EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH

CERN–ECP/95–27
7 December 1995

RESULTS FROM AN ATM-BASED EVENT BUILDER DEMONSTRATOR

M. Costa, J.-P. Dufey, M. Letheren, A. Marchioro, R. McLaren, C. Paillard
CERN, 1211 Geneva 23, Switzerland

L. Gustafsson
Uppsala University, ISV, Uppsala, Sweden

A. Manabe, M. Nomachi
National Laboratory for High Energy Physics, Oho 1-1, Tsukuba 305, Japan

D. Calvet, K. Djidi, P. Le Dû, I. Mandjavidze
CEA Saclay, 91191 Gif-sur-Yvette CEDEX, France

T. Lazrak, Th. Lindblad, H. Tenunen
The Royal Institute of Technology (KTH), Stockholm, Sweden

M. de Prycker, B. Pauwels, G. Petit, H. Verhille
Alcatel Bell Telephone, Antwerp, Belgium

M. Benard
Hewlett Packard, Geneva, Switzerland

Abstract

ATM switching fabrics are good candidates to implement high-performance parallel event builders for the future data acquisition systems in particle physics experiments. We are studying their feasibility through simulations and implementation of event builder demonstrators. We present results from performance measurements made with a demonstrator based on a commercial ATM switch and on network interfaces that we have developed. The measurements are compared with the simulation studies and their scalability is discussed.

Presented at the IEEE Nuclear Science Symposium, San Francisco, 21–28 October, 1995.

1 INTRODUCTION

The RD31 project is evaluating Asynchronous Transfer Mode (ATM) (see for instance Ref. [1]) as a possible technology for implementing high-rate and high-data throughput event builders [2].

We have developed detailed computer simulation models of commercial or generic switches and various software tools that are used to implement models of most types of event builder architecture and data flow that are foreseen for the future experiments. The simulations carried out so far have shown that the ATM technology is a good candidate for several types of event building applications. Results can be found in Refs. [2] and [3].

In parallel with simulation studies, we develop small demonstrators in order to validate our understanding of the standards, to measure the performance of actual implementations, and to evaluate various traffic shaping schemes.

The aim of this contribution is to present performance measurements made with a 4×4 event builder demonstrator based on a commercial ATM switch and on ATM interfaces developed by the RD31 Collaboration. We shall first summarize the main features of ATM. The characteristics of event building over an ATM switching network and the specific problems caused by this type of traffic will be outlined. We shall then present the measured switch performance under various traffic conditions, using different traffic shaping schemes. A comparison of the results with the simulation studies will be made in order to evaluate the scalability of the event builder. The performance of two event-building algorithms will be compared in terms of their impact on global latency and event building frequency.

2 ATM TECHNOLOGY AND EVENT BUILDING WITH AN ATM SWITCH

ATM is a connection-oriented packet switching technology based on fixed length packets, called *cells*, of 53 bytes (5 bytes of header and 48 bytes of payload). Cells are routed through the network via virtual connections (VCs) which define the characteristics of point to point links. The standard requires the sequence of cells to be preserved on a VC. Multiple VCs can be opened simultaneously on a physical link. There is no connection overhead when a source switches from one VC to another.

Above this 'ATM layer', an *adaptation layer* is defined which offers a choice of standard adaptations to various types of services. For data transmission in event building we have selected the ATM Adaptation Layer 5 (AAL5) protocol which specifies the transmission of data packets with variable length up to 64 kbyte. In a source, an AAL5 packet is complemented with an 8 byte trailer that is terminated with a Cyclic Redundancy Check (CRC) and it is segmented into cells. Reassembly and CRC check occur at the destination.

The AAL5 protocol does not include data retransmission in case of error (e.g. cell loss). It is the responsibility of the higher level layers to implement it if needed. At the moment, no flow control is provided by the standard, but the ATM Forum is in the process of standardizing a flow control mechanism for variable bit rate services [4].

The physical layer can be implemented with various technologies that have been standardized by the ITU or the ATM Forum. We have selected the ITU SDH standard [5] which offers bit rates of (roughly) 155, 622 and 2488 Mb/s. ATM cells are placed asynchronously within frames that are emitted every 125 μsec. A small overhead is due to the SDH control bytes (1/27 at 155 Mb/s).

Various commercial chip sets implement the segmentation and reassembly (SAR) of AAL5 packets, the management of the VCs and the physical layer. For switches, the standard defines the services that have to be provided, but the implementation is not specified and many choices are possible, depending on the application domains. For telecommunications applications, where low-latency real-time services are supported, cells may be discarded in case of congestion, and buffering at every node of the switching network must be sufficient to ensure a low cell loss probability under random traffic conditions. For LAN applications, backpressure or credit based flow control mechanisms may be provided in order to avoid cell loss.

2.1 Event building with ATM

An event builder is made up of multiple source and destination modules interconnected via a switching network. For every event the source modules collect the data from a sub-detector to form an event fragment. Fragments are sent to a destination across the switching network. A controller assigns a destination for every event. Typically, for the Level-2 trigger, a sub-set of the sources is requested to send the whole, or a part, of the event fragment to a destination. For the Level-3 trigger, the complete event is assembled in the destination. The switching network supports simultaneous parallel data streams and multiple events are built concurrently.

A few parameters are of interest to characterize an event builder. The *event building latency* is the time between the decision to submit the event fragments to the event builder and the recognition, by the destination, that the event building is completed. The *load factor* is the maximum fraction of the available aggregate bandwidth of the switch that can be used for event building. The *event building frequency* is the rate at which event building can be accomplished. Source and destination *queue occupancies* are also interesting in order to determine the required amount of storage.

