Reprint

A New Method for Implementing Statistical Functions in Hardware and its Application in Communication Traffic Emulation

E. Ziouva, T. Antonakopoulos and V. Makios

The IEEE Singapore International Conference on Signal Processing, Circuits and Systems '95

SINGAPORE, JULY 1995

A New Method for Implementing Statistical Functions in Hardware and its Application in Communication Traffic Emulation

E. Ziouva, T. Antonakopoulos and V. Makios
Laboratory of Electromagnetics, Telecommunications and Information Processing Division
Department of Electrical and Computer Engineering, University of Patras, 26500 Patras, Greece
Tel: +30 (61) 997 286, Fax: +30 (61) 997 342, e-mail: antonako@ee.upatras.gr

Abstract

The Broadband ISDN will support a wide range of services with different characteristics. For synchronous type services, which employ high rate compression algorithms, the most critical timing parameters are the variability of the end-to-end delay and the cell loss rate. This paper presents a method for implementing statistical functions, such as delay and error rate, in hardware and describes how this has been used for implementing a real-time traffic emulator for ATM virtual channels.

1. Introduction

Any Asynchronous Transfer Mode (ATM) based network consists of various switching units which add variable delay to cells passing through, even if the cells belong to the same virtual circuit. Although parts of the introduced delay are the propagation and the processing delay, the most important type of delay, which is called queuing delay, is introduced by the need to store cells whenever contention conditions are met in the output stages of switches [1], [2]. The ATM is based on statistical multiplexing and the temporary buffering is used for decreasing the cell loss rate, due to the variability of traffic conditions. The queuing delay is not constant, but varies with time and it is affected by the network traffic. The delay jitter, which is the variable part of the queuing delay, is of great importance since its limits and its probability density function determine the implementation parameters of most synchronous services. For Constant Bit Rate (CBR) services, the expected delay jitter is related with buffering requirements at the receiving side, in order to be compensated effectively [3], [4], [5].

This work originated from the need to implement a testing tool for emulating the traffic conditions of virtual channels in ATM networks at 155 Mbps [6]. The basic requirement was to find a procedure to introduce specific inter-cells jitter for evaluating the performance of an MPEG decoder under various traffic conditions [7]. The emulation system would also introduce cell loss rate and multiple errors in the cell payload of each virtual circuit. The use of user selectable delay and error patterns allows the testing of various algorithms and services and the comparison of results produced under the same traffic conditions with different application parameters.

Section 2 presents various traffic models used in ATM networks, either for modelling the arrival process or for estimating the total delay. Section 3 gives analytic expressions of how distributions of the delay introduced in the network, affect inter-arrival times. It also provides the equations for defining the distribution function of the inter-cell time, that has to be added using the delay jitter distribution function, and vice versa. The method for implementing an odd statistical function of discrete time is given in Section 4, while Section 5 describes the architecture of a virtual channel emulator.

2. Traffic Models in ATM

Every network is characterized by two statistical variables, the end-to-end delay and the error rate [3]. The probability density function of the delay in an ATM network is calculated by using various models for describing the performance of queues in the switching units and the arrival processes.

According to [2], the timing performance of a newly established connection is determined by the arrival process of this specific service and the superposition of the traffic of the rest connections. The superposed traffic is modelled 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.

In [5], the arrival process, either of a single source or from the superposition of multiple sources, is modelled by the State Dependent Discrete Process (SDDP). Using this assumption it has been proved 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 switching units. But in any case, the jitter probability distribution is an odd function, which can be approached by combining a number of known probability functions in different parts of the jitter scale. The jitter distribution is also affected from the original inter-departure time, since widely spaced cells minimize the cell correlation.

