

Reprint

On Mapping Stochastic Processes into Hardware and Its Application on ATM Traffic Emulation

Th. Antonakopoulos, E. Ziouva and V. Makios

IEEE Transactions on Measurements and Instrumentation

VOL. 52, NO. 3, JUNE 2003, pp. 771-779

Copyright Notice: This material is presented to ensure timely dissemination of scholarly and technical work. Copyright and all rights therein are retained by authors or by other copyright holders. All persons copying this information are expected to adhere to the terms and constraints invoked by each author's copyright. In most cases, these works may not be reposted or mass reproduced without the explicit permission of the copyright holder.

On Mapping Stochastic Processes Into Hardware and its Application on ATM Traffic Emulation

Theodore Antonakopoulos, *Senior Member, IEEE*, Eustathia Ziouva, and Vassilios Makios, *Senior Member, IEEE*

Abstract—Asynchronous transfer mode (ATM) traffic emulators have been designed for demonstrating the performance of various protocols and services in ATM networks under user-defined traffic conditions. Analytical tools and traffic measurements show that specific traffic conditions can be emulated by using the superposition of various stochastic processes. This paper presents a generic method for implementing stochastic processes using dedicated hardware and describes how this method has been used for implementing a real-time traffic emulator of ATM virtual channels.

Index Terms—Asynchronous transfer mode (ATM), field programmable gate arrays, stochastic processes.

I. INTRODUCTION

TESTING equipment is used for emulating the behavior of communication systems. Communication systems support services that require transmission rates ranging from a few kbps up to some hundreds of Mbps, depending on the type of data exchanged and on the actual data rate of the emulated communication system. High data rates require the development of dedicated hardware for implementing traffic patterns, usually modeled using stochastic functions [1], since general-purpose hardware is not fast enough to satisfy the application timing requirements. Although this work originated from the need to develop a testing tool for emulating the traffic conditions of multiple virtual channels in asynchronous transfer mode (ATM) networks at 155 Mbps, a generic method for mapping stochastic functions into hardware was developed that can be applied to any information processing system requiring such functionality.

The basic requirement of such a testing tool was to introduce specific inter-cells jitter for evaluating the performance of an MPEG codec under various traffic conditions in an ATM network and to discard cells according to a user-defined cell rejection distribution. The emulation system should introduce cell loss rate and multiple errors in the payload of the cells of each virtual circuit. According to the ATM specifications, single bit errors in the cell header do not affect the system behavior, due to the single-bit error capability of the error correction mechanism used in the cell header [2]. More erroneous bits in the cell header result in cell rejection at the receiver interface. The cell rejection function was implemented by applying a probabilistic function on each cell stream. The modification of the timing characteristics of a cell stream is a more complicated

procedure and is described in details in the next sections. All these functions were implemented by combining a set of stochastic functions in different sections of a virtual channel emulator. As it is shown in Fig. 1, the emulator receives all cells generated in each virtual circuit, and the rejection function is applied in each cell. All cells that have not been discarded at the first stage are temporarily buffered, and a two-stage delay function is applied in order to determine when they have to be forwarded to their destination. Section II presents the mathematical analysis on how specific traffic conditions are related to the delay introduced by the emulator, while some examples for demonstrating the method's applicability are presented in Section III. Section IV presents a new generic method for implementing stochastic processes using dedicated hardware, and Section V presents how this method has been used for implementing a real-time traffic emulator of ATM virtual channels.

II. TRAFFIC CONDITIONS EMULATION

Every communication network is mainly characterized by some statistical variables, like the end-to-end delay and the error rate [3]. The delay probability density function (pdf) of an ATM network is calculated by using various models for describing the cell arrival processes and the performance of queues in ATM switches. According to [1], the timing performance of a newly established connection is determined by the arrival process of its service and the superposition of the traffic of all other existing connections. The superimposed traffic is modeled either as a two-state Markov modulated Poisson process (MMPP) or as a Markov modulated Bernoulli process (MMBP). In either case, the end-to-end delay is characterized by a set of known moments, since an analytic expression is difficult to be derived.

According to [4], the arrival process, either of a single source or from the superposition of multiple sources, is modeled by the State Dependent Discrete Process (SDDP). Using this assumption it has been proven that the delay jitter is limited to a few slots per switching stage in light traffic loading, while in overload conditions or in load transition phases, the jitter depends on the length of buffers used in the ATM switches. The jitter probability distribution can be approached by combining a number of known probability functions in different parts of the jitter scale. The jitter distribution is also determined from the original inter-departure time, although widely spaced cells minimize the cell correlation.

For the rest of this paper, an *additive inter-cell time* is defined as a random variable which is added to the originally generated *inter-departure times* in order to generate specific *inter-arrival times* at the receiver side, so that specific delay distribution

Manuscript received October 18, 2000; revised February 12, 2003.

The authors are with the Department of Electrical Engineering and Computers Technology, University of Patras, Rio Patras, Greece (e-mail: antonako@ee.upatras.gr).

Digital Object Identifier 10.1109/TIM.2003.814672

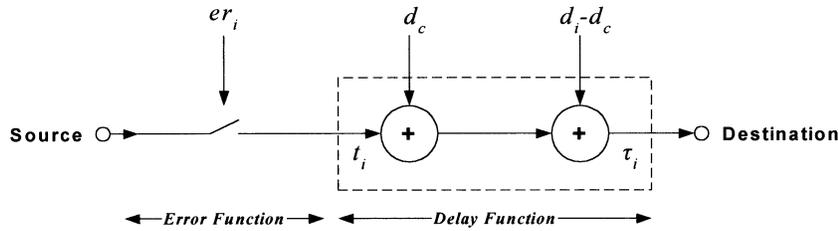


Fig. 1. Error and delay insertion model.

