear. Note that, in general, the standard

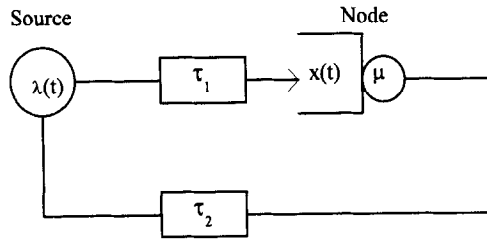


Fig. 1. Single-hop, virtual circuit, fluid communication model with propagation delay.

theory of DDEs does not treat the class of systems resulting from the above modelling.

Fendick et al. used the recursive relation proposed by Mitra and Seery [11], resulting in a linear control function. The queue size is estimated through measurements on packet acknowledgement times. Thus, using linear approximation (indirect Liapounov theorem), the system is proved locally unstable in the neighbourhood of the desired operating point. However, with the appropriate scaling of the parameters associated with the model, the system performs near the optimal theoretical value for a wide range of speeds. Note that, although the above results are quite interesting in terms of model stability, a solution method for the resulting systems of DDEs has not yet been proposed. This is particularly important in real situations where the networks are large and more complex.

In this work we extend the above results of stability in the two representative cases of tandem (Fig. 2) and parallel (Fig. 3) networks. Thus, using the recursive scheme of Mitra and Serry, the above systems are proved locally unstable (in a Liapounov sense) near their desired operating point. Note that in these particular schemes, one may use the analysis proposed by Fendic et al. [10] to identify the appropriate scaling for the parameters and to make each of the above systems perform near their optimal theoretical values for a wide range of speeds. However, in general cases, where the resulting DDE systems are more complex and large, stability in a Liapounov sense is difficult to prove. In addition, a solution method of such DDE systems has not been investigated so far. For this reason, we propose a general method to derive an approximate numerical solution. The approach is based

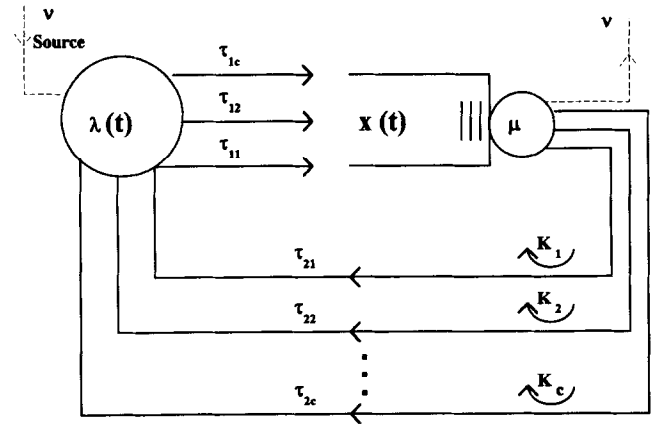


Fig. 3. Multiple virtual circuits with various propagation delays and individual windows over a single hop. The propagation delay and window of the  $j$ th virtual circuit are  $K_j$  and  $\tau_j$ . The cross traffic path is shown by the dotted lines.

on the so-called 'theory of parallel blocks' [17–21], and it is fully discussed. This method is used to solve the system of ODEs (Ordinary Differential Equations), in which the original system of DDEs can be transformed. The basic theory underlying the parallel block methods for the numerical solution of the system of ODEs is also developed. Finally, numerical stability is examined by deducing two and three dimensional, A-stable parallel block methods of second, third and fourth orders.

The paper is organised as follows. In section 2 we briefly review the single node model (Fig. 1) which is based on a set of DDEs. This helps in understanding the formulation of the extended models. We provide the specific equations that approximate the adaptive recursive schemes of Mitra and Seery [11] as continuous deterministic models. We then conclude that the balanced networks proposed in Figs. 2 and 3 can be modelled in a similar way as a set of DDEs, and we derive the specific equations that approximate the above adaptive recursive schemes. It may be observed that both models have strong similarities in their representation as fluid models and in this sense we study their Liapounov stability behaviour. The parallel block method for solving the resulting systems of ODEs is introduced and discussed in section 3. Numerical stability is also presented in this section, where the

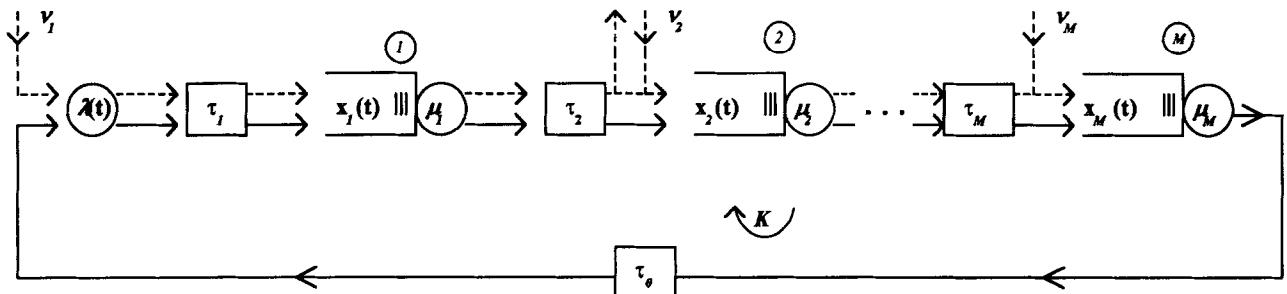


Fig. 2. Model of a multi-hop virtual circuit with propagation delays and sliding window control. The route of the virtual circuit is shown by the solid line.  $M$  is the number of hops and  $K$  is the window size. The dotted lines represent cross traffic streams.

construction of A-stable parallel block methods of a low order of consistency are developed. Lastly, in section 4 we state our conclusions and discuss the prospects of this research.

## 2. The models

### 2.1. The basic model

The basic model we consider is presented in Fig. 1. This is a virtual network consisting of a single session between a source and a destination. It has a single bottleneck node, where  $x(t)$  denotes the amount of data in queue, and  $\mu$  denotes the constant rate of service at the queue. Data experiences some propagation delay  $\tau_1$  before it reaches a bottleneck queue, and acknowledgements experience some propagation delay  $\tau_2$  before returning to the source ( $\tau = \tau_1 + \tau_2$ ). We assume that the source has an infinite amount of data to send, and that it transmits at a rate  $\lambda(t)$  during the life of a session. With this assumption in mind, we isolate the issue of an adaptive control mechanism and may derive the corresponding results for large but finite amounts of data based on the transient and long run behaviour of the system for a source with infinite amounts of data. Note that this model does not cover all the types of traffic, but is restricted only to large file transfers. However, large transfers represent a significant fraction of traffic on a network, which also has different effects on network performance than the other types of transfer, interactive traffic, for example. Clearly, the objective is to determine  $\lambda(t)$  such that the source gets as much throughput from the network as possible without causing excessive congestion.

For this simple model, the rate of transmission  $\lambda(t)$  at the source and the occupancy  $x(t)$  have been modelled as fluids. Acknowledgements carry information about the state of the queue at the bottleneck node, and this information is used to control the rate  $\lambda(t)$  through a control function  $F(x(t), \lambda)$ . The mathematical equations that govern the system are the following:

$$\frac{d}{dt}x(t) = \begin{cases} \lambda(t - \tau_1) - \mu & \text{if } \lambda(t - \tau_1) - \mu > 0 \\ & \text{or } x(t) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

$$\frac{d}{dt}\lambda(t) = \begin{cases} F(x(t - \tau_2), \lambda(t))\lambda(t - \tau) & \text{if } x(t - \tau_2) = 0 \\ F(x(t - \tau_2), \lambda(t))\mu & \text{otherwise.} \end{cases} \quad (2)$$

The above traffic model best suits an environment where the time to transmit a packet is small compared to the total transfer time, and does not vary significantly from packet to packet (e.g. ATM networks, or large transfers

through typical data transport protocols like TCP). Note that Eq. (1) describes the changes at the node, while Eq. (2) describes the adaptation strategy used by the source. Based on the latest feedback information, the transmission rate changes accordingly and the adaptation function  $F(x(t - \tau_2), \lambda(t))$  determines the sign and the magnitude of change to  $\lambda(t)$ . Feedback information on  $x(t - \tau_2)$  is received through acknowledgement packets. Thus, an estimate of  $x(t - \tau_2)$  is obtained through measuring the round-trip times, or through an EFCI bit set by the bottleneck node on a packet that experiences congestion and then copies into a BECI bit of an acknowledgement packet. Provided that acknowledgement packets do not experience queueing, acknowledgements are received at the same rate as they were sent. Otherwise, acknowledgements are received at the queue service rate  $\mu$ . Thus, the faster the acknowledgements are received, the faster the system adapts.