3. Relation between delay jitter and additive inter-cell time.

The delay jitter introduced during transmission, affects the inter-arrival times of consecutive cells. In order to insert a specific delay distribution in the inter-arrival times of cells of the same virtual circuit (VC), it must be

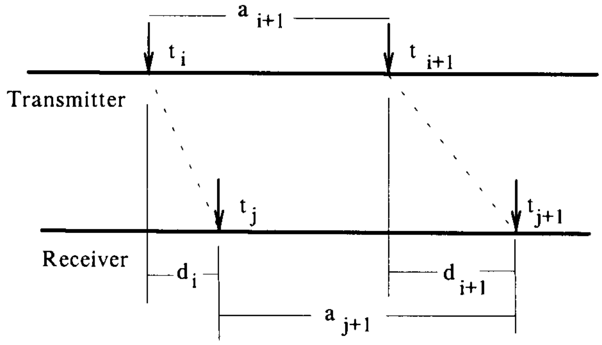

Fig. 1. The timing relation of delay and delay jitter

determined how the inter-arrival times have to be modified.

The delay function is composed of a mean value and the delay jitter. So an equation must be derived which associates the delay jitter density function with the density function of the additive inter-cell time. As additive inter-cell time is defined a random variable which is added to the originally generated inter-arrival times in order to introduce a specific delay distribution in a cell stream.

For deriving the relation between the delay jitter that has to be introduced in each cell transmission time and the additive inter-cell time, the timing diagram depicted in Figure 1 is used. The time required for the i-cell to pass from the transmitting unit to the receiving unit is denoted as d_i, and the inter-arrival times are denoted as a_i.
Then:

$$d_i = t_j - t_i \tag{1}$$

and

$$a_{i+1} = t_{i+1} - t_i \tag{2}$$

The additive inter-cell time, c_{i+1}, is given by:

$$\begin{aligned} c_{i+1} &= a_{j+1} - a_{i+1} \\ &= d_{i+1} - d_i = (d_{i+1} - E[d]) - (d_i - E[d]) \end{aligned} \tag{3}$$

where E[d] is the mean value of the delay distribution function. Equation (3) states that the inter-cell time depends only on the delay jitter and it is independent of the delay mean value and of the cells generation process.

By applying the Laplace transform on equ. (3) and on the condition that the delay jitter density function is an odd function, it can be easily found that:

$$C(s) = D^2(s) \tag{4}$$

where C(s) and D(s) are the Laplace transforms of the delay jitter and of the inter-cell time density functions.

In order to implement statistical functions in hardware, we have to consider discrete value functions and then a reference time unit has to be defined. In ATM networks, the most appropriate time unit is the time required to transmit an ATM cell. Thus having this time unit, equ. (4) is transformed to the following discrete values equation:

$$PC(n) = \sum_{k=-\infty}^{+\infty} PD(k).PD(n-k) \tag{5}$$

where $PC(n)$ is the inter-cell time distribution and $PD(k)$ is the delay jitter distribution. After some lengthy mathematical manipulations it can be proved that if $PD(k)$ is odd and takes (2n+1)-values then, $PC(n)$ is also odd, it takes (4n+1)-values and their relation is given by the following matrix equation:

$$\begin{bmatrix} PC(-2n) \\ PC(-2n+1) \\ PC(-2n+2) \\ \vdots \\ PC(1) \\ PC(0) \end{bmatrix} = \begin{bmatrix} PD(n) & 0 & 0 & \cdots & 0 \\ PD(n-1) & PD(n) & 0 & \cdots & 0 \\ PD(n-2) & PD(n-1) & PD(n) & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ PD(-n+1) & PD(-n+2) & PD(-n+3) & \cdots & 0 \\ PD(-n) & PD(-n+1) & PD(-n+2) & \cdots & PD(n) \end{bmatrix} \begin{bmatrix} PD(-n) \\ PD(-n+1) \\ PD(-n+2) \\ \vdots \\ DP(n-1) \\ PD(n) \end{bmatrix} \tag{6}$$

From this equation it is obvious that if the delay jitter density function is known, the inter-cell time density function can be easily determined. The inter-cell time function is independent of the initial inter-cell times and any density distribution function applicable in the analysis of [2] and [5] can be implemented.