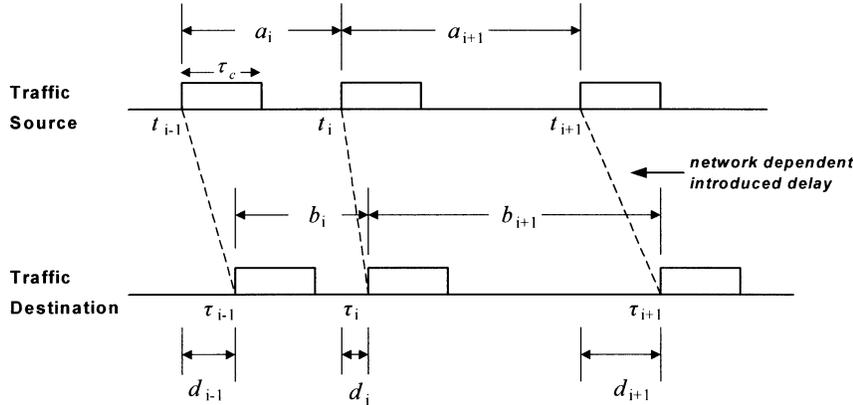


Fig. 2. Inter-arrival time dependence on network introduced delay.

is introduced to the initially transmitted cell stream. The additive delay introduced during transmission affects the inter-arrival times of consecutive cells. In order to form a specific distribution in the arrival of cells of a virtual circuit (VC), we have to determine how the inter-departure times at the source have to be modified and how the delay function is related to the additive inter-cell time function. For deriving the relation between the delay that has to be added in each cell transmission time and the additive inter-cell time, the timing diagram depicted in Fig. 2 is used. The time required for the i -cell to pass from the transmitting unit to the receiving unit is denoted as d_i , the inter-departure time at the source host is denoted as a_i , and the inter-arrival time at the destination host is denoted as b_i . Therefore, $d_i = \tau_i - t_i$, $a_{i+1} = t_{i+1} - t_i$ and $b_{i+1} = \tau_{i+1} - \tau_i$, where t_i is the time when the i -cell is generated at the source and τ_i is the time when the i -cell arrives at the destination. The additive inter-cell time c_{i+1} is given by

$$c_{i+1} = b_{i+1} - a_{i+1} = d_{i+1} - d_i = (d_{i+1} - d_c) - (d_i - d_c) \quad (1)$$

where d_c is a constant offset delay, and its value is equal to $E[d_i]$. Therefore, the network introduced delay is emulated by a two-stage process, where initially a constant delay (d_c) is introduced and then a variable delay function ($d_i - d_c$) is applied having zero mean value.

Equation (1) states that the additive inter-cell time is independent of the constant offset delay, while the introduced delay depends only on the additive inter-cell times. Equation (1) can be written as

$$d_{i+1} = d_i + c_{i+1} \quad (2)$$

and shows that the sequence of random variables $\{d_0, d_1, d_2, \dots\}$ forms a Markov process with stationary

transition probabilities, since the new delay value d_{i+1} depends on the previous sequence of random variables d_n ($n = 0, 1, \dots, i$) only through the most recent value d_i plus a random variable c_{i+1} , which is independent of the random variables d_i .

As stated in [5], such a system is stable if and only if $E[c_i] < 0$, where $E[c_i]$ is the expectation of the additive inter-cell times. Using (1) and the above stability condition, the mean values of the inter-departure and of the inter-arrival processes must satisfy the following inequality

$$E[b_i] < E[a_i]. \quad (3)$$

The cell generation process satisfies the condition

$$t_{i+1} \geq t_i + \tau_c \quad (4)$$

where τ_c is the duration of a cell transmission. Since the initial cell transmission order is preserved, then

$$\tau_{i+1} \geq \tau_i + \tau_c. \quad (5)$$

By following the methodology used in [5], $D(y)$ is defined as the stationary probability distribution function (PDF) for the random variable d_i

$$D(y) = \lim_{i \rightarrow \infty} P[d_i \leq y]. \quad (6)$$

For calculating $D(y)$, we define $C_{i+1}(c)$ as the PDF for the random variable c_{i+1} , that is

$$\begin{aligned} C_{i+1}(c) &= P[c_{i+1} = b_{i+1} - a_{i+1} \leq c] = P[b_{i+1} - a_{i+1} \leq c] \\ &= \int_{a=0}^{\infty} P[b_{i+1} \leq c + a | a_{i+1} = a] dA(a). \end{aligned} \quad (7)$$

However, the inter-arrival time b_{i+1} is independent of the inter-departure time a_{i+1} and, therefore

$$C_{i+1}(c) = \int_{a=0}^{\infty} B(c+a)dA(a) \quad (8)$$

where $A(a)$ and $B(c+a)$ are the PDFs of the inter-departure time a_{i+1} and the inter-arrival time b_{i+1} respectively. Since $C_{i+1}(c)$ is independent of i , then

$$C(c) = C_{i+1}(c) = \int_{a=0}^{\infty} B(c+a)dA(a). \quad (9)$$

We denote as $c(c)$, $b(b)$, and $a(a)$ the pdfs of the additive inter-cell time, the inter-arrival time and the inter-departure time respectively. The integral in (9) is much like a convolution form for $B(b)$ and $A(a)$, and so using the convolution notation (*) and defining, $c(c) = dC(c)/dc$, we get

$$c(c) = a(-c) * b(c). \quad (10)$$

In the above equation, we use $a(-c)$ rather than $a(c)$, because the distribution $C(c)$ represents the difference between the inter-arrival time b_{i+1} and the inter-departure time a_{i+1} rather than their sum. Applying the Laplace transform to (10), we get

$$C^*(s) = A^*(-s) * B^*(s) \quad (11)$$

where $C^*(s)$, $B^*(s)$, and $A^*(-s)$ are the Laplace transforms of $c(c)$, $b(b)$ and $a(a)$, respectively. Equation (11) can be solved by using the G/G/1 queue analysis and spectral methods, as described in [5], and further explained in the Appendix. This equation shows that if the pdfs of the inter-departure and the inter-arrival times are known, we can calculate the pdf of the additive inter-cell time.

III. ATM TRAFFIC EMULATION EXAMPLES

The following two examples apply the above-described methodology on ATM network emulation. ATM networks use asynchronous transfer mode transmission technology that uses fixed-size packets called cells [7]. The data generated by various types of services have different QoS requirements and are adapted by the ATM adaptation layer (AAL) in order to be transmitted efficiently over the ATM network. There are five types of adaptation layer services, designated AAL1, AAL2, etc. AAL1 is used to support synchronous services, like voice, while AAL2 supports variable rate services, like compressed video.

