PERGAMON

Automatica 35 (1999) 1921–1935

www.elsevier.com/locate/automatica

Congestion control in high-speed communication networks using the Smith principle☆

Saverio Mascolo*

Dipartimento di Elettrotecnica ed Elettronica, Politecnico di Bari, Via Orabona 4, 70125 Bari, Italy

Received 6 June 1998; revised 14 March 1999; received in final form 27 May 1999

Classical control theory and Smith's principle are proposed as key tools to design a congestion control law for high-speed data networks. Applications to Asynchronous Transfer Mode networks and to Internet Transmission Control Protocol are illustrated.

Abstract

High-speed communication networks are characterized by large bandwidth-delay products. This may have an adverse impact on the stability of closed-loop congestion control algorithms. In this paper, classical control theory and *Smith's principle* are proposed as key tools for designing an effective and simple congestion control law for high-speed data networks. Mathematical analysis shows that the proposed control law guarantees stability of network queues and full utilization of network links in a general network topology and traffic scenario during both transient and steady-state condition. In particular, no data loss is guaranteed using buffers with any capacity, whereas full utilization of links is ensured using buffers with capacity at least equal to the bandwidth-delay product. The control law is transformed to a discrete-time form and is applied to ATM networks. Moreover a comparison with the ERICA algorithm is carried out. Finally, the control law is transformed to a window form and is applied to Internet. The resulting control law surprisingly reveals that today's Transmission Control Protocol/Internet Protocol implements a Smith predictor for congestion control. This provides a theoretical insight into the congestion control mechanism of TCP/IP along with a method to modify and improve this mechanism in a way that is backward compatible. © 1999 Elsevier Science Ltd. All rights reserved.

Keywords: Flow and congestion control; Smith's principle; High-speed data networks; ATM networks; Internet

1. Introduction

Nowadays, communication networks are among the fastest-growing engineering areas and are driving extraordinary developments in communication industry. An increasing amount of research is devoted to the deployment of new communication networks that merge the capabilities of telephone networks and of computer networks in order to transmit multimedia traffic over a fully integrated universal network. These efforts have lead to the introduction of Broadband Integrated Service Digital Networks (B-ISDNs) and the emerging Asynchronous Transfer Mode (ATM) technology has been retained as the transport technology to be used in B-ISDNs (Varaiya & Walrand, 1996).

ATM networks are a class of *virtual circuit switching* networks conceived to merge the advantages of circuit switched technology (telephone networks) with those of packet switched technology (computer networks). They are connection-oriented in the sense that before two systems on the network can communicate, they should inform all intermediate switches about their service requirements and traffic parameters by establishing a *virtual circuit*. This is similar to the telephone networks, where an exclusive circuit is set up from the calling party to the called party, with the important difference that, in the case of ATM, many virtual circuits can share network resources via *store-and-forward packet switching and statistical multiplexing* (Varaiya & Walrand, 1996). The sharing of network resources allows communication costs

☆This paper was not presented at any IFAC meeting. This paper was recommended for publication in revised form by Guest Editors Venkat Anantharam and Jean Walrand.

*Tel.: +39-080-5460621; fax: +39-080-5460410. The author is temporarily at Boelter Hall, Computer Science Department, University of California at Los Angeles, CA 90095-1596, USA.

E-mail address: mascolo@poliba.it (S. Mascolo)

0005-1098/99/$ - see front matter © 1999 Elsevier Science Ltd. All rights reserved.
PII: S0005-1098(99)00128-4

be drastically reduced and requires sophisticated mechanisms of flow and congestion control to avoid congestion phenomena (Peterson & Davie, 1996). Congestion control is critical in both ATM and Internet networks and it is the most essential aspect of traffic management (Jacobson, 1988; Jain, 1996). Moreover, new control issues are emerging, which aim at ensuring that users get their desired quality of service (QoS). See, for example, Ding (1997), Le Boudec, de Veciana and Walrand (1996), Liew and Chi-yin Tse (1998) and their references.

In the context of ATM networks, the ATM Forum Traffic Management Group defines five service classes to support multimedia traffic, which are the *constant bit-rate* (CBR) class, the *real-time* and *non-real-time variable bit-rate* (VBR) classes, the *unspecified bit-rate* (UBR) class and the *available bit-rate* (ABR) class, which is a *best-effort* class. ABR is the only class that responds to network congestion by means of a feedback control mechanism in order to improve network utilization by minimizing data loss and retransmissions (ATM Forum, 1996; Jain, 1996).

To briefly summarize the algorithms proposed for ABR traffic control, we start by recalling the binary feedback schemes that were first introduced due to their easy implementation (Bonomi, Mitra & Seery, 1995; Fendick, Rodrigues & Weiss, 1992; Ramakrishnan & Jain, 1990; Roberts, 1994; Yin & Hluchyj, 1994). In these schemes, if the queue length in a switch is greater than a threshold, then a binary digit is set in the control management cell. However, they suffer serious problems of stability, exhibit oscillatory dynamics, and require large amount of buffer in order to avoid cell loss. As a consequence, explicit rate algorithms have been largely considered and investigated. See Jain (1996) for an excellent survey. Most of the existing explicit rate schemes lack of two fundamental parts in the feedback control design: (1) the analysis of the closed-loop network dynamics; (2) the interaction with VBR traffic. In Charny, Clark and Jain (1995) and in Jain, Kalyanaraman, Goyal, Fahmy and Viswanathan (1996) an explicit rate algorithm is proposed, which basically computes input rates dividing the measured available bandwidth by the number of active connections. In Zhao, Li and Sigarto (1997), the control design problem is formulated as a standard disturbance rejection problem where the available bandwidth acts as a disturbance for the system. The ABR source rate is adapted to the low-frequency variation of the available bandwidth and H_2 optimal control is applied to design a controller that minimizes the difference between the source input rate and the available bandwidth. A drawback is that the design of the controller depends on the characteristic of the interacting VBR traffic and on the measurements of the available bandwidth, which is difficult to be obtained in practice. In Altman, Basar and Srikant (1998), the problem is formulated as a stochastic control problem where the disturbance is modeled as an autoregressive process. The node has to estimate this process using recursive least squares. In Mascolo (1997), Smith's principle is exploited to derive a controller in case a *first in–first out* (FIFO) buffering is maintained at output links. The control algorithm is executed at bottleneck node and computes an input rate which is fed back to the source. The advantages are that the bottleneck switch maintains FIFO queuing and measures only the queue and not the available bandwidth; the drawback is that more computational and informational burden is placed at the switch. An analytic method for the design of a congestion controller has been proposed in Benmohamed and Meerkov (1993), where the input rate is computed as a linear combination of the past values of the rates and of the queue levels. The goal is to stabilize the queue level at a given threshold. The algorithm requires a complex on-line tuning of control parameters to ensure stability and to damp queue oscillations under changing traffic condition. Due to the complex closed-loop dynamics, the authors were unable to solve the global stability problem and they investigated it only by simulations. In Izmailov (1995), two linear feedback control algorithms have been proposed for the case of a *single connection* with a *constant service rate*, i.e. the interaction with VBR traffic is not considered. Due to the transcendental form of the closed-loop characteristic equations, only the asymptotic properties of the system were analyzed.

In the context of Internet, after the Transmission Control Protocol/Internet protocol (TCP/IP) had become operational, the network was suffering from congestion collapse. TCP/IP congestion control was introduced into the Internet in the late 1980s and has been successful in preventing congestion collapse (Jacobson, 1988). Many improvements have been introduced since that time and intense research activity is going on to improve the efficiency of this control mechanism. See, for example, Floyd and Jacobson (1993), Hahne, Kalmanek and Morgan (1993), Floyd (1994), Villamizar and Song (1995), Brakmo, O'Malley and Peterson (1995), Balakrishnan, Padmanabhan, Seshan and Katz (1996), Jacobson, Braden and Borman (1997), Kalampoukas and Varma (1998), Gerla, Locigno, Mascolo and Weng (1999). Notice that, nowadays, the TCP/IP congestion control algorithm is the only algorithm successfully tested in a real worldwide packet switching network.

In this paper, we address the issue of congestion control in a general packet switching network using *classical control theory*, that is, transfer functions are used to describe the system to be controlled and to design the controller. The dynamic behavior of each network queue in response to data input is modeled as the cascade of an integrator with a time delay. Since propagation delays play a key role in high-speed communication networks, we choose the Smith principle to design a simple congestion control law that is effective over path with any bandwidth-delay product. The "best effort" available bandwidth is modeled as an unknown and bounded *disturbance* input since it is difficult to measure it.

The designed control law is applied to control ABR traffic in ATM networks and a comparison with the well-known Explicit Rate Indication Control Algorithm (Jain et al., 1996) is carried out. Moreover, the proposed control law is applied to TCP/IP. This application surprisingly reveals that today's TCP Internet Protocol implements a Smith predictor to control receiver's buffer and network congestion. This result appears extremely interesting because it gives a theoretical basis to the great success of TCP/IP to control congestion and a very useful insight on how to improve its efficiency.

The paper is organized as follows: Section 2 describes the data network model; Section 3 models the controlled data networks using transfer functions; in Section 4 the controller is designed using Smith's principle; in Section 5 transient and steady-state dynamics are evaluated via mathematical analysis; Section 6 describes the application of the proposed control law to ATM networks whereas Section 7 describes the application to Internet; finally, Section 8 draws the conclusions.

2. The data network model

In this section we develop the model of a general network that employs a *store-and forward packet switching service*, that is, packets or cells enter the network from the source edge nodes, are then stored and forwarded along a sequence of intermediate nodes and communication links, finally reaching their destination nodes (Benmohamed & Meerkov, 1993; Peterson & Davie, 1996; Varaiya & Walrand, 1996). Fig. 1 depicts a *store and forward packet switching network*. Such a network can be considered as a graph consisting of:

(a) A set $N = \{n_i\}$ of nodes (switches or routers), which store and forward the packets along the communication path. A node consists of a set of input queues where incoming packets are stored and of a set of output queues where outgoing packets are stored. It is assumed that the processing capacity of each node is larger than the total transmission capacity of its incoming links so that congestion is caused by transmission capacity only.