However, to solve the above system for  $x(t)$  and  $\lambda(t)$ , it is necessary to provide the control function  $F(x(t - \tau_2), \lambda(t))$ . Since the standard theory of DDEs does not treat the above form of systems, Fendic et al. [10] developed some analytical tools showing that the solutions to systems (1) and (2) are uniformly bounded for a large class of initial conditions. Note that for such a system the solution path is not defined as a single point but as a function defined over a time interval of length  $\tau$ . They also derive tight bounds for a region in terms of the adaptation strategy used, the propagation delays and the bandwidth at the bottleneck node. However, they did not provide numerical solutions of such systems. As we will see, this is a very important issue, particularly in real situations where the networks are much more complex, and our contribution is to provide a general method for the numerical solution of such systems.

For example, one may model the adaptive scheme of Mitra and Seery [11] given by the recursive relation:

$$K_{n+1} = K_n - A[(R_n/\tau - 1)\sqrt{K_n} - b] \quad (3)$$

in terms of Eqs. (1) and (2). In the above,  $K_n$  is the size of the window, namely, the amount of outstanding unacknowledged data at the  $n$ th acknowledgement,  $R_n$  is the round trip delay of the  $n$ th packet,  $b = \sqrt{M}$ , where  $M$  is the number of nodes in the network (here  $M = 1$ ) and  $A$  is a calibrating parameter obtained by simulation. Since the propagation delays dominate the round-trip delay for high speed networks, we may ignore the queueing delay when calculating the window throughput. Thus, by approximating the rate of transmission at the source by the throughput of the widening system, namely  $\lambda(t_n) = K_n/R_n = K_n/\tau$ , we may approximate the adaptation strategy by:

$$\frac{d}{dt}\lambda(t) \approx \frac{\Delta\lambda}{\Delta t} = \frac{K_{n+1} - K_n}{\tau\Delta t} = -\frac{A[(R_n/\tau - 1)\sqrt{K_n} - 1]}{\tau\Delta t}$$

where

$$\Delta t = \begin{cases} \frac{1}{\lambda(t - \tau_1 - \tau_2)} & \text{if } x(t - \tau_0) = 0 \\ 1/\mu & \text{otherwise.} \end{cases}$$

Therefore, by setting  $t = 1$  and  $\gamma = -A$  we may obtain the following DDE system:

$$\frac{d}{dt}x(t) = \begin{cases} \lambda(t - \tau_1) - \mu & \text{if } \lambda(t - \tau_1) - \mu > 0 \\ & \text{or } x(t) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

$$\frac{d}{dt}\lambda(t) = \begin{cases} \gamma \left( \frac{x(t - \tau_2)}{\mu} \sqrt{\lambda(t)} - 1 \right) \lambda(t - \tau_1 - \tau_2) & \text{if } x(t - \tau_2) = 0 \\ \gamma \left( \frac{x(t - \tau_2)}{\mu} \sqrt{\lambda(t)} - 1 \right) \mu & \text{otherwise.} \end{cases} \quad (5)$$

Note that the queueing delay may be obtained through the round-trip delay measurements as  $R_n - \tau = x(t - \tau_1)/\mu$ .

The above system may be simplified further by assuming  $\lambda = \mu + O(\sqrt{\mu})$ . In this case we obtain:

$$\frac{d}{dt}x(t) = \begin{cases} \lambda(t - \tau_1) - \mu & \text{if } \lambda(t - \tau_1) - \mu > 0 \\ & \text{or } x(t) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

$$\frac{d}{dt}\lambda(t) = \begin{cases} \gamma \left( \frac{x(t - \tau_2)}{\sqrt{\mu}} - 1 \right) \lambda(t - \tau_1 - \tau_2) & \text{if } x(t - \tau_2) = 0 \\ \gamma \left( \frac{x(t - \tau_2)}{\sqrt{\mu}} - 1 \right) \mu & \text{otherwise.} \end{cases} \quad (7)$$

#### Remark 1

For the above scheme it can be proved [10], that the points  $(0, 0)$  and  $(\sqrt{\mu}, \mu)$  are equilibrium points. Note that a point is defined as an equilibrium point for a system of DDEs if the solutions remain forever at that point after having been there for at least an amount of time  $\tau = 1$ . Thus, for all well behaved initial conditions, the solution path must enter a neighbourhood of the equilibrium point and can never leave. However, using linear approximation, the equilibrium point  $(\sqrt{\mu}, \mu)$  is locally unstable in the neighbourhood of the desired operating point. In addition,  $\gamma \sim 1/\sqrt{\mu}$  and  $\lambda$  converges to a region bounded by  $\mu \pm O(\sqrt{\mu})$ . Note that similar results hold when  $\gamma$  is general, and the control function  $F(x)$  has the linear form  $F(x) = \gamma f(x/\sqrt{\mu} - 1)$ , where  $f(x)$  is monotone decreasing and  $1 \leq x \leq \infty$ .

In order to solve the above systems (4) and (5), one has to numerically represent the system of DDEs to an equivalent system of ODEs. The procedure is as follows.

Setting:  $z_1(t) = \lambda(a_1(t))$ ,  $z_2(t) = x(a_2(t))$  and  $z_3(t) = \lambda(a_3(t))$  with:  $a_1(t) = t - \tau_1$ ,  $a_2(t) = t - \tau_2$  and  $a_3(t) = t - \tau_1 - \tau_2 = t - 1$ , the resulting system is transformed to the following ODEs system:

$$\frac{d}{dt}x(t) = \begin{cases} z_1(t) - \mu & \text{if } z_1(t) - \mu > 0 \text{ or } x(t) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

$$\frac{d}{dt}\lambda(t) = \begin{cases} \gamma \left( \frac{z_2(t)}{\mu} \sqrt{\lambda(t)} - 1 \right) z_3(t) & \text{if } z_2(t) = 0 \\ \gamma \left( \frac{z_2(t)}{\mu} \sqrt{\lambda(t)} - 1 \right) \mu & \text{otherwise.} \end{cases} \quad (9)$$

In this case, the resulting system is simple and can be solved using well known serial methods.

#### 2.2. The tandem model

We now consider a more general case presented by the network in Fig. 2. This is a multi-hop virtual circuit system having  $M$  nodes in tandem. We assume that the network is balanced, namely:

$$\lambda = \mu_1 - \nu_1 = \mu_2 - \nu_2 = \dots = \mu_M - \nu_M, \quad (10)$$

where  $\mu_i$  is the service rate and  $\nu_i$  is the exogenous cross traffic rate at node  $i$ . Here the parameter  $\lambda$  is the so-called reduced nodal processing rate.  $x_i(t)$  represents the amount of data at node  $i$ ,  $\tau_i$  is the propagation delay from the corresponding source to the bottleneck node  $i$  and  $\tau_0$  is the propagation delay from the last bottleneck node  $M$  to the receiver and back to the source at the beginning ( $\tau_0 + \tau_1 + \dots + \tau_M = \tau$ ). Thus, for  $j = 1, 2, \dots, M$

$$\frac{d}{dt}x_j(t) = \begin{cases} \lambda(t - \tau_j) - \mu_j & \text{if } \lambda(t - \tau_j) - \mu_j > 0 \\ & \text{or } x_j(t) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

$$\frac{d}{dt}\lambda(t) = \begin{cases} F(x_M(t - \tau_0), \lambda(t)) \lambda(t - \tau) & \text{if } x_M(t - \tau_0) = 0 \\ F(x_M(t - \tau_0), \lambda(t)) \mu_M & \text{otherwise.} \end{cases} \quad (12)$$

Mitra [9] has examined the basic mechanism of sliding windows for congestion control of a virtual circuit applied in such high-speed balanced networks using the same recursive algorithm to estimate the queue size. In this balanced network, the difference between the service rate  $\mu_i$  and the rate of exogenous, cross traffic rate  $\nu_i$  in each node  $i$ , equals  $\lambda$ . Note that  $M$  is the number of hops in the network while  $K$  is the window size. The balanced network usually results from an original network by deleting all but the slowest bottleneck nodes. Thus, all the reduced nodal processing rates are almost equal. Mitra formulated and solved the problem of the optimal design of windows and obtained formulas for basic quantities, such as the throughput, delay and moments

of packet queues. He showed that the optimum window size is  $\lambda + O(\sqrt{\lambda})$ . All his results are asymptotic and are based on a design equation which relates the mean response to the window size. This equation is used to adjust the window based on measurements. It is interesting to note that the asymptotic theory developed can be extended to non-balanced networks as well.

