

# A flexible and cost-effective ATM Traffic Generator

*Hontas St., Evangelatos A., Vasileiou D. and Mitrou N.*  
*National Technical University of Athens*  
*Computer Science Division*  
*Heron Polytechniou 9, Zografou*  
*GR 157-73, Athens, GREECE*  
*Phone: +30-1-7721639, Fax: +30-1-7722534*  
*e-mail: mitrou@softlab.ntua.gr*

## **Abstract**

In this paper a flexible burst-level ATM traffic generator is presented. It is a PC-based system with a careful allocation of functions between hardware (PC ISA board) and software, in a way that allows it to work on-line at full-speed (155.52Mbps), on the one hand, and on the other, to be flexible enough to emulate a wide range of ATM traffic profiles. Up to 4 boards can be hosted by a single PC, each being able to generate up to 16 independent streams. Each stream consists of a continuous sequence of “traffic events” (burst-silence cycles), with each event being described by three parameters: the burst size, the silence duration and the inter-cell distance within the burst. The triplets, describing respective traffic events, are generated by software and downloaded on-line (through DMA) to the hardware; thus, in principle, any traffic model can be implemented, the limits being imposed only by the speed of the PC in relation to the total number of independent streams (up to 64) generated concurrently. For example, arbitrary distributions of the three traffic parameters (including experimental histograms) can be independently sampled; or correlated samples according to any law desired can be easily produced. Even further, real traffic can be emulated, provided that a monitoring device (e.g. a LAN traffic monitor) is connected to the PC feeding it with samples of the real traffic stream.

## **Keywords**

ATM test tools, ATM traffic generation, traffic models, bursty traffic

---

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: [10.1007/978-0-387-35353-1\\_28](https://doi.org/10.1007/978-0-387-35353-1_28)

D. Kouvatzos (ed.), *Performance Analysis of ATM Networks*

© IFIP International Federation for Information Processing 2000

## 1 INTRODUCTION

Building test tools for ATM networks is a challenging job from many points of view. Firstly, they have to produce and consume traffic at fairly high rates (155 Mb/s and 622 Mb/s are the two standardised rates at the ATM User-Network Interface, see (ITU-T, 1993A & 1993B)). The problem is not the speed on itself. Since it is rather impractical or impossible to store and process data off-line before injecting it into the network or after capturing it for analysis purposes (due to the huge amount of storage that would be required for that), most functions must be performed on-the-fly, requiring special hardware implementations. Secondly, a multitude of traffic profiles is encountered in an ATM network. In order to produce such diverse profiles artificially and/or analyse them with test tools, a great deal of flexibility is required by the latter, not easily available by hardware implementations.

Another issue is cost. The target users group of test tools is not as large as that of consumer electronics, for example. This fact, combined with the special requirements outlined above, keeps the cost prohibitively high for the common user.

In this paper a flexible and cost-effective solution to the problem of ATM traffic generation is presented. *Flexible*, because the traffic profiles are produced in software and only the time-critical functions are implemented in programmable hardware (FPGA chips). *Cost-effective*, because the whole system is hosted in a standard PC board.

Many traffic sources are bursty in nature, i.e. they produce blocks of information (bursts), rather than continuous-bit-rate streams. The latter are usually encountered within network nodes as a result of successive multiplexing stages or after a rate-shaping operation. Nevertheless, even after shaping or multiplexing which give rise to smooth or even constant-rate traffic flows, the bursts may still exist as distinguished entities within a stream and as such are delivered to their destination.

Apart from the cases where segmentation has been applied (another traffic shaping operation aiming at enforcing contracted traffic profiles), each individual burst corresponds to a separate information unit (packet or block, e.g. a video frame or a Protocol Data Unit from upper layers in the overlying protocol stack). Subsequent processing at the receiver may only start after the entire burst (packet or information block) has been received. In such cases what is of interest from the network side is its performance on a burst level rather than on a cell level.

ATM traffic generators that have appeared so far focus on cell-level performance (ADTECH 1997, ALCATEL STR 1994, WANDEL AND GOLDBERMAN 1994). Even when they are able to emulate higher-level dynamics (e.g. producing Markov-Modulated-Rate-Processes) they do not

provide burst-level information (e.g. burst-sequence numbers) that would enable respective analysis at the receiving end.

This paper presents the functionality and the architecture of an ATM traffic generator enabling burst-level traffic testing (*Burst Level Traffic Generator - BLTG*). Following this introduction, section 2 outlines the main characteristics of the developed traffic generation tool. Section 3 presents the source model of the generated traffic. Section 4 gives functional and architectural details. A high-level user interface, that has been developed allowing the definition by the user of the traffic profile for each stream, is described in section 5. In the same section detailed examples of traffic profiles that can be produced easily by the generator are presented. Finally, section 6 summarises the main features of the *BLTG* and compares it with other similar traffic generators.

## 2 THE MAIN FEATURES OF THE GENERATOR

The proposed generator is a PC-based system (ISA board with the associated driving software) with a careful allocation of functions between hardware and software, in a way that it can work on-line at full speed, on the one hand, and on the other it is flexible enough to emulate a wide range of traffic profiles. Up to 16 independent ATM traffic streams per board, with an aggregate rate up to 155.52Mbs, and up to 4 boards per PC can be supported.

As described in more detail in the next section, each stream is defined in terms of a sequence of "traffic events" (burst-silence cycles), each event being described by three parameters: the *burst size*, the *inter-cell distance* within the burst and the *silence duration*. The software is responsible for producing sample values of the traffic parameters and passing them on-line to the hardware (through Direct Memory Access - DMA), while the hardware produces the actual traffic streams according to the samples generated by the software. For periodic traffic streams, the samples within a period need to be downloaded only once and the hardware keeps on working producing the same desired pattern periodically ("cyclic" mode of operation, in contradistinction to the "refresh" mode of the continuous downloading of traffic events). Moreover, the hardware generates and inserts appropriately additional information (time-stamps, sequence number of the cell within the burst, sequence number of the burst etc.) that is necessary for burst delineation and analysis of basic performance measures (delay, losses) at the receiving side.