In an ATM event builder, VCs link each source to all destinations. These connections are virtual and are kept open permanently so that no connection set-up time overhead degrades the data transfer throughput to a new destination. In the simplest case of a ‘push’ architecture, the sources ‘push’ their event fragments, as AAL5 packets, towards the destination as soon as they are ready. Other schemes are possible such as ‘pull’ architectures, where the destination processors request the data from the sources.

The traffic pattern creates concentration in the switch and can lead to congestion and loss of data if no flow or traffic control mechanism is provided. Our simulations show that congestion can be avoided by using a switch with internal flow control, by applying traffic shaping or by combining both.

3 AN EVENT BUILDER DEMONSTRATOR

We are investigating a push architecture on a square event builder ($N \times N$). Every source has one VC with each of the N destinations. It should be noted that, as no information flows backwards from the destinations to the sources, it is not required to have a module connected on all the active output ports of the switch. The destination assignment (0..N-1) is implicit, the event k being sent to destination k modulo N.

The event builder demonstrator used for the performance measurements presented here is based on an 8×8 telecom switch prototype from Alcatel Bell (Belgium) [6] and on two types of ATM adaptors that we have developed: a VME–ATM interface and a simplified ATM traffic

generator/capture module. They can be combined in various ways to test event building architectures on a small scale (Fig. 1).