Again, one may model the adaptive scheme of Mitra and Serry [11] in terms of Eqs. (11) and (12). The recursive relation is now given by:

$$K_{n+1} = K_n - A[(R_n/\tau - 1)\sqrt{K_n} - \sqrt{M}], \quad (13)$$

where  $K_n$ ,  $R_n$  and  $A$  have been defined previously. From the above it is clear that the rate of acknowledgements will also dictate the rate of adaptation. Thus, when packets find all the queues empty, acknowledgements return at the same rate as the packets that are being acknowledged when transmitted.

Alternatively, when packets find any of the queues non-empty, it is assumed that acknowledgements return at the rate at which packets are served at the  $M$ th queue. Note that, since the propagation delays dominate the round trip delay, the queueing delays at various nodes may be ignored when calculating the window throughput. Therefore,  $\lambda(t_n) = K_n/R_n = K_n/\tau$ . However, the queueing delays may be obtained through the round trip delay measurements at each node; namely at the  $M$ th node:  $R_n - \tau = x_M(t - \tau_0)/\mu_M$ .

Thus, we can approximate the adaptation strategy by:

$$\frac{d}{dt}\lambda(t) \approx \frac{\Delta\lambda}{\Delta t} = \frac{K_{n+1} - K_n}{\tau\Delta t} = \frac{-A[(R_n/\tau - 1)\sqrt{K_n} - \sqrt{M}]}{\tau\Delta t}$$

where the time between adaptations is determined by:

$$\Delta t = \begin{cases} \frac{1}{\lambda(t - \tau_1 - \tau_2 - \dots - \tau_M - \tau_0)} & \text{if } x_M(t - \tau_0) = 0 \\ 1/\mu_M & \text{otherwise.} \end{cases}$$

Taking into account Eq. (10) and setting  $t = 1$  and  $\gamma = -A$  for all  $j = 1, 2, \dots, M$ , we obtain the following DDEs system:

$$\frac{d}{dt}x_j(t) = \begin{cases} \lambda(t - \tau_j) - \mu_j & \text{if } \lambda(t - \tau_j) - \mu_j > 0 \\ & \text{or } x_j(t) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (14)$$

$$\frac{d}{dt}\lambda(t) = \begin{cases} \gamma \left( \frac{x_M(t - \tau_0)}{\mu_M} \sqrt{\lambda(t)} - \sqrt{M} \right) \times \lambda(t - \tau_1 - \dots - \tau_M - \tau_0) & \text{if } x_M(t - \tau_0) = 0 \\ \gamma \left( \frac{x_M(t - \tau_0)}{\mu_M} \sqrt{\lambda(t)} - \sqrt{M} \right) \mu_M & \text{otherwise.} \end{cases} \quad (15)$$

The above system has  $M + 1$  equations and  $M + 1$  unknowns, namely, the  $\lambda(t)$  and  $x_j(t)$ ;  $j = 1, \dots, M$ . It may be simplified further by assuming that  $\mu_j$  are all equal with  $\mu$ , a representative service rate, and  $\lambda = \mu + O(\sqrt{\mu})$ . In this case we obtain:

$$\frac{d}{dt}x_j(t) = \begin{cases} \lambda(t - \tau_j) - \mu & \text{if } \lambda(t - \tau_j) - \mu > 0 \\ & \text{or } x_j(t) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (16)$$

$$\frac{d}{dt}\lambda(t) = \begin{cases} \gamma \left( \frac{x_M(t - \tau_0)}{\sqrt{\mu}} - \sqrt{M} \right) \times \lambda(t - \tau_1 - \dots - \tau_M - \tau_0) & \text{if } x_M(t - \tau_0) = 0 \\ \gamma \left( \frac{x_M(t - \tau_0)}{\sqrt{\mu}} - \sqrt{M} \right) \mu & \text{otherwise.} \end{cases} \quad (17)$$

#### Remark 2

For the above scheme one may show, using linear approximation (indirect Liapounov theorem), that the equilibrium point:

$$\bar{\mathbf{e}} = (\lambda, x_1, x_2, \dots, x_M)^T = (\mu, \sqrt{M\mu}, \sqrt{M\mu}, \dots, \sqrt{M\mu})^T$$

is locally unstable in the neighbourhood of the desired operating point. In addition,  $\gamma \sim 1/\sqrt{\mu}$  and  $\lambda$  converges to a region bounded by  $\mu \pm O(\sqrt{\mu})$ . Note that similar results hold when  $\gamma$  is general and the control function is of the linear form:  $F(x_M) = \gamma f(x_M/\sqrt{\mu} - \sqrt{M})$ , where  $f(x_M)$  is monotone decreasing and  $1 \leq x_M \leq \infty$ .

To solve the above system (14) and (15), one has to numerically represent the system of DDEs to an equivalent system of ODEs. The procedure is as follows:

$$\frac{d}{dt}x_j(t) = \begin{cases} z_j(t) - \mu_j & \text{if } z_j(t) - \mu_j > 0 \\ & \text{or } x_j(t) > 0 \quad \forall j = 1, 2, \dots, M \\ 0 & \text{otherwise} \end{cases} \quad (18)$$

$$\frac{d}{dt}\lambda(t) = \begin{cases} \gamma \left( \frac{z_{M+1}(t)}{\mu_M} \sqrt{\lambda(t)} - \sqrt{M} \right) z_{M+2}(t) & \text{if } z_2(t) = 0 \\ \gamma \left( \frac{z_{M+1}(t)}{\mu_M} \sqrt{\lambda(t)} - \sqrt{M} \right) \mu_M & \text{otherwise,} \end{cases} \quad (19)$$

where  $z_j(t) = \lambda(a_j(t))$ ;  $j = 1, \dots, M$ ;  $z_{M+1}(t) = x_M(a_{M+1}(t))$ ;  $z_{M+2}(t) = \lambda(a_{M+2}(t))$ , and  $a_j(t) = t - \tau_j$ ;  $j = 1, \dots, M$ ;  $a_{M+1}(t) = t - \tau_0$ ;  $a_{M+2}(t) = t - \tau_1 - \dots - \tau_M - \tau_0$ .

As one may observe, in case  $M$  is small, the resulting system is simple and can be solved using well known serial methods. However, in case  $M$  is large, some

more efficient methods have to be derived in order to provide a numerical stable solution. This problem will be faced in the subsequent sections.

### 2.3. The parallel model

We now proceed with the model presented in Fig. 3. This is a single node virtual circuit network with exogenous cross traffic rate  $\nu$  and service rate  $\mu$ . For the theoretical development, the network is assumed to be balanced, i.e.  $\lambda = \mu - \nu$ , with fixed service rate  $\mu$  and Poisson cross traffic rate  $\nu$ . Since the unit of time has been selected to be representative of the round trip propagation delay,  $\lambda$  is large. It is assumed that the virtual circuit aggregates many smaller virtual circuits, say  $C$ , following the same or different paths in the network with various propagation delays. Thus, for the  $j$ th virtual circuit, data experiences some propagation delay  $\tau_{1j}$  before it reaches the bottleneck queue, and acknowledgements experience some propagation delay  $\tau_{2j}$  before returning to the source with  $\tau_j = \tau_{1j} + \tau_{2j}$  for each  $j = 1, 2, \dots, C$ . Note that the  $\tau_j$ 's are assumed to be  $O(1)$  constants, so they do not scale with the large parameter  $\lambda$  under moderate usage regime.