Another important feature of the proposed system (enabled by the flexibility of the followed software-hardware co-design approach) is its ability to inter-work with a real traffic source (e.g. a video card) and emulate

its profile, although with dummy data. All what is required in this case is a traffic monitor, which will take measurements on the real traffic stream. If the real source is also PC-resident, it's a matter of a simple software routine to get these measurements and pass them to the generator's software. In section 5 examples of traffic models that can be emulated by the generator are presented.

In summary, the main features of the *BLTG* are the following:

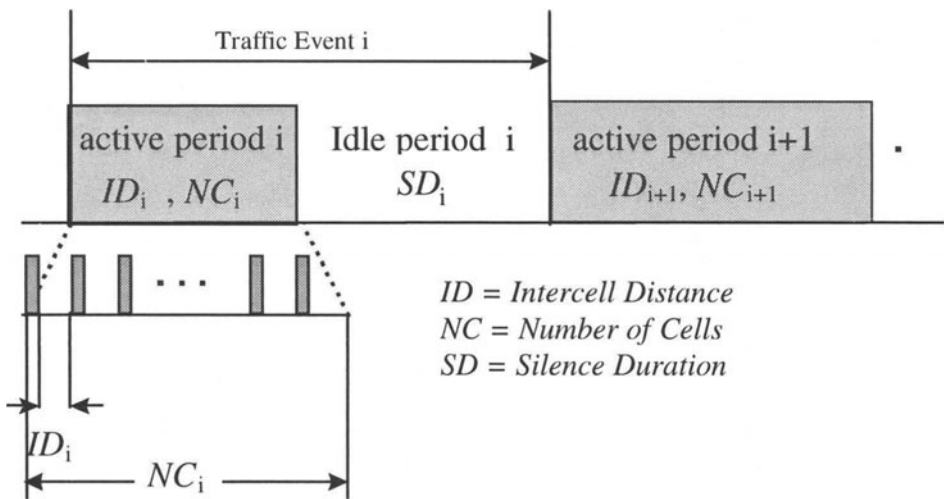
- Up to 4 boards can be hosted by one PC.
- Up to 16 independent ATM cell streams per board, each one with a different traffic profile. Thus a total number of 64 independent streams can be generated by a single PC.
- Supported rate up to 155.52Mbs per card.
- Supported rate for each stream from 4Kbs up to 155.52Mbs.
- Full UTOPIA level 1 compliant interface, 155.52Mbs (ATM Forum 1994B).
- Many physical interfaces can be connected as daughter boards to the generator's mother board (e.g. single-mode or multi-mode physical interface for STM1/STS-3c/OC-3).
- Handling of signalling connections and signalling support (ATM-Forum 3.1 and/or ITU-T Q2931 Access Signalling, and ITU-T Q.2110 Signalling AAL for both user and network sides of attachment, (ITU-T 1994, ITU-T 1995, ATM Forum 1994A)) .
- Handling of bi-directional user data ATM connections with payload exchanged through the PC bus.
- Generation of cell with 53 or 54 bytes length.
- Support of special data fields within the payload in predefined positions: burst identifier, stream identifier, traffic generator card identifier, time stamp.
- Priority mechanism that solves, in real time, collision problems that may occur when more than one stream compete for the same cell-slot of the ATM interface.
- Description of each traffic profile as a series of burst and silence intervals, provided by software. Zero silence values are possible, resulting in continuous streams (of Constant or Variable Bit Rate).
- Any histogram of values, either derived on-line from closed-form expressions or from experimental data, can be sampled for the parameters of each of the 16 streams.
- Ability to configure for CBR, VBR and ABR traffic types in the context of the previous two characteristics. The ABR type of operation, in particular, is configurable as easily as the VBR type, provided that the

PC, hosting the generator board, can accept and process Resource Management (RM) cells.

- Modification of the traffic profiles in real time, according to the user demands.
- Monitoring capabilities of the produced traffic (counting number of transmitted cells: - per cell stream; - over all cell streams in a card).
- Open architecture of design to allow addition of new functionality.

### 3 THE MODEL OF THE GENERATED TRAFFIC

As mentioned before, each one of the sixteen streams, that can be produced by a single *BLTG* board, is described as a series of burst and silence cycles (active and idle periods), named *traffic events*. Each burst is described by two parameters (*Number of Cells -NC-* and *Inter-cell Distance -ID-* within the burst) and each silence period by its duration (*Silence Duration -SD-*). Periods are measured in ATM-slot time units. The traffic stream model and the meaning of the three traffic parameters are schematically shown in figure 1.



**Figure 1** The BLTG traffic model

The software produces a sequence of traffic events for each stream and delivers it to the hardware (batch-wise, as a response to respective

interrupts). The hardware, in turn, produces actual ATM cell traffic corresponding to the downloaded traffic event sequence: for each traffic event it generates and transmits over the physical interface a number of ATM cells equal to  $NC$ , placed  $ID$  ATM slots apart, and then applies a silence for an interval equal to  $SD$  ATM slots. Apparently, this way of traffic stream construction allows in principle the implementation of any desired traffic model. It is up to the software to convert the model parameters into a respective sequence of traffic-event triplets  $[NC_i, ID_i, SD_i]$ . The following cases are already supported by the current generator software (see also section 5).

### **3.1 Independent burst and silence samples from arbitrary distributions**

This is the simplest albeit quite general model of traffic event generation. Two distributions are maintained as look-up tables in the main memory, one for the burst size and another one for the silence duration. The distributions are sampled independently to give the respective parameter values for each new traffic event. The third parameter, namely the intercell distance within the bursts, can be either constant or variable (drawn from a third distribution or determined otherwise, e.g. as a response to RM cells for an ABR-type of operation). The distributions can originate either from known statistical models (e.g. normal, exponential, uniform etc.) or from histograms derived experimentally. As it will be exemplified in section 5, streams of trivial or widely-used profiles (e.g. Constant Bit Rate - CBR, periodic On/Off) can be easily programmed as special cases of this traffic class.

