EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH

CERN / LHCC / 95-14
LCRB Status Report / RD-31
6 March 1995

RD-31 Status report ’95

NEBULAS: High performance data-driven event building architectures based on asynchronous self-routing packet-switching networks

M. Costa, J-P. Dufey[1], M. Letheren[1], A. Manabe[2], A. Marchioro, C. Paillard
CERN, Geneva

D. Calvet, K. Djidi, P. Ledu, I. Mandjavidze
CEA DSM/DAPNIA, Saclay

P. Sphicas, S. Sumorok, S. Tether
Massachusets Institute of Technology, Cambridge, USA

L. Gustafsson, K. Kobylecki
Institute of Radiation Sciences, University of Uppsala, Uppsala

K. Agehed, S. Hultberg, T. Lazrak, Th. Lindblad, C. Lindsey, H. Tenhunen
The Royal Institute of Technology (KTH), Stockholm

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

M. Benard[3]
Hewlett Packard, Geneva

[1] Joint spokespersons.
[2] National Laboratory for High Energy Physics (KEK), Japan
[3] Research Grants Programme, HP European HQ, Geneva.

Table of Contents

1. EXECUTIVE SUMMARY

The goal of the RD-31 project is to demonstrate high-performance, parallel event building architectures that can satisfy the requirements for the level-2 and level-3 trigger systems of the LHC experiments. These architectures can be constructed around commercial or custom-designed parallel, multi-way switching fabrics. Many industrial switching fabrics are now available for switching traffic in broadband telecommunications networks or local area networks based on the Asynchronous Transfer Mode (ATM) standard. High-speed switches for the interconnection of computers and peripherals, based on the Fibre Channel standard, are becoming available also. Within RD31, event building architectures based on ATM switches have been studied extensively. Alternative architectures using custom-designed switching fabrics have also been explored. Investigations on the use of Fibre Channel switches are planned.

RD-31 was approved in November 1992 and the last status report was presented in January 1994. The DRDC assigned as milestones the tasks (1) “Design and simulate full data acquisition protocol for the ATM-based event building, with traffic shaping and internal flow-control options” and (2) “Demonstrate event building from VME microprocessor sources with ATM switch”. It was also recommended to increase contacts with the LHC experiments.

Simulation: the ongoing work on specific models of commercial switches has been complemented by a “generic” model including many options of switching network components and event builder traffic control techniques that allow us to quickly prototype models, based on most of the technologies available now or in preparation, and implementing a large variety of architectures. The flexibility of this tool has allowed us to evaluate quickly new ideas and new products. In addition to the ATM switching technology, a custom-designed switching fabric, optimized for data acquisition, has been proposed and evaluated.

An important issue in using switching fabrics for event building is how to control the traffic patterns so as to avoid internal congestion (depending on the switch architecture, congestion may result in lost data, poor throughput and scaling characteristics). An in-depth investigation of congestion control by the so called “traffic shaping” and “internal link-level flow-control” techniques has been conducted. Interesting results about the switches with internal flow-control have been found and will need confirmation from a demonstrator test bench. The combination of both techniques has also been evaluated. Configurations with large switches (up to 1024x1024) have been studied, whenever possible. The implementation of a model on a parallel machine is under way and should permit to study switches of size up to 2048x2048 ports and to simulate longer real time sequences. A new traffic shaping scheme has been proposed and complements the three that had been proposed and studied earlier.

Most of the data acquisition protocols simulated belong to the class of “push” architecture where the sources, receiving the identifier of a destination push their data through the switch. An investigation of a “pull” architecture, where the destinations play an active role and collect the data selectively, has been initiated with an application to the RoI concept of ATLAS in view. The results are encouraging.

Collaboration with the LHC experiments has led to detailed investigations of their respective event building architectures. This work is continued by groups that are members of the experiments and at the same time members (or collaborators) of RD-31. For CMS, the “full read-out” and the “virtual level 2" architectures have been simulated. For ATLAS, a detailed study of the data flows based on physics simulations and detector read-out scenarios has lead to a proposal for the level 2 and level 3 event building for the calorimeter. An alternative approach, using a “pull” strategy, has shown to be promising.

Event builder demonstrator hardware and software developments: A VME-ATM interface has been developed based on commercially available chip sets which implement the ATM protocols. A prototype has been successfully tested and operates correctly with standard equipment (an ATM switch from Alcatel, and a SONET/ATM tester from HP). It was not possible so far to reach the full performance expected, but we can still implement an event builder based on this interface (a small series is presently being manufactured). Simple data generators have been developed as a cheaper alternative and they can deliver data at full bandwidth through the switch. The software protocol layers and management functions, required for event building, have been developed and tested on the prototype. Some preliminary measurements have been made.

We expect the assembly of the demonstrator to be completed in the coming months, after which it will be possible to complete the implementation and measurement of various event building protocols and traffic shaping techniques. Another demonstrator, based on an internal flow-controlled switch (from AT&T), and using commercial interfaces is planned. Interfacing with the "intelligent" source memories has to be investigated. Studies of event builder management and control using standard ATM signalling protocols should be carried out in collaboration with the experiments in order to provide a user-friendly, self-regulating system. Simulation work should continue and new and more realistic traffic patterns should be studied using data from physics simulations and more detailed information about the detector read-out organization.

2. INTRODUCTION

The RD-31proposal [1] was originally approved on 26 November 1992, and the first status report to the DRDC was presented in January 1994 [2]. This document summarizes the work carried out by the collaboration since the previous status report. We recall here the milestones set by the DRDC for the second year of the project:

- Design and simulate full data acquisition protocol for the ATM-based event building, with "traffic shaping" and "internal flow-control" options.
- Demonstrate event building from VME microprocessor sources with ATM switch.
- In addition it was stated that "the project might benefit from increased contacts with the LHC collaborations".

Two new groups have joined the collaboration: Saclay in the framework of ATLAS and MIT in the framework of CMS. Some further changes in individual collaborators are reflected by the updated list of signatures on the cover page.

In the course of the year it has been recommended that RD31 should also study event building using other switching technologies, in particular Fibre Channel [3-4], by applying a similar method of investigation as the one adopted for ATM-based event building. It has not been possible to carry out any significant work in the domain of Fibre Channel, partly because it is difficult to obtain detailed information about Fibre Channel switches from industry, and partly due to a shortage of available manpower with the requisite skills. We have nevertheless prepared specifications, on request from a manufacturer, for simulation work that they proposed to carry out themselves [5].

We do not give an overview of ATM technology in this report. A summary of those aspects that are relevant to the event building problem can be found in the previous status report [2]. An ATM tutorial [6] and the B-ISDN standards [7] can be consulted by the interested reader.

3. TECHNICAL BACKGROUND

The principle of the parallel event builder architectures we are studying is the use of a switching fabric to interconnect the many front-end physics sources to the multiple "destinations" in which events are built for processing by the level 2 (L2) or level 3 (L3) trigger processor farms. Two standard, commercially available switching technologies seem promising candidates; these are ATM [7] and Fibre Channel [3]. Most of our effort has concentrated on commercially available technologies, and in particular on ATM-based solutions. Nevertheless, we have also investigated a custom-designed conical switching fabric architecture which has been optimized for overall DAQ system simplicity and cost.

If one excludes the switches based on shared media (busses), because they do not offer interesting scaling characteristics, the ATM switching fabrics are built of a number of elementary switching nodes interconnected in a web topology**.** The ideal switch would be an NxN cross-bar, allowing N independent paths to be established in parallel between the N inputs and the N outputs. The complexity of a cross-bar increases like N^2. The large ATM switching fabrics compromise by employing a network of elementary switching nodes in which the traffic of the N independent source-to-destination paths is packetized in ATM cells and asynchronously multiplexed over shared internal links. Contention for the internal links is resolved by introducing cell buffering in each elementary switching node. The complexity of these networks only grows as $N \cdot \log N$.

Internal links are not reserved for specific source-to-destination connections, but cells carry a label that allows them to be routed in hops between the switching nodes. The establishment of routing tables in the internal switching nodes allows the routing of cells according to their labels. Connections set up in this way are said to be *virtual connections*, and the label is called a *virtual connection identifier* (VCI). In principle the number of virtual connections is only limited by the size of the VCI tables in the switching nodes. In summary, the traffic flowing on the virtual connections is statistically multiplexed onto the physical resources of the switch (bandwidth on demand), which makes for efficient use of the hardware when traffic on the individual virtual connections is fluctuating widely.

3.1 The main classes of ATM switches.

From our point of view there are two main classes of ATM switch, which can be differentiated by the strategy they adopt when an internal buffer becomes full:

- those designed for the telecommunications industry, where expandability to large dimensions, low-latency and non-blocking characteristics are important. This class of switch simply drops incoming ATM cells whenever an internal buffer is full; therefore delivery of data is not guaranteed. However, under the “random” traffic pattern resulting from the aggregation of the traffic of a large number of independent subscribers, the probability of data loss is acceptably small (of the same order as the loss probability in a long distance link). For random traffic, the switch’s internal buffers are dimensioned to give a very low probability of loss, typically of the order of 10^{-10} or lower at 80% load on the switch (see for example [8])
- switches which implement a flow-control protocol on the internal links in order to guarantee lossless data transfer under all conditions. These are more likely to be used for LAN applications.

In switches of the first type, even if the network admission control system ensures that the connection characteristics do not exceed, on average, the resources of the switch, traffic burstiness and particular traffic patterns (e.g. concentration of traffic) can still produce overflow in some of the internal buffers. Usually the telecom switches implement some mechanism to indicate internal congestion to the subscribers, who can then use a higher level signalling protocol to slow down traffic at the input. However the reaction time of these higher level flow-control protocols is slow.

Whenever there is any correlation between the traffic flowing on different virtual connections, the probability of internal buffer overflow increases. In this case it is the task of the user-network interface (UNI) to regulate the traffic in such a way as to avoid congestion. This technique is called *traffic shaping* and can be used for event building over a telecom switch.

In switches of the second type an *internal flow-control* protocol is used to prevent buffer overflows in the switch by holding up the traffic flowing towards a nearly full buffer until sufficient buffer space becomes available. In this way, no cells are lost in the switch. However one must consider the case where the buffers of the fabric’s first stage of switching elements overflow and the case where the destination user buffers overflow. The ATM standard does not specify an action in those cases, except for a higher-level flow-control protocol which, as we mentioned above, might not react fast enough to prevent loss of data.

In all cases there are techniques to limit the data losses to acceptable values. A careful evaluation of the switching network by means of simulation is necessary to properly dimension the network and the interface buffers.

4. COMPUTER MODELLING

Computer modelling is an indispensable method to investigate event builder architectures based on large switching networks. The size of the system, the variety of the technological solutions (available presently or within the time scale of the LHC projects) and the abundance of architectural options exclude full scale prototypes. However, practical experimentation on small scale prototypes is a necessary complement to computer modelling. It allows us to confirm or to correct our understanding of the technology and to reveal some limitations that are not emphasized in the usual textbooks or even in the detailed documentation (if it exists at all!). The small prototypes can be used to verify the overall correctness of the simulation for that scale. But, we still depend on the correctness of the model when we use it to extrapolate to the performance of the full scale system. As a consequence, it is extremely important to develop high quality, accurate and reliable models, and to cross-check results with independent models.

This chapter presents the progress accomplished in ATM modelling since the last report. At that time, the development of a detailed model of the Alcatel switch had permitted investigation of data losses and traffic shaping techniques. A first model of an event builder based on the flow-controlled AT&T Phoenix switch had also been developed.