(b) A set $L = \{l_i\}$ of communication links, which connect the nodes to permit the exchange of information. Each link is characterized by the transmission capacity $c_i = 1/t_i$ (packets/s) and the propagation delay t_{di}. For each node $i \in N$, let $O(i) \subset L$ denote the set of its outgoing link and let $I(i) \subset L$ denote the set of its incoming links.

The network traffic is contributed by source/destination pairs $(S, D) \in N \times N$. The pair (S, D) will be referred to as the (S, D) connection, the (S, D) flow or the (S, D) *virtual circuit* (VC). For each (S, D) connection, the source at node S sends packets to the destination node D through a sequence of links referred to as the path of the connection and denoted by $p(S, D)$. A deterministic fluid model approximation of packet flow is assumed, that is, each input flow is described by the continuous variable $u(t)$ measured in packets/sec. In high speed communication networks, the bandwidth delay product t_{dj}/t_j is a key parameter that affects the stability of closed-loop control algorithms. It represents a large number of packets "in flight" on the transmission link. These packets are also called "*in the pipe*" *packets or cells*.

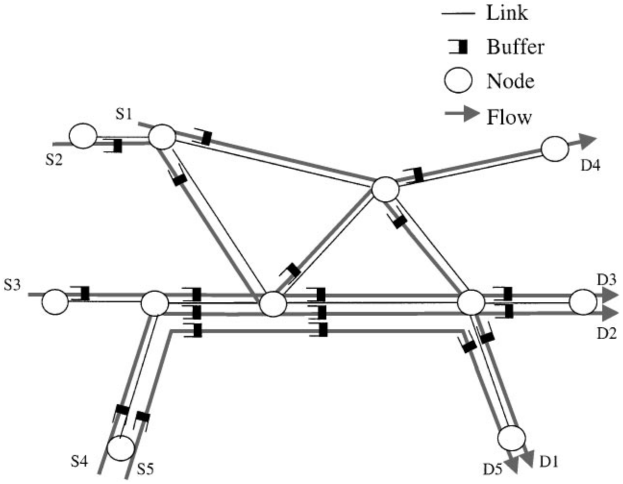


Fig. 1. Store and forward packet switching network. Five (S_i, D_i) connections and per-flow buffering are shown.

2.1. Per-flow versus FIFO buffering

Per-flow buffering is retained to play an important and even necessary role for controlling congestion and the QoS of a flow (Stoica, Shenker & Zhang, 1998; Chow & Leon-Gracia, 1999; Peterson & Davie, 1996; Benmohamed & Wang, 1998; Suter, Lakshman, Stiliadis & Choudhury, 1998; Keshav, 1991). Fig. 2 depicts an output link maintaining per-flow buffering. The assumption of per-flow buffering has many advantages. Congestion control algorithms can be effectively and easily run at the source (see Peterson & Davie, 1996), whereas network nodes are in charge of per-flow queuing. This uncouples the congestion and fairness issues and allows a node to easily enforce max–min fairness (Jaffe, 1981; Jain, 1996; Peterson & Davie, 1996). In contrast to FIFO queuing, per-flow buffering separates packets according to the flow to which they belong. Thus, the dynamics of the flows are completely uncoupled. A flow goes through shared links and exclusive buffers as shown in Fig. 1. All queue levels encountered by a flow are zero except the queue level feeding the bottleneck link. This leads to a

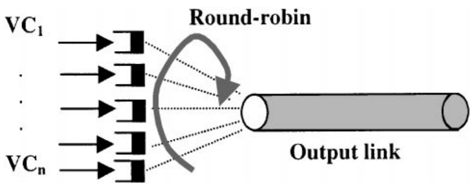


Fig. 2. Per-flow buffering of *n* flows served via Round-robin mechanism.

control problem design that is a single-input/single-output (SISO) problem design, where the input is the connection input rate and the output is the bottleneck queue level. On the other hand, if many flows are aggregated in the same FIFO queue, then the control problem design becomes more complex and requires more computational effort (Mascolo, 1997). In fact, a flow cooperates with many other flows to fill the bottleneck FIFO queue. Therefore, a source should know the input rate of all the other sources sharing the bottleneck queue to exercise the control. But this knowledge is not available at a single source. The only place that can know it is the bottleneck node. As a consequence, the control algorithm is run at network nodes as in Charny et al. (1995), Jain et al. (1996) and Mascolo (1997). Thus, the burden of maintaining per-flow queuing is replaced by the complexity of computing the input rates. Another important drawback of an algorithm executed at network nodes comes from the fact that it is hard for a node to know where the bottleneck of the connection is located. As a consequence, it is difficult to ensure max–min fairness and efficient control. For these considerations, nowadays many switch vendors are implementing per-VC queuing (Benmohamed & Wang, 1998) and per-flow buffering is also auspicated for Internet routers in order to ensure QoS (Stoica et al., 1998; Suter et al., 1998; Peterson & Davie, 1996). The drawbacks of per-flow buffering are that it is not scalable and may be costly to be implemented (Chow & Leon-Gracia, 1999). Therefore, active research is devoted to find simpler solutions which approximate per-flow buffering (Stoica et al., 1998).

2.2. Dynamic model of the bottleneck queue

We assume that each switch/router output link maintains *per-VC first in–first out* (FIFO) queuing. The dynamic model of a network queue is a simple integrator. Let $x_{ij}(t)$ be the queue level associated with connection (S_i, D_i) and link l_j, let $u_i(t) \geq 0$ be the inflow rate due to the ith VC, and let $b_{\mathrm{av},ij}(t) \geq 0$ be the bandwidth which is available at link l_j for the ith VC. By writing the flow conservation equations, the queue level $x_{ij}(t)$, starting at $t = 0$ with $x_{ij}(0) = 0$, is

$$x_{ij}(t) = \int_0^t [u_i(\tau - T_{ij}) - d_{ij}(\tau)]\,\mathrm{d}\tau,$$

where T_{ij} is the propagation delay from the ith source to the jth queue, and $d_{ij}(t) = b_{\mathrm{av},ij}(t)h(x_{ij})$ is the depletion rate of the jth queue, with

$$h(x_{ij}) = \begin{cases} 1 & \text{if } x_{ij}(t) > 0, \\ \alpha(t)/b_{\mathrm{av},ij}(t) & \text{if } x_{ij}(t) = 0, \end{cases}$$

where $\alpha(t) = \min(u_i(t - T_{ij}), b_{\mathrm{av},ij}(t))$.

Notice that the available bandwidth $b_{\mathrm{av},ij}(t)$ depends on the global traffic loading the link and that it results $x_{ij}(t) \geq 0$.

3. Classical control approach to model a flow-controlled data network

In this section, the dynamics of network data queues in response to input traffic is described using a classical control approach based on transfer functions. Data packets go from source to destination. At the destination, control packets are relayed back to the source. Control packets are resource management (RM) cells interleaved with data cells in ATM networks and acknowledgment packets in Internet. They carry the feedback information to the source and make possible to operate a feedback control (ATM Forum, 1996; Jacobson, 1988).

We assume that a connection establishes a virtual circuit with per-VC buffering at the node output links. A particular link along the VC path will be the bottleneck. The queue feeding this link will be the bottleneck queue for the flow, whereas other queues encountered by the flow will be empty. Thus, the control goal reduces to fully utilize the bottleneck link whereas overflow of the bottleneck queue must be avoided. The dynamics of the bottleneck queue level in response to the source input rate is a *single-input–single-output dynamics*. Fig. 3 shows the block diagram of the closed-loop dynamics. In particular, it consists of:

(1) An integrator described in the Laplace domain by the transfer function $1/s$. It models the per-flow buffer which is the bottleneck for the considered flow. The output $x_{ij}(t)$ is the bottleneck queue length of connection i at link j.

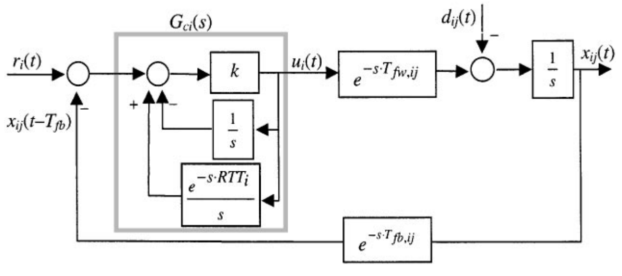


Fig. 3. Block diagram of the dynamics of the controlled bottleneck queue.

(2) A disturbance $d_{ij}(t)$, which models the "best-effort" bandwidth that is available for the ith flow at bottleneck link j. We assume that it is not possible to measure $d_{ij}(t)$. Notice that $d_{ij}(t)$ is modeled as an unknown deterministic function rather than a stochastic process. By means of such a disturbance $d_{ij}(t)$, all traffic scenarios are easily modeled.
(3) The transfer function $e^{-sT_{\mathrm{fw},ij}}$, which models the propagation time $T_{\mathrm{fw},ij}$ from the VC_i source to the bottleneck queue at link j.
(4) The transfer function $e^{-sT_{\mathrm{fb},ij}}$, which models the propagation time $T_{\mathrm{fb},ij}$ from the bottleneck queue at link j to the destination and then back to the source i.
(5) The controller transfer function $G_{\mathrm{ci}}(s)$.
(6) The source input rate $u_i(t)$.
(7) The reference signal $r_i(t)$.