### **3.2 Independent burst samples, dependent silence intervals.**

This scenario may arise in cases of burst-level traffic shaping (see e.g. Mitrou N., 1998), where the silence enforced after the transmission of a burst is deterministically calculated as a function of the preceding burst size. Again, a distribution is sampled for the burst size, independently for each new event, while the silence to follow is calculated according to the desired shaping law.

### **3.3 Correlated traffic**

Any model giving correlated type of traffic (e.g. Auto-Regressive Moving Average - ARMA) can be implemented by the software to give sequences of

traffic events. The rest (sequence downloading and generation of the cell stream by the hardware) is performed as in the other cases.

### 3.4 Periodic traffic

Periodic On/Off traffic can be produced as a sub-case of 3.1 above, by choosing constant values for the three parameters. More general patterns, however, can be directly defined by the user as sequences of triplets describing respective traffic events. Provided that the number of such triplets is a sub-multiple of the batch size which is downloaded to the hardware (512 in the current implementation), the batch is downloaded only once and periodically processed by the hardware. This mode of (periodic) traffic generation is enabled by marking the respective traffic streams, during initialisation, as being in the CYCLIC condition (see section 5.1, User Interface).

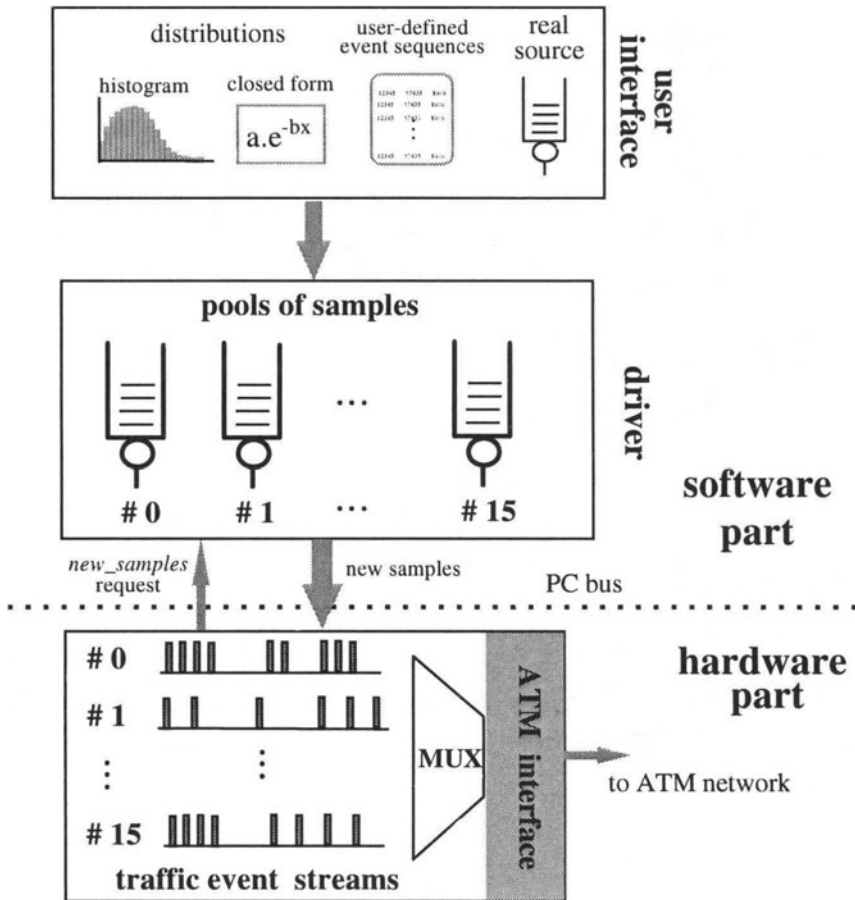
## 4. FUNCTIONAL DESCRIPTION AND ARCHITECTURAL DETAILS

Figure 2 gives a top-level block diagram of the *BLTG*. As stated above, the software part is responsible for maintaining a pool of parameter samples derived either from closed-form distributions, experimental histograms, user-defined sequences or by interfacing with real-traffic monitors. The hardware part pumps data out of the pool and forms bursts of cells accordingly, which are sent to the network over the Physical ATM interface. In order to avoid “sample starvation” at the hardware, double buffering is essentially performed on it: a *new\_samples* request is issued whenever the buffer of a particular stream is half-emptied.

### 4.1 The software

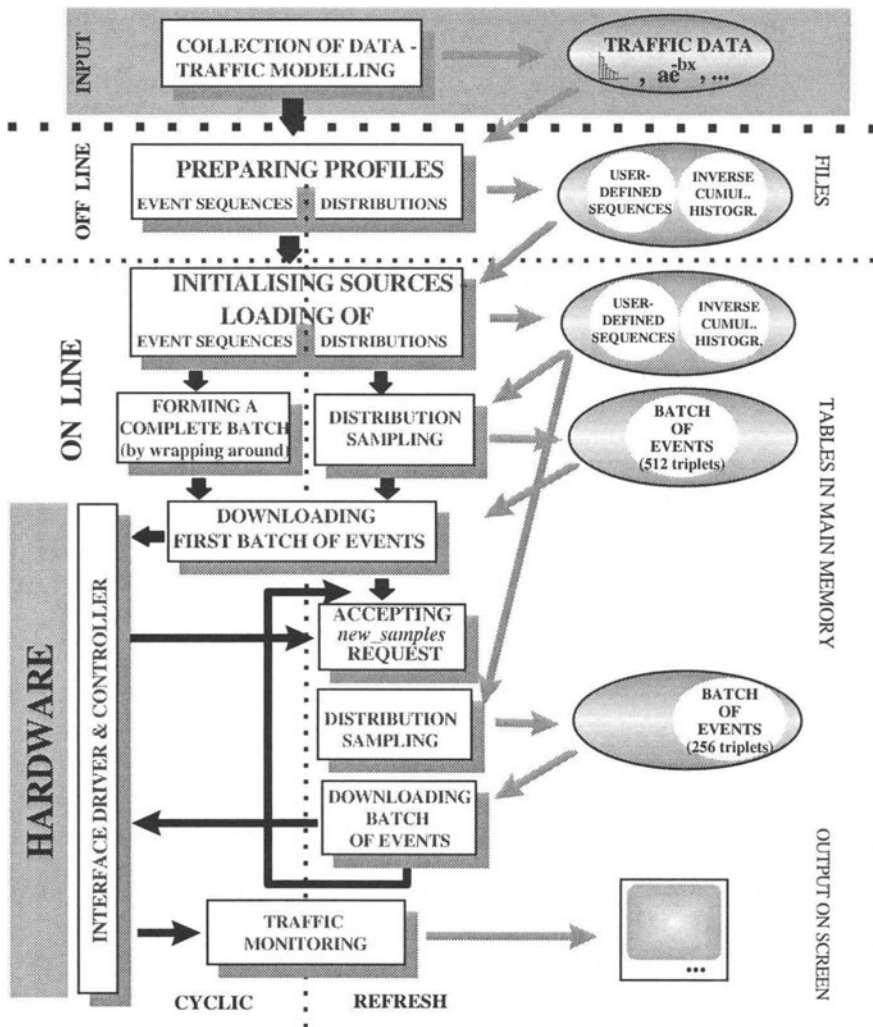
A functional description of the generator’s software is shown in fig. 3. The software consists of two successive parts: one part aiming at the preparation of the traffic profiles (parameter distributions or user-defined event sequences) in the desired format, which is run *off-line*, and a second part which implements the *on-line* functions necessary for initialising the traffic streams and for driving the hardware. Vertically, the software splits in two branches: one for the distribution-based streams, where new batches of traffic events are continuously downloaded to the hardware (REFRESH-