Since then we have recognised the need for a flexible modelling tool that would allow us to apply to the architectural research the variety of technologies that will be available in the time scale of the LHC projects, as well as the numerous methods of traffic control that can be envisaged. This has led to the development of a “generic model” described in section 4.1. This tool has been used to investigate large event builders with a variety of link speeds and switching element technologies, operated with various traffic shaping methods or with link-level flow-control. This model has revealed some interesting and unexpected behaviour of the flow-controlled switches and we expect to be able to observe these effects on a real system soon.

The development of a switch model using the technique of parallel simulation is now under way. It promises to allow us to simulate very large switches (2048 x 2048) and to run over much longer real-time sequences than can be achieved on sequential machines.

A detailed investigation of a custom switching fabric optimized for the event building task has been undertaken and has delivered very valuable results

The modelling of the proposed architectures for ATLAS and CMS has occupied a large fraction of our efforts, and has led to contributions to the technical proposals and to the publication of back-up reports. A couple of models of event builders realised in MODSIM have been our contribution to the global model of the ATLAS data acquisition system. In addition, a detailed investigation of the probable data flow scenarios for the ATLAS L2 trigger has led to the proposal of original architectural concepts. For CMS, the “virtual level 2" architecture has been studied and encouraging results have been obtained.

The use of standard simulation tools has been extended. Apart from μC++, models are available in C++ and MODSIM. As mentioned before, cross-checking of simulation results obtained with independent models, possibly in different languages, has proved to be necessary in order to remove bugs and also to show that unexpected behaviour was not due to modelling approximations adopted in a specific model.

4.1 A generic event builder model

The modelling activity aims at evaluating and optimizing the performance of a particular event building architecture; poor performance is due to bottlenecks and, in order to obtain the desired performance, many parameters of the system can be adjusted. Each object constituting an event building system has a complex behaviour which depends not only on its own architecture but also on the behaviour of the other objects. Examples of such sub-systems are: the distribution of event data fragments among the sources, their size distribution, the discipline which sources follow while segmenting and sending the event fragments to destinations, the architecture of the switching network, the protocol which allows a destination to determine when the event building process for an event is finished, strategies for assigning events to destinations, etc. We need to understand how those parameters (and many others) influence the performance of the overall system.

Consequently, it is desirable to have flexible tools within the model which allow easy modification of the architecture and investigation of ways to optimize its performance. To this end, a generic event builder model, shown in figure 1, has been developed. It is flexible enough and allows easy change of various system parameters via screen menu or from a parameter file. In order to facilitate the debugging process it can update the statistics on-line, on the user screen, as the simulation task is executing. We give below a description of each module with the options offered (*in italic*) by the generic model.

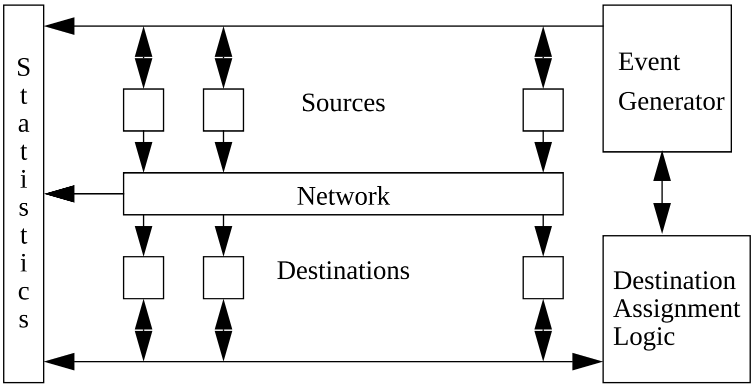


Fig. 1 Block diagram of the generic event builder model

The ***Event Generator*** decides when a new event has to be created. The inter-trigger delay can be *constant* or follow a *negative exponential* or *geometric distribution*. The value of the mean inter-trigger delay parameter determines the average trigger rate. Also the minimum time between two events can be specified. For each event the event generator creates event data fragments and distributes them among the sources. The size of event data fragments can be *constant* or follow various distributions (*flat, exponential, erlang, normal*). The mean, min., max and variance determine the shape of the distributions. When using one of the above distributions, the sizes of the event data fragments in different sources are assumed to be uncorrelated. In order to study source correlation effects, the event generator can read event data *from a file*, which can contain, for example, events generated by Monte Carlo. From the Destination Assignment Logic the event generator obtains a destination's identifier to which the event has to be sent and it passes this information to the sources.

The ***Destination Assignment Logic*** assigns events to destinations following one of various possible strategies: *simple periodic (sequential or butterfly), random* or a more complex assignment which *takes into account the status of the destinations* (current occupancy, number of events being scheduled to the destination, etc.). A suitable destination assignment strategy in conjunction with some traffic shaping scheme can significantly reduce the congestion probability in the switching networks [8].

In the ***Source Modules*** event fragments are associated with the virtual connection to the assigned destination. If necessary, the sources provide the necessary buffering and queuing of the event fragments. The event fragments are packed in the ATM AAL5 packet format and then segmented into cells, which are injected in the switching fabric. In this way the corresponding ATM and AAL5 overheads are modelled. The segmentation and cell injection strategies can be selected from *a simple FIFO* scheduling (no traffic shaping), or a *traffic shaping scheme (Cell Based Barrel Shifter [2], True Barrel Shifter [9] or Randomizer [2])*. Recently a couple of other flow-control methods have been added to the source module behaviour, such as *Static Rate Control* and *Dynamic Rate Control techniques* (their effect on the event builder performance is a subject for future studies). Of course, the number of source modules in the event building system is variable.

The ***Destination Modules*** receive, cell-by-cell, the event fragments sent by the sources, and they reassemble them. The event fragments are associated with the event structure to which they belong. Several events can be built simultaneously in a destination (especially, when source traffic shaping schemes are used to reduce contention in the fabric). One of the following protocols, which allow a destination to determine when the event building process is finished, can be chosen: *Known Sources, Empty Records or Time-Out [10].* The number of destination modules in the event building system is variable.

The ***Switching Network*** is built with a regular interconnection topology of the switching elements that can be either *Banyan or Omega*. Switching elements can be of variable size *(2x2, 4x4, 2x4, 8x4, etc)*. Contention resolution in the switching elements can be selected from one of the following methods:

- *shared media switching element with no link-level flow-control* (Fore Systems type [11])
- *shared memory with no link-level flow-control* (Alcatel/HSS type [12])
- *output queueing with link-level flow-control* (AT&T/Phoenix type [13])
- *shared memory with link-level flow-control* (IBM/Prizma type [14])

The switching element queue and buffer sizes are variable and can even be infinite. The switching element link rates are also variable and can be chosen to be either *160Mbit/s, 320Mbit/s, 640Mbit/s, 1.28Gbit/s or 2.56Gbit/s*. In the generic model the switching elements operate on a cell, or transmission unit, of length 64 bytes, which carries 56 byte user payload. When studying ATM switching fabric implementations by particular vendors one finds, in most cases, that the internal switching elements operate on cells with a proprietary format (e.g. 55 byte cells for AT&T Phoenix). The size of the event builder switching fabric is variable.

During simulation runs various interesting statistics are gathered and stored in the form of distributions, tail distributions or tables. The statistics can be visualized on-line or stored in ASCII files for off line analysis. The source and destination buffer occupancies, the event building latency, the network load, the switching elements’ buffer occupancy are a few examples from the list of all the variables whose behaviour is monitored and analysed.

It is worth mentioning that the generic event building model is restricted to the data flow aspects and does not model possible control traffic due to the event building protocols. Presently the generic model simulates one unique switching fabric that interconnects the source and destination modules. Recently provisions have been made for allowing the event building switching network to be formed by several interconnected fabrics. The interoperability and performance of cascaded fabrics will be a subject for future study.

The generic model has been developed in μC++ [15], an extension of the object oriented C++ language towards concurrency and simulation. When possible, the results derived from the simulations have been compared and successfully cross-checked with the results of other modelling activities which use C++ [16] and Modsim [17] as the simulation environment. A C++ model is under development [18] and will provide easy portability, in addition to several other advantages.

4.2 Comparative performance evaluation of traffic shaping versus flow-control techniques.

4.2.1 Traffic shaping: the true barrel shifter.

As was mentioned elsewhere [2], the continuous strong concentration of data streams, typical of the event building traffic, creates congestion in switching fabrics. In the networks, which do not exploit link-level flow-control (WAN switches), unacceptably high cell losses occur due to buffers overflowing in the switching elements. Thus the traffic originating from the sources has to be shaped before being injected into an event building switch. The effect of a well-chosen traffic shaping method is to reduce data loss probabilities in WAN-type networks to acceptably low values, to avoid unpredictably long event building latencies and to limit the required memory in the front-end source modules.

Traffic shaping has been described extensively [2, 19]. To summarize it consists of:

a) allocating an average bandwidth to all virtual connections between sources and destinations in such a way that the aggregate average bandwidth seen at each destination does not exceed the available bandwidth at the output port;

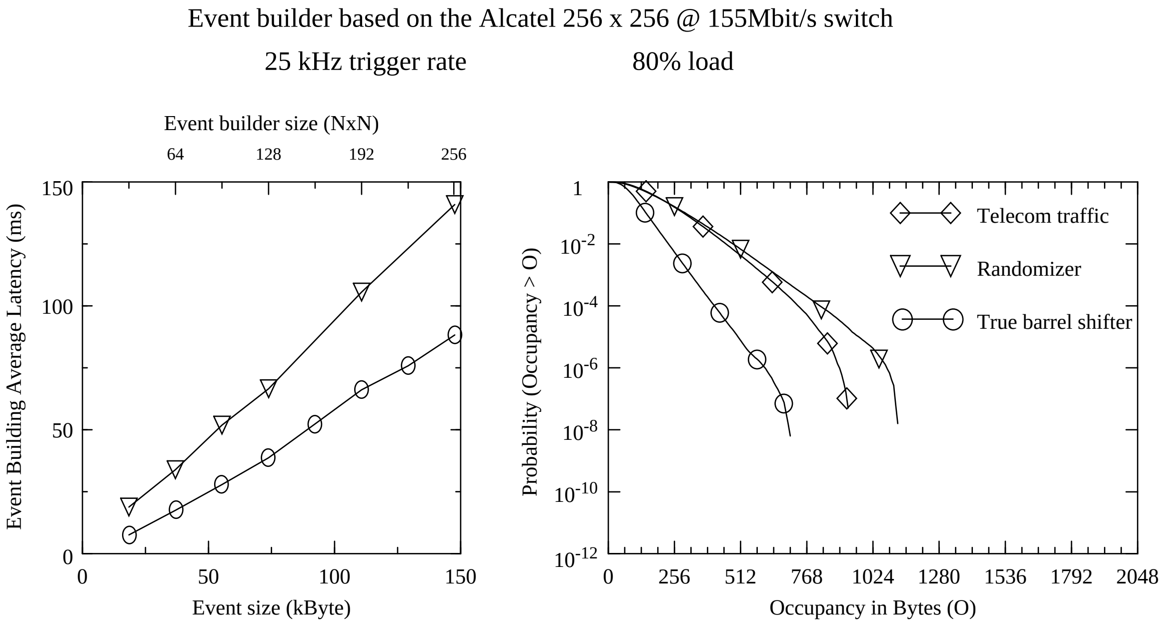
b) breaking the instantaneous time correlation between cells emitted from all the sources towards the same destination, as a result of the trigger.

The need for hardware traffic shaping, may preclude the use of commercial ATM interfaces. Thus it is very important to find a traffic flow-control scheme which presents the following characteristics:

- guarantees acceptable data loss probabilities due to residual congestion in the WAN ATM fabric;
- results in low event building latencies and source/destination buffer occupancies;
- needs a minimum of (or preferably no) specialized hardware to be added to a detector front-end source ATM interface;
- requires minimum (or preferably no) centralized control of detector front-end sources.