4. The Hardware Implementation

The basic idea of the method proposed for implementing statistical functions in hardware, is that the delay density function must be expressed in such a way that it can be easily implementable. This can be achieved by using the distribution function, as it is shown in Figure 2. The probability distribution function of a random variable takes values from 0 to 1, and it can be considered as the normalization of an arbitrarily defined function with minimum and maximum values. If a pseudo-random generator is considered as the function that produces values between a minimum and a maximum, then this pseudo-random generator can be easily associated with every distribution function.

The shape of the distribution function is projected to an axis from 0 to 1 and splits the axis into different areas of discontinuity points, where each area represents a probability. As the pseudo-random generator produces normalised values from 0 to 1 uniformly and the length of each interval defined between two successive discontinuity points depends on the initial probability density function, the number of generator values which map to such an area is variable. The ratio of a number of values which belong

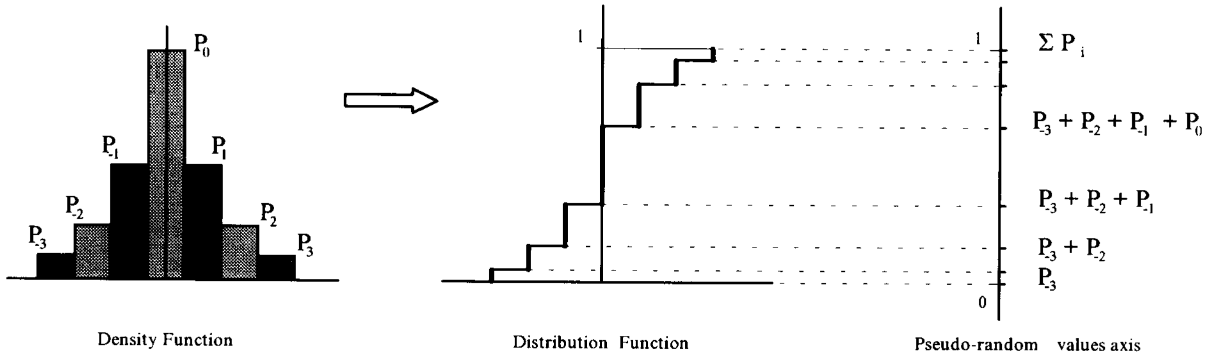

Fig. 2 The mapping procedure from density function to pseudo-random values.

to an area to the total number of generator values is equal to the probability of this value. Since the mapping of statistical values to their probability and to their distribution is unique, the above described projection can be used for generating statistical values following specific probabilities. From the above description it is obvious that the method is generic and independent of the type of the density function.

Due to the high transmission rate of ATM networks, the cell inter-arrival times could be very short (≥2.5μsec) and the whole function must be implemented in dedicated hardware. For implementing the above method in hardware two techniques can be used:

RAM based technique

This technique uses a RAM for storing the probability distribution values and follows the successive approximation approach for determining the delay value (Fig. 3). The accuracy of these techniques depends on the number of bits used for presenting the distribution values and on the number of different subareas the density function is divided. The output of the pseudo-random generator is compared consecutively with the RAM values, starting with the lowest one. Whenever the RAM value becomes greater than the pseudo-random value, the procedure terminates and the RAM value determines the inter-cell value.

CAM based technique

This technique uses a Content-Addressable Memory (CAM) for speeding up the comparison process (Fig. 4). Each CAM register contains the probability distribution values which are the projection of the distribution function discontinuity points. Due to the CAM shell array architecture, simultaneous comparison to the pseudo-random generator value is performed. When two consecutive comparators activate their complementary outputs (*'less'* and *'greater or equal'*), they determine the pointer of the value that has to be used. The pointer is decoded and the address of the stored value is generated.

The CAM based technique is much faster than the RAM based, its response time is independent of the pseudo-random generator value, but its hardware complexity increases as long as the number of stored values increases or when more accuracy is required.