The feedback control scheme consists of two inputs: the reference signal $r_i(t)$ and the disturbance $d_{ij}(t)$. Referring to the reference signal $r_i(t)$, the source compares $r_i(t)$ with the delayed bottleneck queue level $x_{ij}(t - T_{\mathrm{fb},ij})$ and then inputs the difference into the controller $G_{\mathrm{ci}}(s)$, which calculates the input rate. The input rate, which is computed at the source, reaches the queue after the forward propagation delay $T_{\mathrm{fw},ij}$, whereas the queue level reaches the destination and is relayed back to the source after the backward propagation delay $T_{\mathrm{fb},ij}$. Notice that the *round trip time* (RTT_i) of the VC_i connection is always $RTT_i = T_{\mathrm{fw},ij} + T_{\mathrm{fb},ij}$ wherever the connection bottleneck link j may be positioned along the VC_i path. This means that the proposed *control scheme models also the general case of bottleneck that moves along the VC path.* In case the contribution of the bottleneck queuing time to the RTT cannot be neglected, then the queuing time can be considered part of $T_{\mathrm{fb},ij}$. Referring to the disturbance $d_{ij}(t)$, it directly empties the bottleneck queue.

Due to the possibly large propagation delays in the control loop, queue level dynamics might exhibit oscillations, and even become unstable. Therefore, the design of the linear controller $G_{\mathrm{ci}}(s)$ must be carried out carefully.

In the next section, we propose to design the controller $G_{\mathrm{ci}}(s)$ following the Smith principle (Åström & Wittenmark, 1984; Marshall, 1979). The Smith predictor is well known as an effective dead-time compensator for a stable process with large time delay. The main advantage of this technique is that the time delay is eliminated from the characteristic equation of the closed-loop system. Thus the design problem for the process with delay can be transformed to the one without delay. A problem is that it cannot be used for processes having an integral mode since a constant disturbance will result in a steady-state error. An interesting new modified Smith's predictor has been proposed in Åström, Hang and Lim (1994) to uncouple the setpoint response from the disturbance response. This allows the disturbance be rejected without measuring it. In this paper we choose the classical Smith's predictor for the following reasons: (1) the rejection of the disturbance is not appropriate in the context of the design problem herein discussed (see Section 4); (2) Smith's predictor can be easily transformed to discrete-time form (Åström & Wittenmark, 1984).

3.1. The reference signal and the disturbance

The controlled system reported in Fig. 3 is a SISO system with a disturbance. The input is the reference signal $r_i(t)$, which sets a threshold for the bottleneck queue length. The output is the queue level $x_{ij}(t)$. We consider the reference signal $r_i(t) = r_i^o(t - T_{\mathrm{fb}})$. Thus, we can think that the source receives the feedback $(r_i^o(t - T_{\mathrm{fb}}) - x_{ij}(t - T_{\mathrm{fb}}))$ from the bottleneck. This value represents the space that is free at the bottleneck buffer.

The bottleneck link transmission capacity is normalized to unity so that, if all link bandwidth is suddenly available for a single flow a $t = t_0$, then the available bandwidth $b_{\mathrm{av}}(t)$ is equal to the step function[1] $1(t - t_0)$. Coexisting flows reduce the bandwidth that is available for the considered flow. Letting $b(t) \le 1$ be the bandwidth used by coexisting flows, it results

$$0 \le b_{\mathrm{av}}(t) \le 1(t) - b_m 1(t) = a1(t),$$

where $b_m = \min_t \{b(t)\}$, $0 \le b_m \le 1$ and $a = (1 - b_m) \le 1$.

Thus, we model the available bandwidth via a deterministic, unknown and bounded function that represents a *worst-case disturbance*. More precisely, we consider the general worst-case disturbance of the form

$$d_{ij}(t) = \sum_{i=1}^{p} a_i 1(t - T_i) \tag{1}$$

where $T_j > T_i$ if $j > i$, $a_1 \in (0,1]$, $a_i \in [-1,0) \cup (0,1]$ for $i > 1$, and $0 \le \sum_{i=1}^{h} a_i \le a$, $\forall h \in N$ with $h \le p \in N$.

Disturbance (1) represents a general piece-wise constant available bandwidth with values that suddenly change at instants T_i. From a practical point of view, it can model any traffic scenario loading the bottleneck link.

4. The control law

The objective of the control law is to guarantee that the source input rate promptly utilizes all available bandwidth. At the same time, buffer overflow must be avoided. These goals can be formally stated via the two following conditions:

(1) *Stability condition*:

$$x_{ij}(t) \le r^o \quad \text{for } t > 0, \tag{2}$$

[1] The step function is

$$1(t) = \begin{cases} 1 & \text{if } t \ge 0, \\ 0 & \text{if } t < 0. \end{cases}$$

where r^o is the bottleneck queue capacity, which guarantees that this queue is always bounded, i.e. no packet loss, and

(2) *Full link utilization*:

$$x_{ij}(t) > 0 \quad \text{for } t \geq RTT, \tag{3}$$

that guarantees full utilization of the bottleneck link, i.e. $b_{\mathrm{av},ij}(t) = d_{ij}(t)$, because the link has always data to send. The reason of RTT in condition (2) is that the feedback cannot reach the source in a time less than the backward propagation delay and the input rate cannot reach the bottleneck queue in a time less than the forward propagation delay. As a consequence, it is never possible to reject the disturbance during the transient, which is the normal operation mode of communication networks. For these considerations and in contrast to Charny et al. (1995), Jain et al. (1996) and Zhao et al. (1997), our control goal is not disturbance rejection but is to guarantee that the bottleneck queue level is always greater than zero and less than the queue capacity.

The plant consists of a forward delay in cascade with an integrator, and of a backward delay. We assume that delays are known[2] so that a controller can be successfully designed following the Smith principle (Marshall, 1979).

The Smith principle suggests to look for a controller such that the closed-loop controlled system, which contains a delay in the control loop, becomes equivalent to a system with the delay pushed out of the control loop. In this case our idea is to look for a controller $G_{\mathrm{ci}}(s)$ so that the input–output dynamics of the controlled system reported in Fig. 3 becomes equal to the input–output dynamics of the system reported in Fig. 4. The desired input–output dynamics reported in Fig. 4 has been carefully chosen so that:

(a) the closed-loop part of the desired system is delay free, that is, the delay is pushed out of the feedback loop and does not affect stability;
(b) the desired system is a *simple first-order system* with a delay in cascade, that is, the transfer function is $[k/(s+k)]\mathrm{e}^{-T_{\mathrm{fw}}\cdot s}$. The first-order system is obtained by choosing a simple proportional controller k for the delay-free plant ($1/s$). Notice that this system is asymptotically stable for any $k > 0$ and has no overshoots.
(c) a controller $G_{\mathrm{ci}}(s)$ exists that renders the input–output dynamics of the system reported in Fig. 3 equal to the input–output dynamics of the target system reported in Fig. 4.

Now we derive the controller that gives the desired input–output dynamics reported in Fig. 4.

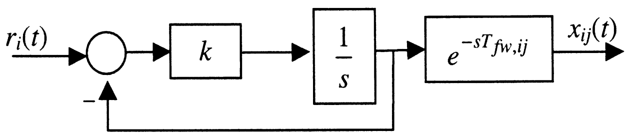


Fig. 4. Desired input–output dynamics.

Proposition 1. *The transfer functions $X(s)/R(s)$[3] of the systems shown in Figs.* 3 *and* 4 *are made equal by using the controller described by the transfer function*

$$G_{\mathrm{c}}(s) = \frac{k}{1 + (k/s)(1 - \mathrm{e}^{-RTT\cdot s})}. \tag{4}$$

Proof. By equating the transfer functions of the systems reported in Figs. 3 and 4

$$\frac{(G_{\mathrm{c}}(s)/s)\mathrm{e}^{-T_{\mathrm{fw}}\cdot s}}{1 + (G_{\mathrm{c}}(s)/s)\mathrm{e}^{-RTT\cdot s}} = \frac{k/s}{1 + k/s}\mathrm{e}^{-T_{\mathrm{fw}}\cdot s},$$

controller (4) is derived after a little algebra. □

Looking at controller (4) shown in Fig. 3, it is easy to write the rate control equation

$$u(t) = k\left(r(t - T_{\mathrm{fb}}) - x(t - T_{\mathrm{fb}}) - \int_{t-RTT}^{t} u(\tau)\,\mathrm{d}\tau \right). \tag{5}$$

This equation can be intuitively interpreted as follows: the computed input rate is proportional, through the coefficient k, to the free space in the bottleneck queue, that is $r(t - T_{\mathrm{fb}}) - x(t - T_{\mathrm{fb}})$, decreased by the number of cells released by the source during the last round trip time RTT. It is worth noting that the role of the Smith predictor is to take into account the "in flight" cells given by the integral $\int_{t-RTT}^{t} u(\tau)\,\mathrm{d}\tau$.

In the next section we show via mathematical analysis that the proposed control law guarantees bottleneck queue stability and full link utilization during both transient and steady-state condition.

5. Mathematical analysis of transient dynamics and steady state

Classical control theory provides an established set of tools that enables us to design algorithms whose performance can be predicted analytically, rather than relying on simulations. To analyze the performance of the proposed algorithm is sufficient to use standard Laplace transform technique. The important advantage of mathematical analysis is that allows us to demonstrate the

[2] We will further discuss this assumption when we will apply the designed control law to ATM networks and Internet.

[3] From now on, the subscripts in $x_{ij}(t)$, $d_{ij}(t)$, $G_{\mathrm{ci}}(t)$, $u_i(t)$, RTT_i, which are used to refer the jth output link and the ith connection, are dropped.

properties of the proposed control law in a general setting, whereas the validation via computer simulations is inevitably restricted to the simulated scenarios. Notice that the analysis of transient dynamics is extremely important in the context of communication networks because these systems never reach a steady-state condition due to continuous joining and leaving of connections.