Among the traffic shaping schemes that have been studied, one suffers from low bandwidth utilization (the event-based barrel shifter), another requires a strict synchronization of sources (the cell-based barrel shifter), while still another needs dedicated hardware (the Randomizer) thus excluding the use of commercial ATM interfaces in the sources.

A new, conceptually simple, traffic flow-control scheme, referred to as “True Barrel Shifter”, has been proposed [9]. It presents almost all the desirable characteristics mentioned above. The scheme is very suitable for the level 3 event building processes. To summarize, the event building system operates as a barrel shifter which changes its states after a time period approximately equal to the emission time of an average event fragment size. A strict synchronization of the sources is not necessary: at the transition, and during a short period of time corresponding to the inaccuracy of the synchronisation, it can happen that some destinations receive data from 2 sources. But this small bursty traffic can easily be smoothed out by an event building ATM switching fabric. Commercially available ATM interface chips offer all the necessary features to configure the source modules with the true barrel shifter requirements (hardware-maintained linked lists of virtual connections, activation-deactivation of virtual connections, segmentation processes, switching from one virtual connection to another with no overhead). It has been shown [9] that event builder systems which use the “true barrel shifter” traffic shaping scheme, scale linearly to large dimensions (figure 2(a)), while maintaining low cell loss probabilities (figure 2(b) shows that the probability of overflow the 2 kByte buffer of a switching element is less than ~10^{-12}).



a) Scaling of event builder

b) Cell loss probability for the 256x256 event builder

Fig. 2 Simulation results for an event builder with true barrel shifter traffic flow-control scheme

It has also been shown that the performance of an unbalanced system (a system with sources generating event fragments with different average sizes) does not depend too much on the barrel shifting time period: the event building latency remains the same whether the time period is equal to the smallest, the average or the biggest event fragment transition time ([9, 20]).

4.2.2 Use of traffic shaping in a flow-controlled switch (AT&T Phoenix).

The combination of link-level flow-control and the traffic shaping technique has been studied for an event builder based on the AT&T Phoenix switch [21]. The switching element operates at 400 Mbit/s link speed and transports 55 Bytes long cells. Network adapters, based on the ALI chip [22], are used as inlets and outlets in order to match the external 155 Mbit/s link speed with the internal 400 Mbit/s rate, and to perform the conversion between the external ATM cell format and the internal proprietary cell format. For reasons of simplicity, the network adapters have been modelled as FIFO queues of cells. Backpressure, originating from the output network adapters or from the switching elements can propagate as far back as the input network adapters (as shown in fig 3), but there is no hardware signal to transmit the backpressure to the source. If the backpressure persists for a long time, while cells are still delivered to the congested inlet, the buffer in the network adapter can overflow and the cells can be lost. Therefore, although the cell loss probability is zero in the fabric, cell losses can still occur in the network adapters. One could try to estimate, by simulation, the required buffer size in the network adapters in order to achieve low cell loss probabilities.

Simulations have been conducted for an event builder which operates at 10 kHz trigger rate. Sources generate event data fragments with an average size of 1000 Bytes and a maximum size of 5000 Bytes. The resulting bandwidth utilization is 60% of the 155 Mbit/s links, but due to bandwidth expansion inside of the fabric, the core of the switching fabric operates at approximately 25% load. For the 1024x1024 event builder, the total event size is equal to 1024*1000 = 1Mbyte.

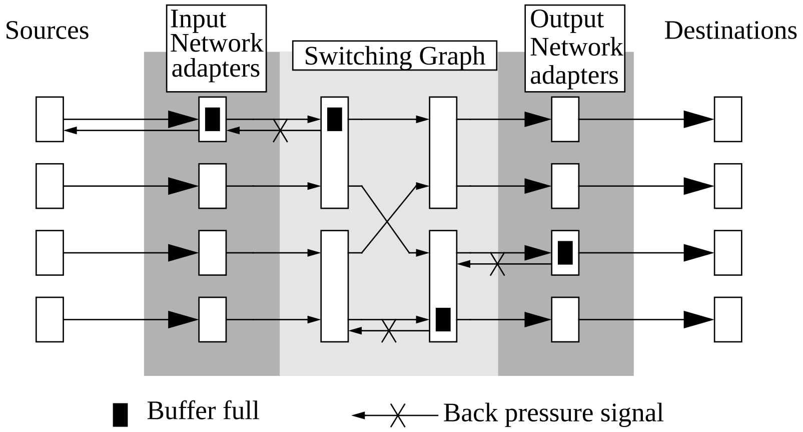


Fig. 3 Buffer overflow and backpressure signals

Initially no source traffic shaping has been applied. The source buffers were simple FIFO queues of event fragments injecting cells at full 155 Mbit/s speed. Event builder systems up to 256x256 have been modelled. Larger systems could not be simulated because of the required memory space and long execution times. The event building latency and the buffer space required in the input network adapter scales linearly with the size of the event builder and, for the 256x256 event builder, each network adapter had to buffer 5000 Cells (270 kBytes) in order to guarantee a cell loss probability lower than 10^{-6}. From the simulation results one could expect that a network adapter buffer size of 1 Mbyte would be necessary for the 1024x1024 event builder system. The effect of the rate control technique has not been studied and this is a subject for future investigation. We expect that rate control of sources will decrease the level of congestion in the fabric and, as a consequence, the network adapter occupancy.

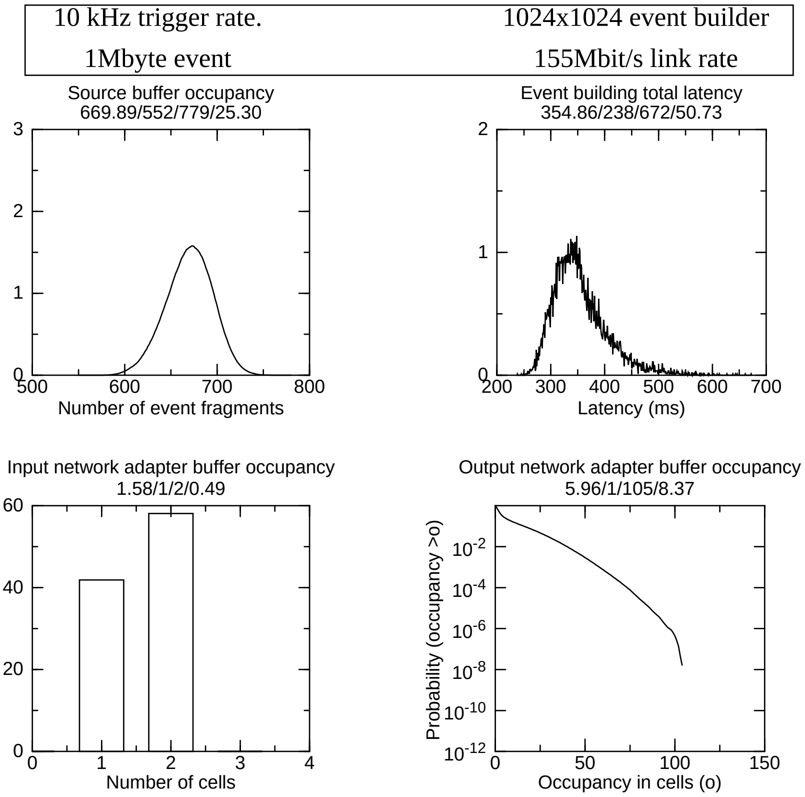
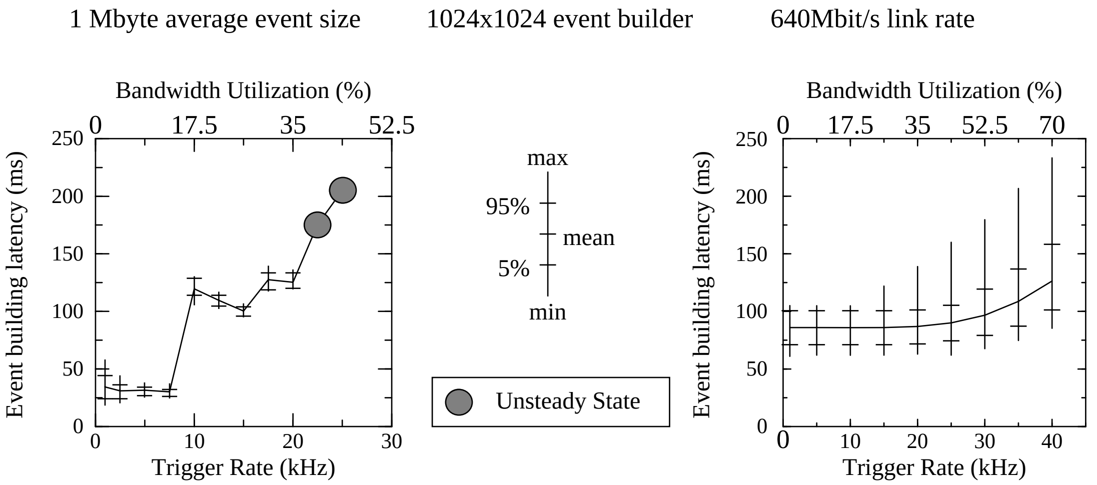


Fig. 4 Simulation results for event builder which combines flow-control and traffic shaping techniques.

When traffic shaping (randomizer) was applied at the sources, it became possible to simulate a 1024x1024 event builder. Figure 4 presents the simulation results. On average 355 ms were necessary to build events of 1 Mbyte in a destination. This time includes also the queuing time of the event fragments in the source modules. The source modules have to buffer on average 700 event fragments. A buffer space of 1 Mbyte in the sources (easy to provide) would guarantee a very low probability of buffer overflow. On the other hand, the network adapter occupancy charts (along with some other statistics) indicate that contention in the fabric never propagates back to the input network adapters (the input network adapter occupancy never exceeded two cells for approximately 350x10^6 cells injected in the fabric). From the tail distribution of the output network adapter buffer occupancy (see figure 4) one can estimate that a buffer of 150 cells in this adapter will guarantee a cell loss probability much lower than 10^{-6} (today, network adapters can buffer up to 256 cells and more).

4.2.3 Study of the flow-control and traffic shaping techniques based on the generic event builder model.

We have seen, in the previous section, that the simulation of large switches with pure flow-control (without traffic shaping) was not possible with the exact model of the Phoenix switch. Hence, the generic event builder model described in section 4.1 has been used for this study. Sources generate on average 1 kByte of event fragment data. A 1024x1024 network was interconnecting source and destination modules and was constructed from 4x4 switching elements with 16 kByte of memory each. The fabric operates with 640Mbit/s links, therefore the aggregate bandwidth equals 640 Gbit/s. Figure 5 shows the event building latency versus trigger rate.



a) flow-control extended up to sources

b) Source traffic shaping

Fig. 5 Simulation results for the generic event builder

In one case (a) no traffic shaping has been applied to the source modules, but backpressure, if necessary, could propagate as far back as the sources and prevent them from transmitting data. For trigger rates below 10 kHz, there is very weak dependency of the event building latency on the trigger rate. In the destination modules only one event is built at a time. For trigger rates in the range from 10 to 15 kHz there are always two events being built concurrently in a destination and event building latency stabilizes around 110 msec. The third state (if one can describe the system behaviour in terms of states) is characterized by three events being built concurrently in a destination and is observed in the 17-20 kHz range. Event building latency continuously increased and system steady state could not be reached for the trigger rates above 22 kHz, even though the load on the switching fabric was not too high (around 40% of the available aggregate bandwidth). The same behaviour has been observed for the event builders which operate at 2.56 Gbit/s link rates, the maximum reachable trigger rate for a stable system being 80 kHz. Those results were qualitatively confirmed by means of two other simulation programs, one in C++ (for conic event builders) and one in Modsim (for the event builders based on the AT&T switching fabric).