conditioned streams), and a second one for the cyclic streams. For the latter, a single batch of traffic events is downloaded during activation, which thereafter is repeated periodically by the hardware(CYCLIC-conditioned streams).



**Figure 2** Block diagram of the Burst-Level Traffic Generator



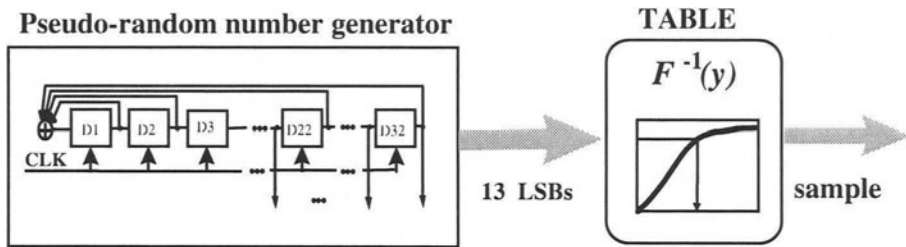


**Figure 3** Block Diagram of the BLTG Software

*Preparation of parameter distributions*

The method of producing samples in accordance with a specific Probability Distribution Function (*PDF*), say  $F(x)$ , is the one of randomly sampling the inverse of the desired *PDF*,

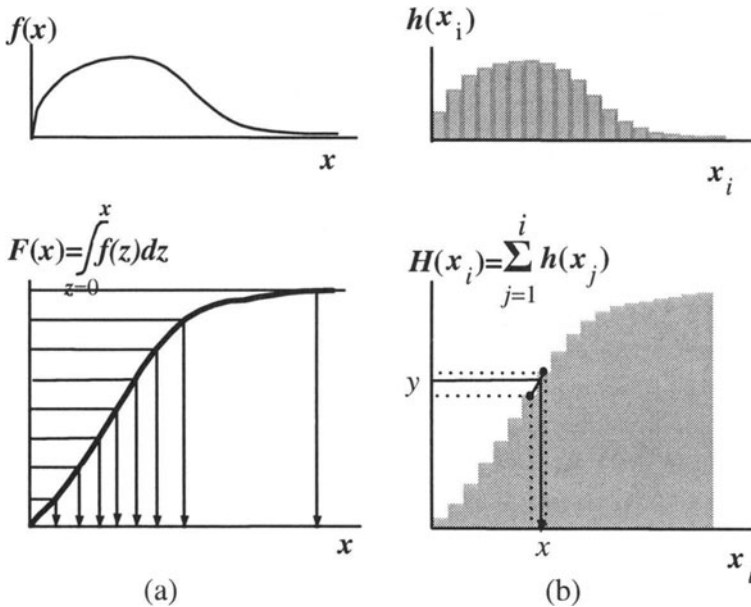
$F^{-1}(y)$  (see e.g. Law A.M. and Kelton W.D., 1982)). For the random-number generator one can either use the standard random function available in almost all programming languages, or implement his own pseudo-random generator (for better accuracy and/or speed). Fig. 4 depicts such a generator, implemented as a linear-feedback shift register, and how it is used to give samples of the distribution  $F(x)$ . The 13 LSBs of the shift register are used as the address of a look-up table, containing ( $2^{13}$  entries of) the inverse *PDF*,  $F^{-1}(y)$ .



**Figure 4** A linear-feedback pseudo-random number generator and PDF sampling

The off-line software has to produce tables of the inverse *PDF* (inverse cumulative histograms). In some cases (like for exponential or uniform distributions) such an inverse function is provided in closed form and the job is fairly simple. In other cases, however, like that of a normal or an Erlang-type distribution,  $F^{-1}(y)$  is not given in closed form. Even worse, the user may have available only a histogram from experimental measurements. If the distribution  $y=F(x)$  is provided in closed form, the inversion is performed at discrete points of the  $y$ -axis by a trial-and-error procedure (e.g. through dividing the interval of the  $x$ -axis, containing the sought value, successively by two and deciding on the new interval through comparisons; the final  $F^{-1}(y)$  value is determined by a simple linear interpolation). If even the  $F(x)$  is not available in closed form (e.g. in the case of experimental histograms), a cumulative histogram is calculated and inverted at desired points through an inversion procedure similar to the above. An example of a *PDF*, a discrete histogram and the way of their approximate inversion is schematically depicted in fig.5.

In all cases, the output of this part of the software is a file with the desired inverse cumulative histograms for the three traffic parameters.