A. AAL1 Example

Let us consider a constant bit rate (CBR) service that generates fixed length data with constant inter-departure times. In this scenario, we consider uncompressed voice transmission at 64 Kbps, supported by AAL type 1. The first byte of the cell payload is used for AAL1 control purposes, and the remaining 47 bytes are used for transmitting the actual user data. Therefore, the inter-departure times at the CBR source are constant and equal to $T = 5.875$ ms. The PDF, the pdf, and the Laplace

transform of the random variable a_i that describes the inter-departure times at the source node are, respectively

$$A(t) = \begin{cases} 0 & t < T \\ 1 & t \geq T \end{cases}, \quad a(t) = \delta(t - T), \quad A^*(s) = e^{-sT} \quad (12)$$

where $\delta(t)$ is the unit impulse function.

We also consider that the inter-arrival times at the destination node have exponential pdf with average arrival rate μ cells/s. In this case the PDF, the pdf, and the Laplace transform of the random variable b_i that describes the inter-arrival times are, respectively

$$B(t) = 1 - e^{-\mu t}, \quad b(t) = \mu e^{-\mu t}, \quad B^*(s) = \frac{\mu}{s + \mu}. \quad (13)$$

Using the previously described methodology and based on the distributions and transforms of (12) and (13), we have to estimate what delay distribution has to be introduced and what additive inter-cell times have to be implemented in the traffic emulator in order to get the desired traffic behavior. According to (3), the system is stable if $E[b_i] = 1/\mu < E[a_i] = T$ and thus $\mu > 170$ cells/s. Substituting the Laplace transforms of a_i and b_i in (11), we get

$$C^*(s) = \frac{e^{sT}\mu}{s + \mu}. \quad (14)$$

Using (14), the PDF and the pdf of the random variable c_i , which describes the additive inter-cell times, are given by

$$\begin{aligned} C(u) &= B(u + T), \quad u \geq -T \\ c(u) &= b(u + T), \quad u \geq -T. \end{aligned} \quad (15)$$

The term $A^*(-s)B(s) - 1 = (e^{sT}\mu/s + \mu) - 1$ is difficult to be expressed in rational functions; therefore we use (21) of the Appendix in order to calculate the delay PDF

$$D(y) = \int_{-\infty}^y D(y-u)dC(u) = \int_{-\infty}^y D(y-u)c(u)du$$

$$D(y-T) = \int_{-\infty}^{y-T} D((y-T)-u)c(u)du.$$

Since $u \geq -T$ and using (15)

$$D(y-T) = \int_{-T}^{y-T} D(y-T-u)b(u+T)du.$$

The above equation is easily transformed into

$$D(y-T) = \int_0^y D(y-u)b(u)du \quad (16)$$

by replacing $u+T$ with u' and finally by using u instead of u' .

By combining equations (13) and (16), and by considering $D(y) = 1 + l_1 e^{l_2 y}$ as a possible solution (l_1 and l_2 are parameters that have to be determined), then

$$\frac{1}{\mu} + \frac{l_1}{\mu + l_2} = 0 \quad \text{and} \quad \frac{\mu}{\mu + l_2} = e^{-l_2 T}.$$

Since $D(0) = 1 + l_1$ and using the above relations, we get that $1 - D(0) = e^{-\mu D(0)T}$, which has a zero root and a nonzero root. Let ρ be the nonzero root of the above equation. Then the PDF of delay is given by

$$D(y) = 1 - (1 - \rho)e^{-\mu \rho y}, \quad y \geq 0.$$

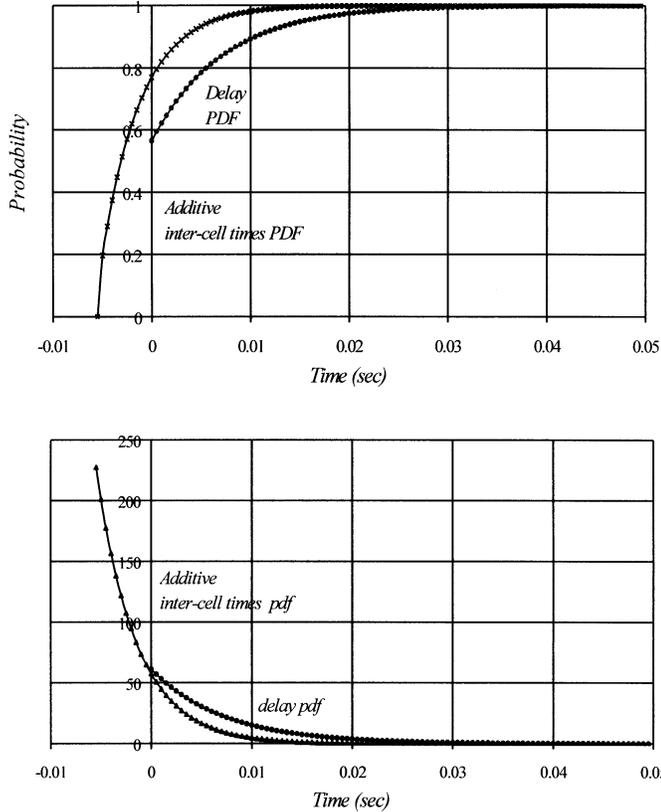


Fig. 3. AAL1 scenario distributions for delay and additive inter-cell times.

Fig. 3 shows the introduced delay and the additive inter-cell time distributions for the AAL1 scenario.