Figure 6 shows how the event building latency changes with time, starting with an empty system.

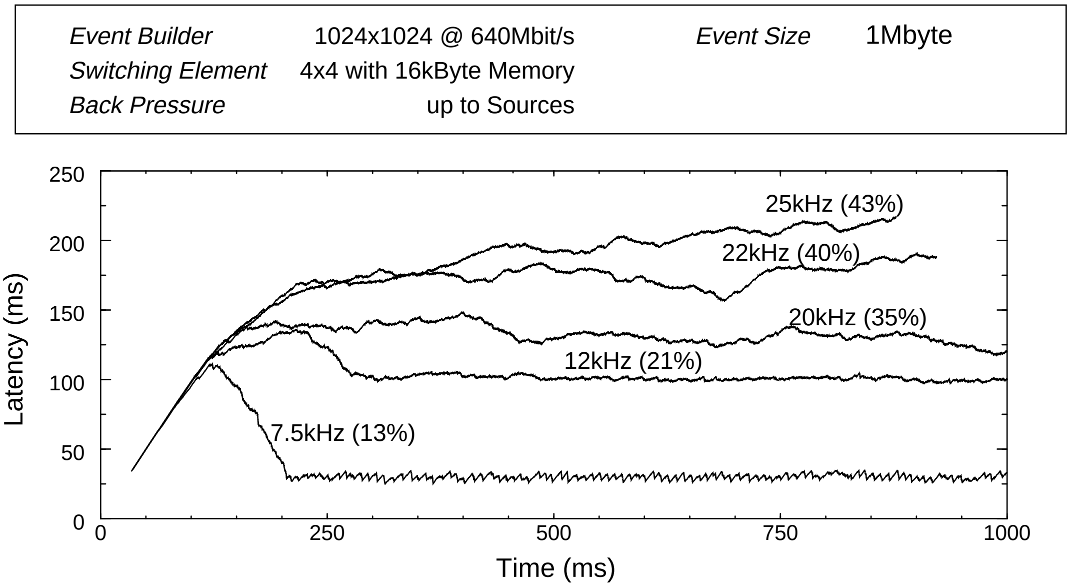


Fig. 6 Latency profile in a fabric with link level flow-control

For low loads, one can see that the latency grows up to a maximum value and then drops to a lower value where it remains stable. This stable value grows with the load applied to the switch, but it grows by steps, and not in a continuous way. Beyond some value of load, the latency keeps growing with time: the system cannot stabilize and the event builder is not usable under this condition.

In order to understand better the observed behaviour of flow-controlled systems, a more detailed study of a small (16x16) event builder has been carried out. It has shown that, due to backpressure, some kind of self-organization in the system leads to minimizing contention in the switching fabric. For example, by analysing the occupancy of sources it has been found that they are divided into groups which never send data to the same destination at a given time. A more detailed study of the observed phenomena could be a subject of future work.

Scaling characteristics of the event builders with flow-control, depend on the switching network architecture and event building traffic characteristics (trigger rate, event size, network load) [23].

It should be mentioned that the effect of using the rate control technique on event builders with flow-control has not been studied and remains a subject for future investigation. One can expect, that rate control of sources will decrease the level of congestion in the fabric and, as a result, will lead to higher bandwidth utilization.

In the second case (case (b) on figure 5) the randomizer traffic flow-control scheme was used to prevent congestion in the switching fabric. Available system bandwidth utilization up to 70% has been observed without significant performance degradation. These simulation studies confirm results obtained previously, which have been successfully cross-checked against queuing theory [20]. It has also been shown that event builders which exploit traffic shaping techniques can be characterized by good scalability: event building latency depends linearly on the system size [23]. Moreover, assuming that the traffic shaping guarantees the same level of cell loss probability for two different types of network architecture, the event building latency, the source/destination buffer occupancies do not depend on the particular network architecture, but are determined by the traffic shaping technique.

4.3 Parallel simulation for large switches.

As already mentioned, the simulation of event builders based on switches with internal flow-control requires a very large memory space and long execution times. Thus, our simulations of event builders based on the AT&T Phoenix switching element (when we do not apply traffic shaping) have been limited to fabrics not exceeding the size 256x256.

We intend to use a parallel simulation environment, SIMA[24] to simulate large ATM switches on parallel computers. We expect to be able to simulate very large switches (up to 2048x2048) on a Convex multiprocessor system. The parallel approach will also allow runs simulating longer real time spans than those of the order of one second, which we typically achieve after many hours of computing on a high-performance uniprocessor workstation.

4.4 A custom-designed conic switching fabric

A custom-designed conical switching fabrics, employing internal link-level hardware flow-control, has been proposed as an alternative architecture to those based on square commercial switching fabrics. The conical fabric has M inputs and N outputs, where M>N, and it has been proposed specifically as a simple, optimized solution for event building [1, 16, 25-26]. The conical fabric connects directly to the front-end modules via a large number of low speed input ports. It simultaneously performs the cell switching function and a data multiplexing function. It has the advantage of providing a homogeneous data acquisition system architecture, whereas in the case of the square commercial switches, the multiplexing function must be provided by specific dedicated hardware upstream of every input port.

A model of a proposed conic event builder [25] for the Euroball experiment [27] has been developed. It was based on an optimized, partially interconnected, 6-stage Banyan network with 1536 inputs (connected directly to front-end modules) and 64 outputs. The proposed fabric could be constructed from very simple custom switching ASICs, each with 4 inputs and 2 outputs.

4.5 Modelling of the ATLAS architecture.

4.5.1 MODSIM model of the switches

A model of an ATM switch, based on the PHOENIX AT&T switching element [21] has been developed [17], using the MODSIM language [28]. It can be plugged into the ATLAS DAQ model [29]. This work also had the goal of comparing the performances of switches with and without the use of internal flow-control. It has confirmed that, due to the bandwidth expansion inside of the switch, rather high loads (75%) can be applied to the Phoenix based switching fabric. Event building latencies are significantly lower than in the case of a switch operated with traffic shaping.

The MODSIM simulation allowed us to model switches up to size 128x128 and the μC++ program has confirmed the favourable scaling of latency up to a size of 256 x 256, but it has shown that the buffering space required in the network interfaces is probably too high.

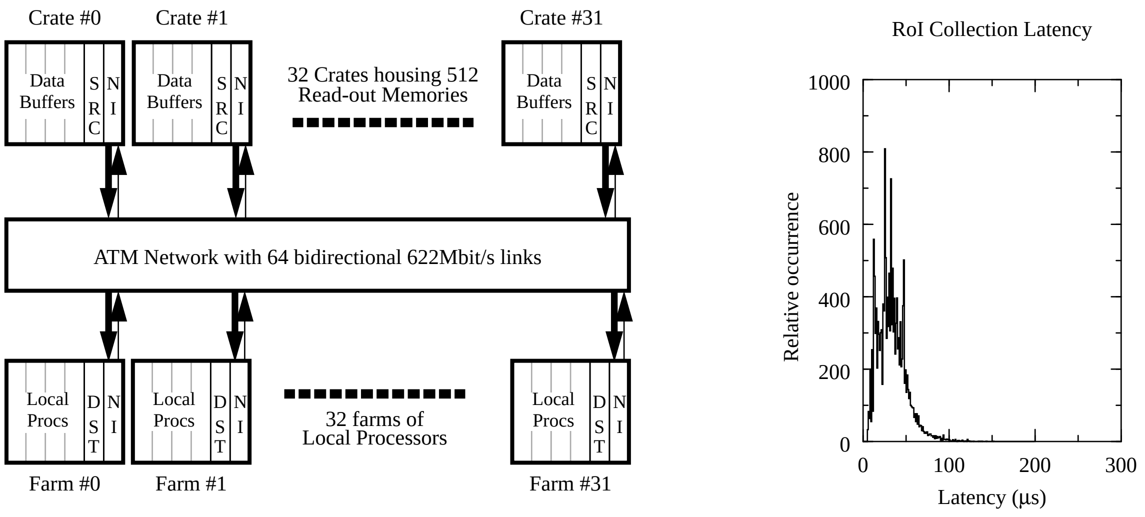
4.5.2 Study of the Atlas Calorimeter system

The Saclay group, within the RD31 collaboration, has initiated a detailed study of the Atlas level 2 and level 3 triggering systems. Oriented towards the calorimeter subdetector part of the Atlas DAQ, it is nevertheless general enough to cover other subdetectors as well.

According to the read-out scheme proposed in [30], for each event accepted by level 1, data are transmitted from the calorimeter (PS, EM, HAC) front-end boards to the Intelligent Read-out Memories. The read-out of the calorimeter is organized in towers of 0.1 x 0.1 in the η, ϕ space. The detector consists of 64 ϕ by 60 η towers. In our model we assume that 512 Intelligent Read-out Memories will be used to store the data during level 2 and level 3 decision latencies. The memories will be housed in 32 crates, 8 crates, for example, mapping the barrel part of the EM calorimeter, each one covering 1.4 x 1.6 in the η, ϕ space. One link per crate can be used to transmit the data from the Regions of Interest (RoI), required for the level 2 decision, into the feature extraction (local) processors via a switching network.

We started with studies of the data volumes and with an evaluation of the bandwidth required by the system. Starting from physics simulation data and, with simple numerical evaluation, we found that, for a trigger rate of 100 kHz from level 1, an aggregate bandwidth of 5 Gbit/s is required for collecting the calorimeter data for level 2 processing. Assuming that the event building traffic for the next selection level will use the same links, a total aggregate data bandwidth of approximately 7 Gbit/s is required. This results in a data throughput of 225 Mbit/s per crate. Assuming that standard ATM links at 622 Mbit/s are used to interconnect the read-out crates with the switching network, the links will be utilized at ~40% (including the overhead due to the ATM protocols), which is a reasonable value.

The simulated architecture is shown in figure 7a). The local processors are grouped in farms. At the output of the switch, the level 2 data are delivered to each farm through a single link. 32 processing farms have been used in our model. In this first modelling study we are only interested in the latency due to the network (RoI collection latency). Therefore the execution time of the Feature extraction (FEX) algorithm, though used in the model, has been chosen to be constant. We plan to introduce more realistic FEX algorithm time distributions at a later stage.



a) simulated architecture

b) RoI collection time distribution

Fig. 7 Initial simulation studies for the Atlas calorimeter L2 subsystem

In our model, a generic ATM Multistage Interconnection Network (MIN), based on switching elements with a size that can be varied, provides a data path between the read-out crates and the farms of local processors (see section 4.1 for more details about switching network).

Two completely independent simulation programs have been developed in concurrent object oriented languages: Modsim [28] and μC++ [15]. The results derived from both programs have been compared in the same conditions and have shown to be in good agreement with each other.

From our initial simulation studies we have found that the average crate occupancy (the probability that, for a given event, a part of at least one RoI will fall in a crate) amounts to 27%. For 100 kHz level 1 trigger rate this means that each crate must be able to collect, format and send level 2 data at a rate of 27 kHz. Up to four sources may contain data for a given RoI. Therefore one, two or four packets have to be collected in the destination to reconstruct a Region of Interest from the calorimeter. The distribution of the time necessary for this operation is shown on the “RoI Collection Latency” histogram in figure 7b). The average RoI collection latency is ~50 µs. The different peaks observed on the time distribution correspond to the different types of RoI’s and their distribution among the crates.