**Figure 5** Probability Distribution and discrete histogram inversion

#### *Preparation of user-defined sequences*

As an alternative to the procedure of 4.1.1 above, the user may directly enter sequences of parameter values (in triplets  $[NC, ID, SD]$ , one for each traffic event). This can be done through using any text editor and storing the sequences in files with an appropriate naming convention. The size of these user-defined sequences of events should be a sub-multiple of 512 (i.e. a power of 2, from 0 to 9), so that, by repeating itself, to fill up a complete batch of 512 events, as required by the hardware.

#### *The on-line procedures*

The on-line part of the software starts with the initialisation of the traffic sources, by associating a traffic profile to each one of them. The respective profiles are then loaded into the main memory as tables containing the inverse cumulative histograms or the user defined sequences of events (three such tables for each stream). A first batch of traffic events (512 triplets of  $NC, ID$  and  $SD$  samples) for each initialised stream is downloaded to the hardware, which is then activated. On a hardware request basis, new batches (256 triplets of  $NC, ID$  and  $SD$  samples) are formed and downloaded for the distribution-based (REFRESH-conditioned) streams.

The *Interface Driver & Controller* module, shown in fig. 3, is used for the communication between the hardware and the software. It recognises interrupts generated by the hardware and downloads batches of traffic event samples for the respective traffic streams.

## 4.2 The hardware

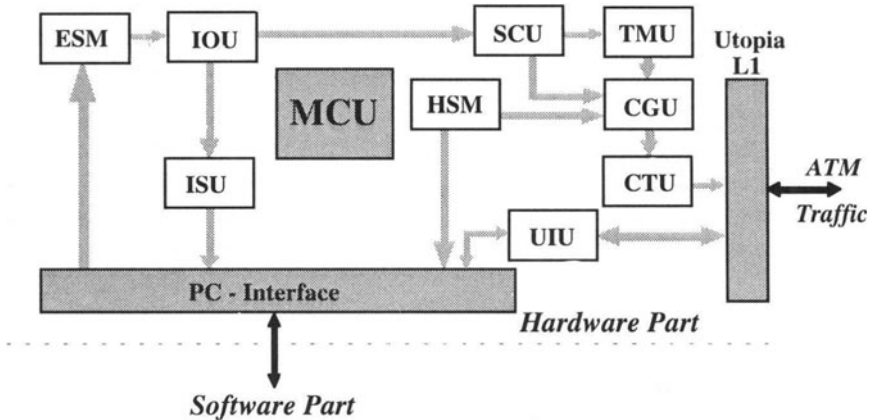
The architecture of the Burst Level Traffic Generator (BLTG) hardware is presented in figure 6. It consists of the following logical modules:

### *PC Interface Module*

A DMA-based PC interface for a 16-bit ISA bus used to provide high rate data exchange from the host to the TGC. This module is also responsible for passing interrupts from the TGC to the host processor. Interrupts are generated when one or more of the storage units associated with one stream, are half-empty.

### *Event Storage Memory (ESM)*

This is the memory where *NC*, *ID*, *SD* triplets for a number of traffic events (512) for each stream are stored. The ESM is divided into 16 different segments each one associated with one stream. This memory is implemented as a high-speed Dual Port RAM.



**Figure 6** Hardware part architecture

### *Input Output Unit (IOU)*

This module reads sequentially from the ESM each *NC, ID, SD* triplet for the current traffic event for each one of the sixteen streams, stores internally the address of the next traffic event and generates an interrupt when half of the traffic events (i.e. 256) for one stream have been already processed.

### *Interrupt Storage Unit (ISU)*

This is a temporary storage space intended to queue the interrupts generated by the IOU and used by the PC interface module. The *ISM* is implemented as a FIFO memory.

### *Master Control Unit (MCU)*

This provides control and the synchronisation of the whole system.

### *Source Control Unit (SCU)*

This module performs all functions for the traffic generation for each one of the sixteen streams. The architecture of this module is presented in figure 7.

### *Header Storage Memory (HSM)*

This is a memory where a five-byte-header for each one of the 16 different streams is stored by the host, during initialisation. The CRC header field is calculated by the software part.

### *Traffic Multiplexing Unit (TMU)*

This module solves collisions when more than one stream attempt to transmit a cell in the same cell-slot.

### *Cell Generation Unit (CGU)*

The cell selected by the TMU is generated in this module. As already mentioned, the cell payload comprises of some special fields as shown in figure7. These fields are used for analysis purposes at the traffic analyser side.

### *Cell Transmission Unit (CTU)*

This module transmits the cell generated by the CGU to a standard UTOPIA interface, as recommended by ATM Forum.

### *User Information Unit (UIU)*

This unit is responsible for sending and receiving ATM cells with user defined information field passing through the ISA bus. The UIU has priority over the CTU in accessing the UTOPIA i/f. Note that the CTU module only

sends cells to the UTOPIA i/f (generation process), while UIU sends and receives cells conveying user defined payload.

### 4.3 Generated traffic cell format

As already mentioned, the *BLTG* inserts some special fields in the ATM cell payload as produced by the CGU (this is not the case for the UIU, where the ATM cell payload is user defined and passed through the ISA bus). These fields (inserted by the GCU, Cell Generator Unit) are depicted in fig. 7 and described below.

#### *Stream Code (SC)*

This is a 4-bit field containing the identifier of the stream that generates this cell. Value xxxx is source #0, value yyyy source #15.

#### *Generator Code (GC)*

This is a 4-bit field and contains the identifier for the specific *BLTG* board.

<b>Header</b> (5 bytes)	
<b>SC</b> (4 bits)	<b>GC</b> (4 bits)
<b>ID</b> (1 byte)	
<b>NoC</b> (1 byte)	
<b>CSN</b> (1 byte)	
<b>ST</b> (2 bytes)	
<b>TS</b> (3 bytes)	
<b>BSN</b> (1 byte)	
<b>DM</b> (38 or 39 bytes)	

**Figure 7** Cell format provided by the *BLTG*

Through the SC & GC fields each source cell stream can be identified not only through their VP/VC identifier (resident in the header) but also inside the payload (16 cell sources in 16 different *BLTG* boards).