Let  $\theta_j(t); j = 1, 2, \dots, C$ , denote a set of fractional allocation parameters of bandwidth in which  $\theta_j$  is the fraction of the total throughput which is to be allocated to the  $j$ th virtual circuit, that is,  $\theta_j > 0, j = 1, 2, \dots, C$ , with  $\sum_{j=1}^C \theta_j = 1$ . If we denote by  $\lambda_j(t), \mu_j$  and  $x_j(t)$  the transmission rate, the service rate and the occupancy of the  $j$ th virtual circuit, the mathematical equations that govern the system we are analysing are defined for each  $j = 1, 2, \dots, C$  by:

$$\frac{d}{dt}x_j(t) = \begin{cases} \lambda_j(t - \tau_{1j}) - \mu_j & \text{if } \lambda_j(t - \tau_{1j}) - \mu_j > 0 \\ & \text{or } x_j(t) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (20)$$

$$\frac{d}{dt}\lambda_j(t) = \begin{cases} F_j(x_j(t - \tau_{2j}), \lambda_j(t))\lambda_j(t - \tau_j) & \text{if } x_j(t - \tau_{2j}) = 0 \\ F_j(x_j(t - \tau_{2j}), \lambda_j(t))\mu_j & \text{otherwise.} \end{cases} \quad (21)$$

Clearly,  $\theta_j$  is allocated to the  $j$ th virtual circuit, and therefore:

$$\lambda_j(t) = \theta_j \lambda(t), \quad \mu_j = \theta_j \mu \quad \text{and} \quad x_j(t) = \theta_j x(t) \quad (22)$$

$$\lambda(t) = \sum_{j=1}^C \lambda_j(t), \quad \mu = \sum_{j=1}^C \mu_j \quad \text{and} \quad x(t) = \sum_{j=1}^C x_j(t). \quad (23)$$

Note that acknowledgements carry information about the state of the queue at the bottleneck node, and this information is used to control the rates  $\lambda_j(t)$  through the control functions  $F_j(x_j(t - \tau_{2j}), \lambda_j(t))$ .

The above system of DDEs consists of  $2C$  equations with  $3C$  unknowns, namely the  $x_j(t), \lambda_j(t)$  and  $F_j(x_j(t - \tau_{2j}), \lambda_j(t))$ . The solution should also satisfy Eqs. (22) and (23). To model the adaptive scheme of Mitra and Seery in terms of the above system of equations, one has to use the following recursive relations (see Ref. [11]):

$$K_{j,n+1} = K_{j,n} - a_j \left[ \left( \frac{R_{j,n}}{\tau_j} - 1 \right) \sqrt{K_{j,n}} - b_j \right] \quad (24)$$

$$\text{with } b_j = \sqrt{\frac{\frac{\theta_j}{\tau_j}}{\sum_{i=1}^C \frac{\theta_i}{\tau_i}}}.$$

In the above,  $K_{j,n}$  represents the window size of the  $j$ th virtual circuit, by means of the amount of outstanding unacknowledged data of the  $j$ th virtual circuit at the  $n$ th acknowledgement, and  $R_{j,n}$  is the round trip delay of the  $n$ th packet of the  $j$ th virtual circuit. The use of simulation has been suggested [11] to obtain the constants  $a_j, j = 1, 2, \dots, C$ , but as we shall see later, these constants may be calculated directly for the cases of interest. Using similar queueing arguments to those developed in the analysis presented earlier in this section, we may approximate the rate of transmission of the source by the throughput of the windowing system,  $\lambda_j(t_n) = K_{j,n}/R_{j,n} = K_{j,n}/\tau_j$ , where  $t_n$  is the time when the source receives the  $n$ th acknowledgement of the  $j$ th virtual circuit. Again, the queueing delays have been ignored when calculating the window throughput, since the propagation delays dominate the round-trip delay for the high speed networks we are studying. However, the queueing delay of each virtual circuit can be obtained through the round-trip delay measurements:

$$R_{j,n} - \tau_j = \frac{x_j(t - \tau_{2j})}{\mu_j}, \quad j = 1, 2, \dots, C$$

$$\text{or } \frac{R_{j,n}}{\tau_j} - 1 = \frac{x_j(t - \tau_{2j})}{\mu_j \tau_j}; \quad j = 1, 2, \dots, C.$$

In scheme (24) the source adapts every time an acknowledgement is received. Thus, the rate of acknowledgements dictates the rate of adaptation. Note that when packets find that the queue of the  $j$ th virtual circuit is empty, acknowledgements return at the same rate as the packets that are being acknowledged when transmitted. On the other hand, when packets find the queue of the  $j$ th virtual circuit non-empty, acknowledgements return at the rate at which packets are served at the bottleneck queue. Thus, we may approximate the adaptation

strategy for each  $j = 1, 2, \dots, C$  by:

$$\frac{d}{dt} \lambda_j(t) \approx \frac{\Delta \lambda_j}{\Delta t} = \frac{K_{j,n+1} - K_{j,n}}{\tau_j \Delta t}$$

$$= \frac{\gamma_j \left[ \frac{x_j(t - \tau_{2j})}{\mu_j \sqrt{\tau_j}} \sqrt{\lambda_j(t)} - b_j \right]}{\tau_j \Delta t}$$

where:

$$\Delta t = \begin{cases} \frac{1}{\lambda_j(t - \tau_{1j} - \tau_{2j})} & \text{if } x_j(t - \tau_{2j}) = 0 \\ 1/\mu_j & \text{otherwise} \end{cases}$$

and  $\gamma_j = -a_j$ .

Using the above, we may re-write Eqs. (20) and (21) as:

$$\frac{d}{dt} x_j(t) = \begin{cases} \lambda_j(t - \tau_{1j}) - \mu_j & \text{if } \lambda_j(t - \tau_{1j}) - \mu_j > 0 \\ & \text{or } x_j(t) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (25)$$

$$\frac{d}{dt} \lambda_j(t) = \begin{cases} \gamma_j \left[ \frac{x_j(t - \tau_{2j})}{\mu_j \sqrt{\tau_j}} \sqrt{\lambda_j(t)} - b_j \right] \tau_j^{-1} \\ \quad \times \lambda_j(t - \tau_{1j} - \tau_{2j}) & \text{if } x_j(t - \tau_{2j}) = 0 \\ \gamma_j \left[ \frac{x_j(t - \tau_{2j})}{\mu_j \sqrt{\tau_j}} \sqrt{\lambda_j(t)} - b_j \right] \tau_j^{-1} \mu_j & \text{otherwise.} \end{cases} \quad (26)$$

Observe that this system may be simplified further by assuming that  $\lambda_j = \mu_j + O(\sqrt{\mu_j}) \forall j = 1, 2, \dots, C$ . Thus

$$\frac{d}{dt} x_j(t) = \begin{cases} \lambda_j(t - \tau_{1j}) - \mu_j & \text{if } \lambda_j(t - \tau_{1j}) - \mu_j > 0 \\ & \text{or } x_j(t) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (27)$$

$$\frac{d}{dt} \lambda_j(t) = \begin{cases} \gamma_j \left[ \frac{x_j(t - \tau_{2j})}{\sqrt{\mu_j \tau_j}} - b_j \right] \tau_j^{-1} \lambda_j(t - \tau_{1j} - \tau_{2j}) & \text{if } x_j(t - \tau_{2j}) = 0 \\ \gamma_j \left[ \frac{x_j(t - \tau_{2j})}{\sqrt{\mu_j \tau_j}} - b_j \right] \tau_j^{-1} \mu_j & \text{otherwise.} \end{cases} \quad (28)$$

### Remark 3

Working out with the linear function scheme:

$$F_j(x_j(t - \tau_{2j}), \lambda_j(t)) = \gamma_j f_j \left( \frac{x_j}{\sqrt{\mu_j \tau_j}} - b_j \right),$$

$$\forall j = 1, 2, \dots, C,$$

where  $f_j(-b_j) \sim O(1)$ ,  $f_j(x_j) \downarrow$ , and  $f_j(0) = 0$ , we may conclude:

$$1. \vec{p} = (\lambda_1, x_1; \lambda_2, x_2; \dots; \lambda_C, x_C)^T$$

$$= (\mu_1, B_1; \mu_2, B_2; \dots; \mu_C, B_C)^T,$$

where  $B_j = b_j \sqrt{\mu_j \tau_j}$ , is an unstable equilibrium point.