The simulation studies described here constitute a first approach and the status of our research is evolving rapidly. In Section 5.2 we introduce a possible scheme for the level 2 and level 3 event building of ATLAS, which uses the same switching network for data and control flows, and we present our latest simulation results.

4.6 Modelling of the CMS architecture

The feasibility of using packet switching networks for the DAQ architecture of CMS has been studied. For events accepted by level 1, the CMS collaboration currently considers two types of event read-out schemes: full and partial [32]. The full read-out scheme corresponds to sending the full event from the front-end dual port memories to a destination (processing farm) via the switching network. The simulation results of the generic event builders, presented in the section 4.2.3, are applicable to this scheme and will not be repeated here. This section will focus only on the partial read-out simulation architecture. A detailed description of the modelling for both architectures can be found in [33].

4.6.1 Partial read-out architecture (“Virtual level 2")

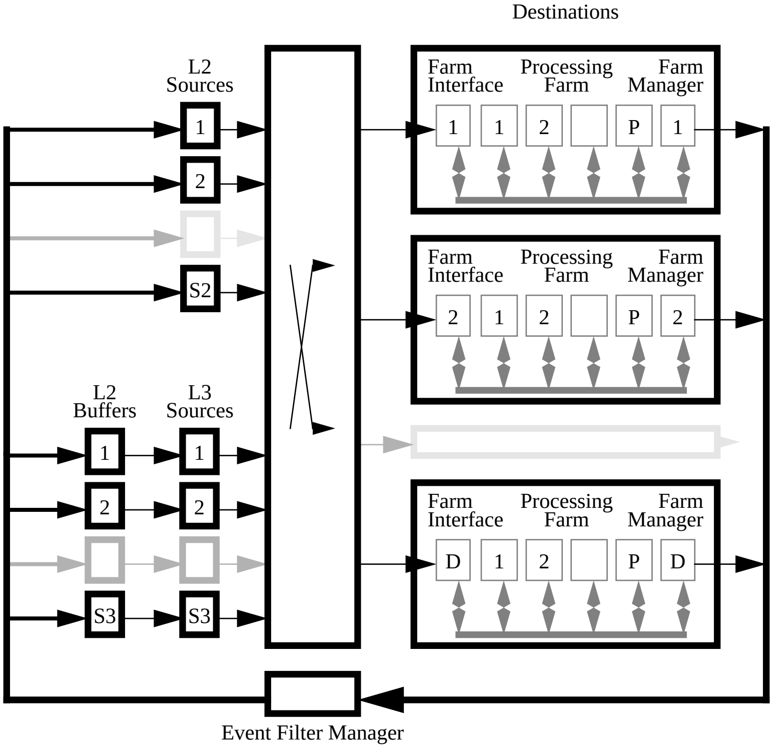
Partial read-out corresponds to transmitting only the information needed by the level 2 trigger for every event. In case of acceptance, the rest of the event data is sent to the destination processor farm. The model of the partial read-out scheme is shown in figure 8.

The 1024 sources are interconnected with the 1024 destination farms via a switching network. Only 256 source modules (level 2 sources) participate in the level 2 decision, sending on average 400 bytes of data to a destination processing farm for every event accepted at level 1. The rest of the event data, which does not participate in the level 2 decision, are stored in the L2 buffers where they wait for completion of level 2 event building and decision. In the case of a negative decision the event data fragments are flushed from the L2 buffers. In the case of a positive decision the event data fragments are transferred to the level 3 sources, which then send them to the same destination farm that made the level 2 decision. The amount of data sent by each level 3 source is 1 kByte on average. Both level 2 and level 3 data fragment sizes follow normal distributions.

The same 1024x1024 network model, which was used for generic event builder studies, was used for the CMS partial read-out simulation. The network was constructed from 4x4 switching elements with 16 kByte of memory in each switching node. The fabric operates with 640 Mbit/s links, giving an available aggregate bandwidth of 640 Gbit/s. The source traffic shaping technique was applied in order to minimize congestion in the fabric. Figure 9 shows the event building latencies for level 2 and level 3 as a function of the trigger rate (a) and as a function of the variance of the event fragment size distribution (b). In the simulations, a level 2 rejection factor of 10 was assumed.

Compared to the full read-out scheme the partial read-out approach substantially reduces (by a factor of 9) the data throughput. At a trigger rate of 100 kHz, only 30% of the available aggregate bandwidth is utilized. Both level 2 and level 3 latencies have approximately a flat distribution as a function of the trigger rate. Therefore the safety factor of the system is high.

The event data generator, used in the simulations, is very simple and assumes uncorrelated front-end sources which generate normally distributed event fragments. As can be seen in Fig 9(b),



S2 - number of L2 Sources

S3 - number of L3 Sources

D - number of destination farms

P - number of processors in a farm

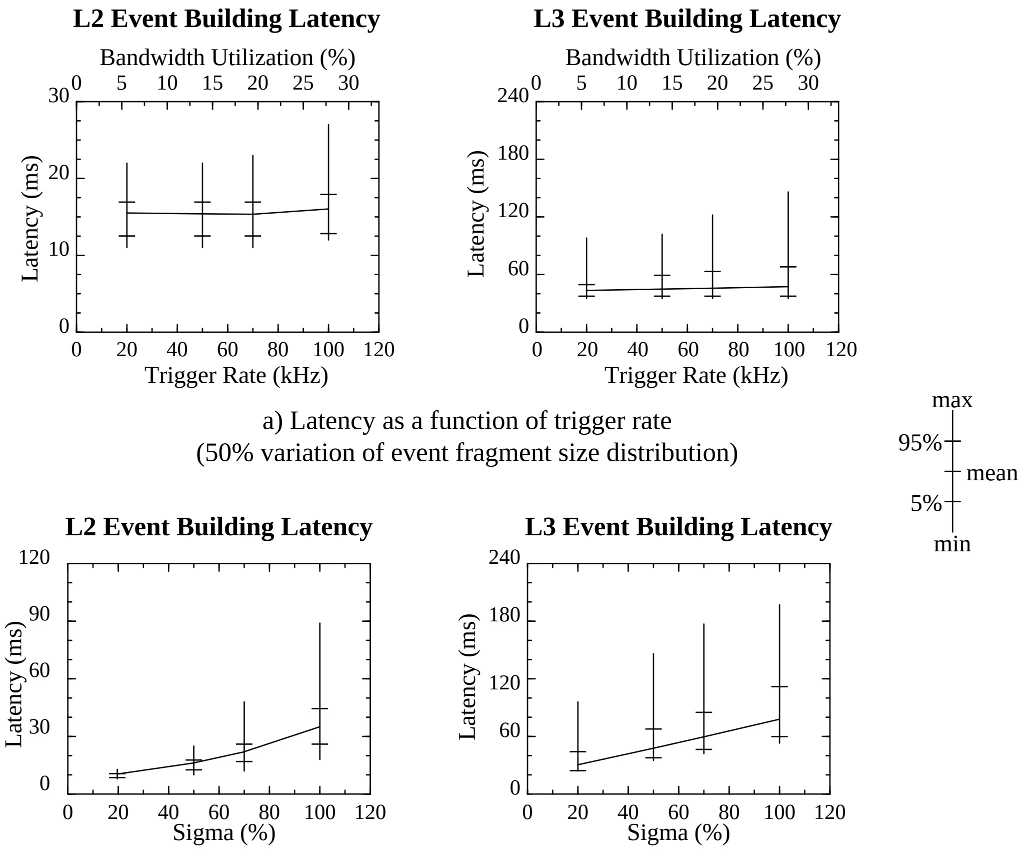
Fig. 8 Simulation model of the CMS partial read-out architecture.

level 2 and level 3 latencies depend not only on the average size of the event data fragments, but also on the statistical distribution of the size: the larger the variation of the event fragment size, the longer it takes to build an event. What is even more important is that the tail distribution of the latencies also becomes longer. In the future, more realistic simulation studies should be based on the input data derived from physics simulations and from the detailed read-out architecture of the detectors. This will allow us to study the effects of the correlations between the sources.

5. EVENT BUILDING PROTOCOLS AND RELATED SOFTWARE DEVELOPMENT

The ATM-based event building process has been defined as a layered structure of protocols. These layers include the standard ATM layers and are complemented by higher-level layers that implement the event building functions. Most of the basic software required to run the full event building process has now been written. It is complemented by various programs which provide monitoring, event building control and data generation. This software has already been used extensively during the development of the VME-ATM board, the tests of interoperability with the switch and, currently, the development of the event builder demonstrator. A detailed presentation of the concepts and the software is given in [34].

We have defined additional layers, on top of the ATM protocol layers, to implement a full function event builder. See ref. [34] for a detailed review. Most of the work has been done in the field of a “push” architecture, whereas some new work is starting now to investigate the possibilities of “pull” architectures, mainly for application to the ATLAS level-2 trigger (see section 5.2).



a) Latency as a function of trigger rate
(50% variation of event fragment size distribution)

b) Latency as a function of event fragment size distribution
(100kHz trigger rate)

Fig. 9 Simulation results of the CMS partial read-out architecture.

5.1 The layered structure of the event builder architecture based on an ATM switching network.

5.1.1 The architecture of protocol layers.

Figure 10 shows the layers of protocols as they are proposed and have been implemented. Layers 1 to 3 are the standard ATM layers [7]. The top “event building” layer is sub-divided into 2 sublayers:

- the *Event fragment sublayer* ensures the independence of the layer(s) above it from the network-specific layers (1 to 3). It allows the handling of event-fragments which are longer than the maximum packet length defined by the underlying technology (in the AAL5/ATM case the maximum packet length is 64 kBytes).
- the *Event sublayer* has the task of linking together the received event-fragments to form an event (because they traverse different paths through the fabric, the event fragments reach the destination out of order). It must also recognize when an event is completely assembled. Several methods have been proposed for this latter task [10] and can be selected according to the particular conditions of operation of an event builder. The event sublayer is only required in the destinations, where it must be able to build several events concurrently.

Of course the event fragment sublayer introduces some overhead, but it is necessary in order to allow event fragments to be larger than the maximum 64 kByte AAL5/ATM packet. It is expected that the ALICE experiment will have event fragments much longer than 64 kByte. It will perhaps also be needed on other experiments when collecting calibration data. It can also optimize memory by allowing us to define the buffers in the memory interface to be smaller than the largest possible event fragment.

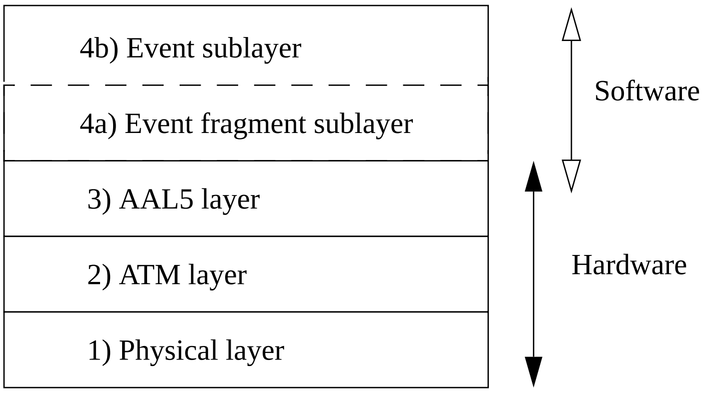


Fig. 10 Protocol Layer structure of the event builder architecture

In the protocol stack described, there is no OSI transport layer. The CRC within an AAL5 packet provides error detection, but there is no mechanism to retransmit an errored packet. Actually most transport protocols suppose that sources and destinations are both able to send and to receive. In the current implementation of the event builder demonstrator system (see section 7.1), sources are emulated using traffic generators which can neither receive nor process data. This implies that no transport layer can be used. However the use of a transport layer has to be evaluated, considering the trade-off between the overhead in the network interfaces and the increase of the traffic caused by retransmitted packets and the consequences on the reliability of the event builder. For limiting the protocol overhead the transport functionality could be included into the event fragment sublayer. The algorithms for avoiding further congestion due to the transport protocol have still to be investigated taking in account the recommendations of the ATM Forum.