*Inter-cell Distance (ID), Number of Cells (NoC) and Silence Duration (SD)*  
These fields fully describe the current burst of the cell stream Traffic Event in Fig. 1. These fields are the same for all the cells within the same burst.

*Cell Sequence Number (CSN)*  
This field specifies the sequence number of the cell in the current burst. It ranges from 0 to 255.

*Time-Stamp (TS)*  
This 3-byte time stamp field gives the indication of the generation time by enumerating ATM slots and inserting the current number in the cell occupying the particular slot.

*Burst Sequence Number (BSN)*  
This field specifies the burst sequence (Active Period in a Traffic Event) number within the train of Traffic Events of the particular source. This field is used for burst delineation in the side of an analyser and is the same for all cells of the specific source in the current burst.

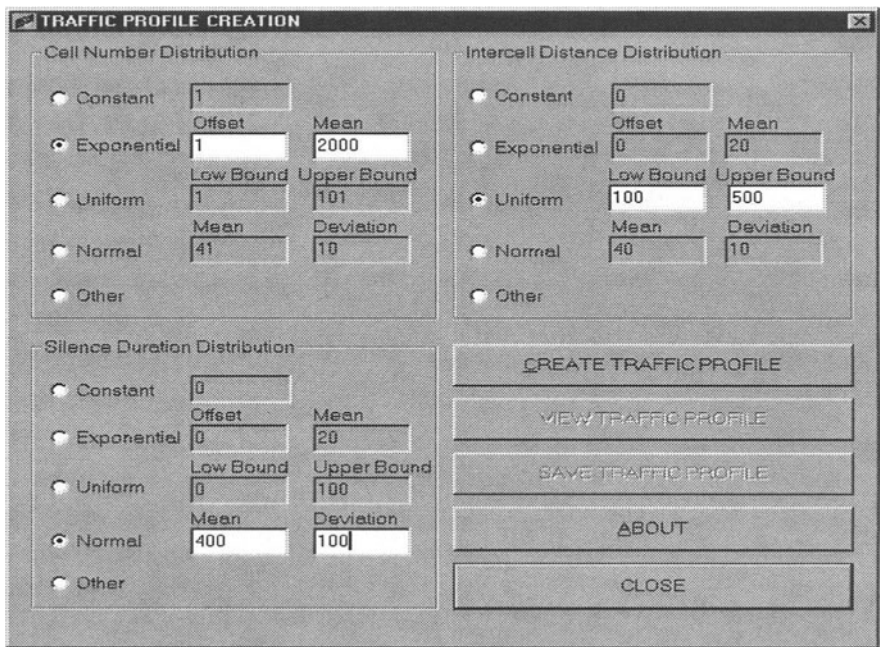
*Dummy Data (DM)*  
This field is created by a pseudo-random number generator. The length of this field is 38 or 39 bytes depending of the whole length of the cell (53 or 54 bytes).

## 5 USER INTERFACE, TRAFFIC PROFILE EXAMPLES AND TYPICAL TESTING CONFIGURATIONS

### 5.1 User interface

A friendly interface has been developed to help the user in creating traffic profiles and programming the *BLTG* hardware. Fig. 8 is an instance of the *Traffic Profile Creation* dialog, part of this interface. As shown in figure 10, the user can select among 4 known distributions (constant, exponential, normal and uniform), independently for each of the three traffic event parameters of each stream. A fifth choice (bullet "Other", in fig. 8) gets as input an arbitrary distribution, stored as a discrete histogram in a file. Such

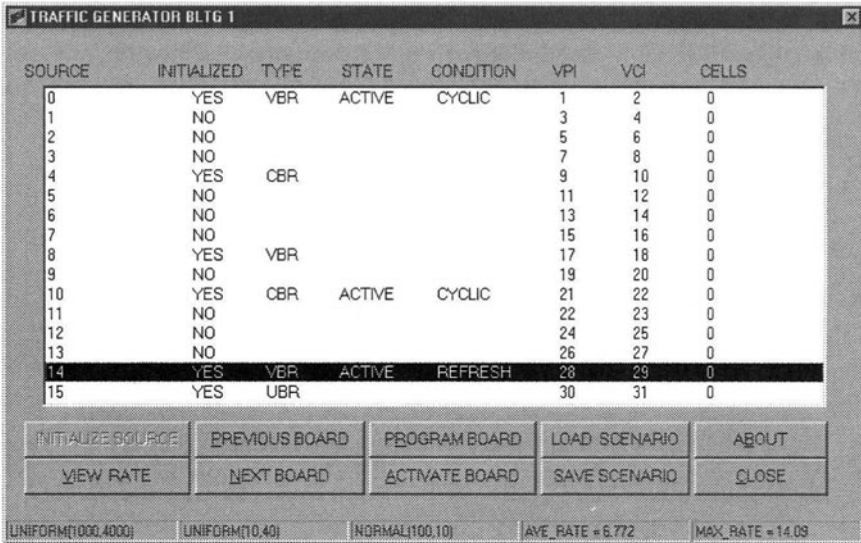
histograms may be derived in various ways (e.g. from closed-form probability density functions, experimental measurements, etc.). In all cases, the off-line software will automatically create the respective inverse cumulative histograms and store them in files for later use by the on-line part. The VIEW TRAFFIC PROFILE option in fig. 8 gives the user a visual representation (e.g. histograms) of the created traffic profile.



**Figure 8** Example of Traffic Profile Creation dialog

The user interface for the on-line software is also friendly and flexible. It allows the user to view and handle the complete list of one board's sources, including their type (CBR, VBR, UBR), status (ACTIVE or not), condition (CYCLIC or REFRESH) and VPI-VCI assignments (fig. 9). On the same screen the user has some on-line monitoring facilities of the actual traffic streams generated by the hardware: on the last column of the screen, as in fig. 9, the user can see the number of cells that actually transmitted by each source from its activation. In order to support large observation periods, 64-bit counters have been used (number of cells between 0 and 1.84e+19)





**Figure 9** Traffic Generator status and control dialog

## 5.2 Examples of traffic profiles

In this section some of the most known ATM traffic models are translated into the *BLTG* model. The selected examples aim at demonstrating the flexibility of our traffic generator and not to exhaustively address its scope of applicability.