B. AAL2 Example

A variable bit rate (VBR) voice source is modeled by an interrupted Poisson process (IPP), which is alternatively turned on for an exponentially distributed period of time (active period) and turned off for another independent exponentially distributed period of time (silent period). Let $1/\sigma_A$, $1/\sigma_S$ and λ denote the average duration of the active and silent periods and the packet-generation rate during the active period, respectively. According to [1], an IPP with parameters $(\sigma_A, \sigma_S, \lambda)$ is equivalent to the hyper-exponential distribution with parameters $(p_1, \lambda_1, p_2, \lambda_2)$, where

$$\begin{aligned} \lambda_1 &= 0.5 \left\{ (\lambda + \sigma_A + \sigma_S) + [(\lambda + \sigma_A + \sigma_S)^2 - 4\lambda\sigma_A]^{1/2} \right\} \\ \lambda_2 &= 0.5 \left\{ (\lambda + \sigma_A + \sigma_S) - [(\lambda + \sigma_A + \sigma_S)^2 - 4\lambda\sigma_A]^{1/2} \right\} \\ p_1 &= (\lambda - \lambda_2) / (\lambda_1 - \lambda_2) \\ p_2 &= 1 - p_1. \end{aligned}$$

So, the PDF, the pdf, and the Laplace transform of the random variable a_i are, respectively

$$\begin{aligned} A(t) &= 1 - p_1 e^{-\lambda_1 t} - p_2 e^{-\lambda_2 t} \\ a(t) &= p_1 \lambda_1 e^{-\lambda_1 t} + p_2 \lambda_2 e^{-\lambda_2 t} \\ A^*(s) &= \frac{p_1 \lambda_1}{\lambda_1 + s} + \frac{p_2 \lambda_2}{\lambda_2 + s}. \end{aligned}$$

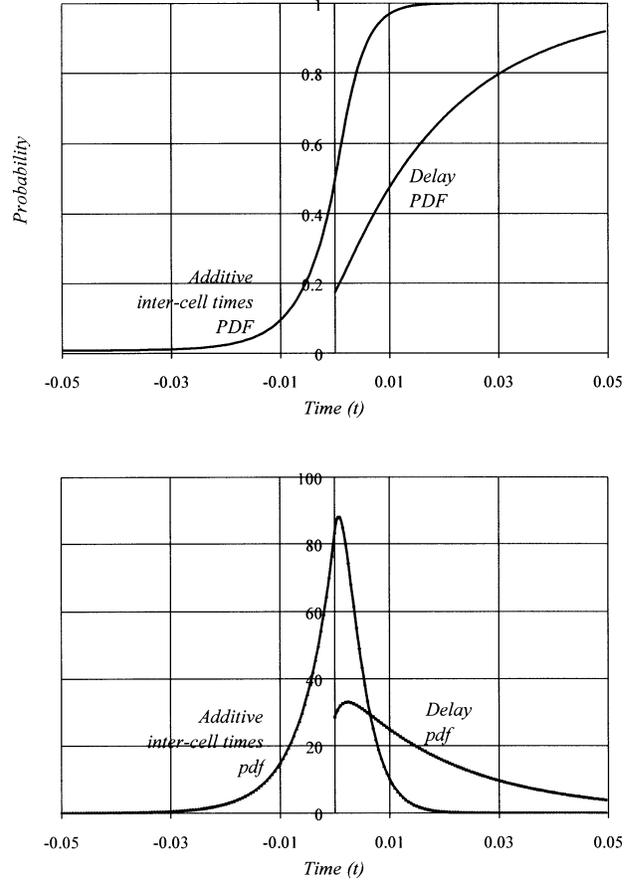


Fig. 4. AAL2 scenario distributions for delay and additive inter-cell times.

For the inter-arrival times at the destination, we consider a two-stage Erlangian distribution with PDF, pdf and Laplace transform, respectively

$$\begin{aligned} B(t) &= 1 - (1 + 2\mu t)e^{-2\mu t} \\ b(t) &= 2\mu(2\mu t)e^{-2\mu t} \\ B^*(s) &= \frac{(2\mu)^2}{(2\mu + s)^2}. \end{aligned}$$

The traffic emulator behaves as the $H_2/E_2/1$ system, and it is stable if $E[b_i] = 1/\mu < E[a_i] = \sum_{i=1}^2 p_i/\lambda_i$. Following the presented method and according to (23) of the Appendix

$$\frac{\Psi_+(s)}{\Psi_-(s)} = A^*(-s)B^*(s) - 1 = \left(\frac{p_1 \lambda_1}{\lambda_1 - s} + \frac{p_2 \lambda_2}{\lambda_2 - s} \right) \frac{(2\mu)^2}{(2\mu + s)^2} - 1. \quad (17)$$

From (17), we have that $\Psi_+(s)/\Psi_-(s) = -sP(s)/(\lambda_1 - s)(\lambda_2 - s)(2\mu + s)^2$, where $P(s)$ is a polynomial of third order

$$P(s) = s^3 + [4\mu - (\lambda_1 + \lambda_2)]s^2 + [\lambda_1 \lambda_2 - 4\mu(\lambda_1 + \lambda_2) + 4\mu^2]s + 4\mu[\lambda_1 \lambda_2 - \mu(p_2 \lambda_1 + p_1 \lambda_2)]$$

and we have to find the roots of $P(s)$. According to Cartesius, $P(s)$ may have up to α positive real roots if the signs of the $P(s)$

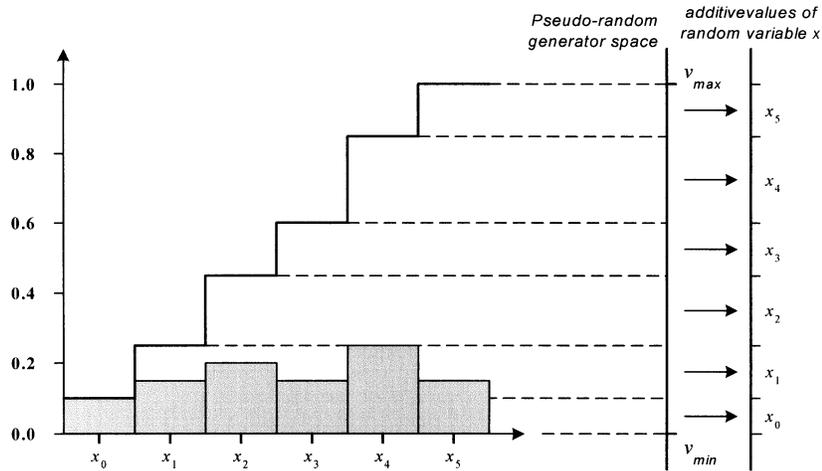


Fig. 5. Projection procedure.

coefficients present α alternations, and $P(s)$ may have up to β negative real roots if the signs of the $P(-s)$ coefficients present β alternations. Using the stability condition for our system, we get

$$\begin{aligned} 4\mu - (\lambda_1 + \lambda_2) &> 0 \\ \lambda_1\lambda_2 - 4\mu(\lambda_1 + \lambda_2) + 4\mu^2 &> 0 \text{ or } < 0 \\ 4\mu[\lambda_1\lambda_2 - \mu(p_2\lambda_1 + p_1\lambda_2)] &< 0. \end{aligned}$$