The architecture proposed so far is a “push” architecture where the destinations have no means to direct the collection of data. It should be preferred in all cases where it is adequate because it is certainly simpler to implement than a “pull” architecture, in which the destination requests the data from the sources. However, we have just started investigating the “pull” architecture for applications which have a sparse distribution of data in sources (e.g. for systems collecting data from “Regions Of Interest”, as discussed in section 5.2). This might be a subject for further research.

5.1.2 Data format, data structures.

Figure 11 shows the data format for the event fragment sub-layer. At the sending side, an event fragment PDU (Protocol Data Unit) is formed from a payload and a PCU (Protocol Control Unit) with information about event number, destination and optional event building control information (e.g. the event sequence number which may be required for the event building completeness algorithm). It is segmented in AAL5 packets, each one being in turn complemented with a PCU containing the source number, a fragment sequence number and a segment type (continuation of message or last packet indication (LPI)). The PCU’s are used in the destination to check for the completeness of every event fragment (Event Fragment SAR and AAL5 PCU’s) and for the completeness of the event building (event fragment PCU).

The event building algorithms are implemented with linked lists of descriptors to keep track of the various segments received and possibly belonging to different events. The event building is per-

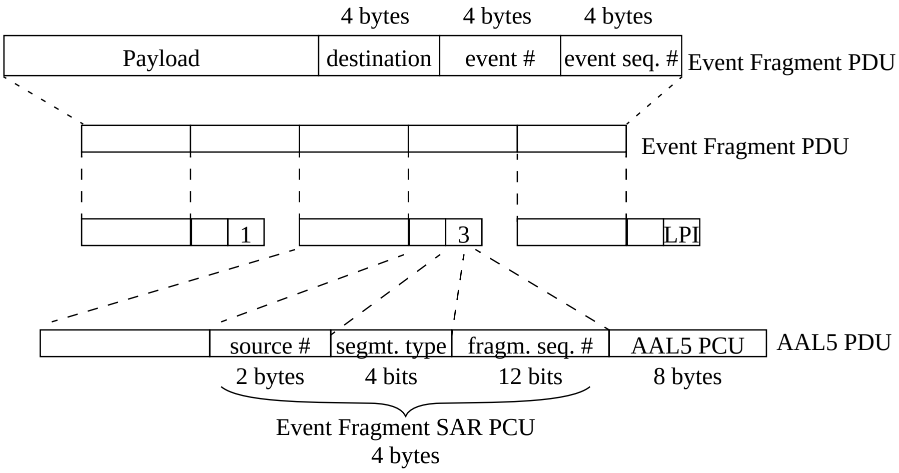


Fig. 11 Event fragment protocol data unit (PDU) and its segmentation into AAL5 packets

formed keeping track of fragments using pointers and, as far as there are enough free buffers in the network interface memory, without moving data into the processor main memory.

5.1.3 Software structure

The software has been designed taking into consideration its portability and reusability. These issues are particularly important because some aspects of the ATM standard as well as many other aspects of the event builder are not completely defined.

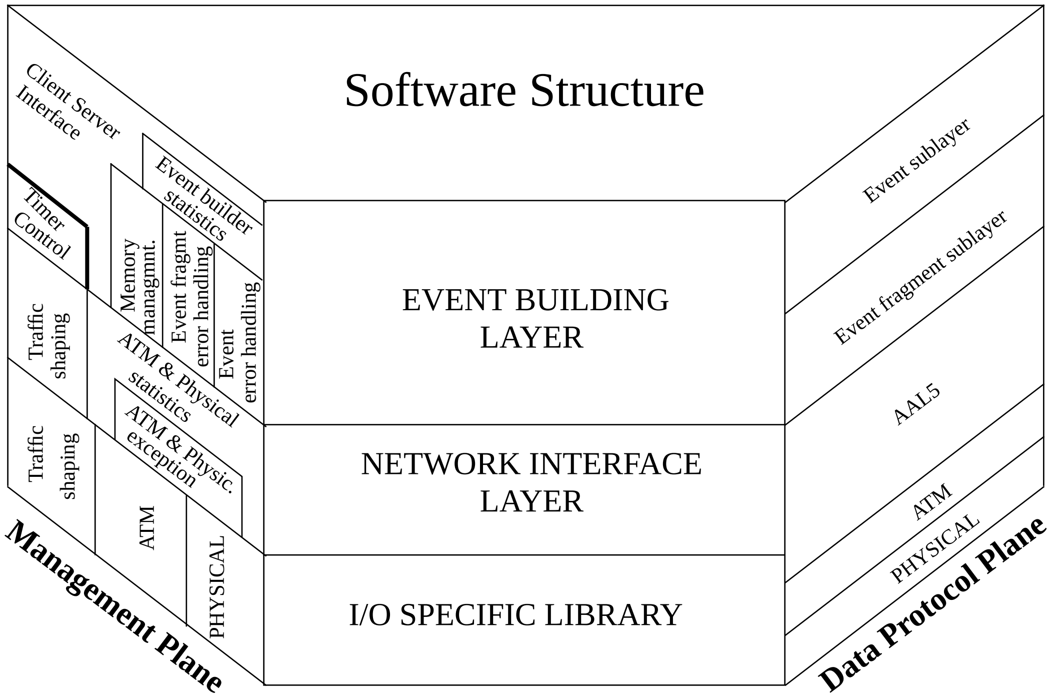


Fig. 12 Software structure

The structure of the software can be divided into three layers and two planes, as shown in fig. 12. The **Event building Layer** implements the event building layer protocol and controls and manages the system in order to satisfy the requests from sources or destinations. The **Network Interface Layer** implements the communication protocol, manages the network interface resources and monitors the network performance. The **I/O Specific Library** provides a set of functions to access the interface.

The **Data protocol** plane implements the event building protocol and the communication protocol; the **Management** plane manages the system resources, handles errors and monitors the system performance. More details on the functions shown in fig. 12 can be found in [34].

In the current implementation the software runs in stand-alone mode on a CES RIO module [35]. It works as a server executing tasks requested from a client controller program running on a CES RAID. The two programs communicate using a message passing mechanism via their FIFOs. Due to limitations in VME data transfer characteristics, no event building data is actually transferred between RIO and RAID.

5.2 Protocol traffic transport via the switch

Until now most of the performance studies of event builders have been focused on the data flow aspects. Recently, the collaboration has started to investigate various scenarios of control flow and data flow; in particular, we are evaluating the merits of “Push” and “Pull” architectures. In any DAQ architecture control information should be exchanged between the various parts of the system, such as the data sources, the destination processors, etc. The control flow can use the same medium (network) as the one used for data transmission. The main advantages of this approach are:

- a unique switching network for all types of traffic (data and control),
- a single network adapter per node,
- standard network protocols, available from industry. Therefore, assuming bidirectional control flows, and using bidirectional links one can take advantage of industry developments, thus greatly simplifying the error detection and recovery issues.

We propose a control scheme for the level 2 and level 3 triggers of ATLAS based on the considerations above. The principles are shown in Figure 13.

We assume that the information on each event accepted by the level 1 (number of RoIs and position in η, ϕ space) will be delivered to the trigger supervisor via a dedicated path. One of the tasks of the supervisor is to allocate resources for processing this event, e.g. assign a processor per RoI and a processor for the global decision. Currently we propose a very simple destination processor assignment scheme, namely that a sequential allocation should be adequate. More sophisticated algorithms are not excluded. Each allocated (local or global) processor receives a notification message (one cell) from the supervisor (message flow (1) in fig.13a)). The message contains the Event ID, the RoI ID, the Global Processor ID, etc. We also envisage the possibility to replace flow 1b by flow 1c.

With this information, the global processor knows from which local processor it has to expect features data, and a Local Processor knows which sources contain data for a particular RoI. The local processor will then send a request message (one cell) to each source concerned (flow 2 in fig. 13b)). In response, the sources send the requested data (3). When all data for a given RoI have been delivered to the local processor, it executes the feature extraction algorithm. The result is then sent (4) to the global processor. When all features of the event have been collected the global processor executes a global algorithm and generates the level 2 decision.

We consider two possibilities for the continuation after the level 2 decision. In one case, the source modules are notified only if the event has been accepted. The level 2 decision “Yes” is sent to the level 3 supervisor (flow 5 in fig. 13c)), which then multicasts it to all sources (flow 6). No immediate action is taken in the sources for the events which didn’t pass the level 2 selection. The oldest event is simply overwritten in the source buffer when a new event is being read from the front-end modules. This scheme is based on the consideration that the event buffer in the source modules has anyway to be designed to be sufficient for the longest possible level 2 decision latency. This scheme is attractive

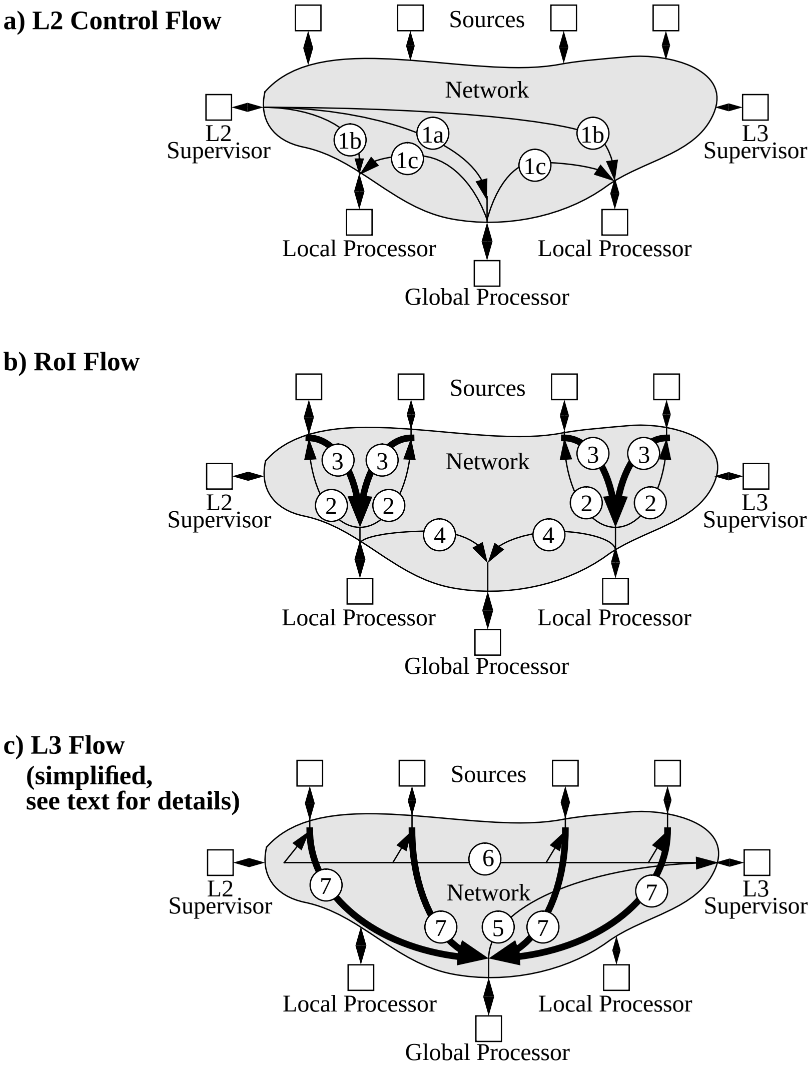


Fig. 13 Possible scheme for the L2 and L3 levels of the ATLAS trigger system.

because it does not generates unnecessary traffic in the network (99% of the level 2 decisions are expected to be “No”), it simplifies the control logic of the data sources and it requires less actions in the system per event.