Due to the linearity of the controlled system, we can determine the output queue length by superposing the responses $x_r(t)$ and $x_d(t)$ to the inputs $r(t)$ and $d(t)$, respectively. In order to do this, we compute the transfer functions

$$\frac{X_r(s)}{R(s)} = \frac{1}{(1 + s/k)} \mathrm{e}^{-T_{\mathrm{fw}} \cdot s}$$

and

$$\frac{X_d(s)}{D(s)} = -\frac{1}{s} + \frac{k}{s(s+k)} \mathrm{e}^{-RTT \cdot s}.$$

Considering the reference signal $r(t) = r^o \cdot 1(t - T_{\mathrm{fb}})$ and the disturbance $d(t) = a_1 1(t - T_1)$, where $T_1 \geq RTT$ is the instant when the bandwidth a_1 is suddenly available, and by transforming back to time domain $X_r(s)$ and $X_d(s)$, it results

$$x_r(t) = r^o(1 - \mathrm{e}^{-k(t-RTT)})1(t - RTT), \tag{6}$$

$$\begin{aligned} x_d(t) = &- a_1(t - T_1)1(t - T_1) \\ &+ a_1(t - T_1 - RTT)1(t - T_1 - RTT) \\ &- \frac{a_1}{k}(1 - \mathrm{e}^{-k(t-T_1-RTT)})1(t - T_1 - RTT). \end{aligned} \tag{7}$$

It can be noted that $0 < x_r(t) < r^o$ for $t > RTT$, $x_r(RTT) = 0$, $x_d(t) < 0$ for $t > T_1$ and $x_d(T_1) = 0$.

Remark 1. The queue dynamics $x(t) = x_r(t) + x_d(t)$ is characterized by the time constant $\tau = 1/k$. The transient dynamics can be considered exhausted after the time $T_{\mathrm{tr}} = RTT + T_1 + 4\tau$. Thus k can be chosen to influence T_{tr}.

Now we are ready to show that the proposed controller ensures queue stability, i.e. bounded queue, in presence of any bounded, piecewise constant disturbance expressed in the general form (1).

Theorem 1. *Considering the reference signal* $r(t) = r^o \cdot 1(t - T_{f0})\,(t - T_{\mathrm{fb}})$, *disturbance* (1), *and controller* (4), *the bottleneck queue is* stable, *that is* $x(t) < r^o$ *for* $t \in [0, \infty)$.

Proof. Since the system is linear time invariant, the disturbance response is the superposition of the responses to each disturbance $a_i 1(t - T_i)$ for $i = 1, p$, that is

$$\begin{aligned} x_d(t) = \sum_{i=1}^{p} \Big(&- a_i(t - T_i)1(t - T_i) \\ &+ a_i(t - T_i - RTT)1(t - T_i - RTT) \\ &- \frac{a_i}{k}(1 - \mathrm{e}^{-k(t-T_i-RTT)})1(t - T_i - RTT)\Big). \end{aligned}$$

For $t \in [0, T_1]$, $x_d(t) = 0$.

For $t \in (T_1, T_1 + RTT)$, $x_d(t) = \sum_{i=1}^{j} - a_i(t - T_i) < 0$[4] where $T_j < T_1 + RTT$ and $j \geq 1$.

Considering a generic time $t \geq T_1 + RTT$ with $T_j \leq t < T_{j+1}$ and $RTT + T_l \leq t < T_{l+1} + RTT$, $1 \leq l \leq j \leq p$ it results

$$\begin{aligned} x_d(t) = &\sum_{i=l+1}^{j} - a_i(t - T_i) - RTT \sum_{i=1}^{l} a_i \\ &- \tau \sum_{i=1}^{l} a_i(1 - \mathrm{e}^{-k(t-T_i-RTT)}) \end{aligned}$$

with $\sum_{i=l+1}^{j}() = 0$ if $l = j$. From the Proposition A.1 in the appendix, with $\Delta_i = t - T_i$, $\Delta = RTT > \Delta_{l+1} = t - T_{l+1}$, it follows that

$$\sum_{i=l+1}^{j} - a_i(t - T_i) - RTT \sum_{i=1}^{l} a_i < 0$$

and from the Proposition A.2 in the appendix, with $\Delta_i = 1 - \mathrm{e}^{-k(t-T_i-RTT)}$, it follows that

$$-\tau \sum_{i=1}^{l} a_i(1 - \mathrm{e}^{-k(t-T_i-RTT)}) < 0.$$

Therefore, it results $x_d(t) < 0$ for $t > T_1$ and $x_d(t) = 0$ for $0 < t \leq T_1$. Then,

$$x(t) = x_r(t) + x_d(t) \leq x_r(t) < r^o \quad \text{for } t \geq 0$$

that is, the stability condition is always satisfied. □

Remark 2. Theorem 1 states that network queues are stable for any bottleneck buffer capacity r^o. Therefore, data loss can be avoided using any small buffer capacity.

Now we show that the proposed controller ensures full utilization of network links in the presence of disturbance (1) if the bottleneck queue capacity is at least equal to the connection bandwidth-delay product.

Theorem 2. *Considering the reference signal* $r(t) = r^o \cdot 1(t - T_{\mathrm{fb}})$, *disturbance* (1) *with* $T_1 \geq RTT$, *and controller* (4), *the bottleneck link is fully utilized for* $t > RTT$ *if the bottleneck buffer capacity is* $r^o > a(RTT + \tau)$.

[4] From Proposition A.2 in the appendix, with $\Delta_i = t - T_i$, it follows that $\sum_{i=1}^{j} - a_i(t - T_i) < 0$.

Proof. For $t\in[0,RTT]$, $x(t)=x_r(t)=0$. For $t\in(RTT,T_1]$ $x(t)=x_r(t)>0$. For $t\in(T_1,T_1+RTT)$:

$$x(t)=x_r(t)-\sum_{i=1}^{j}a_i(t-T_i)$$
$$>r^o(1-\mathrm{e}^{-k(t-RTT)})-a(t-T_1)\text{[5]}$$

which is greater than zero for $T_1>\tau\ln(r^o/(r^o-aRTT))$.

Considering a generic time $t\geq T_1+RTT$ with $T_j\leq t<T_{j+1}$ and $RTT+T_l\leq t<T_{l+1}+RTT$, $1\leq l\leq j\leq p$, from Proposition A.3 shown in the appendix, with $\Delta_i=t-T_i$, $\Delta=RTT>\Delta_{l+1}=t-T_{l+1}$ it follows that

$$\sum_{i=l+1}^{j}-a_i(t-T_i)-RTT\sum_{i=1}^{l}a_i>-aRTT$$

and from Proposition A.4, where now $\Delta_i=1-\mathrm{e}^{-k(t-T_i-RTT)}$ it follows that

$$-\sum_{i=1}^{l}a_i(1-\mathrm{e}^{-k(t-T_i-RTT)})>-a(1-\mathrm{e}^{-k(t-T_1-RTT)}).$$

Thus it results

$$x(t)=x_r(t)+x_d(t)>r^o(1-\mathrm{e}^{-k(t-RTT)})-aRTT$$
$$-a\tau(1-\mathrm{e}^{-k(t-T_1-RTT)}).$$

If $-r^o\mathrm{e}^{-k(t-RTT)}+a\tau\mathrm{e}^{-k(t-T_1-RTT)}\geq 0\Leftrightarrow T_1\geq\tau\ln(r^o/a\tau)$, then

$$r^o(1-\mathrm{e}^{-k(t-RTT)})-aRTT-a\tau(1-\mathrm{e}^{-k(t-T_1-RTT)})$$
$$\geq r^o-a(RTT+\tau),$$

that gives

$$x_r(t)+x_d(t)\geq r^o-a(RTT+\tau)>0$$
$$\text{for } r^o>a(RTT+\tau).$$

Note that, if $r^o=a(\tau+RTT)$ and $T_1=RTT$, the relation $T_1>\tau\ln(r^o/a\tau)$ is always true. In fact, $T_1/\tau>\ln(1+T_1/\tau)\Leftrightarrow\mathrm{e}^{T_1/\tau}>1+T_1/\tau$. Moreover it results: $\tau\ln(r^o/(r^o-aRTT))=\tau\ln(r^o/a\tau)$.

Now we consider the case $T_1<\tau\ln(r^o/a\tau)$.

For $t\in[T_1,RTT+T_1)$, $x_d(t)=-\sum_{i=1}^{k}a_i(t-T_i)>-a(t-T_1)$ and $x(t)=x_r(t)+x_d(t)>x_r(t)-a(t-T_1)=z_1(t)$.

Noting that $z_1(t=T_1)=x_r(T_1)\geq 0$ and that $\dot{z}_1(t)=kr^o\mathrm{e}^{-k(t-RTT)}-a>0$ for $T_1<\tau\ln(r^o/a\tau)$, it results $z_1(t)\geq 0$ and it can be concluded that $x(t)>z_1(t)\geq 0$.

Considering a generic time $t\geq T_1+RTT$ with $T_j\leq t<T_{j+1}$ and $RTT+T_l\leq t<T_{l+1}+RTT$, $1\leq l\leq j\leq p$, from Propositions A.3 and A.4 it results

$$x(t)=x_r(t)+x_d(t)>r^o(1-\mathrm{e}^{-k(t-RTT)})-aRTT$$
$$-a\tau(1-\mathrm{e}^{-k(t-T_1-RTT)})=z_2(t),$$
$$z_2(t=T_1+RTT)=z_1(T_1+RTT)>0$$

and

$$\dot{z}_2(t)=kr^o\mathrm{e}^{-k(t-RTT)}-a\mathrm{e}^{-k(t-T_1-RTT)}>0$$
$$\text{for } T_1<\tau\ln(r^o/a\tau)\Rightarrow z_2(t)>0$$

and it can be concluded that $x(t)\geq z_2(t)>0$. Thus, the queue is always greater than zero for $t>RTT$ if the bottleneck buffer capacity is $r^o>a(RTT+\tau)$. □

Remark 3. The result of Theorem 2 is well known in TCP community (Villamizar & Song, 1995), although, at our best knowledge, it has not been derived via mathematical analysis. Notice that the condition $r^o=a(\tau+RTT)$, which gives the steady-state queue level $x_s=x(\infty)=r^o-a(\tau+RTT)=0$, also ensures full link utilization. In fact, control law (5) gives the following steady-state input rate $u_s=u(\infty)=k(a(\tau+RTT)-u_sRTT)\Rightarrow u_s=a$. Moreover notice that, in this case, the queuing delay is zero.