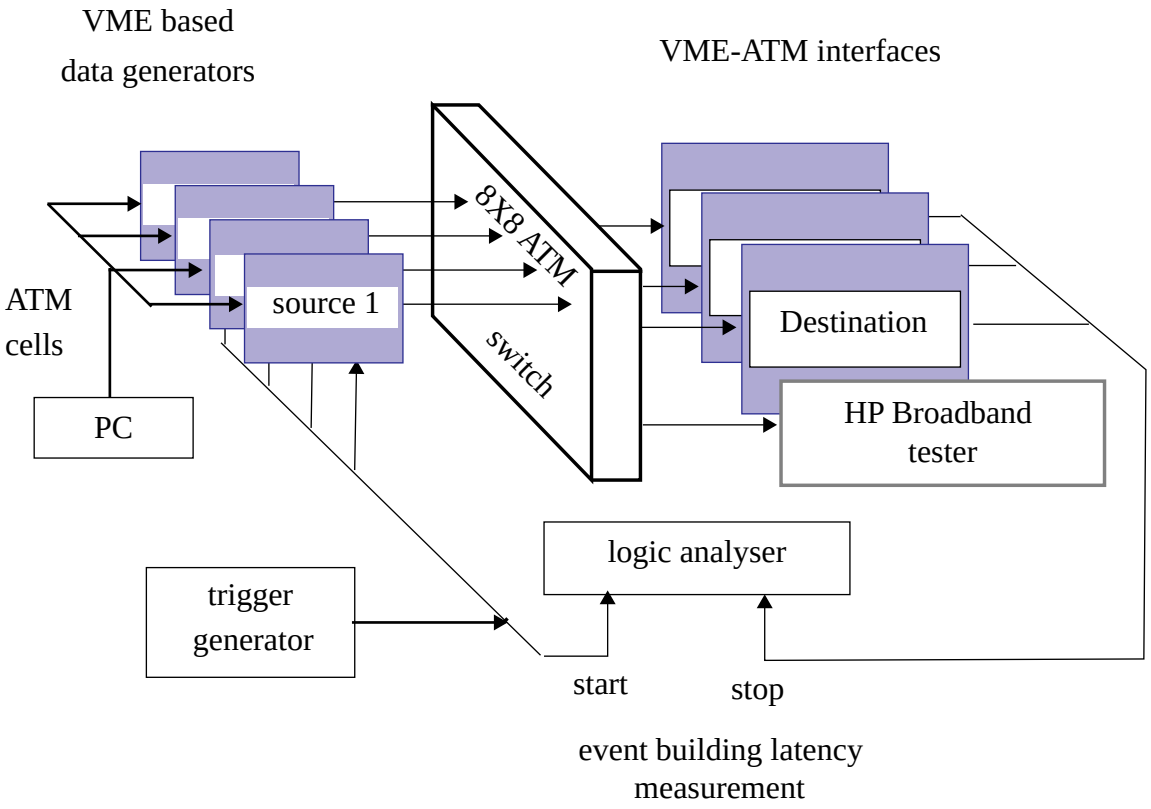


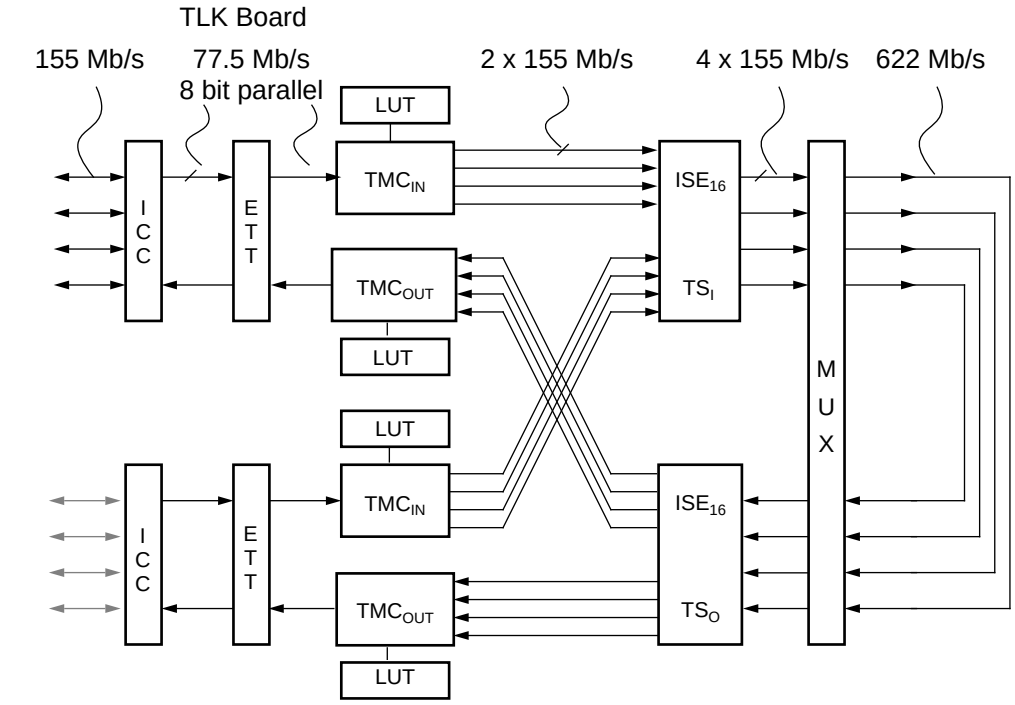
Fig. 1: Example of configuration of the event builder demonstrator.

A Hewlett Packard broadband tester [7] is connected to the switch and can act as a traffic generator or as a traffic analyser, measuring throughput and latencies and signalling the errors.

This section describes the hardware and software components of the demonstrator. Performance measurements will be presented in the next section.

3.1 The switch

A detailed description of the switch architecture can be found in Ref. [6] and a summary of the main features in Ref. [8]. It is a Multi-Path Self-Routing (MPSR) broadband switching fabric developed by Alcatel for public network applications. The architecture allows expansion of the switch to very large aggregate bandwidths in incremental steps. Figure 2 shows the layout of the eight-port version that we are using. Our present set-up is limited to the use of four ports only.



ICC - Interface Conversion Chip (Serial to parallel)

TMC - Transmission Mode Conversion (ATM to Multi Slot Cells)

LUT - Look Up Table

ETT - External Transmission Termination (SONET/SDH to ATM)

ISE - Integrated Switching Element (Switching of MSC-s)

Fig. 2: Switch layout.

3.2 The ATM adaptor

The VME–ATM adaptor developed by the RD31 Collaboration is a full duplex interface that implements the AAL5 segmentation and reassembly and the ATM layer (including the management of VCs) using the SARA-S and SARA-R chips from Transwitch [9]. The SONET/SDH physical layer, at 155 Mb/s, is based on the SUNI chip from PMC Sierra [10]. The SARA chipset implements the rate-division traffic shaping discussed later, and provides error checking of the data received. The ATM network adaptor is plugged into a VME RIO board from CES [11] which includes, as host CPU, a 25 MHz MIPS R3000.

3.3 The ATM traffic module

A simple VME module, that can be a generator or receiver of ATM traffic, has been developed with the aim of providing a low cost source and destination module. It uses the same physical layer implementation as the ATM adaptor. The ATM/AAL5 layer is replaced by a memory interconnected to the physical layer and to VME.

At the moment, the traffic module is used mainly as an event builder source and allows one to simulate easily several types of traffic shaping. The ATM cells, with the AAL5 structure, are created by a general-purpose program on the basis of specifications of the required global traffic pattern and then downloaded into the traffic module memory. An external trigger, sent simultaneously to all sources, initiates the delivery of ATM cells into the network. A VME-based PC can control several of these modules to download, analyse, define and edit ATM cells.

3.4 The protocol stack and the event building software

The software that drives the ATM interface is structured in layers. Figure 3 shows a simplified representation of the functional structure.

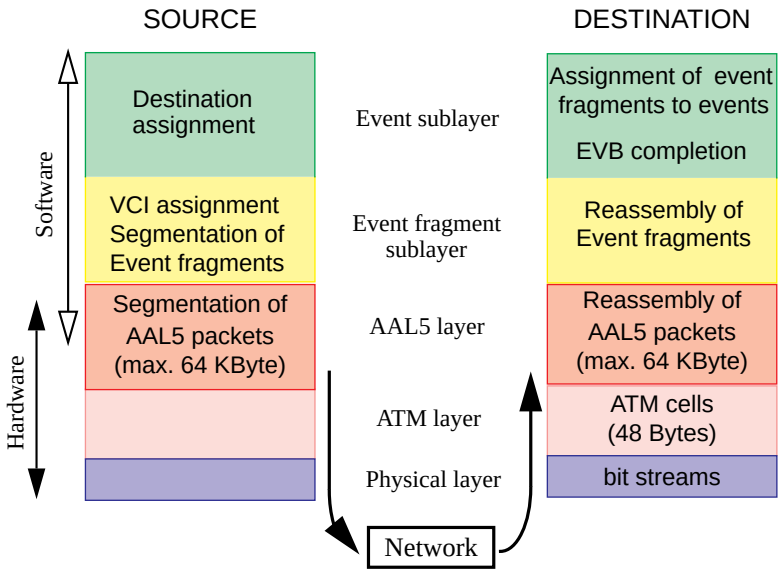


Fig. 3: Software layer structure.