The other solution consists of sending either of the level 2 decisions, “Yes” or “No”, to the level 3 supervisor (flow 5). In order to minimize the decision broadcast traffic, several (around 10) decisions are packed in one ATM cell which is then multicast to the sources (6). If necessary, information about the accepted events is sent from the level 3 supervisor immediately, and only “No” decisions for consecutive events are packed together.

For the continuation of the work on the level 2 selection, we are studying two possibilities: parallel and sequential. The first one requires all RoI data to be sent to the local processors and examined in parallel for all subdetectors and the results to be combined in the global processor. In the second case, the RoI data for the particular subdetector are sent to the local processors only if they are required by the subsequent steps of level 2 selection algorithm (e.g. TRT data for a RoI is requested only if the decision based on the calorimeter data of the RoI was positive). The sequential flow of the level 2 selection can significantly reduce the aggregate bandwidth requirements for the switching network.

For level 3, it is not decided yet, whether the full event data is needed for the selection algorithm or whether partial event data will be sufficient. In the latter case the level 3 selection step will be followed by the event building. It is also possible that the same processor, which performs the level 2 Global decision will continue to work on the level 3 selection for the same event (as it already possesses a substantial amount of information about this event). The figure 13c) represents a simplified scenario of the level 3 data flow, assuming that full event reconstruction happens in the same global processor. In principle, if necessary, the same steps, which have been performed for the level 2 decision, can be followed for the level 3 selection, namely a dedicated processor can be assigned for the event, which will request the necessary data from the sources, will perform either partial or full reconstruction of the event, will execute the level 3 algorithm, etc.

Figure 14 represents our model which allows us to simulate the behaviour of the level 2 and level 3 selection systems described above. The studies have been performed for the Atlas calorimeter subdetector. The 64x64 switching fabric interconnects a supervisor module, 32 front-end data sources, 16 farms of local processors and 15 farms of global processors. Each farm consists of 8 processing elements. The switching fabric is a multistage interconnection network of AT&T Phoenix-like switching elements and operates at 622 Mbit/s.

In our simulation studies we used the same level 2 input data as described in the section 4.5.2. In addition, the global processors requested the sources to send level 3 data (16 kByte per source, fixed) with an average rate of 1 kHz (level 2 acceptance rate). The level 3 traffic adds approximately 20% load on the Source-Destination data path. Compared to the average level 2 message size of the order of 500 bytes per source, the level 3 messages are long and can significantly delay level 2 packets in the sources, if level 2 and level 3 data are serviced in the same FIFO queue of the sources. Apart from that, if no precautions are taken, the level 3 traffic creates congestion in the switching fabric and therefore, delays not only the level 2 traffic, but also the protocol traffic. There are several methods which permit to reduce the congestion in the switching fabric and minimize level 2 and protocol traffic latencies. First, in the source modules each type of traffic can be serviced with different priorities. Protocol data will be sent prior to level 2 and level 3 data, while the level 2 packets will overtake level 3 packets. Another possibility is to apply the rate division technique to the level 3 traffic. Most existing SAR chips implement both the prioritised servicing and the rate division techniques. Some ATM switching fabrics support routing priorities. Also, different service classes can be used for protocol (CBR) traffic and data (VBR). All these techniques and their combinations are currently under investigation.

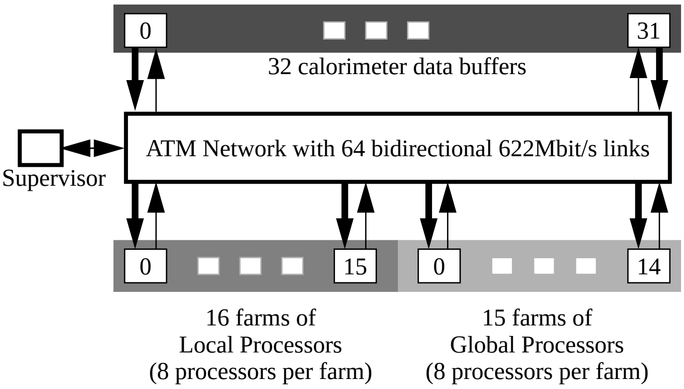


Fig. 14 Simulation model for the Atlas calorimeter system.

The simulation results obtained from the Modsim and μC++ models have been compared and cross-checked against each other. The good agreement of the results is shown in figure 15, which represents the protocol traffic latency tail distributions. On average, 12 μs are necessary for the level 2 supervisor to provide the RoI information to the local processor (the “local processor notification latency” graph). The protocol traffic from the supervisor towards the local processors is disturbed by the level 2 data traffic from the sources towards the local processors. From the "local processor notification latency" tail distribution we observe that, due to the contention inside the fabric, this time can be as big as 40 µs with a probability of the order of 10^{-3}. The time necessary for the RoI data request cell from the local processor to reach the source module is shown on the "Source RoI data request latency" graph. In average it amounts to 7.5 µs. Even though the data request stream travels in opposite direction to the data stream (figure 13.b), inside the switch they share internal links, which can be overloaded. As a result, the data request latency can be as big as 30 µs with a probability of the order of 10^{-3}.

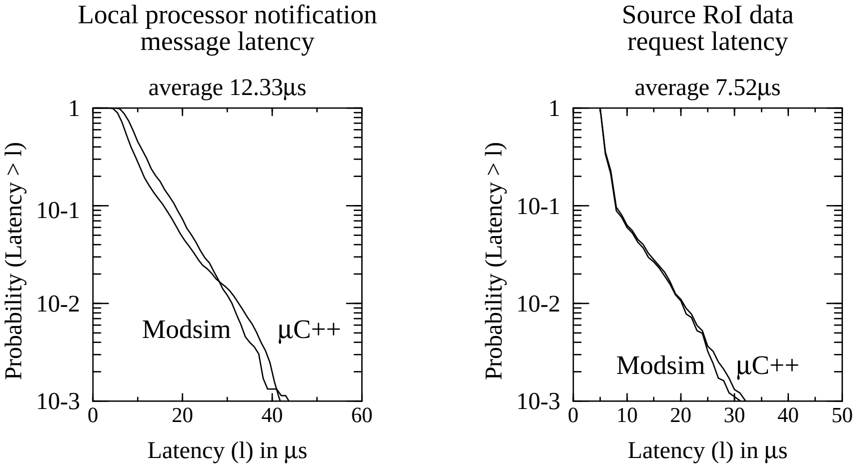


Fig. 15 Tail distributions of the protocol traffic latency.

During the simulation runs, the utilization of the switching fabric's available bandwidth has been monitored. Thus, for example, the level 2 data traffic (RoI) from the 32 sources to the 16 local processor farms require 25% of the source output links (622 Mbit/s) and 50% of the destination input links respectively. As was mentioned above, level 3 data traffic adds another 20% load on the source output links and requires about 45% of the global processor input link bandwidth. The traffic which delivers features from the local processors to the global processor uses 19 Mbit/s of the global input links and increases their load up to 50%. 250 Mbit/s are necessary in order to distribute the notification messages from the supervisor module to the local and global processor farms (40% of 622 Mbit/s rate). The RoI data request traffic requires 13 Mbit/s bandwidth on the input links of the Sources (2% of the 622 Mbit/s rate). In average 25% of the available switching fabric aggregate bandwidth utilization has been observed.

Based on the developed models and their future versions we are going to perform extensive studies of the relative merits and disadvantages of the "Push" and "Pull" data flow-control strategies. We plan to feed our simulations with more realistic input parameters derived from the physics simulations. The simulation code can be used for similar studies for other detector types, like the Muon detector, the TRT, etc.

6. HARDWARE DEVELOPMENT

In order to set up an event builder demonstrator, we are developing a VME-ATM interface, traffic shaping hardware and an ATM data generator to produce flexible traffic patterns.

6.1 ATM SONET physical layer board

An implementation of the ATM physical layer has been realised in the form of a piggy-back daughter card [36] that plugs on top of the ATM SAR part of the ATM-VME adaptor card described below. This interface can be configured to comply either with STS-OC3 SONET [37] or with STM-1 SDH [38], and it transmits and receives over serial optical-fibre at bit-rates of 155 Mbit/s.

The Physical layer interface can be used to form the basis of an ATM data generator that will be integrated in the event builder demonstrator to emulate data sources.

Figure 16 shows the lay-out of the physical interface. It is built around the SUNI chip from PMC Sierra [39] which implements the CCITT standard I.432 specification. The interface with the ATM layer part must include a bi-directional data path and a control path for SUNI set-up and for synchronization signals. The board implements 2 different interfaces to the ATM layer. One complies with the UTOPIA standard [40] and the other is custom defined and was adopted to simplify the design of our ATM board.

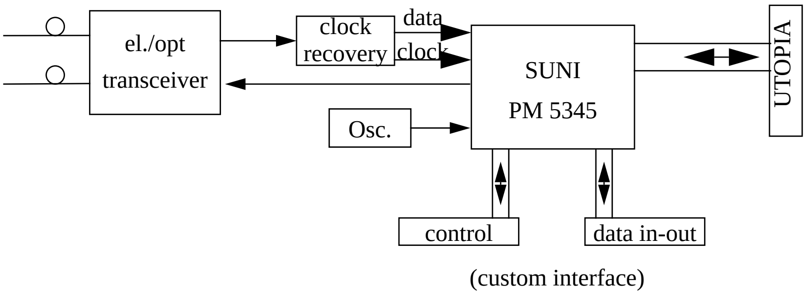


Fig. 16 Lay out of the physical interface

6.2 VME - ATM adapter

The development of a VME-ATM interface has been undertaken to provide the source and destination modules for the event builder demonstrator. This activity helped us to gain experience with ATM technology, and also to check if and how the functionality and performance needed for event building could be implemented using commercially available chip sets designed for building ATM host-interfaces. A custom development was necessary in order to integrate the Randomizer traffic shaping hardware [41] (a function specific to event building).

A prototype has been realised [42] as a daughter board which plugs into a CES RIO module [35]. It has been tested successfully as regards its functionality and interoperability with the Alcatel ATM switch [43] and the HP broadband tester [44]. However, the theoretically achievable data transfer rates have not yet been reached, neither in the ATM interface itself, nor in the data transfers via the VME bus. Nevertheless, the performances obtained are adequate to proceed with the implementation of the event builder and we have launched the production of a small series of a printed circuit board version of the interface for this purpose. We have gained through this development effort a deep knowledge of the technology and a very good understanding of the critical issues in ATM interface design.

6.2.1 Implementation of the VME-ATM adaptor

The CES RIO module is used for the VME interface; it includes a 25 MHz RISC processor which will run the software implementing the higher layers of the protocol stack. The lower layers are implemented in hardware on a daughter board that communicates with the processor via the system bus.The architecture of the ATM adapter hardware is shown in figure 17. During the prototyping phase we actually have three separate hardware plug-in modules. One implements the B-ISDN AAL5 and ATM protocol layers; the second implements the SONET physical layer (and was described in section

6.1), and the third is an optional randomizer module that includes special hardware (see [41] and section 6.2.2) to perform the traffic shaping required for event building over telecommunication switches.