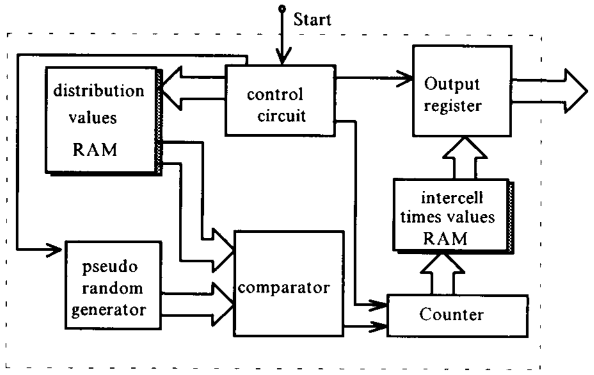

Fig. 3 The RAM based technique block diagram.

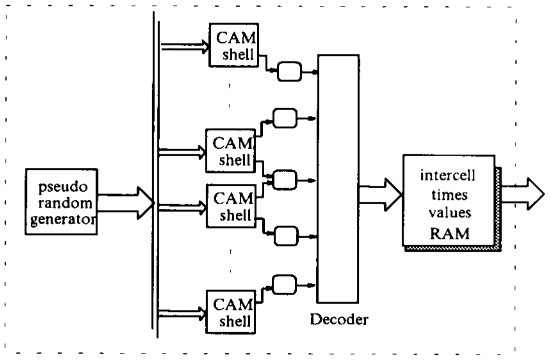

Fig. 4 The CAM based technique block diagram.

5. The Emulation Unit

The developed method has been used in the Emulation unit of a real-time test bed for ATM based networks. This test bed consists of an Emulation node and various User-Network Interface nodes. The Emulation node has been built using a UNI interface for implementing the ATM functions of the lower layers and multiple Emulation Units interconnected via a dedicated bus. The functionality of each Emulation Unit is determined via an external host computer. All Emulation

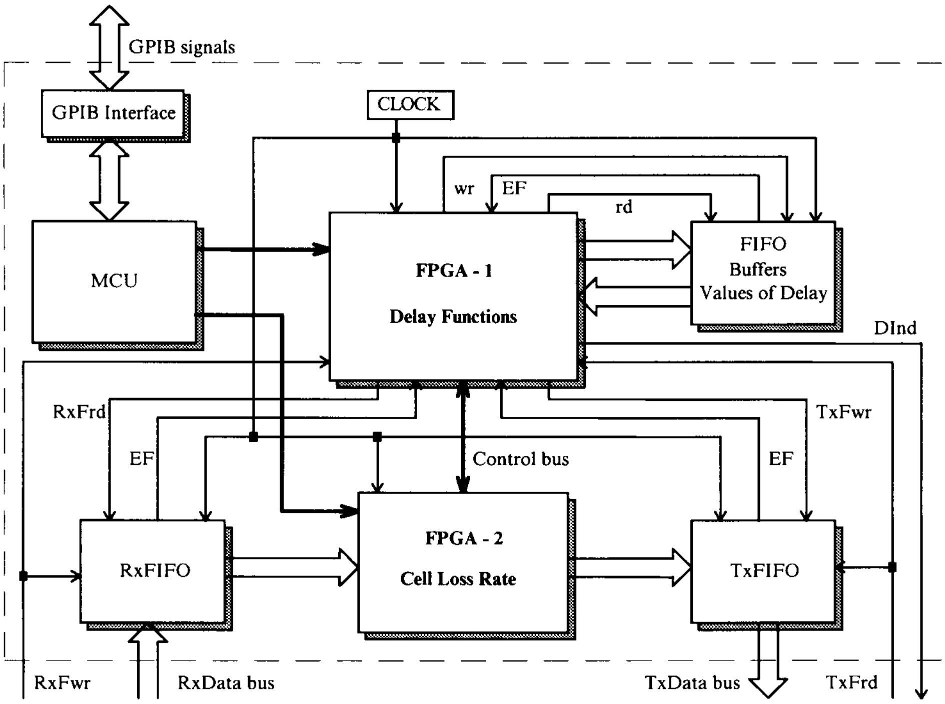

Fig. 5 The VC traffic emulation board block diagram.