So, $P(s)$ may have one positive real root and up to two negative real roots. Furthermore, if $a_1 < a_2$ and $P(a_1)P(a_2) < 0$, there is at least one root that belongs to the interval (a_1, a_2) . Since $P(0)P(-2\mu) < 0$ and $P(\lambda_1)P(\lambda_2) < 0$, $P(s)$ has a positive root ρ_1 , where $\lambda_2 < \rho_1 < \lambda_1$ and a negative root $-\rho_2$, where $-2\mu < -\rho_2 < 0$. Let $-\rho_3$ be the third root. Then, in order to satisfy the method's conditions, the functions $\Psi_+(s)$ and $\Psi_-(s)$ are defined as

$$\begin{aligned} \Psi_+(s) &= \frac{s(s + \rho_2)(s + \rho_3)}{(s + 2\mu)^2} \text{ and} \\ \Psi_-(s) &= \frac{-(\lambda_1 - s)(\lambda_2 - s)}{s - \rho_1}. \end{aligned} \quad (18)$$

Substituting $\Psi_+(s)$ from (18) to (24), we find that $\Phi_+(s) = (\rho_2\rho_3/4\mu^2)((s + 2\mu)^2/s(s + \rho_2)(s + \rho_3))$ and the PDF of the delay is equal to

$$D(y) = 1 + \frac{\rho_3(2\mu - \rho_2)^2}{4\mu^2(\rho_2 - \rho_3)} e^{-\rho_2 y} - \frac{\rho_2(2\mu - \rho_3)^2}{4\mu^2(\rho_2 - \rho_3)} e^{-\rho_3 y}, \quad y \geq 0.$$

Using (11), the PDF of the additive inter-cell time is given by the equation at the bottom of the page. Fig. 4 shows the introduced delay and the additive inter-cell time distributions for the AAL2 scenario for $1/\sigma_A = 0.352$, $1/\sigma_S = 0.650$ and $\lambda = 170.21$ cells/sec.

IV. MAPPING METHOD

The basic idea of the method proposed for implementing a stochastic process in hardware is that its pdf must be expressed in such a way that it can be easily realizable in hardware. The PDF of a random variable takes values from 0 to 1 and can be considered as the normalization of an arbitrarily defined function with minimum v_{\min} and maximum v_{\max} values. If a pseudo-random generator produces values between v_{\min} and v_{\max} , then this pseudo-random generator can be associated to every distribution function.

The shape of the PDF is projected to a normalized axis (from 0 to 1) and splits the axis into different areas of discontinuity points, where each area represents a cumulative probability (Fig. 5). The pseudo-random generator produces normalized values from 0 to 1 uniformly distributed, and the length of each interval defined between two successive discontinuity points depends on the shape of the initial pdf. Thus, the ratio of the number of random values that belong to an area, to the total number of the generator values, is equal to the ratio of the length of that interval to the total projection interval, and it is equal to the probability of this value. Since the mapping of stochastic values to their probability distribution is unique, this projection can be used for generating stochastic values following any distribution function. This mapping procedure is independent of the type of the pseudo-random generator and the number of projection intervals, although these two parameters determine the method's accuracy.

The pseudo-random generator is implemented using the n -stage linear feedback shift register (LFSR) circuit with a primitive polynomial having the fewest number of terms [8]. The LFSR is a collection of unit delays and modulo-2 adders and scalar multipliers that implements a finite state machine,

$$C(u) = \begin{cases} \frac{4\mu^2 p_1}{(\lambda_1 + 2\mu)^2} e^{\lambda_1 u} + \frac{4\mu^2 p_2}{(\lambda_2 + 2\mu)^2} e^{\lambda_2 u}, & u < 0 \\ 1 - 2\mu \left(1 - \frac{2\mu p_1}{\lambda_1 + 2\mu} - \frac{2\mu p_2}{\lambda_2 + 2\mu} \right) u e^{-2\mu u} - \left[1 - \frac{4\mu^2 p_1}{(\lambda_1 + 2\mu)^2} - \frac{4\mu^2 p_2}{(\lambda_2 + 2\mu)^2} \right] e^{-2\mu u}, & u \geq 0. \end{cases}$$

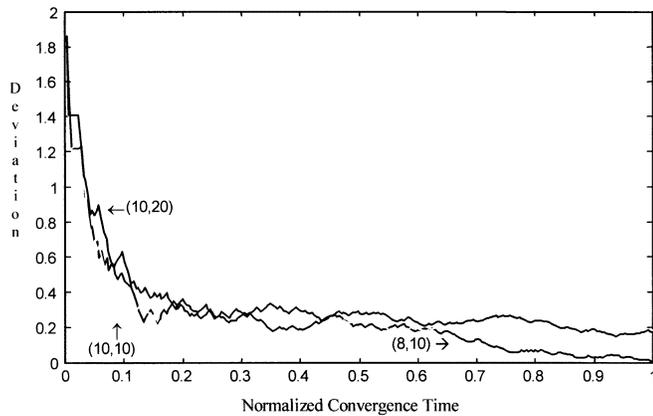


Fig. 6. Method's convergence for various (n, m) values.

where each state is uniquely determined from its previous state and feedback connections defined by its characteristic polynomial. The number of projection intervals of an arbitrarily defined function is defined as parameter m . Fig. 6 has been derived by using simulation and shows how fast the method converges according to the (n, m) parameters. The convergence time has been normalized to the period of each pseudo-random sequence. The convergence time is independent of the number of the projection intervals, and the method approaches its best performance when the time is equal to the pseudo-random sequence period.

The mapping procedure can be implemented by using either the ROM-based technique or the CAM-based technique, as described below.

A. ROM-Based Technique

The ROM-based technique uses a ROM for storing the cumulative probability values and a state machine that implements the successive approximation approach. The output of the pseudo-random generator is compared consecutively with all values stored in the ROM, starting with the lowest one. Whenever the ROM value becomes greater than the pseudo-random value, the scanning process terminates. The ROM address is used as a pointer to another ROM for determining the additive inter-cell delay value. The use of dual-port RAMs, instead of ROMs, allows the modification of the stored values during system operation, thus allowing the modification of the distribution parameters or the application of a completely new distribution.