### *Constant Bit Rate (CBR) streams*

This type of traffic can be easily produced by using any distribution for the burst size, a constant inter-cell distance that corresponds to the desired rate and a zero silence. Obviously, a *CYCLIC* stream with just a single triplet [*NC* any, *ID*, *SD*=0] written in the user-defined-sequence file does the job with the minimum resources.

### *CBR with jitter*

This pattern arises when a CBR stream passes through successive multiplexing stages. With the assumption of a large number of independent such stages, a normal distribution is a good model of the resulted jitter (= deviation from the ideal CBR cell positioning). Jittered CBR streams can be produced by the *BLTG* through using: *NC*=1, *ID*=0 and *SD* distributed according to the intercell distance of the jittered stream, e.g. *SD*=max{0,

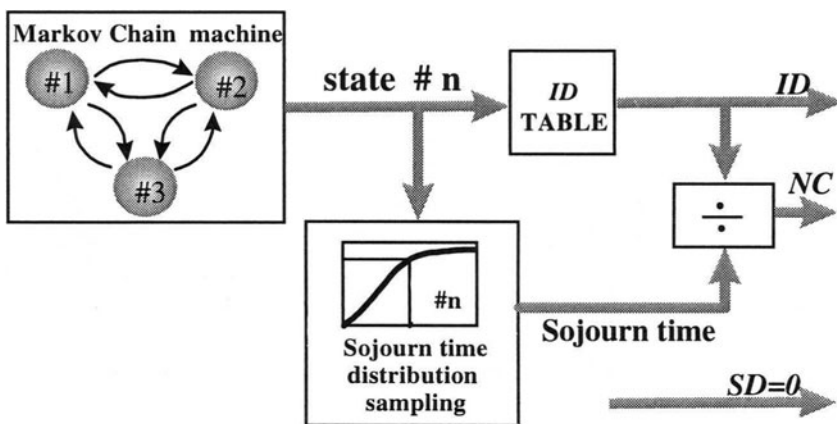
$\lceil 1/r-1+d \rceil$ , where  $r$  is the desired constant rate in cells/slot,  $d$  a normally distributed variable with zero mean and  $\lceil y \rceil$  denotes the integer part of  $y$ .

*Periodic traffic streams*

Any periodic cell stream, with a pattern that can be mapped onto a sequence of *BLTG* events  $[NC_i, ID_i, SD_i]$ ,  $i=1,2,\dots,M$ ,  $M=2^m$ ,  $m \in \{0,1,\dots,9\}$ , can be produced by the *BLTG* as a *CYCLIC* stream. Notice that the period duration, expressed as  $D = \sum_{i=1}^M (ID_i + 1) * NC_i + SD_i$ , may become extremely large, depending on the values of the individual event parameters.

*Markov-Modulated Rate Processes (MMRP)*

Such processes have been widely used in the literature for modelling ATM traffic, due to their flexibility and analytical tractability (Stern T.E. and Elwalid A.I. 1991, Elwalid A.I. and Mitra D. 1993). An underlying Markov chain drives the visiting of the source to different states, each being characterised by a certain cell rate. The sojourn time at each state is exponentially distributed (Markovian behaviour). An *MMRP* can be easily mapped on the *BLTG* traffic model by forming a traffic event for the visit of the source to a new state. The Markov chain is implemented to give the transitions between the states and, hence, the *ID* values for the respective *BLTG* events. An exponential distribution is sampled for the corresponding sojourn times, which, divided by the respective *ID*, give the *NC* values. The silence duration, *SD*, of all the events is set equal to zero. The procedure of realising an *MMRP* stream is schematically shown in fig. 10.



**Figure 10** Producing traffic events of MMRP streams

### *Exponential On/Off*

This is a special *MMRP*, as above, featuring only two states: the On state, with the source transmitting at a constant rate, and the Off state, with the source being silent. The duration of both states is assumed exponentially distributed. This model is directly implemented on the *BLTG*, by using two exponential distributions for the *NC* and *SD* parameters, respectively, and a constant-valued *ID* (corresponding to the desired rate of the On state).

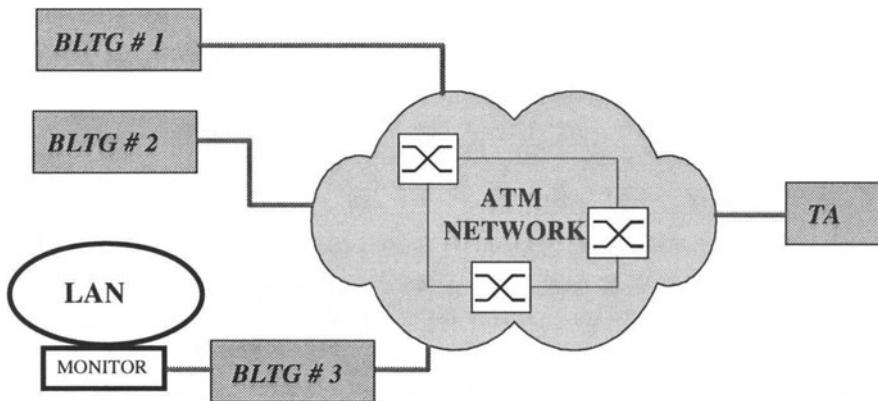
### *Traffic with correlated burst and silence duration*

As mentioned in section 3.2, this type of profile may arise when applying burst-level traffic shaping. For example, if we want to control the mean rate, we should enforce, after each burst, a silence proportional to the size of the preceding burst. Similarly, if we want to keep the *effective rate* constant (as defined through appropriate burst-level modelling), a silence calculated directly in terms of the preceding burst size and the desired effective rate and Quality of Service (QoS) figures should be enforced (Mitrou N. 1998, Mitrou N. and Kavidopoulos K. 1998). In such cases, the table holding the *SD* distribution is directly calculated by applying the desired *SD-NC* dependency law on the *NC* table. The latter is derived from the respective distribution as usual. During the on-line procedure of forming traffic event samples, both tables are sampled at once (with the same random index) to give suitable pairs of *NC,SD* values.