The *event fragment sublayer* ensures independence from the packet size defined by AAL5 and more generally from the network technology. In particular, packets can be of any length. The *protocol data unit* (PDU) in which an event fragment is encapsulated is illustrated in Fig. 4. An *event fragment PDU* is formed by complementing the payload with the event number and the destination identifier. It is then segmented in one or more *event fragment SAR* (segmentation and reassembly) *PDUs*, which must include a source identifier and a fragment sequence number. This structure is then encapsulated in an AAL5 packet.

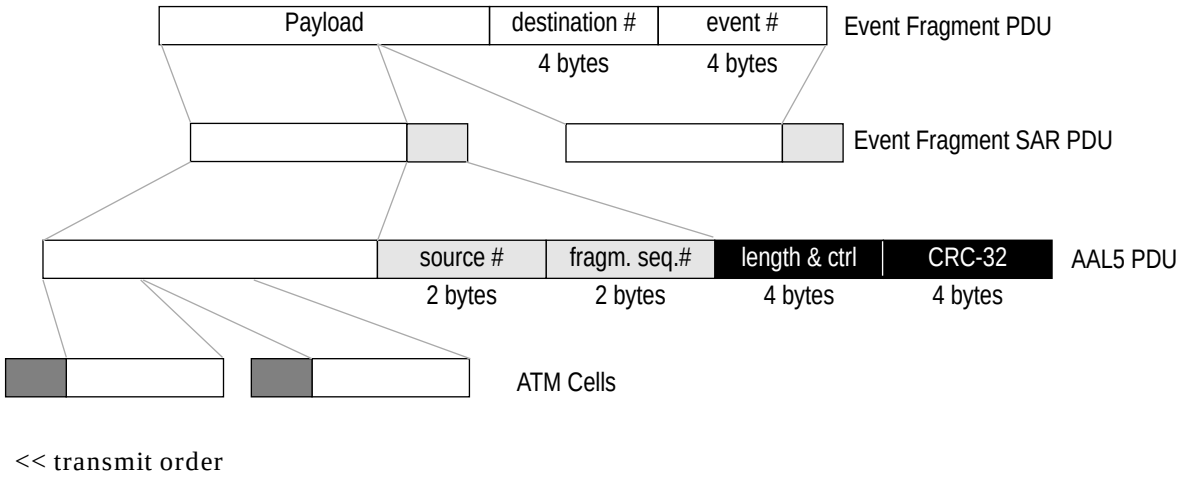


Fig. 4: ATM-based event building protocol data units (PDUs).

In the *event sublayer* of the destination, two algorithms have been implemented to determine when the building of an event is completed, namely the *time out* and the *notification* algorithms [12]. They will be briefly described in the next section.

Currently, no transport layer has been implemented to handle retransmission in case of errors. Errors are detected by checking the AAL5 CRC. Events with one or more errored packets are simply discarded. In the demonstrator, which contains many prototype boards, the probability of receiving an errored packet has been measured and is around 10^{-6} for packets of 1 kbyte. If a transport layer were considered as a necessity, it could be provided either by a standard network protocol like TCP/IP or, more simply, by a dedicated function in the event fragment sublayer.

As the destination will receive several event fragments, it is important for the scaling of the system that the performance is a linear function of the number (N) of fragments. This requirement has implications on the choice of data structures and algorithms, in particular on the algorithms used to search for fragments and to recognize the end of event building. In our demonstrator the time spent on one event is of the form: $t_0 + N \cdot t_f$, where t_0 is a fixed overhead for the global event building operation and t_f is the overhead for one fragment, including the data transfer.

In order to keep the overheads as low as possible, the event building software is designed to rely only on semaphore polling while interrupts are used only when errors occur.

4 PERFORMANCE MEASUREMENTS

4.1 Performance of the adaptor

The performance of the VME–ATM adaptor has been measured in transmission (source) and reception (destination). For the measurements there was no transfer of data between VME and the packet memory (it is assumed that suitable dual-ported memories and DMA block transfers will be available to feed the packet memory without additional overhead).

Figure 5 shows the measured performance for the ATM throughput and the fragment rate at emission and reception. The ATM throughput measures the amount of transmitted cell data, including the cell headers. After subtraction of the SONET/SDH overhead, the maximum theoretical limit is 149.76 Mb/s. The fragment rate measures the maximum frequency at which the interface can send, or receive event fragments of a given size.

In the source, for small packet sizes (up to 2 ATM cells, or 88 bytes of user data), the throughput is limited by software and hardware overheads (~ 12 μs). Although the bandwidth utilization is rather poor (~ 25%), the frequency is high and indicates that event building of small event fragments can proceed at high trigger rates. The lower performance for the receiver side, and consequently the lower frequency (38 kHz), is due to a higher software overhead at reception, where the event building protocol has more tasks to carry out.

For larger packets the software overheads no longer play a role since they proceed in parallel with the data transfer. The hardware link performance is determinant in this case. When the interface is used as a source, it can transmit fragments of 1 kbyte at 95% of the link bandwidth. When it is used as a destination, the interface hardware can receive at the maximum ATM speed. This guarantees that the interface can absorb bursty traffic without losing cells.