2. Assume  $\forall j = 1, 2, \dots, C: f_j(x_j) \downarrow, -b_j \leq x_j \leq \infty$  and  $\gamma_j$  is general. Then, there are functions  $H_j(\mu_j)$ ,  $\Lambda_j^-(\mu_j)$ ,  $\Lambda_j^+(\mu_j)$ , which are positive, bounded, and independent of initial conditions, such that:

(a)  $0 \leq x_j(t) \leq H_j(\mu_j)$

(b)  $\mu_j - \Lambda_j^-(\mu_j) \leq \lambda_j(t) \leq \mu_j + \Lambda_j^+(\mu_j)$ ,

$\forall t$  large enough,  $t > T$ , where  $T$  depends on initial conditions.

(c) For optimal scaling  $\gamma_j = \tau_j \sqrt{\tau_j} / \sqrt{\mu_j}$ ,  $\exists m_j, n_j > 0$ ;  $j = 1, 2, \dots, C$ , independent of initial conditions such that:

$$H_j(\mu_j) = m_j \sqrt{\mu_j}$$

$$\text{and } \Lambda_j^+(\mu_j) = \Lambda_j^-(\mu_j) = n_j \sqrt{\mu_j}.$$

Finally, to numerically solve the above system (25) and (26), one has to numerically represent the system of DDEs to an equivalent ODE system. The procedure is as follows:

$$\frac{d}{dt} x_j(t) = \begin{cases} z_j(t) - \mu_j & \text{if } z_j(t) - \mu_j > 0 \text{ or } x_j(t) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (29)$$

$$\frac{d}{dt} \lambda_j(t) = \begin{cases} \gamma_j \left[ \frac{w_j(t)}{\mu_j \sqrt{\tau_j}} \sqrt{\lambda_j(t)} - b_j \right] \tau_j^{-1} r_j(t) & \text{if } w_j(t) = 0 \\ \gamma_j \left[ \frac{w_j(t)}{\mu_j \sqrt{\tau_j}} \sqrt{\lambda_j(t)} - b_j \right] \tau_j^{-1} \mu_j & \text{otherwise.} \end{cases} \quad (30)$$

where

$$z_j(t) = \lambda_j(f_j(t)), \quad w_j(t) = x_j(g_j(t)),$$

$$\text{and } r_j(t) = \lambda_j(h_j(t)) \quad (31)$$

with

$$f_j(t) = t - \tau_{1j}, \quad g_j(t) = t - \tau_{2j},$$

$$\text{and } h_j(t) = t - \tau_{1j} - \tau_{2j}. \quad (32)$$

In the above,  $j = 1, \dots, C$ , and in reality  $C$  may be large. In addition, the control functions may be computationally expensive. Thus, serial methods may not be efficient. It is therefore necessary to provide new, more efficient methods to solve the produced systems of DDEs. These methods will be considered in the following section. However, the analysis presented above is quite general, and may be applied in some interesting practical situations of a single hop high speed data network. To



facilitate the examination, the following classification has been adopted.

*1st case: No of nodes: 1, No of VCs: C, unequal propagation delays, unequal allocation of bandwidth.*

This is the case presented above. It corresponds to the most general situation of a single node network. The adaption algorithm for the  $j$ th VC,  $j = 1, 2, \dots, C$ , following the path with propagation delay  $\tau_j$  and having a fraction of the total throughput  $\theta_j$  is given by Eq. (24). Note that for each VC,  $j = 1, 2, \dots, C$ , the parameters  $\tau_j$  and  $\theta_j$  may all be different. Clearly, the corresponding system of ODEs is that of Eqs. (29) and (30), while the solution should also satisfy Eqs. (31) and (32). However, an interesting case may allow groups to be considered as well. As an example, we may consider the case of a single node network with two paths, 'long' and 'short' propagation delays  $\tau_L$  and  $\tau_S$  ( $\tau_L/\tau_S = 2$ ), and many VCs in each path, say  $C_L$  and  $C_S$ , respectively ( $C = C_L + C_S$ ).

*2nd case: No of nodes: 1, No of VCs: C, unequal propagation delays, fair allocation of bandwidth.*

This is the case in which a virtual circuit aggregates many smaller virtual circuits, each of which follows a different path with different propagation delay  $\tau_j$ ;  $j = 1, 2, \dots, C$ , and shares an equal allocation of bandwidth. The adaptation algorithm, and the resulting system of ODEs is derived from Eqs. (24) and (29) to (32), with  $\theta_j = 1/C$ ;  $j = 1, 2, \dots, C$ .

*3rd case: No of nodes: 1, No of VCs: C, equal propagation delays, unfair allocation of bandwidth.*

This is the case in which a virtual circuit aggregates many smaller virtual circuits, each of which follows the same path in the data network and shares an unequal allocation of bandwidth. In such a case the propagation delays  $\tau_j$ ,  $j = 1, 2, \dots, C$ , are all equal (say equal to 1), and the fractional bandwidth allocation parameters  $\theta_j$  are all different. For each  $j = 1, 2, \dots, C$ , the adaptation algorithm (24) is reduced to:

$$K_{j,n+1} = K_{j,n} - \gamma_j \left[ (R_{j,n} - 1) \sqrt{K_{j,n}} - \sqrt{\theta_j} \right] \quad (33)$$

while in a similar way, we may derive the following ODE system:

$$\frac{d}{dt} x_j(t) = \begin{cases} z_j(t) - \mu_j & \text{if } z_j(t) - \mu_j > 0 \text{ or } x_j(t) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (34)$$

$$\frac{d}{dt} \lambda_j(t) = \begin{cases} \gamma_j \left[ \frac{w_j(t)}{\mu_j \sqrt{\tau_j}} \sqrt{\lambda_j(t)} - \sqrt{\theta_j} \right] \tau_j^{-1} r_j(t) & \text{if } w_j(t) = 0 \\ \gamma_j \left[ \frac{w_j(t)}{\mu_j \sqrt{\tau_j}} \sqrt{\lambda_j(t)} - \sqrt{\theta_j} \right] \tau_j^{-1} \mu_j & \text{otherwise} \end{cases} \quad (35)$$

where

$$z_j(t) = \lambda_j(f_j(t)), \quad w_j(t) = x_j(g_j(t)), \quad \text{and} \quad r_j(t) = \lambda_j(h(t)) \quad (36)$$

with

$$f_j(t) = t - \tau_{1j}, \quad g_j(t) = t - \tau_{2j}, \quad \text{and} \quad h(t) = t - \tau_{1j} - \tau_{2j} = t - 1. \quad (37)$$

*4th case: No of nodes: 1, No of VCs: C, equal propagation delays, fair allocation of bandwidth.*

This case is similar to the 3rd case, except that we assume a fair share of bandwidth, and therefore the fractional allocation parameters  $\theta_j$  are all equal to  $1/C$ ,  $j = 1, 2, \dots, C$ . The application is straightforward. The adaptation algorithm is derived from Eq. (33), while the system of ODEs representing this scheme is derived from Eqs. (34) and (35). The solution also satisfy Eqs. (36) and (37).

*5th case: No of nodes: 1, No of VCs: 1.*

This is a trivial case.

### 3. Numerical analysis

#### 3.1. Introduction to parallel block methods for ODEs

In the sequel we propose a new method for solving the ODEs systems produced from the above modelling. The method is particularly useful in complex cases, where the resulting systems are large. As one may observe, the resulting systems of ODEs can be viewed as systems of equations of the form:

$$\frac{d}{dt} \vec{\Psi} = Q(\vec{\Psi}) \quad \text{where} \quad \vec{\Psi} = (\lambda_1, x_1; \lambda_2, x_2; \dots; \lambda_C, x_C)^T \quad (38)$$

and  $Q$  is a functional form of our system of equations. Such problems may not be solvable in reasonable time on a serial machine because of their magnitude. However, parallelism may also be exploited in many other ways. For example, one may be interested to solve this model over a long time scale, or if solutions are needed in real time, and finally, if function evaluations are expensive.

One natural way to exploit parallelism associated with such large systems is by splitting the evaluation of the function components amongst the processors available. A second source of parallelism arises if the problem to be solved is stiff. Stiffness is a very difficult property to characterise explicitly, but problems that are stiff typically arise from models which have widely differing time components. These two natural ways of exploiting parallelism suggest that it is not possible to develop parallel algorithms in isolation from the parallel environment.