6. Application to ATM networks

In this section the control law (5) is applied to control ABR traffic in ATM networks. Moreover, a discussion on the dynamics of a bottleneck queue controlled via the ERICA algorithm, proposed by Jain et al. (1996), is carried out using transfer functions.

The ATM Forum (1996) prescribes that an ABR source must send one RM cell (Resource Management cell) every NRM data cells ($NRM=32$) for conveying the feedback information. We assume that RM cells have priority over data cells at the queues. With this assumption the queuing time is zero and, as a consequence, the *round trip time is constant* and equal to the propagation time. The round trip time is measured by marking RM cells with a timestamp. Each switch encountered by the RM cell along the VC path stamps on the RM cell the available buffer space $(r^o(t-T_{\mathrm{fb}})-x(t-T_{\mathrm{fb}}))$. At the destination, the RM cell comes back to the source conveying the minimum available buffer space encountered along the path. Upon receiving of this information, the source updates its input rate. Notice that the feedback information is relayed in cells and thus not available in continuous time, but rather in sampled form. This is not a problem since the discrete time implementation of the Smith predictor is simpler than the continuous one (Åström & Wittenmark, 1984). A problem arises because the feedback supplied by RM cells is not received at

[5] From Proposition A.4 in the appendix it follows that $-\sum_{i=1}^{j}a_i(t-T_i)>-a(t-T_1)$.

a constant rate but rather at a frequency that is proportional to the input rate (i.e. input rate/NRM), whereas digital control assumes that signals are sampled at constant rate. As shown in the next subsection, the simplicity of the proposed control law allows this problem to be easily solved.

6.1. Discrete-time control equation

To discretize the continuous control equation (5), we simply invoke the Shannon sampling theorem and the rule of thumb reported in Åström and Wittenmark (1984) which requires that, in order to have a "continuous like" performance of the system under digital control, the ratio of the system time constant $\tau = 1/k$ over the sampling time T_s must belong to the interval [2,4], that is $T_s \in [\tau/4, \tau/2]$. However, the mechanism of conveying feedback via RM cells cannot guarantee a constant sampling rate. To overcome this problem we propose to *estimate* the feedback in case it is not available after that the sampling time is elapsed. Thus the algorithm must compute the input rate at least every T_s regardless whether the source gets the feedback information or not. Two cases need to be considered:

(1) If the source receives at $t = t_h$ the RM cell, then the input rate is computed as

$$u(t_h) = k\left(r^o(t_h - T_{fb}) - x(t_h - T_{fb}) - \sum_{i=1}^{m} u(t_{h-i})\Delta_i \right),$$

where $\Delta_i \le T_s$, and $t_{h-m} = t_h - RTT$.

(2) If the time T_s since the last RM cell was received at time t_{h-l} expires and no RM cell has been yet received, then the source performs a *worst-case estimate* of the feedback information at $t_h = t_{h-1} + T_s$. To be conservative and to prevent cell loss, we assume that in the time interval $[t_{h-1}, t_{h-1} + T_s]$ the queue has zero output rate. Thus, the worst-case estimate of the bottleneck free space is the last received or the last estimate minus what the source has pumped into the network during the interval $[t_{h-1} - RTT, t_{h-1} - RTT + T_s]$. For the sake of simplicity and without loss of generality we assume that in this interval the rate is constant and equal to $u(t_{h-1} - RTT)$. Thus the estimate is

$$\begin{aligned} & r^o(t_h - T_{fb}) - x(t_h - T_{fb}) \\ & \quad = r^o(t_{h-1} - T_{fb}) - x(t_{h-1} - T_{fb}) - u(t_{h-1} - RTT)T_s \end{aligned}$$

and the rate is computed as

$$\begin{aligned} u(t_h) &= k\Big(r^o(t_{h-1} - T_{fb}) - x(t_{h-1} - T_{fb}) \\ &\quad - u(t_{h-1} - RTT)T_s - \sum_{i=1}^{m} u(t_{h-i})\Delta_i \Big) \\ &= u(t_{h-1}) - ku(t_{h-1})\cdot \mathrm{T}_s. \end{aligned} \tag{8}$$

If the feedback is not received in the interval $[t_{h-m}, t_h]$, then the calculated rate is

$$u(t_h) = u(t_{h-1})(1 - kT_s) = u(t_{h-m})(1 - kT_s)^m. \tag{9}$$

Noting that $kT_s \in [0.25, 0.5]$, it results that $(1 - kT_s) < 1$. Therefore, in case of missing feedback, Eq. (9) implements a *multiplicative decrease algorithm*. When a new feedback is received the rate will increase because the actual $(r^o(t_h - T_{fb}) - x(t_h - T_{fb}))$ can never be smaller than the worst-case estimate.

Remark 4. The behavior of the multiplicative decrease equation (9) is similar to the multiplicative decrease phase of some proposed ABR algorithms, see Bonomi et al., (1995) and Yin and Hluchyj (1994), and to the multiplicative decrease behavior of the TCP congestion control window when a loss is detected, see Jacobson (1988). Herein we have rigorously derived it as a consequence of the fact that in communication networks the delivery of the feedback at a constant rate is not guaranteed.

Remark 5. Per-flow queuing is not scalable and may be costly to be implemented in high-speed switches at the backbone. Thus, VC merging is still important. To this purpose a good and approximate implementation of control equation (5) can be obtained using FIFO queuing. In this case a switch must supply its total available buffer space B divided by the number of active connections n. With this approximation the max–min fairness can no more be ensured, whereas no data loss and full utilization are still preserved.

6.2. Simulation results

We report the dynamic behavior of a controlled VC characterized by a bandwidth-delay product of 10,000 cells. The bottleneck link capacity is normalized to unity, so that $RTT = 10{,}000$ in time slots. This can correspond to a cross-country connection (≈ 24,000 km round trip) through a Wide Area Network (WAN) with a typical bottleneck link capacity of 155 Mb/s. We choose the constant gain $k = 1/750$ and the sampling time $T_s = (2/5)\tau = 300$. The available bandwidth is $b_{av}(t) = 0.9\cdot \mathrm{I}(t - 10{,}000) - 0.7\cdot \mathrm{I}(t - 45{,}000) + 0.5\cdot \mathrm{I}(t - 65{,}000)$ and the buffer capacity is $r^o = 9700 > 0.9\cdot(10{,}000 + 750) = 9675$. Fig. 5(a) shows that all available bandwidth is utilized, that is $b_{av}(t) = d(t)$. Fig. 5(b) shows that the queue length is upper bounded by 8000 cells and always greater than zero, i.e. 100% link utilization is ensured. Notice that when $d = 0.9$ the queue level is almost zero (from mathematical analysis it results $x_s = 9700 - 9675$) and that when $d = 0.7$ the queue level is in accordance with the theoretical value $x_s = 9700 - 0.7(10{,}750) = 2175$.

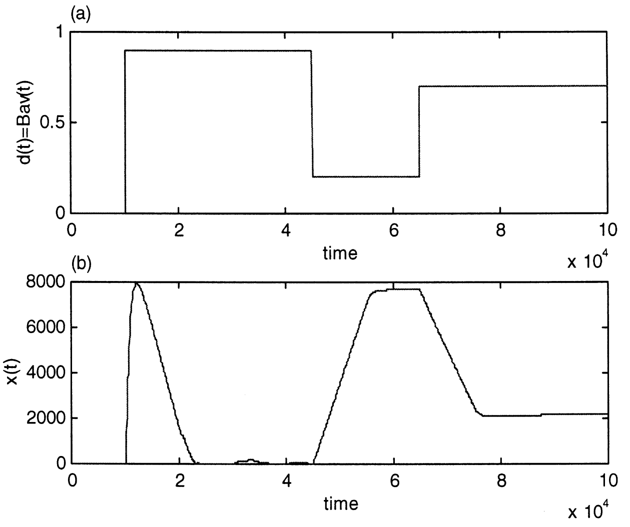


Fig. 5. (a) Per-VC available bandwidth $b_{uv}(t) = d(t)$; (b) bottleneck queue length $x(t)$.

6.3. Discussion on the dynamics of the ERICA algorithm

ERICA is the most known and significant example of explicit rate congestion control algorithm (Jain et al., 1996). Like most of the explicit rate algorithms, it lacks of the analysis of the closed-loop dynamics. We start giving a brief description of ERICA and then we discuss its dynamics using transfer functions.

ERICA maintains FIFO queuing and requires that switches compute an explicit rate, which is fed back to the sources. Thus, the complexity of maintaining per flow buffering is here replaced by the burden of computing the explicit rate. A particularly hard task is the measurement of the available bandwidth ABR_{BW}

$$ABR_{\mathrm{BW}} = \text{Target utilization}$$

$$\times (\text{Link capacity} - (VBR + CBR)_{\mathrm{BW}})$$

which is bursty due to interaction with VBR traffic. Typical values of the target utilization are $0.9 \div 0.95$. Another difficult task is to keep track of the number n of active connections. This number is necessary to compute the Fair-Share $= (ABR_{\mathrm{BW}}/n)$ of each VC.