Units are connected to the host computer via a GPIB interface.

The Emulator board, which is shown in Fig. 5, uses two FIFOs for storing the user cells before and after their processing. The RxFIFO stores the incoming cells, which are transferred to the TxFIFO under the supervision of the Delay Functions module. It uses a local MCU for setting up the board configuration, loading the FPGAs and for communicating with the host computer. The board has a number of FPGAs for implementing the interface to the internal bus, the cell delay functions and error rate control. The FPGAs architecture is controlled by the local MCU and can be reprogrammed during system operation, for implementing various delay functions, mean delay values and delay jitter bounds. The Delay Functions module implements the inter-cell time function and generates the timing of the output stream based on measured inter-arrival times and on the locally generated timing values. The Cell Loss Rate FPGA uses the specified 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 errors are inserted in the cell payload when it is transmitted from the Rx FIFO to the Tx FIFO using a XOR plane, controllable by the Cell loss Rate FPGA.

The inter-cell distribution has been implemented, as a part of the Delay Functions module, in an XC4005 FPGA, as it is shown in Figure 6. The circuit measures the inter-arrival times of consecutive cells at the ATM to the ATM Adaptation Layer interface and adds to their value the value generated by the inter-cells time generator. The emulator uses a double buffering scheme for handling the timing relation of consecutive cells and a cell payload is transferred from the input buffer to the output buffer when the inter-departure time expires. The developed emulator uses the RAM based technique, since it is easily implementable using FPGAs.

The Inter-arrival Times Counter uses the RxFIFO control signals for measuring the time elapsed between the arrivals of consecutive cells. Concurrently, the Distribution of Inter-cell Times state machine is triggered for generating a new value. The outputs of these two modules are algebraically added and are stored in a local FIFO. Whenever a cell payload has to be transferred from the RxFIFO to TxFIFO, the respective inter-departure time is recalled from the local FIFO and the Inter-departure Times circuit is fed. Upon expiration of the inter-departure time, the cell payload is transferred to the TxFIFO. The three modules that implement the Delay Functions operate independently by using timing information received from the lower layers of the ATM network.

6. Conclusions

A new method for implementing various statistical functions in hardware was presented in this paper. The method originated from the need to develop a testing environment for traffic emulation of ATM virtual channels, but the proposed method is general and applicable in many areas. This work developed a technique for implementing any discrete distribution function, irrespective of its specific shape.

The main advantage of the developed architecture is that the hardware is generic, since it is independent of the implemented probabilistic function, and it is programmable, since different values with different accuracy can be loaded and various statistical functions can be implemented, using the same architecture.