B. CAM-Based Technique

The CAM-based technique uses content addressable memory (CAM) cells for speeding up the comparison process (Fig. 7). Each CAM cell uses a register for storing the cumulative probability values and has two outputs (" $<$ " and " \geq "), which indicate if the compared value is "less" or "greater or equal" to the value stored in the CAM cell. Due to the CAM cell array architecture, simultaneous comparison of the random value to the pre-stored values is performed. When two consecutive comparators activate their complementary outputs, they determine a pointer that is used as an address indicator to another memory that contains the additive inter-cell delay values.

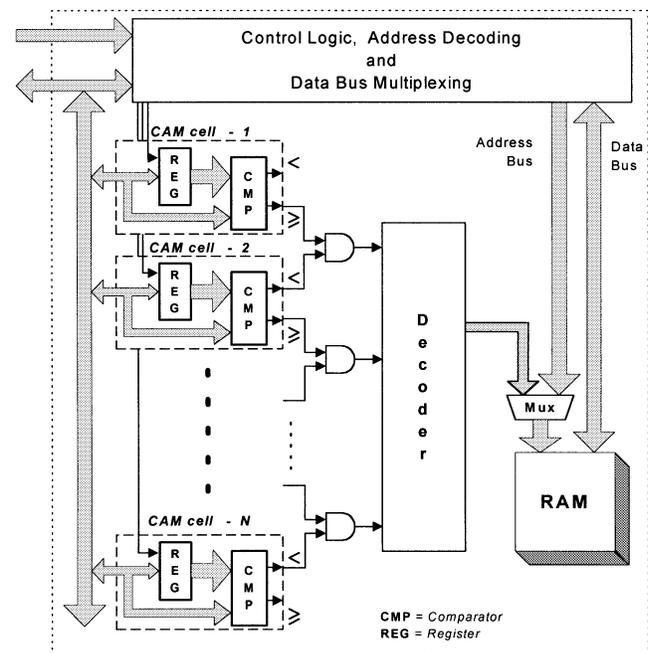


Fig. 7. CAM-based technique.

TABLE I
PERFORMANCE RESULTS OF THE SUCCESSIVE APPROXIMATION APPROACH
(IN MICROSECONDS)

	Mwave DSP C-code	Mwave DSP Assembly-code	ARM7Thumb C-code
rand()	0.85	-	7.20
LFSR	-	0.31	1.80
Maximum execution time (10 projection intervals)	6.46	2.64	28.90
Maximum execution time (64 projection intervals)	31.38	13.72	134.20
Maximum execution time (128 projection intervals)	60.92	26.85	259.00
Maximum execution time (256 projection intervals)	120.00	53.10	508.60

The CAM-based technique is much faster than the ROM one, requiring just a single processing cycle. Its response time is independent of the form of the pseudo-random generator and the number of projection intervals, but its complexity increases as the number of stored values increases and when more accuracy (projection intervals) is required. The processing time of the ROM-based technique is variable and may become unacceptably high, when high accuracy is required, while its hardware complexity is slightly affected by the number of the distribution density values.

Table I shows some comparative performance results for various software implementations of the proposed method using the ROM-based technique (successive approximation approach), which is the only one of the proposed techniques that can be implemented in software. For these measurements, we used the 39 MIPS version of the MWave DSP of IBM Corp., and an ARM7Thumb (40 MHz, μ PD97022) microcontroller of ARM Ltd. For the MWave DSP we implemented the proposed

method both in C (using the *rand()* function) and in assembly (using the LFSR and optimized code for exploiting the compound instruction capability of the MWave architecture). Due to the high transmission rate of ATM networks, the minimum cell inter-arrival time is very short ($2.73 \mu\text{sec}$ at 155 Mbps), and only the MWave with optimized assembly code can support the high cell rate by restricting the projection intervals to 10, which affects the method's accuracy. When a higher number of projection intervals are required, no software implementation can satisfy the ATM cell stream timing requirements. At higher data rates, where the time between successive cells becomes even shorter ($0.68 \mu\text{sec}$ at 622 Mbps), only the CAM-based technique implemented in hardware can support real-time ATM traffic emulation.

V. EXPERIMENTAL PROTOTYPE

The delay introduced in any ATM network can be approached by combining a constant delay value and a number of known probability functions in different parts of the jitter scale [9]. Although both techniques have been prototyped and tested, the CAM-based technique has been used in the virtual circuit emulator (VCE) of an ATM real-time test bed, due to its fast and predictable response.

The real-time ATM emulator consists of two network interface cards (NICs) with multiple VCE boards in order to support traffic conditioning on multiple virtual channels in each direction. A host processor is used to control the emulator's functionality and to configure the emulator boards. Each NIC uses a STM-1 Framer and an ATM controller implemented using reprogrammable logic. The ATM controllers manage the emulator's internal busses and multiplex cells from different sources. The controllers have been designed in order to implement the cell assembly/disassembly following the B-ISDN ATM layer functionality and to support emulator specific measurements.

The VCE board is a multifunctional board for implementing network traffic conditions and node management functions. The VCE uses an incoming and an outgoing FIFO for storing the cells of a virtual circuit, two field programmable gate arrays (FPGAs) for implementing the delay and cell error functions, and a local CPU for setting up the board configuration, loading the FPGAs and for communicating with the host computer. The error rate control FPGA uses the user-defined BER of the emulated virtual channel for estimating the cell rejection rate and inserts errors in the cell payload by estimating the single-bit error rate probability. The cell delay generator implements the cell delay function and generates the timing of the output stream based on measured inter-cell times, provided by the ATM Controller and on locally generated additive inter-cell delay values.