For the larger packet sizes, the lower throughput of the source, compared with the destination, is due to a small link inefficiency between the segmentation chip and the physical layer (a 16 bit link is required instead of 8 bits as implemented). This has been corrected on the destination side, where it is crucial to reach full bandwidth to avoid cell losses. In this case, Fig. 5 shows that the throughput curve has a perfect shape, increasing linearly in the interval dominated by the software overheads (~ 25 μs) and reaching full link capacity above.

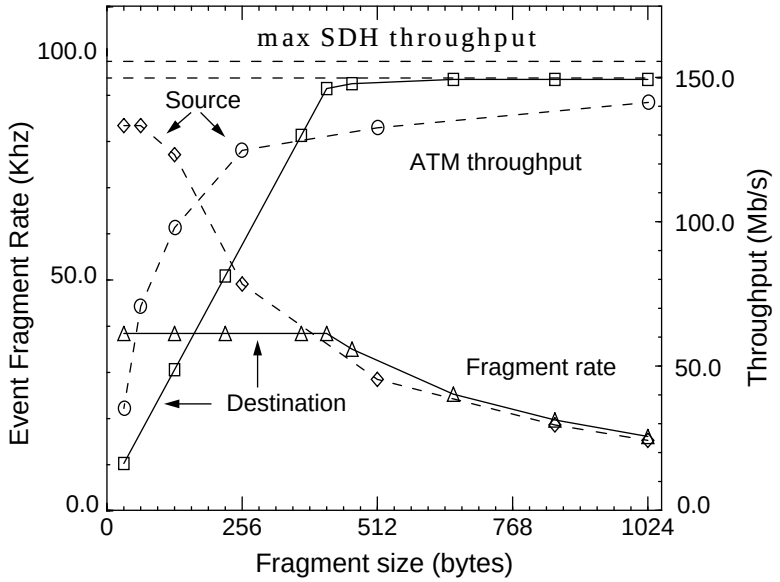


Fig. 5: Event builder source a nd destination performance.

4.2 Performance of a 4 × 4 event builder

In the present status of our development, the event builder test system uses four ATM traffic modules as sources, one ATM–VME interface and the HP broadband tester as destinations, while the other two destinations receive event fragments but are left open, which, as already pointed out, has no effect on the performance measurement. The event building latency is determined with a logic analyser that measures the time elapsed between the first cell of an event fragment submitted to the physical layer in the source module and a signal generated by the software in a destination, when the event has been completed. The broadband tester checks that no cell loss occurs, measures the fragment rate and throughput as well as the latency of the switch.

4.2.1 Congestion avoidance using traffic shaping

The four sources send event fragments of equal size to four destinations. We compared the behaviour of the switch for different traffic patterns by measuring the cell latencies through the switch and checking for cell losses. The traffic patterns implemented are:

a) ***no traffic shaping:*** each source sends the event fragments one after the other to the destination (FIFO of event fragments); the traffic has been tested at two different physical link loads (50% and 82%);

b) ***rate division:*** this is the traffic control provided by default by the SAR chip. The source maintains one queue of event fragments for each VC and one cell is extracted from each queue in a round-robin manner at a defined rate;

c) ***barrel shifter:*** the sources are synchronized by an external signal. A logical FIFO queue is maintained for every destination. A source extracts cells from the same queue during the time interval T_c between two signals and changes queue, at a new signal, in a well-defined way such that no two sources can send to the same destination simultaneously.

Figure 6 shows the results for all these cases and a comparison with a simulation model of the switch. If no traffic shaping is applied, the latency grows linearly with the size of the event fragment. This is an indication that the buffers in the switch are accumulating cells as a consequence of the concentration of data. In fact the last point of measurement in the top graphs is the limit beyond which cell loss occurs. Reducing the mean load per link from 82% to 50% has practically no effect in avoiding congestion.