The computed Fair-Share can be effectively used only if no connection is bottlenecked downstream, see Charny et al. (1995) and Jain et al. (1996). Notice that this is a drawback of all control algorithms executed at network switches. In fact, a switch can never know where the bottleneck of a connection is located. To overcome this problem ERICA defines the load factor $z = ABR_{\mathrm{Inputrate}}/ABR_{\mathrm{BW}})$ and computes the $VC_{\mathrm{Share}} = (CCR/z)$, where CCR is the source current cell rate stored in the RM cells. The load factor z is an indicator of the congestion level of the link. The optimal operating point is $z = 1$. The goal of the switch is to maintain the network at unit overload. Finally, the explicit rate (ER) is computed as follows:

$$ER_{\mathrm{calculated}} \leftarrow \mathrm{Min}(ABR_{\mathrm{BW}}, \mathrm{Max}(FairShare, VC_{\mathrm{Share}})),$$

$$ER_{\mathrm{calculated}} \leftarrow \mathrm{Min}(ER_{\mathrm{in\ RM\ cell}}, ER_{\mathrm{calculated}}).$$

ERICA depends upon the measurements of ABR_{BW} and n which are both difficult to determine in practice. In particular, we note that the Shannon sampling theorem does not allow a signal with high-frequency components, like is the ABR bandwidth, to be reconstructed using the sampling frequency supplied by RM cells. If there are errors in these measurements, queues may become unbounded and the capacity allocated to drain queues becomes insufficient. An enhancement, obtained by using queue length as a secondary metric, has been proposed in Jain et al. (1996). Other drawbacks of ERICA are: (1) an oscillatory behavior due to the decrease/increase mechanism introduced by the load factor; (2) the selection of the target utilization and of the switch measurement interval; (3) cases in which the algorithm does not converge to max–min fair allocation (Jain et al., 1996).

Now we exploit transfer functions to get an insight into the closed loop dynamics of ERICA. To this purpose we assume that a switch is able to perform an exact calculation of the bottleneck Explicit Fair Rate. The block diagram of the ERICA control algorithm is shown in Fig. 6. It contains: (1) the FIFO queue; (2) the block $F(s)$ that measures the low-pass filtered available bandwidth $b_{\mathrm{avL}}(t)$ (Zhao et al., 1997; Li & Hwang, 1995), and allocates the Fair Share b_{avL}/n to each connection bottlenecked at the queue; (3) the forward and backward propagation delays. Looking at the block diagram in Fig. 6, it is clear that ERICA performs a direct *disturbance rejection*. Moreover, it is easy to see that, due to propagation time, *a bottleneck buffer capacity equal to the bandwidth delay product is necessary to guarantee no cell loss*. In fact, consider n connections with the same round trip time RTT, each one getting the bandwidth a/n. Let us assume that at time t_0 the available bandwidth goes to zero. Then the queue level will increase up to the value

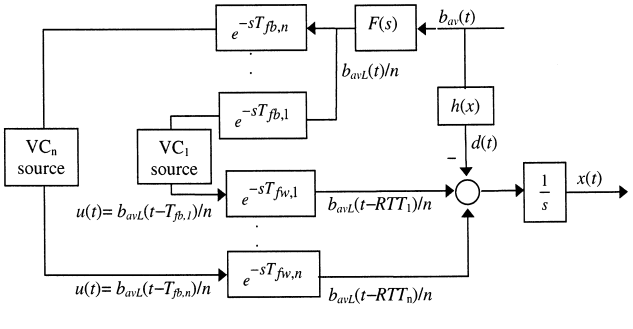


Fig. 6. Block diagram of the FIFO queue dynamics controlled by the ERICA algorithm. The queue is shared by n flows.

$a \cdot RTT$. Thus, even if we assume that switches perform an exact measurement of the Fair Share, which is a hard task in practice, a bottleneck buffer capacity at least equal to the bandwidth delay product is required to guarantee no data loss.

7. Application to the Internet Transfer Control Protocol

The application of the control law (5) to Internet shows that today's Transfer Control Protocol/Internet Protocol (TCP/IP) already implements a Smith's predictor to control congestion. This surprising result gives a theoretical insight into the great success of TCP to control congestion in Internet. Moreover gives useful guidelines to improve the efficiency of the TCP congestion control algorithm. For the sake of completeness, we first give a brief description of the TCP/IP congestion control algorithm and then we apply Eq. (5) to Internet.

7.1. The TCP/IP congestion control algorithm

The TCP congestion control mechanism maintains two variable: the *Advertised window*, which measures the congestion status of the receiver buffer and the *Congestion window* which measures the congestion status of the network. In particular, let *MaxRcvBuffer* be the size of the receiver buffer in bytes, *LastByteRcvd* the last byte received and *NextByteRead* the next byte to be read. To avoid the overflow of the receiver buffer, on the receive side TCP advertises a window size of

$$AdvertisedWindow = MaxRcvBuffer - (LastByteRcvd - NextByteRead)$$

that represents the amount of free space remaining in the receiver buffer. The measure of an appropriate value for the *Congestion Window* is the core of today's TCP/IP. In fact, TCP/IP obeys the principle that the network is a "black box" that cannot supply any *explicit feedback* information to the source. Thus the issue is how TCP, on the send side, must learn an appropriate value for the *Congestion Window*. Van Jacobson (1988) defines an increase/decrease mechanism to throttle the size of the congestion window. In particular, the end-systems probe the network state by gradually increasing the window of packets that are outstanding in the network until the network becomes congested and drops packets. When a packet drop is detected, first the window is shrunk and then it starts to increase until a packet drop is again detected. Finally, the *Effective Window W* that limits how much outstanding data the source can send is computed as follows (see Peterson & Davie, 1996)

$$MaxWindow = MIN(CongestionWindow, AdvertisedWindow), \quad (10)$$

$$W = MaxWindow - (LastByteSent - LastByteAcked) = MaxWindow - OutstandingPackets. \quad (11)$$

Notice that the congestion window size oscillates around its equilibrium value because packet losses are intentionally provoked in order to probe network capacity. This causes link under-utilization when the congestion window is small, and packet loss when the congestion window is large. Moreover, increasing the congestion window until packets are lost fills the buffers and increases delays which are harmful for delay sensitive applications.

7.2. A modified backward compatible TCP/IP congestion control

Now we apply the control equation (5) to TCP/IP. Since TCP/IP uses a window-based control equation, we start by transforming to a window-based control equation the rate-based control equation (5). Eq. (5) can be rewritten as

$$u(t)\tau = r^o(t - T_{\mathrm{fb}}) - x(t - T_{\mathrm{fb}}) - \int_{t-RTT}^{t} u(\tau)\,\mathrm{d}\tau. \quad (12)$$

We can interpret the amount of data $u(t)\tau$ as a *Window* (W) of unacknowledged data that can be sent at time t. The window W represents an impulse of data sent every sampling time τ.

The integral in (12) represents the packets sent by the source and not yet acknowledged, i.e. the *outstanding packets*

$$OutstandingPackets = \int_{t-RTT}^{t} u(\tau)\,\mathrm{d}\tau.$$

The quantity $(r^o(t - T_{\mathrm{fb}}) - x(t - T_{\mathrm{fb}}))$ is the space that is free at the bottleneck buffer. It is the *minimum free space* over all the buffers encountered along the connection path, including the Receiver's Buffer. In other words, letting B_{i} be the free space remaining in the ith buffer we let $(r^o(t - T_{\mathrm{fb}}) - x(t - T_{\mathrm{fb}}))$ be the *Generalized Advertised Window* (GAW)

$$GAW = r^o(t - T_{\mathrm{fb}}) - x(t - T_{\mathrm{fb}}) = \min_{\substack{i \in \text{connection} \\ \text{path}}} \{B_i, AdvertisedWindow\}. \quad (13)$$

Every packet, at the destination, carries stamped on its header the minimum amount of free space remaining in the buffers it traverses along the forward path. The minimum between this value and the *Advertised Window* is the *GAW*, which is relayed back to the source in the TCP header of the returning ACK packet. At this point control equation (12) can be rewritten in window form

$$W = GAW - OutstandingPackets. \quad (14)$$

Notice that the window control equation (14) takes into account the jitter in *RTT*. In fact, the number of outstanding packets inherently takes into account the fact that *RTT* is time varying due to queuing time. By setting

$$MaxWindow = MIN(CongestionWindow, GAW), \quad (15)$$

where *GAW* reduces to the *Advertised Window* in today's TCP, Eq. (11) utilizes the *GAW* in a way that is completely backward compatible with the existing TCP, which is important to enable interoperability. In fact, the *GAW* feedback is stored where today's TCP stores the *Advertised Window* so that the TCP on the send side reads the *GAW* instead of the *Advertised Window* and runs completely unchanged. Moreover, by substituting the *GAW* in (15) with an *estimated*(*GAW*) obtained from any defined function of the congestion status of the network, the resulting congestion control algorithm is still backward compatible. For instance, a router of total storage capacity *B* which maintains FIFO queuing could estimate the *GAW* as follows:

$$estimated(GAW) = B/N, \quad (16)$$

where *N* is the number of queued flows, or the source could infer an estimate of the *GAW* using round trip time measurements.

Finally notice that Eq. (11) is identical to Eq. (14) except for the *GAW* that in (11) is estimated via the Eq. (10). Thus, it can be concluded that the TCP already implements a Smith's predictor to avoid the overflow of the receiver's buffer and to control network congestion with the only difference that the *GAW* is estimated using the (10). This surprisingly result gives a theoretical insight into today's TCP congestion control algorithm and into its success to control congestion since, at our best knowledge, it is the first derivation of the TCP window control equation which is based on control theory. Moreover, the fact that TCP implements a Smith predictor gives an important validation to the control law designed in this paper because nowadays TCP is the only congestion control algorithm running and successfully tested in a real wide-world network.

Remark 6. In the context of Internet, differentiated services (DS) have been designed for scalability through handling aggregates of traffic instead of individual flows as in the integrated services. However, Chow and Leon-Gracia, (1999) has observed that the DS mechanism can hardly achieve the desired QoS and may be unfair. Stoica et al. (1998) remark that Fair Queuing have many desirable properties for congestion control in the Internet. However, such mechanism needs to manage buffers and maintains state on a per-flow basis and this complexity may prevent it from being cost-effectively implemented in routers. For these considerations Stoica et al. (1998) propose an interesting architecture to approximately achieve fair bandwidth allocation. The approach distinguishes between edge routers, which maintain per-flow state, and high-speed core routers which do not maintain per-flow state and use FIFO queuing (see Fig. 7). In our case, Eqs. (11) and (15) can be directly implemented in such architecture where edge routers are allowed to supply *GAW* and coexist with core routers non *GAW* enabled, or enabled to supply an estimated *GAW*, for instance, (16). Finally, notice that *GAW* enabled router could be used for connecting wired Internet to mobile hosts (Balakrishnan et al., 1996).