However, currently there are no general purpose parallel scientific libraries, and while there have been some attempts at creating codes which can be ported between different parallel environments, this has only focused on the basic elements of array manipulation. Note that, according to the taxonomy in Ref. [21], four basic types of parallelism can be identified, that is, SISD (Single Instruction stream, Single Data stream), SIMD (Single Instruction stream, Multiple Data stream), MISD (Multiple Instruction stream, Single Data stream) and MIMD (Multiple Instruction stream, Multiple Data stream). In addition, new architectures are being proposed which are hybrid in nature, having elements from many different categories.

Both Gear [19] and Franklin [20], in their review papers, gave a taxonomy of available modes of parallelism in the numerical solution of first order Initial Value Problems (IVP) of the form:

$$\frac{d}{dt} \bar{Y} = \bar{F}(t, \bar{Y}(t)), \quad \bar{Y} : [t_0, T] \times \mathbf{R}^m \rightarrow \mathbf{R}^m \quad \bar{Y}(t_0) = \bar{Y}_0. \quad (39)$$

Their classifications include parallelism across space, equation segmentation methods, methods in which specific subsystems in Eq. (39) are assigned to separate processors, parallel predictor-corrector methods, parallelism across time, parallel block methods, and finally, other methods which accommodate some mode of parallelism, or they are adaptable to parallel computations. Gear [19] identified the major impediments to parallelism in ODE solvers as:

- The narrowness of the computational lattices inherit in most conventional algorithms which adopt error and stepwise control.
- The extensive communications between processes, particularly in the case of stiff algorithms which are implicit and which are solved through some variants of Newton's scheme introducing independence of the components of the dependent variable.

Block methods turned out to be efficient methods for solving, on parallel computers, the first order scalar IVP presented in Eq. (39) and corresponding to the systems of ODEs in which we are particularly interested. The general structure of the new block methods described here is a direct generalisation of the implicit one-step method:

$$y_{n+1} = ay_n + hbf(y_n) + hcf(y_{n+1}), \quad n = 0, 1, \dots, \quad (40)$$

where  $h$  is the stepsize and  $y_n$  an approximation to  $y(t_n)$ . We introduce the block vectors  $\bar{Y}_{n+1} = (y_{n,1}, \dots, y_{n,k})^T$ ,  $\bar{d} = (d_1, \dots, d_k)$  and  $d_k = 1$ , where the components  $y_{n,j}$  denote numerical approximations to the exact solution values  $y(t_n + d_j h)$  at  $k$  points. Since, for scalar ODEs  $(dy(t))/dt = f(t, y(t))$ ,  $y(t_0) = y_0$  is a scalar test

equation, we may define the block method:

$$\bar{Y}_{n+1} = A\bar{Y}_n + hB\bar{f}(\bar{Y}_n) + hC\bar{f}(\bar{Y}_{n+1}), \quad n = 0, 1, \dots, \quad (41)$$

where  $A$ ,  $B$  and  $C$  are  $k$ -by- $k$  matrices. Here we use the convention that, for any given vector  $\bar{u} = (u_i)$ ,  $\bar{f}(\bar{u})$  denotes the vector with entries  $f(u_j)$ . The method (41) can be considered as the block analogue of (40), and it is straightforwardly extended to systems of ODEs, and thus to non-autonomous equations.

A characteristic of these methods is that, unlike conventional BLM (Block Linear Multistep) methods, the block point vector  $\bar{d}$  is allowed to have  $k - 1$  non-integer components. Note that, in the block methods considered here, the components of the block vector represent approximations to the exact solution are not necessarily equally spaced block points  $t_n + d_j h$ . Thus, we obtain additional parameters for increasing the order of accuracy of the method. In the derivation of these methods, it turns out to be convenient to start with a Runge–Kutta (R-K) method. Then, in analogy with BLM methods, we may replace the  $y$ -values generated by the method by vectors, the components of which represent approximations to the exact solution. Since these vectors are  $k$ -dimensional, the R-K parameters are replaced by  $k \times k$  matrices.

The points  $t_n$  and  $t_n + d_j h$ ;  $j \neq k$ , will be called *step points* and *block points*, respectively. Block points coincide with step points if the corresponding value of  $d_j$  is an integer. Upon completion of the integration process, the order of accuracy of the numerical solution obtained is not necessarily the same at all points  $t_n + d_j h$ . Points where the corresponding components of  $\bar{Y}_{n+1}$  have the same order as the components corresponding to the step points are called *output points*. To start the method, we need the initial vector  $\bar{Y}_0$ , which requires, in general, as many starting values as there are distinct values  $d_j$  ( $j = 1, 2, \dots, k$ ). Notice that the last component of  $\bar{Y}_{n+1}$  contains the step point value  $y_{n+1}$ . Furthermore, we remark that in general,  $y_{n,i} = y_{m,j}$  even if  $n + d_i = m + d_j$ .

The method (41) is suitable for direct use on parallel computers if the matrix  $C$  is *diagonal*, since such a form uncouples the various components as far as implicitness is concerned. In such cases, the corresponding methods will be called *parallel block methods*. Let the exact solution be substituted into Eq. (41). Then, in general, the order conditions are derived by requiring that the residual vector is of order  $h^{p+1}$  for all components (that is, we require that all components of  $\bar{Y}_{n+1}$  are  $p$ th order approximations to the exact solution values). In this way, we obtain the following order consistency:

$$(I - zC) \exp(z\bar{d}) - (A + zB) \exp(z\bar{d} - z\bar{e}) = O(z^{p+1}) \quad (42)$$

where  $\bar{\mathbf{e}} = (1, 1, \dots, 1)^T$  denotes the unit vector. By defining the error vectors:

$$\begin{aligned}\bar{\mathbf{D}}^{(0)} &:= \mathbf{A}\bar{\mathbf{e}} - \bar{\mathbf{e}} \\ \bar{\mathbf{D}}^{(1)} &:= \mathbf{A}(\bar{\mathbf{d}} - \bar{\mathbf{e}}) + \mathbf{B}\bar{\mathbf{e}} - \mathbf{C}\bar{\mathbf{e}} - \bar{\mathbf{d}} \\ \bar{\mathbf{D}}^{(j)} &:= \mathbf{A}(\bar{\mathbf{d}} - \bar{\mathbf{e}})^j + j[\mathbf{B}(\bar{\mathbf{d}} - \bar{\mathbf{e}})^{j-1} - \mathbf{C}\bar{\mathbf{d}}^{j-1}] - \bar{\mathbf{d}}^j \\ j &= 2, 3, \dots\end{aligned}\quad (43)$$

the conditions for  $p$ th order consistency take the form:

$$\bar{\mathbf{D}}^{(j)} = [\mathbf{D}_i^{(j)}] = \bar{\mathbf{0}} \quad i = 1, 2, \dots, k \quad j = 0, 1, \dots, p. \quad (44)$$

Here, powers of vectors (apart from those in parenthesis) are meant to be component-wise powers. Note that in the construction of high-order formulas, it is convenient to specify the matrix  $\mathbf{A}$  in advance. This is because the eigenvalues of  $\mathbf{A}$  should lie in a zero-stable configuration, and therefore they should be on the unit disk, those on a unit circle being simple. Such a zero-stability condition is difficult to be satisfied simultaneously with the order conditions unless  $k$  is sufficiently small.

However, the linear stability of block methods can be investigated by applying the method to the test equation  $(d/dt)y = \rho y$ . Thus, we obtain the following recursive equation:

$$\begin{aligned}\bar{\mathbf{Y}}_{n+1} &= \mathbf{M}(w)\bar{\mathbf{Y}}_n, \quad \mathbf{M}(w) = [\mathbf{I} - w\mathbf{C}]^{-1}[\mathbf{A} + w\mathbf{B}], \\ w &= \rho h\end{aligned}\quad (45)$$

where  $\mathbf{M}$  is the *amplification matrix* and its eigenvalues are the *amplification factors*. Note that, if we require the elements of the diagonal matrix  $\mathbf{C}$  to be positive, then the matrix  $\mathbf{I} - w\mathbf{C}$  is non-singular for all  $w$  on the negative axis. Thus, we may assume that the elements of  $\mathbf{C}$  are positive, and following the familiar stability definitions used for Runge–Kutta and linear multistep methods we may introduce the following definitions:

**Definition 1** The region where the amplification matrix  $\mathbf{M}(w)$  is power bounded, is called the *stability region* of the block method.