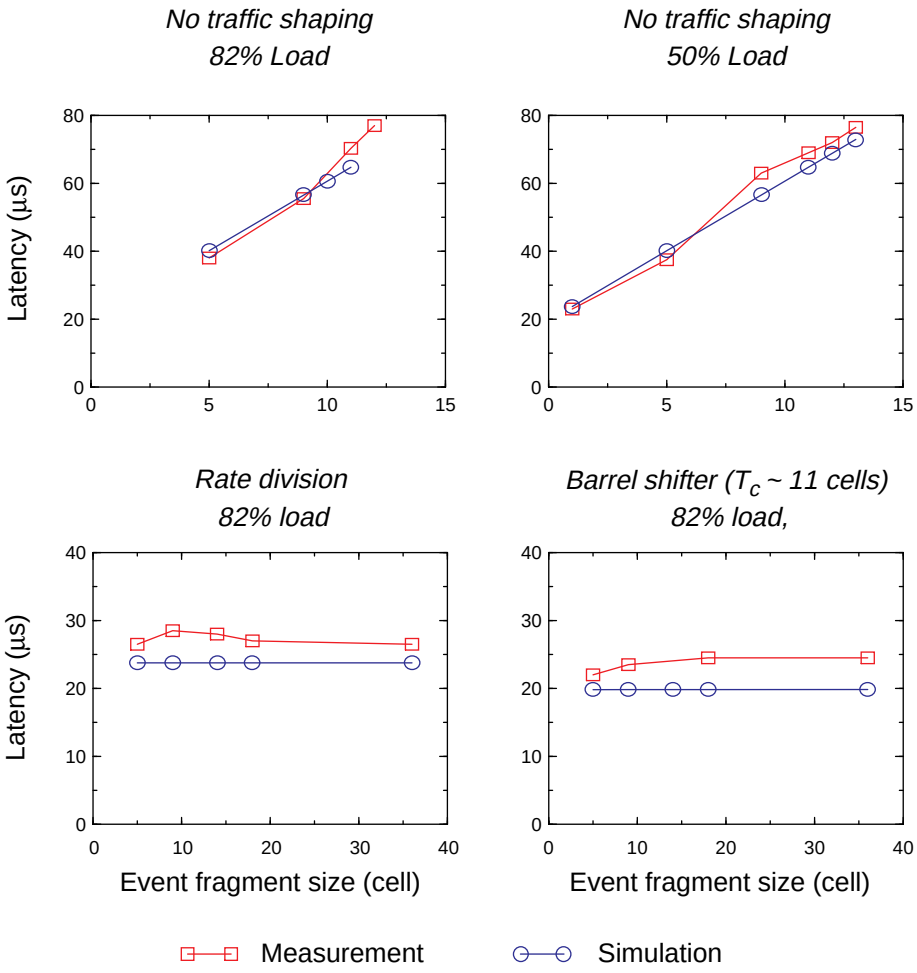


Fig. 6: Cell latency for different traffic shaping schemes.

The rate division and the barrel shifter schemes provide a good distribution of the traffic and do not result in accumulation of cells in the switch. For our experimental conditions, these two traffic shaping methods give the same results. However, one can expect a difference between the two schemes for larger switches because the rate division alone does not break the correlation between the sources and many of them can, at the same time, send a cell to the same destination. In fact the simulation shows that for a 16×16 event builder the rate division is not sufficient to avoid cell losses.

4.2.2 Full event building

In a full event builder, for each event, all sources send a fragment. Figure 7 shows measurements of the maximum event building rate and of the aggregate user throughput as a function of the fragment size. Rate division traffic shaping is applied for these measurements.

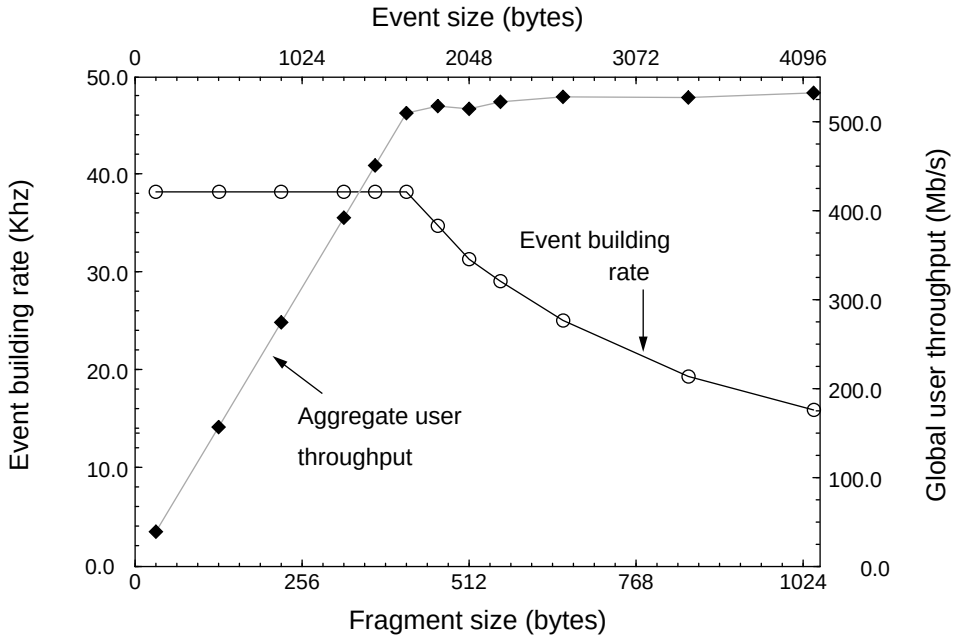


Fig. 7: 4×4 Event builder rate and throughput.

The event building rate remains constant (38 kHz) for fragments smaller than 412 bytes because the software overhead in the destinations is the limiting factor. For bigger fragments the link throughput determines the maximum rate, as already discussed. The maximum aggregate throughput of user data is 527 Mb/s (66 Mb/s). We should observe that, although the links are loaded at 99%, the switch does not lose cells. This is because only half of the inputs are in operation, resulting in a 50% total load factor of the switch. Using all the ports of the switch will lead to lower performance values because it will not be possible to use 100% of all links simultaneously. The maximum load factor will depend on the switch performance and on the efficiency of the traffic shaping.

In first approximation, for a 'square' $N \times N$ event builder, the rate performance does not depend on N: the rate that a destination can sustain varies like $1/N$ and the number of destinations is N. However, the scalability may depend on the acceptable load factor of the switch as a function of N.

We have also measured the event building latency for two traffic shaping methods: the rate division and the barrel shifter. Figure 8 shows the results as a function of the event fragment size.

When rate division is applied, the event building latency increases roughly linearly with the event fragment size. The destination receives one cell from each source in round robin and the event building time is proportional to the event fragment size (or to the size of the largest fragment in the case of fragments of variable size). If N event fragments of equal size are

expected, none of them is completed before the *N*-th round robin. Consequently, the operations on the event fragments cannot start before this delay.