7.3. Simulation results

Here we report simulation results of TCP using *GAW*. The bandwidth-delay product of the considered TCP connection is 1000 packets, which can correspond to 1000 outstanding TCP segments of 1200 bytes each, for instance, along a high-capacity packet satellite channel or along a terrestrial fiber-optical path of roughly 6.000 km (Jacobson et al., 1997). The bottleneck link capacity is normalized to unity so that the round trip time is $RTT = 1000$ time slots. We assume a sampling time $T_s = RTT$, a gain $k = 1/2500$, a buffer capacity $r^o = 1 \cdot (\tau + RTT) = 3500$ packets. Fig. 8(a) shows the bandwidth that is available for the considered flow. Fig. 8(b) shows that the bottleneck queue length is less than 3500 packets and greater than zero except during two very short intervals. Fig. 8(c) shows that the bandwidth is unused during these two short intervals. Fig. 8(d) shows the control window time behavior. The steady-state value of the control window is $W_s = 1000$, which corresponds to a throughput of $W_s/T_s = W_s/RTT = 1$.

Remark 7. In digital control systems the sampling is performed via impulse modulation and a hold operation, which maintains the input rate constant between sampling instants (Åström & Wittenmark, 1984). In the case of TCP window control equation (14), the hold operation is omitted and the input rate is impulsive. As a consequence, the queue level oscillates in steady-state condition. This is a well-known aspect of today's TCP traffic

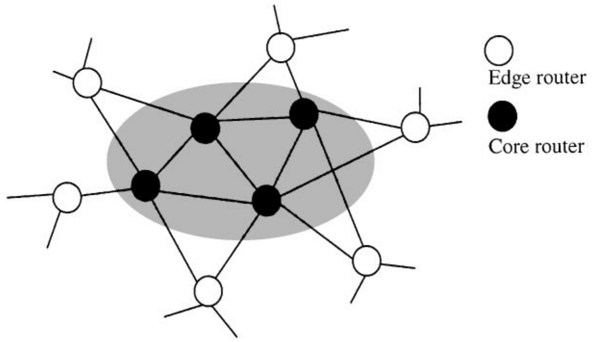


Fig. 7. Edge routers GAW-enabled inter operate with high-speed core routers non-GAW-enabled.

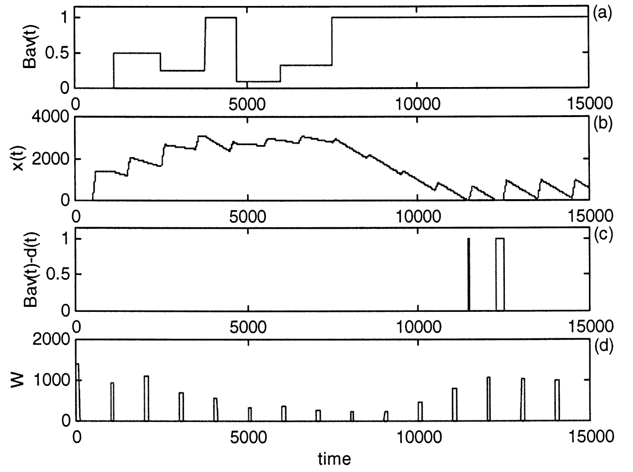


Fig. 8. (a) Available bandwidth $b_{\mathrm{av}}(t)$; (b) bottleneck queue length $x(t)$; (c) unused bandwidth $b_{\mathrm{av}}(t) - d(t)$; (d) effective Window W.

that is bursty due to its window-based control. Bursty traffic originates intense buffer oscillations with the drawback of overflows and link under-utilization.

Remark 8. Gerla et al. (1999) contains extensive simulations of TCP using *GAW* with a head-to-head comparison with TCP-Reno, with TCP-RED and with TCP using ECN. Two versions of TCP-*GAW* have been simulated, one that uses FIFO queuing and the other that uses per-flow buffering. Both these implementations of TCP-*GAW* perform better than the other schemes providing smoother network operation and increased fairness.

8. Conclusions

Classical control theory has been proposed for modeling the dynamics of high-speed communication networks. Smith's principle has been exploited to design a congestion control law which guarantees no data loss and full link utilization over communication paths with any bandwidth-delay product. The properties of the proposed control law have been demonstrated via mathematical analysis in a realistic network scenario consisting of multiple "best-effort" flows, characterized by different round trip times, which share available bandwidth with high priority traffic. The advantages of the proposed algorithm can be summarized as follows: (1) it is a simple algorithm; (2) it allows us to analyze transient and steady-state behavior via mathematical analysis; (3) it ensures fast exponential convergence of input rates to stationary values without oscillations or overshoots; (4) it allows us to prove the global stability of queues along with the full and fair utilization of links during both transient and steady-state condition; (5) it does not require the tuning of any control parameter to react to the changing traffic condition; (6) it does not require the measurement of available bandwidth; (7) it can be applied both to ATM networks and Internet.

Acknowledgements

I would like to thank the anonymous reviewers for their constructive comments, which helped to improve the quality of the paper. I would also thank Prof. Bruno Maione for introducing me to time-delay systems and Prof. Mario Gerla for initiating me to flow control in high-speed data networks.

Appendix

Proposition A.1. *If* $0 \le \sum_{i=1}^{h} a_i \;\forall h \in N$, *with* $h \le p \in N$, *then* $0 < \sum_{i=1}^{l} a_i \Delta + \sum_{i=l+1}^{p} a_i \Delta_i$, *where* $1 \le l \le p-1$, $0 < \Delta_i < \Delta_j$ *for* $i > j$, $\Delta \ge \Delta_{l+1}$.

Proof.

$$a_1 + a_2 + \cdots + a_p \ge 0 \Rightarrow -a_p \le a_1 + a_2 + \cdots + a_{p-1}$$

$$\Rightarrow -a_p\Delta_p < (a_1 + a_2 + \cdots + a_{p-1})\Delta_{p-1},$$

$$(a_1 + a_2 + \cdots + a_{p-2})\Delta_{p-1} + a_{p-1}\Delta_{p-1}$$
$$< (a_1 + a_2 + \cdots + a_{p-2})\Delta_{p-2} + a_{p-1}\Delta_{p-1},$$

$$\cdots\;\cdots\;\cdots\;\cdots\;\cdots\;\cdots$$

$$(a_1 + a_2 + \cdots + a_l)\Delta_l + a_{l+1}\Delta_{l+1} + \cdots + a_{p-1}\Delta_{p-1}$$
$$\le (a_1 + a_2 + \cdots + a_l)\Delta + a_{l+1}\Delta_{l+1}$$
$$+ \cdots + a_{p-1}\Delta_{p-1},$$

that is

$$-a_p\Delta_p < (a_1 + a_2 + \cdots + a_l)\Delta + a_{l+1}\Delta_{l+1}$$
$$+ \cdots + a_{p-1}\Delta_{p-1}$$

or

$$0 < \sum_{i=1}^{l} a_i\Delta + \sum_{i=l+1}^{p} a_i\Delta_i. \qquad \Box$$

Proposition A.2. *If* $0 \le \sum_{i=1}^{h} a_i \;\forall h \in N$, *with* $h \le p \in N$, *then* $0 < \sum_{i=1}^{p} a_i \Delta_i$, *where* $0 < \Delta_i < \Delta_j$ *for* $i > j$.

Proof.

$$a_1 + a_2 + \cdots + a_p \ge 0 \Rightarrow -a_p \le a_1 + a_2 + \cdots + a_{p-1}$$

$$\Rightarrow -a_p\Delta_p < (a_1 + a_2 + \cdots + a_{p-1})\Delta_{p-1},$$

$$(a_1 + a_2 + \cdots + a_{p-2})\Delta_{p-1} + a_{p-1}\Delta_{p-1}$$
$$< (a_1 + a_2 + \cdots + a_{p-2})\Delta_{p-2} + a_{p-1}\Delta_{p-1},$$

$$\cdots\;\cdots\;\cdots\;\cdots\;\cdots\;\cdots$$

$$-a_p\Delta_p < a_1\Delta_1 + a_2\Delta_2 + \cdots + a_{p-1}\Delta_{p-1}$$

or

$$0 < \sum_{i=1}^{p} a_i \Delta_i. \qquad \square$$

Proposition A.3. *If* $\sum_{i=1}^{h} a_i \leq a$ *with* $a > 0$, $\forall h \in N$, $h \leq p \in N$, *then* $\sum_{i=1}^{l} a_i \Delta + \sum_{i=l+1}^{p} a_i \Delta_i < a\Delta$, *where* $1 \leq l \leq p-1$, $0 < \Delta_i < \Delta_j$ *for* $i > j$, $\Delta \geq \Delta_{l+1}$.

Proof.

$$\begin{aligned}
&a - a_1 - a_2 - \cdots - a_p \geq 0\\
&\quad \Rightarrow a_p \leq a - a_1 - a_2 - \cdots - a_{p-1}\\
&\quad \Rightarrow a_p \Delta_p < (a - a_1 - a_2 - \cdots - a_{p-1})\Delta_{p-1},\\
&(a - a_1 - a_2 - \cdots - a_{p-2})\Delta_{p-1} - a_{p-1}\Delta_{p-1}\\
&\quad < (a - a_1 - a_2 - \cdots - a_{p-2})\Delta_{p-2} - a_{p-1}\Delta_{p-1},\\
&\ldots\ldots\\
&(a - a_1 - a_2 - \cdots - a_l)\Delta_{l+1} - a_{l+1}\Delta_{l+1}\\
&\qquad - \cdots - a_{p-1}\Delta_{p-1}\\
&\quad \leq (a - a_1 - a_2 - \cdots - a_l)\Delta - a_{l+1}\Delta_{l+1}\\
&\qquad - \cdots - a_{p-1}\Delta_{p-1},
\end{aligned}$$

or

$$\sum_{i=1}^{l} a_i \Delta + \sum_{i=l+1}^{p} a_i \Delta_i < a\Delta. \qquad \square$$

Proposition A.4. *If* $\sum_{i=1}^{h} a_i \leq a$, *with* $a > 0$, $\forall h \in N$, *with* $h \leq p \in N$, *then* $\sum_{i=1}^{p} a_i \Delta_i < a\Delta_1$, *where* $0 < \Delta_i < \Delta_j$ *for* $i > j$.