It should be emphasised again that the above cases provide only some typical examples of the *BLTG* usage. Since the traffic events of each stream are calculated by software, any traffic model can be implemented, in principle, provided that it doesn't impose excessively heavy computations that could violate the time limits set by the on-line procedures.

## **5.3 Typical testing configurations**

In the following figure 11 one can see how the *BLTG* can be used in experimental configurations for performance testing. A number of *BLTG* boards can be used to load the network (or a particular network component) with *background traffic*, while other board(s) will produce *test traffic*. The test traffic can be of an artificial type, or emulation of a real traffic source (e.g. of a LAN). An ATM Traffic Analyzer (*TA*), connected to the network, will conduct measurements on the received test traffic concerning various measures of interest (cell or burst losses, cell or burst delays etc.).



**Figure 11** Typical performance testing configuration using the BLTG

## 6 CONCLUSION AND COMPARISONS

A new ATM traffic generation tool was presented. It is a PC-based system (ISA board with the associated user interface and driving software), with a careful allocation of functions between hardware and software. The followed software-hardware partitioning allows functioning at full speed, on the one hand, and a flexible implementation of a variety of traffic models, on the other. The emphasis is on the burst-level modelling, hence the name *Burst Level Traffic Generator (BLTG)*. A traffic stream is defined as a sequence of burst-silence cycles (*traffic events*) described by three parameters, which are downloaded on-line and processed by the hardware. This architecture allows not only the implementation of sophisticated traffic models (e.g. MMRPs of any order or correlated traffic), but also the generation of traffic in a “dynamic” style. For example, the generator can emulate real traffic streams by communicating on-line with appropriate monitoring devices; or it can change the traffic profiles according to feedback information from the network (e.g. ABR-type of operation), provided that appropriate facilities for receiving and processing this feedback (RM cells) are installed on the PC hosting the *BLTG*.

Compared to other ATM traffic generation tools (ADTECH 1997, ALCATEL STR 1994, WANDEL AND GOLDBERMAN 1994), the *BLTG* features the following advantages:

- Up to 16 independent traffic streams can be produced and multiplexed on the same physical interface (compare this, for example, with the 8 independent streams available by the AX/4000 (ADTECH 1997)). Four such boards can be hosted by a single PC, leading up to 64 independent sources per PC.
- The traffic streams produced by the three tools mentioned above are static, in the sense that, once programmed (during initialisation), they follow the same statistics. With the *BLTG*, due to its pioneering design with suitable partitioning of functions between software and hardware, the user has the flexibility to change the traffic profiles "on the fly" (i.e. without re-initialising the generator).
- The previous feature becomes even more important in cases of testing ATM services which are adaptable to network feedback (e.g. ABR-type of service). The authors believe that the current architectures of the other tools mentioned above could not support ABR, while in the *BLTG* case it is a matter of simple software to interpret RM cells and adjust the profiles accordingly.
- The *BLTG* directly supports two levels of traffic statistics, the cell level and the burst level, by inserting appropriate information into the payload of the transmitted cells, allowing respective analysis at the receiving side. The other tools mentioned above, operating basically on a cell level, can incorporate only implicitly (e.g. with the three-state model supported by the AX/4000) burst-level statistics.
- The power of the *BLTG* lies on its feature to produce the traffic profiles on-line by software, without sacrificing the full-speed operation. A sound example of its flexibility is the potential to interwork with a traffic monitoring device or a real traffic source (e.g. a real-video card) and pass samples of traffic parameters to its hardware for real-traffic emulation.

## REFERENCES

- ADTECH (1997), "The ADTECH AX/4000 - The Art of ATM Testing"  
(see also  
[www.adtech-inc.com](http://www.adtech-inc.com))
- ALCATEL STR, "ALCATEL 8643, ATGA User Manual" (see also  
[www.snh.ch/environment/measurement/alcatel-a8643.html](http://www.snh.ch/environment/measurement/alcatel-a8643.html)).
- ATM Forum (1994A), *ATM User-Network Interface Specification Version 3.1*, Sept. 1994.
- ATM Forum (1994B), *UTOPIA Specification Level 1, Version 2.01*, March 21, 1994.
- Elvalid A. and Mitra D. (1993), "Effective Bandwidth of General Markovian Sources and Admission Control of High-speed Networks," *IEEE/ACM Transactions on Networking*, **1**, 329-343.
- ITU-T (1993A), Rec. I.413, *B-ISDN User-Network Interface*, Helsinki, March 1-12, 1993.
- ITU-T (1993B), Rec. I.432, *B-ISDN User-Network Interface - Physical-Layer Specification*, Helsinki, March 1-12, 1993.
- ITU-T (1994), Rec. 2.110, *B-ISDN ATM Adaptation Layer - Service Specific Connection Oriented Protocol (SSCOP)*, July 1994.
- ITU-T (1995), Rec. 2.931, *B-ISDN Application Protocols for Access Signalling*, Feb. 1995.
- Mitrou N. and Kavidopoulos K. (1998), "ATM traffic engineering applications using a class of M/G/1 models", submitted for publication in the *J. of Net. & Comp. Applications* (Academic Press).
- Mitrou N. (1998), "Traffic control in ATM: a review, an engineers critical view and a novel approach", *Computer Networks and ISDN Systems*, to appear.
- Law Averil M. and Kelton W. David (1982), *Simulation Modelling and Analysis*, McGraw- Hill, chapter 6.
- Nellas V., Lykouropoulos N., Konstantoulakis G. and Mitrou N. (1994) "Experiments with artificial ON-OFF PC-produced traffic, with arbitrary, programmable ON and OFF sojourn time distributions", *EXPLOIT Traffic Workshop*, Basel, 14-15 Sept. 1994.
- Stern T. E. and Elvalid A. I. (1991) "Analysis of separable Markov-Modulated Rate models for information-handling systems", *Adv. Appl. Prob.* **23**, 105-139.
- WANDEL AND GOLTERMAN, "ATM-100, Description and Operation Manual" (see also  
[www.snh.ch/environment/measurement/wg-atm100.html](http://www.snh.ch/environment/measurement/wg-atm100.html)).