The delay distribution has been implemented in an XC4005 FPGA. A counter measures the inter-departure time between consecutive cells, while the device that uses the CAM-based technique estimates a new inter-cell time that has to be algebraically added to the inter-departure time in order to generate a pre-defined inter-arrival distribution at the destination node. When the inter-arrival time expires, the cell is transmitted into the network. It has been measured that using a 12-bit LFSR on a

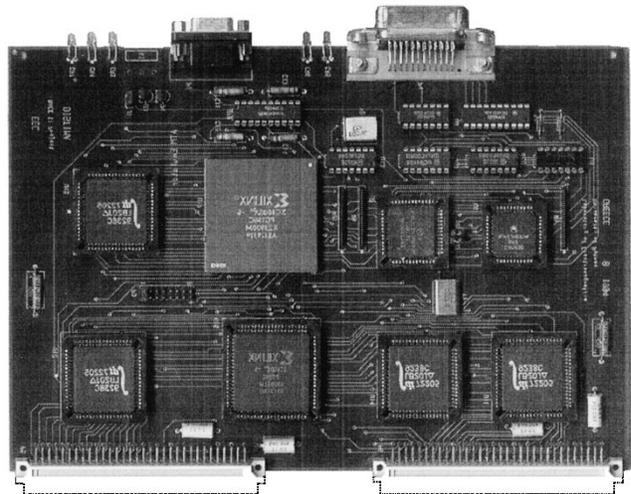


Fig. 8. Emulator's board.

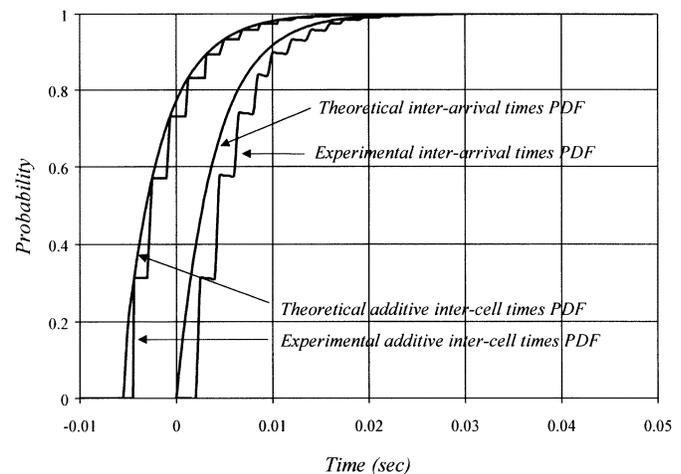


Fig. 9. AAL1 example: Theoretical versus experimental results.

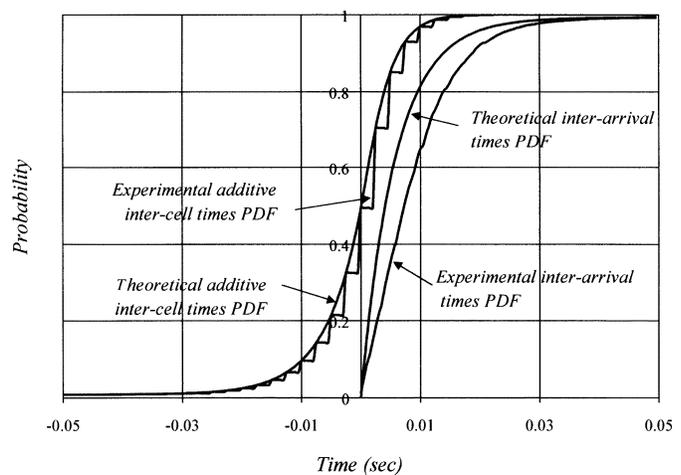


Fig. 10. AAL2 example: Theoretical versus experimental results.

155 Mbps network emulator, the system provides reliable results 180 msec after its initialization. Fig. 8 shows the board that has been developed for emulating the ATM traffic. Multiple boards can be used in such a testing tool for emulating multiple VCs using the same or different probability functions.

Experimental results of the two traffic scenarios presented in Section III are shown in Figs. 9 and 10. We used the CAM-based circuit, and we implemented the PDFs of the additive inter-cell times on 32 projection intervals, according to the mapping method. During our experiments, we recorded the additive delay values and the inter-cell times at the emulator's output of 40 960 cells. As it is shown in these Figures, the statistical analysis of the experimental measurements confirms the validity of the proposed method, since the experimental results approach the expected theoretical behavior. More accurate results can be obtained by increasing the number of projection intervals.

VI. CONCLUSION

A new method for implementing stochastic functions in hardware was presented. The method is independent of the implemented stochastic function and can easily support different accuracy requirements. The method was prototyped using FPGAs, and it has been used in an ATM real-time traffic emulator for measuring the performance of various services under different traffic conditions.

APPENDIX

ADDITIVE INTERCELL TIME DISTRIBUTION CALCULATION

The delay distribution of the i -th cell is given by $D_i(y) = P[d_i \leq y]$ and for $y \geq 0$

$$\begin{aligned} D_{i+1}(y) &= P[d_i + c_{i+1} \leq y] \\ &= \int_{0^-}^{\infty} P[c_{i+1} \leq y - u | d_i = u] dD_i(u) \\ &= \int_{0^-}^{\infty} C_{i+1}(y - u) dD_i(u). \end{aligned} \quad (19)$$

However, according to (6), the limit of the above distribution is equal to $D(y)$, and finally we get the following integral equation

$$D(y) = \int_{0^-}^{\infty} C(y - u) dD(u), \quad \text{for } y \geq 0. \quad (20)$$

Further, it is clear that $D(y) = 0$ for $y < 0$. Combining the last two equations, we have Lindley's integral equation [6], which defines the limiting distribution of delay jitter for G/G/1 queuing systems (queuing systems with general inter-departure and inter-arrival time distributions), which finally can be written as

$$D(y) = \begin{cases} \int_{-\infty}^y D(y - w) dC(w), & y \geq 0 \\ 0, & y < 0. \end{cases} \quad (21)$$

Since $D(y)$ is defined as the stationary PDF of a random variable d_i , $D_-(y)$ is defined as its "complementary" function for

negative values of y . $\Phi_-(s)$ and $\Phi_+(s)$ are the Laplace transforms of $D_-(y)$ and $D(y)$ respectively, and the following expressions are valid:

$$D^*(s) = s\Phi_+(s) \text{ and } \Phi_-(s) = \Phi_+(s) [A^*(-s)B^*(s) - 1]. \quad (22)$$