**Definition 2** If the stability region contains the origin, the method called *zero-stable*.

**Definition 3** Let the characteristic equation:

$$P(H, r) = \det[\mathbf{A} + H\mathbf{B} - r(\mathbf{I} - H\mathbf{C})] \quad (46)$$

of the block method, where  $H = \rho h$  with  $\rho$  defined by the scalar test equation  $(d/dt)y = \rho y$ . Then, the block method is *A-stable* if the roots of polynomial  $P(\infty, r)$  are on the unit disk, those on the unit circle being simple.

**Definition 4** If the amplification matrix of an *A-stable* method has vanishing eigenvalues at infinity, the method is called *L-stable*.

**Definition 5** The region where  $\|\mathbf{M}^n\|$  tends to zero will be called the *strong stability region*.

**Definition 6** If the strong stability region of a block method contains the left half plane, the block method is called *strongly A-stable*.

### 3.2. Construction of A-stable parallel block methods

In the sequel we consider A-stable block methods of two and three dimensions.

#### 3.2.1. Methods of order two ( $k = 2$ )

Consider the case  $k = 2$  and choose the coefficients matrices of the form:

$$\begin{aligned}\mathbf{A} &= \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}, \\ \mathbf{C} &= \begin{pmatrix} c_1 & 0 \\ 0 & c_2 \end{pmatrix}, \quad \text{and} \quad \bar{\mathbf{d}} = (d, 1)^T.\end{aligned}\quad (47)$$

Using Eq. (44) we may obtain the conditions of second order consistency:

$$\begin{aligned}\bar{\mathbf{D}}^{(0)} &:= \mathbf{A}\bar{\mathbf{e}} - \bar{\mathbf{e}} = \bar{\mathbf{0}} \\ \bar{\mathbf{D}}^{(1)} &:= \mathbf{A}(\bar{\mathbf{d}} - \bar{\mathbf{e}}) + \mathbf{B}\bar{\mathbf{e}} - \mathbf{C}\bar{\mathbf{e}} - \bar{\mathbf{d}} = \bar{\mathbf{0}} \\ \bar{\mathbf{D}}^{(2)} &:= \mathbf{A}(\bar{\mathbf{d}} - \bar{\mathbf{e}})^2 + 2[\mathbf{B}(\bar{\mathbf{d}} - \bar{\mathbf{e}}) - \mathbf{C}\bar{\mathbf{d}}] - \bar{\mathbf{d}}^2 = \bar{\mathbf{0}}.\end{aligned}\quad (48)$$

The above equations may be re-written explicitly ( $i = 1, 2$ ) as:

$$\begin{aligned}a_{i1} &= a_i \\ a_{i2} &= 1 - a_i \\ b_{i1} &= \frac{1}{2}(1 - d)a_i + \frac{d_i(2c_i - d_i)}{2(1 - d)} \\ b_{i2} &= d_i + (1 - d)a_i - b_{i1} - c_i.\end{aligned}\quad (49)$$

Note, that  $d_1 = d$ , and  $d_2 = 1$ . Thus, we may express the coefficients of the matrix  $\mathbf{B}$  in terms of the five parameters  $d, a_1, a_2, c_1$  and  $c_2$ .

The components  $D_i^{(j)}$  of the vector  $\bar{\mathbf{D}}^{(j)}$ , for  $j \geq 3$  are given by:

$$\begin{aligned}D_i^{(j)} &= (1 - \frac{1}{2}j)(d - 1)^j a_i + j d_i^{j-1} c_i \\ &\quad + \frac{1}{2} j d_i (d_i - 2c_i)(d - 1)^{j-2} - d_i^j \quad i = 1, 2.\end{aligned}\quad (50)$$

It is easy to observe that  $D_i^{(3)}$  vanishes if:

$$\begin{aligned}a_i &= \frac{d_i}{(d - 1)^3} [3(d - 1)(d_i - 2c_i) + 2d_i(3c_i - d_i)] \\ i &= 1, 2\end{aligned}\quad (51)$$

and  $D_i^{(4)}$  also vanishes if:

$$c_1 = \frac{d}{2(d+1)}, \quad c_2 = \frac{d-2}{2(d-3)}. \quad (52)$$

The characteristic equation of the method is given by:

$$P(H, r) = \det[\mathbf{A} + H\mathbf{B} - r(\mathbf{I} - H\mathbf{C})] \quad (53)$$

with  $H = \rho h$ , where  $\rho$  is defined by the scalar test equation  $(d/dt)y = \rho y$ . Thus, the condition of zero-stability (i.e. the condition that the eigenvalues 1 and  $a_1 - a_2$  of the matrix  $\mathbf{A}$  are on the unit disk) leads to the condition:  $-1 \leq a_1 - a_2 < 1$ . Also, the condition of A-stability (i.e. the roots of polynomial  $P(\infty, r)$  are on the unit disk) leads to the conditions:

$$[b_{11}c_2 + b_{22}c_1] \leq c_1c_2[c_1c_2 + \det(\mathbf{B})], \quad \det(\mathbf{B}) \leq c_1c_2. \quad (54)$$

Based on the above, we may now present the A-stable parallel block methods of second order consistency by choosing the free parameters  $a_i$  and  $c_i$  such that the matrix  $\mathbf{B}$  in Eq. (49) vanishes while satisfying the A-stability property. As an example in case where  $d = 0$  we have:

$$\mathbf{A} = \begin{pmatrix} 0 & 1 \\ -\frac{1}{3} & \frac{4}{3} \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, \quad \mathbf{C} = \begin{pmatrix} 0 & 0 \\ 0 & \frac{2}{3} \end{pmatrix},$$

$$\vec{d} = (0, 1)^T.$$

To present the A-stable, parallel block methods of three-order consistency, we choose  $D_1^{(3)} = D_2^{(3)} = 0$  in Eq. (50). Thus we find:

$$a_1 = \frac{d(d^2 - 3d + 6c_1)}{(d-1)^3},$$

$$a_2 = \frac{3d + 12c_2 - 6dc_2 - 5}{(d-1)^3}$$

$$b_{11} = \frac{d^2 - 2dc_1 - d^2c_1}{(d-1)^2},$$

$$b_{12} = \frac{d - 2dc_1 - c_1}{(d-1)^2}$$

$$b_{21} = \frac{2 - 5c_2 - d + 2dc_2}{(d-1)^2},$$

$$b_{22} = \frac{(d-2)^2 - c_2(d^2 - 6d + 8)}{(d-1)^2}$$

where  $d, c_1, c_2$  are free parameters. Taking into account the conditions of zero-stability and A-stability, we find  $d = 0.917387$ ,  $c_1 = 0.319523$ ,  $c_2 = 0.347067$ .

### 3.2.2. Methods of dimension three ( $k = 3$ )

Consider the case  $k = 3$  and choose the coefficients matrices of the form  $\mathbf{A} = [a_{ij}]$ , with  $a_{i3} = 1 - a_{i1} - a_{i2}$ ,  $i = 1, 2, 3$ ,  $\mathbf{B} = [b_{ij}]$ ,  $\mathbf{C} = [c_{ij}]$ , diagonal, with  $c_{ii} = c_i$  and  $\vec{d} = (d_1, d_2, 1)^T$ . Clearly, the condition of the first order accuracy is satisfied, since  $\vec{D}_0 = \vec{0}$ . Requiring the conditions of fourth-order accuracy the following linear systems must be satisfied:

$$b_{i1} + b_{i2} + b_{i3} + c_i = d_i - a_{i1}(d_1 - 1) - a_{i2}(d_2 - 1)$$

$$(d_1 - 1)b_{i1} + (d_2 - 1)b_{i2} + d_i c_i$$

$$= \frac{1}{2}[d_i^2 - a_{i1}(d_1 - 1)^2 - a_{i2}(d_2 - 1)^2]$$

$$(d_1 - 1)^2 b_{i1} + (d_2 - 1)^2 b_{i2} + d_i^2 c_i$$

$$= \frac{1}{3}[d_i^3 - a_{i1}(d_1 - 1)^3 - a_{i2}(d_2 - 1)^3]$$

$$(d_1 - 1)^3 b_{i1} + (d_2 - 1)^3 b_{i2} + d_i^3 c_i$$

$$= \frac{1}{4}[d_i^4 - a_{i1}(d_1 - 1)^4 - a_{i2}(d_2 - 1)^4].$$

To ensure zero-stability, we require that  $\mathbf{A}$  has the two extra eigenvalues within the unit circle. Writing the characteristic equation of  $\mathbf{A}$  in the form  $P(z) = (z - 1) \times (z^2 + t_0 z + s_0) = 0$ , we find that we have zero-stability if:

$$|t_0| < s_0 + 1, \quad s_0 < 1, \quad t_0 = a_{31} + a_{32} - a_{11} - a_{22}$$

$$s_0 = a_{11}a_{12} + a_{31}a_{12} + a_{32}a_{21} - a_{11}a_{32} - a_{21}a_{12} - a_{22}a_{31}.$$