Proof.

$$\begin{aligned}
&a - a_1 - a_2 - \cdots - a_p \geq 0\\
&\quad \Rightarrow a_p \leq a - a_1 - a_2 - \cdots - a_{p-1}\\
&\quad \Rightarrow a_p \Delta_p < (a - a_1 - a_2 - \cdots - a_{p-1})\Delta_{p-1},\\
&(a - a_1 - a_2 - \cdots - a_{p-2})\Delta_{p-1} - a_{p-1}\Delta_{p-1}\\
&\quad < (a - a_1 - a_2 - \cdots - a_{p-2})\Delta_{p-2} - a_{p-1}\Delta_{p-1},\\
&\ldots\ldots\\
&(a - a_1)\Delta_2 - a_2\Delta_2 - \cdots - a_{p-1}\Delta_{p-1}\\
&\quad < (a - a_1)\Delta_1 - a_2\Delta_2 - \cdots - a_{p-1}\Delta_{p-1},
\end{aligned}$$

or

$$\sum_{i=1}^{p} a_i \Delta_i < a\Delta_1. \qquad \square$$

References

Altman, E., Basar, T., & Srikant, R. (1998). Robust rate control for ABR sources. *Proceedings of infocom'98*, San Francisco, CA, USA, vol. 1 (pp. 166-173).

Åström, K. J., Hang, C. C., & Lim, B. C. (1994). A new Smith predictor for controlling a process with an integrator and long dead-time. *IEEE Transactions on Automatic Control*, *39*(2), 343–345.

Åström, K. J., & Wittenmark, B. (1984). *Computer controlled systems.* Englewood Cliffs, NJ: Prentice Hall.

ATM Forum Technical Committee TMWG (1996). ATM forum traffic management specification version 4.0. *af-tm-0056.000.* Available at http://www.atmforum.org.

Balakrishnan, H., Padmanabhan, V. P., Seshan, S., & Katz, R. H. (1996). A comparison of mechanism for improving TCP performance over wireless links. *Proceedings of sigcomm 96* in *ACM computer communication review*, vol. 26, no. 4 (pp. 256–269).

Benmohamed, L., & Meerkov, S. M. (1993). Feedback control of congestion in packet switching networks: The case of a single congested node. *IEEE/ACM Transactions on Networking*, *1*(6), 693–708.

Benmohamed, L., & Wang, Y. T. (1998). A control-theoretic ABR explicit rate algorithm for ATM switches with per-VC queueing. *Proceedings of the IEEE Infocom'98*, S. Francisco, CA, USA, vol. 1 (pp. 183–191).

Bonomi, F., Mitra, D., & Seery, J. B. (1995). Adaptive algorithms for feedback-based flow control in high-speed, wide-area ATM networks. *IEEE Journal on Selected Areas in Communications (JSAC)*, *13*(7), 1267–1283.

Brakmo, L. S., O'Malley, S. W., & Peterson, L. L. (1995). TCP Vegas: End-to-end congestion avoidance on a global internet. *IEEE Journal on Selected Areas in Communications (JSAC)*, *13*(8), 1465–1480.

Charny, A., Clark, D. D., & Jain, R. (1995). Congestion control with explicit rate indication. *Proceedings of IEEE ICC'95*, Seattle, WA, USA, vol. 3 (pp. 1954–1963).

Chow, H. K., & Leon-Gracia, A. (1999). A feedback control extension to differentiated services. *IETF Draft.* Available at http://www.ietf.org.

Ding, W. (1997). Joint encoder and channel rate control of VBR video over ATM networks. *IEEE Transactions on Circuits and Systems for Video Technology*, *7*(2), 266–278.

Fendick, K. W., Rodrigues, M. A., & Weiss, A. (1992). Analysis of a rate-based feedback control strategy for long haul data transport. *Performance Evaluation*, *16*, 67–84.

Floyd, S. (1994). TCP and explicit congestion notification. *ACM Computer Communication Review*, *24*(5), 10–23.

Floyd, S., & Jacobson, V. (1993). Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, *1*(4), 397–413.

Gerla, M., Locigno, R., Mascolo, S., & Weng, W. (1999). *Generalized window advertising for TCP congestion control.* UCLA Technical Report 990012. Available at www.cs.ucla.edu/NRL/.

Hahne, E. L., Kalmanek, C. R., & Morgan, S. P. (1993). Dynamic window flow control on a high-speed wide-area data network. *Computer networks and ISDN systems*, vol. 26 (pp. 29–41). Amsterdam, North-Holland.

Izmailov, R. (1995). Adaptive feedback control algorithms for large data transfer in high-speed networks. *IEEE Transactions on Automatic Control*, *40*(8), 1469–1471.

Jacobson, V. (1988). Congestion avoidance and control. *Proceedings of sigcomm88* in *ACM computer communication review*, vol. 18, no. 4 (pp. 314–329).

Jacobson, V., Braden, R., & Borman, D. (1997). TCP extensions for high performance. *Network Working Group Internet-Draft.* Obsoletes *Request for Comments*: 1323 (1992). Available at http://www.ietf.org.

Jaffe, J. (1981). Bottleneck flow control. *IEEE Transaction on Communications*, *29*(7), 954–962.

Jain, R. (1996). Congestion control and traffic management in ATM networks: Recent advances and a survey. *Computer Networks and ISDN Systems*, *28*(13), 1723–1738.

Jain, R., Kalyanaraman, S., Goyal, R., Fahmy, S., & Viswanathan, R. (1996). The ERICA switch algorithm for ABR traffic management in ATM networks, Part I: Description. Available at http://www.cis.ohio-state.edu/ ~ jain/papers.html.

Kalampoukas, L., & Varma, A. (1998). Explicit window adapatation: A method to enhance TCP performance. *IEEE Proceedings of Infocom 98*, vol. 1 (pp. 242–251).

Keshav, S. (1991). A control-theoretic approach to flow control. *Proceedings of ACM sigcomm91* in *ACM computer communication review*, vol. 21, no. 4 (pp. 3–15).

Le Boudec, J., de Veciana, G., & Walrand, L. (1996). QoS in ATM: Theory and practice. *Proceedings of IEEE conference. on dec. and control'96*, Kobe, Japan, vol. 1 (pp. 773–778).

Li, S. Q., & Hwang, C. (1995). Link capacity allocation and network control by filtered input rate in high speed networks. *IEEE/ACM Transactions on Networking*, *3*(1), 10–15.

Liew, S. C., & Chi-yin Tse, D. (1998). A control-theoretic approach to adapting VBR compressed video for transport over a CBR communication channel. *IEEE/ACM Transactions on Networking*, *6*(1), 42–55.

Marshall, J. E. (1979). *Control of time-delay systems*, London: Peregrinus (Peter).

Mascolo, S. (1997). Smith's principle for congestion control in high speed data networks. *Proceedings of IEEE conference. on dec. & control '97*, S. Diego, CA, vol. 5 (pp. 4595–4600).

Peterson, L. L., & Davie, B. S. (1996). *Computer networks.* San Francisco, CA: Morgan Kaufmann Publishers.

Ramakrishnan, K., & Jain, R. (1990). A binary feedback scheme for congestion avoidance in computer networks with a connectionless network layer. *ACM Transactions on Computer Systems*, *8*(2), 158–181.

Roberts, L. (1994). Enhanced PRCA (Proportional Rate-Control Algorithm). *af-tm 94-0735R1*. Available at http://www.atmform.org.

Stoica, I., Shenker, S., & Zhang, H. (1998). Core-stateless fair queueing: achieving approximately fair bandwidth allocation in high speed networks. *Proceedings of sigcomm98* in *ACM computer communication review*, vol. 28, no. 4 (pp. 118–130).

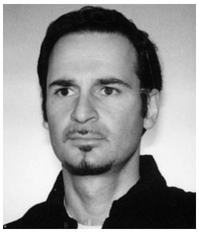
Suter, B., Lakshman, T. V., Stiliadis, D., & Choudhury, A. K. (1998). Design considerations for supporting TCP with per-flow queueing. *IEEE proceeding of infocom 98*, S. Francisco, CA, USA, vol. 1 (pp. 299–306).

Villamizar, C., & Song, C. (1995). High performance TCP in ANSNET. *ACM Computer Communication Review*, *24*(5), 45–60.

Varaiya, P., & Walrand, J. (1996). *High-performance communication networks.* San Francisco, CA: Morgan Kaufmann Publishers.

Yin, N., & Hluchyj, M. G. (1994). On closed-loop rate control for ATM cell relay networks. *Proceedings of infocom 94*, Los Alamitos, CA, USA, vol. 1 (pp. 99–108).

Zhao, Y., Li, S. Q., & Sigarto, S. (1997). A linear dynamic model for design of stable explicit-rate ABR control schemes. *Proceeding of IEEE infocom97*, Kobe, Japan, vol. 1 (pp. 283–292).



Saverio Mascolo was born in Bari, Italy in 1966. He received the Laurea degree, cum laude, in Electronic Engineering in 1991 and the Ph.D. in 1995, both from Politecnico di Bari. During 1995 he was visiting scholar at the Computer Science Department of the University of California at Los Angeles. Since 1996 he is Assistant Professor in Automatic Control at the Electrical and Electronic Department of Politecnico di Bari. His main research interests include flow control in high speed data networks, control of nonlinear systems, modeling and control of discrete event systems and deadlock avoidance. He is a member of IEEE, of IEEE Control System Society and of IEEE Communications Society.