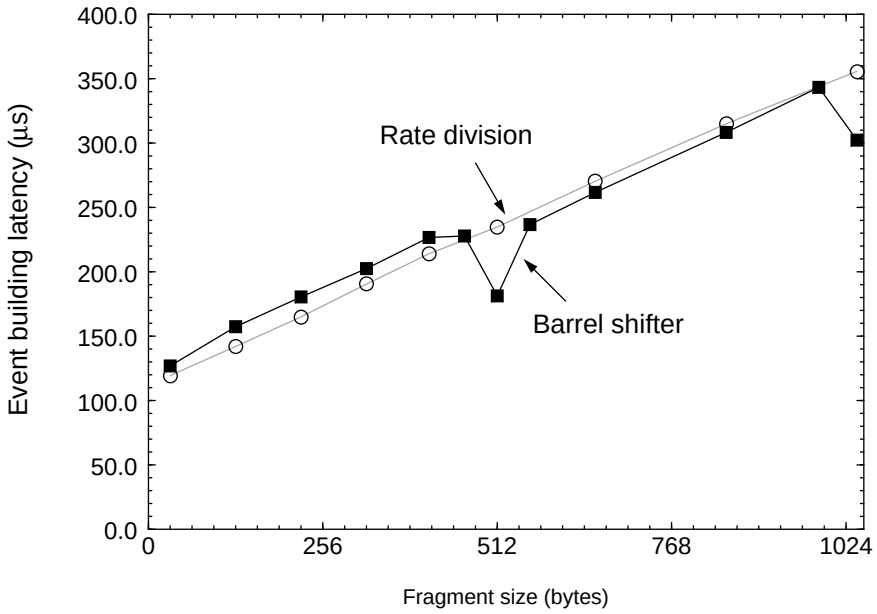


Fig. 8: 4×4 Event builder latency using the rate division and barrel shifter traffic shaping schemes.

In the barrel shifter case, we have chosen a constant time period (T_C) for all packet sizes, namely the time needed to transmit 11 cells. The particular case when event fragments have a size of exactly 11 cells is easy to understand: the event fragments arrive exactly one after the other and the operation for each one can proceed immediately and in parallel with the arrival of the next one. An additional overhead is needed to complete the event building. This efficient operation mode explains the dip observed for a size of 512 bytes (a similar effect of synchronization can be anticipated for sizes equal to a multiple of 11 cells and is in fact visible at 1024 bytes). For event fragments with size different from a multiple of 11 cells, the latency varies periodically as function of the event number (see Fig. 9).

As the event builder has a pure push architecture the throughput of the system does not depend on the latency and is the same using both traffic shaping schemes.

4.2.3 Event building algorithms

In many applications of event building, only a subset of the sources have data to send for a given event and some mechanism must be provided in order to determine when all non-empty fragments have been received in the destination.

We have implemented two of the proposed algorithms to determine the completion of the event building: *on time-out* and *by notification* [12]. In the time-out case, after reception of the first fragment of a new event, one waits a time sufficiently long to have a low probability of missing data. In this implementation we use, as approximation of time, the arrival of a certain number of event fragments (four in our test). In the notification case, when a source has no fragment for a given event, it sends instead a notification cell. In order to minimize software overheads, the notification is encapsulated in an Operation And Maintenance (OAM) F5 cell [5] instead of an AAL5 packet.

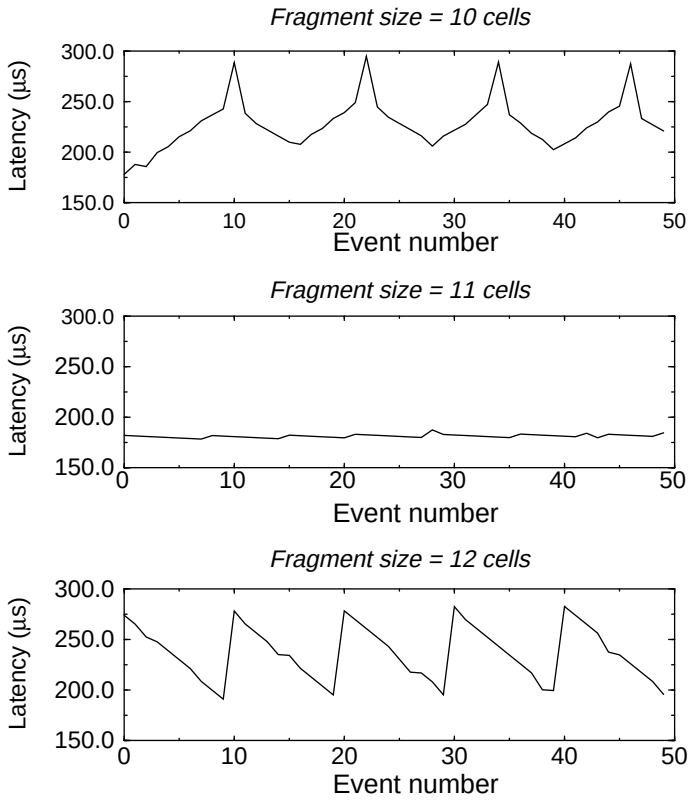


Fig. 9: 4 × 4 Event builder latency profile using the barrel shifter traffic shaping schemes for different event fragment sizes.

In both cases the rate division traffic shaping has been used. The event builder runs in sparse mode, corresponding to the Level-2 conditions: in our test, two sources (chosen randomly for each event) send event fragments of a fixed size while the other two either do not send any data (time-out algorithm) or send a notification cell (notification algorithm).

The event building latency and the maximum event building rate have been measured, for both cases, as a function of the event fragment size. The results are given in Fig. 10.