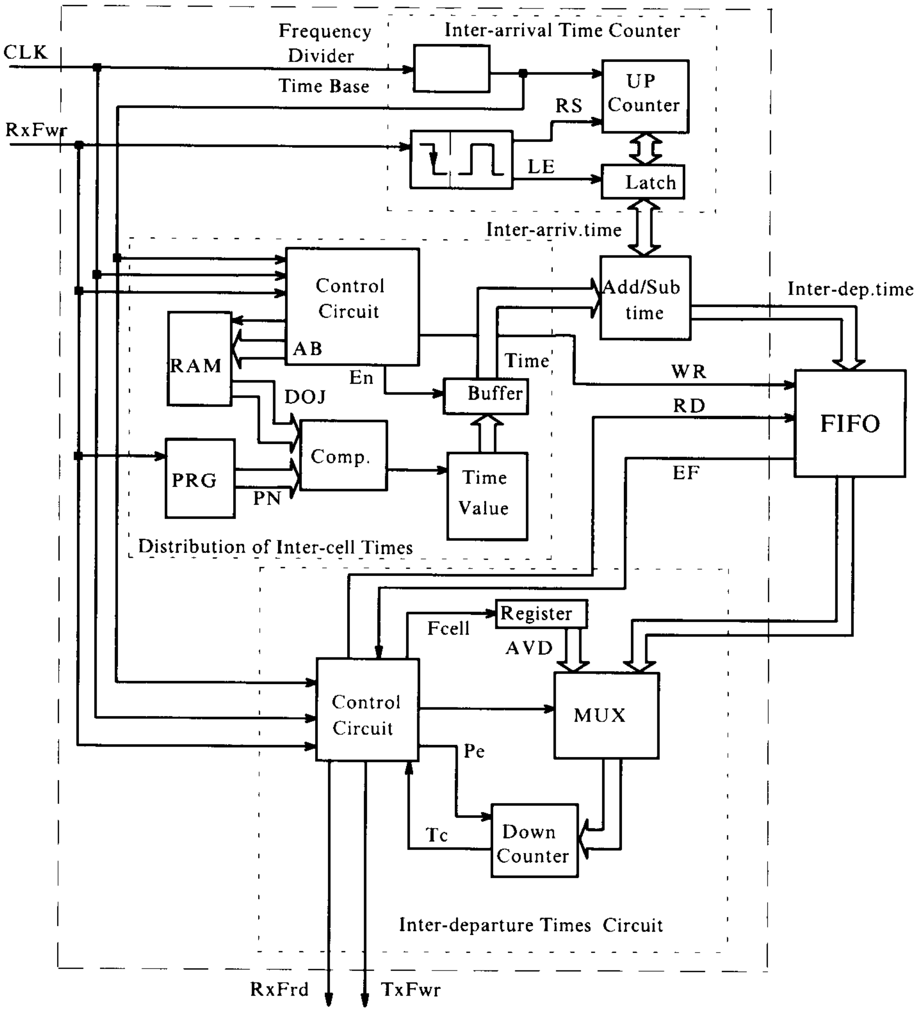

Fig. 6 The Delay Functions module block diagram.

The method has been used in an ATM real-time traffic emulator for implementing various end-to-end delay and cell loss rate distributions, and it was prototyped using Field Programmable Gate Arrays.

Acknowledgement

This work was partially supported by the "Research for Advanced Communications in Europe" (RACE) program, R2045 DISTIMA project.

References

[1] Martin de Prycker: "Asynchronous Transfer Mode: Solution for Broadband ISDN", Ellis Horwood Ltd, 1991.

[2] Raif O. Onvural: "Asynchronous Transfer Mode Networks: Performance Issues", Artech House, 1994.

[3] David A. Hughes, "Temporal Loss and Jitter Characteristics of ATM Networks", *Proceedings of the ASIA-Pacific Conference on Communications,* Taejon, Korea, 1993, pp. 382-386.

[4] M. Kawarazaki, H. Saito and H. Yamada, "An Analysis of Statistical Multiplexing in an ATM Transport Network", *Proceedings of the IEEE International Conference on Communications,* Atlanta, 1990, pp. 478-482.

[5] H. Kroner, M. Eberspacher, T. H. Theimer, P. J.Kuhn and U. Briem, "Approximate Analysis of the End-to-End Delay in ATM Networks", *Proceedings of the 11th IEEE INFOCOM Conference*, Florence, Italy, 1992, pp. 978-986.

[6] C. Papadopoulos, A. Maniatopoulos, T. Antonakopoulos and V. Makios: "A Real-time Test-bed for Prototyping Cell-based Communication Networks", *The 5th IEEE International Workshop on Rapid System Prototyping,* Grenoble, France, June 1994.

[7] DISTIMA project: "Annual Project Review Report", September 1994, 45/SIE/SN6/DS/B/A1-94/b1.