According to [5], we assume that $A^*(s)$ and $B^*(s)$ are suitably approximated with rational functions of s , so the term $(A^*(-s)B^*(s) - 1)$ can be expressed as

$$A^*(-s)B^*(s) - 1 = \frac{\Psi_+(s)}{\Psi_-(s)} \quad (23)$$

$\Psi_+(s)$ and $\Psi_-(s)$ must satisfy the following conditions:

Condition 1: $\Psi_+(s)$ be an analytic function of s with no zeroes for $\text{Re}(s) > 0$ and

$$\lim_{|s| \rightarrow \infty} \frac{\Psi_+(s)}{s} = 1$$

Condition 2: $\Psi_-(s)$ be an analytic function of s with no zeroes for $\text{Re}(s) < D$ and

$$\lim_{|s| \rightarrow \infty} \frac{\Psi_-(s)}{s} = -1.$$

By applying the Liouville's theorem, we get

$$\Phi_+(s) = \frac{K}{\Psi_+(s)} \text{ while } K = \lim_{s \rightarrow 0} \frac{\Psi_+(s)}{s}. \quad (24)$$

The above equations provide the means for calculating $\Psi_+(s)$ and K and finally $\Phi_+(s)$. Using equations (11), (22)–(24), we can determine the pdf of the additive inter-cell times and of the introduced delay if we know the pdf of the inter-arrival times and of the inter-departure times.

REFERENCES

- [1] R. O. Onvural, *Asynchronous Transfer Mode Networks: Performance Issues*. Boston, MA: Artech House, 1994.
- [2] A. Maniopoulos, T. Antonakopoulos, and V. Makios, "Single-bit error correction circuit for ATM interfaces," *Proc. Inst. Elect. Eng., Electron. Lett.*, vol. 31, no. 8, pp. 617–618, 1996.
- [3] D. A. Hughes, "Temporal loss and jitter characteristics of ATM networks," in *Proc. ASIA-Pacific Conf. Communications*, Taejon, Korea, 1993, pp. 382–386.
- [4] H. Kroner, M. Eberspacher, T. H. Theimer, P. J. Kuhn, and U. Briem, "Approximate analysis of the end-to-end delay in ATM networks," in *Proc. 11th IEEE INFOCOM Conf.*, 1992, pp. 978–986.
- [5] L. Kleinrock, *Queueing Systems—Volume 1: Theory*. New York: Wiley-Interscience, 1976.
- [6] D. V. Lindley, "The theory of queues with a single server," in *Proc. Cambridge Phil. Soc.*, vol. 48, 1952, pp. 277–289.
- [7] H. J. R. Dutton and P. Lenhard, "Asynchronous transfer mode (ATM): Technical overview," in *IBM Redbooks*. Englewood Cliffs, NJ: Prentice-Hall, 1995.
- [8] P. H. Bardell and W. H. McAnney, *Built-in Test for VLSI: Pseudorandom Techniques*. New York: Wiley, 1987.
- [9] M. Kawarazaki, H. Saito, and H. Yamada, "An analysis of statistical multiplexing in an ATM transport network," in *Proc. IEEE Int. Conf. Commun.*, 1990, pp. 478–482.



Theodore Antonakopoulos (M'88–SM'99) was born in Patras, Greece, in 1962. He received the electrical engineering diploma degree in 1985 and the Ph.D. degree in 1989 from the Department of Electrical Engineering, University of Patras.

In September 1985, he joined the Laboratory of Electromagnetics, University of Patras, participating in various R&D projects for the Greek Government and the European Union, initially as a research staff member and subsequently as the Senior Researcher of the Communications Group. Since 1991, he has been on the faculty of the Electrical Engineering Department at the University of Patras, where he is currently an Associate Professor. His research interests are in the areas of data communication networks, LANs and wireless networks, with emphasis on performance analysis, efficient hardware implementation, and rapid prototyping. He has more than 80 publications in the above areas and is actively participating in several R&D projects of the European Union.

Dr. Antonakopoulos is a member of the Technical Chamber of Greece. He serves in the Program Committee of the IEEE International Workshop on Rapid System Prototyping.



Eustathia Ziouva was born in Athens, Greece, in 1970. She received the Dipl.Eng. degree in 1993 from the Electrical Engineering Department, University of Patras, Patras, Greece, and the Ph.D. degree in the area of wireless networks in 2001.

In September 1993, she joined the Laboratory of Electromagnetics, University of Patras, where she worked on R&D projects for the Greek Government and the European Union for three years as a Research Member.



Vassilios Makios (M'67–SM'74) was born in Kavala, Greece. He received the Dipl.Eng degree in electrical engineering from the Technical University in Munich, Munich, Germany, in 1962, and the Ph.D degree (Dr. Ing.) from the Max Planck Institute for Plasmaphysics and the Technical University, Munich, in 1966.

From 1962 to 1967, he was a Research Associate at the Max Planck Institute for Plasmaphysics, where he was associated with microwave interaction studies of plasmas. He served as Assistant Professor from 1967 to 1970, Associate Professor from 1970 to 1973, and Full Professor from 1973 to 1977 in the Department of Electronics, Carleton University of Ottawa, Ottawa, ON, Canada, where he was involved with teaching and research in microwave and optical communications radar technology, remote sensing, and Co2 laser development. Since 1977, he has been an honorary Research Professor at Carleton University. Since 1976, he has been Professor of engineering and Director of Electromagnetics Laboratory in the Electrical Engineering Department, University of Patras, Patras, Greece, where he is involved in teaching and research in microwave and optical communications, data communications networks, LANs, MANs, and B-ISDN, with emphasis on efficient hardware implementations and rapid prototyping. He is also involved in research in photovoltaic systems. He has published over 150 papers and holds numerous patents in the above mentioned fields. He served as the Dean of Engineering from 1980 to 1982 and from 1997 to 2000.

Dr. Makios is the recipient of the silver and gold medal of the German Electrical Engineering Society (VDE). He is a member of the Canadian Association of Physicists, the German Physical Society, and the VDE, and he is a Professional Engineer of the Province of Ontario and the Greek Technical Chamber. He has participated in the organizing committees of numerous IEEE and European Conferences and was the Technical Program Chairman of the 5th Photovoltaic European Community Conference in Athens, Greece, 1983 and Co-Chairman of the EURINFO 1988 Conference of the European Community.