The event building latency is mainly due to the data transfer time (which increases linearly with the event fragment size). The switch contributes for a constant amount in the case of rate division traffic shaping, as seen before. The time-out algorithm, in its present implementation, adds a latency proportional to the event fragment size. The event building latency is expected to grow with the size of the event builder.

The event building frequency has the characteristic shape seen before. In the case of a sparse event builder, the time-out algorithm does not process information from each source and the software overhead is smaller than for the notification algorithm, allowing higher event building frequency. The curves of Fig. 10 show that, with the present software overheads, the cutoff occurs at ~ 256 bytes for the time-out algorithm and at ~ 512 bytes for the notification algorithm.

The more complicated software explains the lower performance of the notification algorithm compared to the full event building.

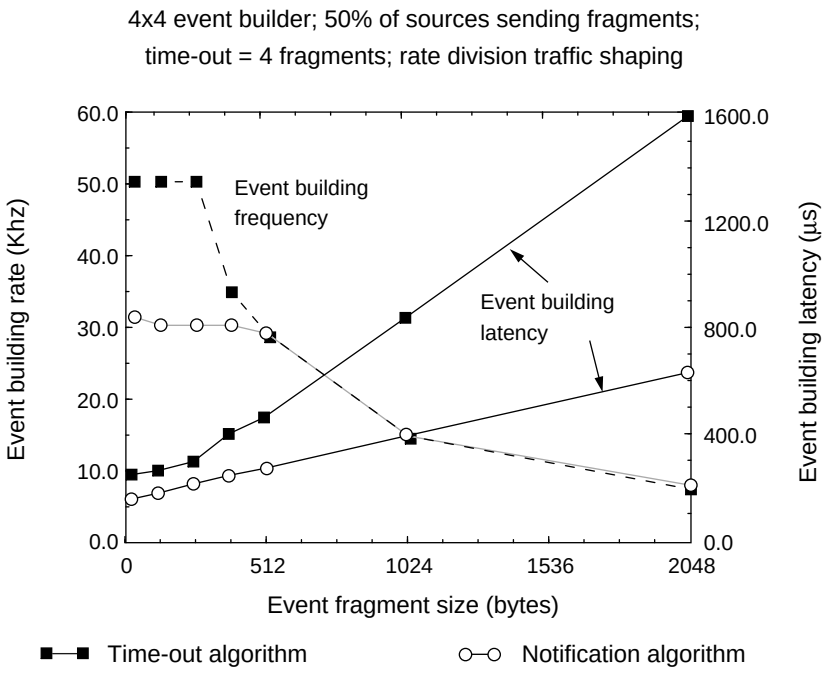


Fig. 10: Performance comparison of 2 event building algorithms.

5 CONCLUSION

A first small event builder using ATM has been set up and its performance measured. A complete event building software has been implemented in the source and destination modules. The system is completely decentralized in order to be scalable to very large event builder systems.

In the current implementation, using 25 MHz RISC processors, sources can send small event fragments at a frequency up to 80 kHz and destinations can receive them at 38 kHz (or higher if the number of destinations is superior to the number of sources). For small fragments the rate is limited by the software overheads. For larger fragment sizes the link throughput is the limiting factor.

Measurements of performance under various traffic patterns confirm the need for traffic shaping, as was previously shown through simulation. The implementation of different event building algorithms has allowed us to verify advantages and disadvantages of each of them in terms of global throughput and latency.

References

[1] M. de Prycker, “Asynchronous Transfer Mode”, 2nd ed., Ellis Horwood Series in Computers and their Applications, 1993.

[2] M. Costa et al., “NEBULAS - A high performance data-driven event building architecture based on an asynchronous self-routing packet-switching network”, CERN/LHCC/95–14.

[3] D. Calvet et al., “A study of performance issues of the ATLAS event selection system based on an ATM switching network”, Ninth Conference on Real-Time Computer Applications in Nuclear, Particle and Plasma Physics (RT95), Michigan State University, May 22–25, 1995. To be published in the Conference issue of IEEE Transactions on Nuclear Science.

[4] J.B. Lyles, “Definition of ABR Service Model”, ATM Forum document 94–0709, 18 July 1994.

[5] The International Telecommunication Union (ITU); recommendations G.707, G.708, G.709.

[6] M. Henrion, et al., “Technology, distributed control and performance of a multipath self-routing switch”, *in* Proceedings of the XIV International Switching Symposium, Yokohama, Japan, October 1992, vol. 2, pp. 2–6;
Th.R. Banniza, et al., “Design and technology aspects of VLSI’s for ATM switches”, IEEE J. Selected Areas Commun. **9**, No. 8. Oct. 1991.

[7] Hewlett Packard, Broadband Series Test System, 1994.

[8] M. Letheren et al., “An asynchronous data-driven event-building scheme based on ATM switching fabrics”, IEEE Trans. Nucl. Sci. **41**, No.1, Feb. 1994.

[9] Transwitch Corp., Shelton, Connecticut, USA, SARA chip set, Technical Manual, version 2.0, Oct. 1992.

[10] PMC-Sierra Inc., the PMC5345 Saturn User Network Interface (SUNI) manual, May 1993.

[11] Creative Electronic Systems SA Geneva, RIO 8260 and MIO 8261 RISC I/O processors. User’s manual, version 1.1, March 1993.

[12] I. Mandjavidze, “Software protocols for event builder switching networks”, International Data Acquisition Conference, Fermilab, Batavia, Ill., USA, Oct. 26–28, 1994.