Taking this constraint into account, and using a package of symbolic computation, we search over the free parameters ( $a_{i1}, a_{i2}$ ,  $i = 1, 2, 3, d_1, d_2$ ), to obtain the A-stable fourth-order method:

$$\mathbf{A} = \begin{pmatrix} -1 & \frac{1}{2} & \frac{3}{2} \\ \frac{1}{2} & 1 & -\frac{1}{2} \\ -1 & \frac{1}{2} & \frac{3}{2} \end{pmatrix},$$

$$\mathbf{B} = \begin{pmatrix} \frac{2795}{2048} & \frac{15161}{3168} & \frac{103501}{92160} \\ -\frac{73}{126} & -\frac{467}{126} & -\frac{259}{702} \\ \frac{80345}{129024} & \frac{54419}{30240} & \frac{41927}{55296} \end{pmatrix},$$

$$\mathbf{C} \text{ diagonal with } c_1 = \frac{16939}{28160}, \quad c_2 = \frac{277}{234}, \quad c_3 = \frac{16001}{23040}$$

$$\text{and } \vec{d} = (5, 13/4, 1)^T.$$

## 4. Conclusions

In this paper we have developed a general method for solving some basic delayed feedback schemes used in long haul, high speed, virtual circuit networks in which the propagation delay cannot be neglected. A major

problem in such networks is the statistical interaction among the traffic flows to be controlled. Two basic virtual circuit networks of a balanced form are examined. These are the single-hop networks which aggregate many virtual circuits in parallel, and the  $M$ -hop tandem network. These networks are presented as systems of DDEs with unknown control functions, in which the rate of transmission and the queue occupancy are modelled as fluids. We used the well known adaptive algorithms of Mitra and Seery [11] to dynamically adjust the window size and to impose linear structure control. As a result, the examined networks are presented as linear systems of some DDEs.

Using linear approximation, the above schemes are proved unstable (in a Liapounov sense). However, we identified the appropriate scaling for the parameters so as to make the above schemes to perform near their optimal theoretical values for a wide range of speeds. Since the standard theory of DDEs does not treat the class of systems we considered, stability in a Liapounov sense is very difficult to prove. Thus new tools have to be developed. In addition, and more importantly, a solution method of such DDE systems has not been investigated so far, particularly in general cases, where the resulting DDE systems are more complex and large. For this reason, we proposed a general method, by means of parallel block methods, and we derived an approximate numerical solution. This method is used to solve the systems of the ODEs in which the original systems of DDEs have been transformed. The basic theory underlying the parallel block methods for the numerical solution of the system of ODEs has been developed and fully discussed. Finally, we examined the numerical stability by deducing two and three dimensional, A-stable parallel block methods of second, third and fourth order consistency.

The results found leave open the question of whether further improvements are possible. Since this work directly applies to the adaptive control of frame relay and ATM networks, both of which provide feedback to users on congestion, the results found are even more important. However, one may observe that the principal concerns here are the accuracy of the results and the time required to obtain them. Of course, on the analytical side, accuracy is an issue only if the model or the solution method is approximate. In either case, one should seek a means of determining bounds on the errors of approximations, as they reflected in the performability values that are ultimately determined. If both the model and the solution method are approximate (as in this case), there is a need to understand the interaction between the two types of approximation; the concern being in incompatibility which causes excessive errors when the two are used together. Therefore, further study is needed

in order to provide more accurate models, so that any improvement in the numerical stability, by means of providing high-order A-stable parallel block methods, will become viable.

## References

- [1] M. Schwartz, *Telecommunications Networks: Protocols, Modeling and Analysis*. Addison-Wesley, Reading, 1987.
- [2] D. Bertsekas and R. Gallager, *Data Networks*. Prentice-Hall, Englewood Cliffs, NJ, 1987.
- [3] D. Brownning, Flow control in high-speed communication networks. *IEEE Trans. Commun.*, 42 (1994) 2480–2489.
- [4] L. Dittmann, S. B. Jacobsen and K. Moth, Flow enforcement algorithms for ATM networks. *IEEE J. Select. Areas Commun.*, 9 (1991) 343–350.
- [5] R. Jain, Congestion control in computer networks: Issues and trends. *IEEE Network Mag.* (May 1990) 230–244.
- [6] S.-Y.R. Li, Algorithms for flow control and call set-up in multi-hop broadband ISDN. *IEEE INFOCOM*, 1990, pp. 889–895.
- [7] R. Dighe, C. J. May and C. Ramamurthy, Congestion avoidance strategies in broadband packet networks. *INFOCOM*, 1991, pp. 295–303.
- [8] W.E. Leland, Window-based congestion management in broadband ATM networks: The performance of three access-control policies. *GLOBECOM*, 1989, pp. 1794–1800.
- [9] D. Mitra, Asymptotically optimal design of congestion control for high speed data networks. *IEEE Trans. Comm.*, 40 (1992) 301–311.
- [10] F. W. Fendick, M. A. Rontrigues and A. Weiss, Analysis of a rate-based feedback control strategy for long haul data transport. *Perfor. Evaluat.*, 16 (1992) 67–84.
- [11] D. Mitra and J.B. Seery, Dynamic adaptive windows for high speed data networks with multiple paths and propagation delays. *Computer Networks & ISDN Systems*, 25 (1993) 663–679.
- [12] R. Jain, A delay-based approach for congestion avoidance in interconnected heterogeneous computer networks. Rep. DEC-TR-556, Digital Equipment Corp., 1988.
- [13] V. Jacobson, Congestion avoidance and control. *Proc. ACM SIGCOMM*, Stanford, CA, 1988, pp. 314–329.
- [14] K.K. Ramkrishnan and R. Jain, A binary feedback scheme for congestion avoidance in computer networks with a connectionless network layer. *Proc. ACM SIGCOMM*, Stanford, CA, 1988, pp. 303–313.
- [15] J.C. Bolot and U.A. Shankar, Dynamical behaviour of rate-based flow-control mechanisms. *Comput. Comm. Rev.*, 40(2) (1990) 35–49.
- [16] E.L. Hahne, C.R. Kalmanek and S.P. Morgan, Fairness and congestion control on a large ATM data network with dynamically adjustable windows. 13th Int. Teletraffic Conf., Italy, July 1991.
- [17] K. Burrage, Parallel methods for initial value problems. *Applied Numerical Mathematics*, 11 (1993) 5–25.
- [18] P. Amodio and D. Trigiante, A parallel direct method for solving initial value problems for ordinary differential equations. *Applied Numerical Mathematics*, 11 (1993) 85–93.
- [19] C.W. Gear, The potential for parallelism in ODEs. In: S.O. Fatunla (ed.) *Computational Mathematics II*. Boole Press, Dublin, Ireland, 1987, pp. 33–48.
- [20] M.A. Franklin, Parallel solution of ODEs. *IEEE Trans. Comput.*, 27 (1978) 413–420.
- [21] M.F. Flynn, Some computer organisations and their effectiveness. *IEEE Trans. Comput.*, 23 (1972) 948–960.



*Theodore A. Tsiligirides received his BSc in mathematics from the University of Athens, Greece, his MSc in probability and statistics from the Manchester-Sheffield University, UK, and his PhD in telecomms from the University of Strathclyde, Scotland. In 1981 he joined the Mathematics and Computer Science Division of the Agricultural University of Athens, first as a research/teaching assistant, and then as a lecturer. Since 1993 he has worked as an Assistant Professor in the Informatics Laboratory of the same division. His research interests include modelling and traffic control of high speed networks, access methods in mobile and broadband networks, queueing networks and parallel algorithms.*