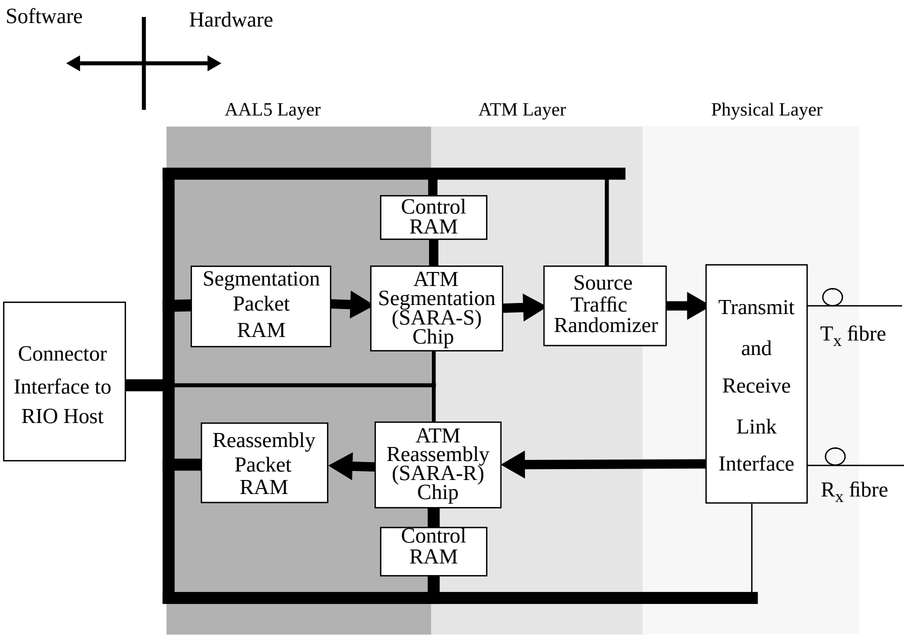

Software
Hardware
AAL5 Layer
ATM Layer
Physical Layer
Control RAM
Segmentation Packet RAM
ATM Segmentation (SARA-S) Chip
Source Traffic Randomizer
Transmit and Receive Link Interface
T_x fibre
Connector Interface to RIO Host
Reassembly Packet RAM
ATM Reassembly (SARA-R) Chip
R_x fibre
Control RAM

Fig. 17 Block diagram of the interface hardware supporting the AAL, ATM and Physical layers of the B-ISDN protocol.

A commercial chip set [45] performs in hardware the segmentation and reassembly of data packets, in the AAL5 format (up to 64 kByte long), into/from ATM cells. These segmentation and reassembly (SARA) chips require two dual-ported memories each. The first one, the packet memory, stores the actual data packet to be transmitted (or that has been received and reassembled). In order to sustain the full 155 Mbit/s rate, this memory is accessed by the SARA via a 32-bit port, and 12 memory accesses are required per ATM cell. The second port is also 32-bit wide and connects to the host's system bus. The port arbitration logic assigns equal priority to both ports. Currently we transfer data between VME bus and the packet memory using programmed I/O. Some improvements to the current design are required in order to be able to support block transfer mode between VME address space and the packet memory.

The second type of memory contains packet descriptors that point to the location of AAL5 packets in the packet memory and specify their length, the virtual connection index (VCI) and its associated traffic metering parameters. The segmentation chip implements sophisticated procedures to segment the packet when multiple VCIs are concurrently active. We measured that, for every ATM cell generated and passed to the physical layer, not less than 23 control memory accesses are required for this management. Each SARA chip can support up to 64k different VCIs, and can simultaneously segment/reassemble 8k packets, which is sufficient to construct very large event builders. The current design uses 512 kByte packet memories and 256 kByte control memories.

The physical layer hardware is included in fig. 17, and has already been described in section 6.1. The interface between the physical layer board and the board with the AAL layer is a custom protocol rather than the standard UTOPIA (which would have been more difficult to implement). FIFOs of 4 cells on both sender and receiver path are provided by the SUNI chip. Their role is to make the transition between the asynchronous ATM and the synchronous physical layers. The receive FIFO can also smooth out some burstiness in the cell rate, but the current size of 4 cells is not sufficient when the effective rate on the AAL layer is as low as it is in our case (see section 6.2.3 for more details).

6.2.2 Traffic Randomizing hardware

Source traffic shaping can be used to control congestion within the switching fabric by regulating the bandwidth assignment to virtual connections, and by modulating the time at which cells are injected into the switch. Figure 18 shows the principle of the randomizer traffic shaping hardware developed for event building applications. Each source module in the event builder must maintain one logical FIFO queue of event data for each destination (in fig. 18 these logical FIFO queues are labelled with unique Virtual Connection identifiers associating them with a specific destination). The SARA segmentation chip services the packet queues in round robin, picking one cell from the head of each packet queue in each round robin cycle. Rate metering is effectively imposed by SARA applying a programmable delay between each service cycle.

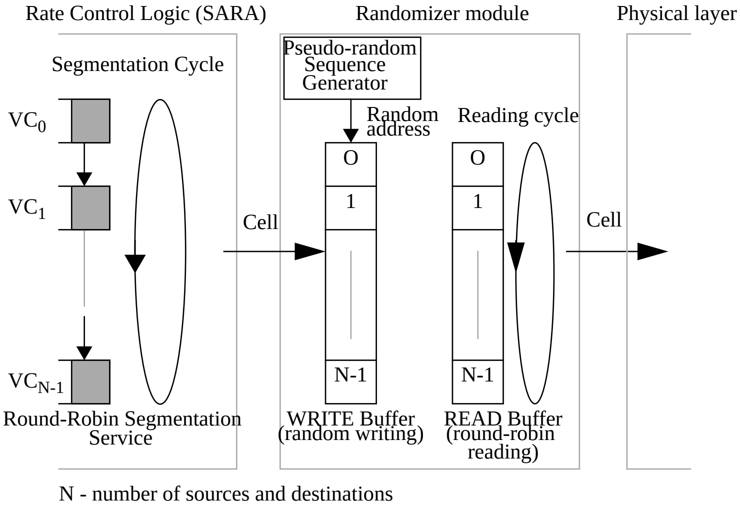

Rate Control Logic (SARA)
Randomizer module
Physical layer
Segmentation Cycle
Pseudo-random Sequence Generator
Random address
Reading cycle
VC$_0$
VC$_1$
VC$_{N-1}$
Cell
Cell
O
1
N-1
O
1
N-1
Round-Robin Segmentation Service
WRITE Buffer (random writing)
READ Buffer (round-robin reading)
N - number of sources and destinations

Fig. 18 The principle of operation of the randomizer traffic shaping hardware.

The randomization of a cell injection time, which breaks the correlation between traffic from different sources and therefore minimizes congestion inside the fabric, is performed by the randomizer module [41]. The randomizer contains two cell buffer memories (a “write” buffer and a “read” buffer). It operates by writing the ATM cells sent out by SARA during a segmentation cycle into pseudo-random locations in the write buffer. During the next segmentation cycle the write and read buffers are switched. The cells from the read buffer are always read out by scanning the memory sequentially, thus effectively adding a random delay to the injection time of cells on a given VC. The algorithm guarantees that cell sequencing within each VC is preserved.

A modification to the original design of the randomizer had to be performed, due to the lower than expected bandwidth achieved by the interface. The new functional protocol between the AAL layer and the randomizer is described in [46].

6.2.3 Tests and Performance measurements

The VME-ATM prototype has been extensively tested to ensure that it worked properly with other ATM standard equipment, namely the Alcatel switch and the HP broadband tester. In addition, the various layers (ATM, AAL and physical) have been tested individually in loop-back mode. The randomizer part is still under development and has not yet been integrated in the global tests.

Currently we achieve transfer rates of 50 Mbit/s between VME bus and the packet memories using programmed I/O. The bit-rate on the optical fibre is 155 Mbit/s, but after subtracting SONET framing protocol overheads the theoretically available bandwidth is 149.7 Mbit/s. In loop-back mode, when packet data are transferred between packet memories (but not to the VME bus), we achieve a sustained effective data transfer rate of 95 Mbit/s. However, when the board sends data through the switch, we have to reduce the rate to 70 Mbit/s on the sender line because the switch output ports deliver the cells in bursts that cause the receive FIFOs in the physical layer card to overflow (i.e when we send data at 95 Mbit/s bursts of up to 8 consecutive cells are delivered, which of course overflow the 4-cell deep FIFOs of the SUNI). Further optimization of the design is required in order to sustain the full bandwidth offered by the 155 Mbit/s bit-rate of the fibre optic transmission standard.

6.2.4 What we have learned

a) Link with VME: real DMA block transfer must be provided by the mother card. Future versions of RIO, implementing PCI, should solve this problem.

b) The AAL, ATM and physical layer chip sets must be chosen in order to have the best match of their individual characteristics. Our choice was not optimized, mainly due to the limited availability of components when we started.

c) An ATM interface must be able, on the receiver side, to accept bursty traffic at full 155 Mbit/s bandwidth, even if its maximum average bandwidth performance is lower.

6.3 ATM Data Generator

A simple ATM data generator has been developed with the aim of providing a low cost source module for event builder demonstrators [36]. It is based on the ATM physical interface card, described in sect. 6.1, to which a memory is attached. It is controlled through a connection to a PC (Fig. 19).

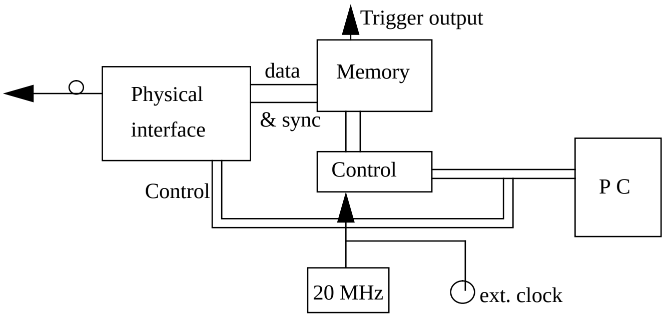


Fig. 19   Data generator lay-out

In its present version the Data Generator can store a sequence of 1230 cells in its 64 kByte memory. Those cells can be delivered continuously at the maximum available bandwidth supported by the 155 Mbit/s SONET standard. One can insert empty cells in order to reduce the ATM rate (e.g. to 77.5 Mbit/s by inserting an empty cell after every ATM cell. By varying the frequency of the external clock, any frequency can be achieved.

The control program running on the PC provides the functionalities to define the cells and the characteristics of the traffic. It includes an ATM cell editor which allows to define the VCI, VPI, PT and CLP fields. It is used to control the SUNI and display the error messages. In addition, a general purpose program (running under UNIX) can generate ATM cell sequences according to global data and traffic characteristics and frame them within an AAL5 structure. A file is used as an intermediate storage medium [47].

A couple of data generator modules as described above have been produced. It is intended to implement several enhancements in future versions, in order to facilitate the use of the Data Generator in the event builder demonstrator. The new features will include an implementation in VME format, the possibility to use an external trigger to launch the emission of the next available AAL5 packet and to increase the size of the memory to store longer data sequences.

7. INTEGRATION OF EVENT BUILDER DEMONSTRATORS

7.1 The Alcatel ATM-based event builder demonstrator

In order to test the largest configuration at minimum cost, we are planning to use the traffic generator (see section 6.3) in the event builder demonstrator. The system (Figure 20, left) consists of a number of ATM traffic generator sources, an ATM switch and one or more RIO/ATM interfaces (see section 6.2).

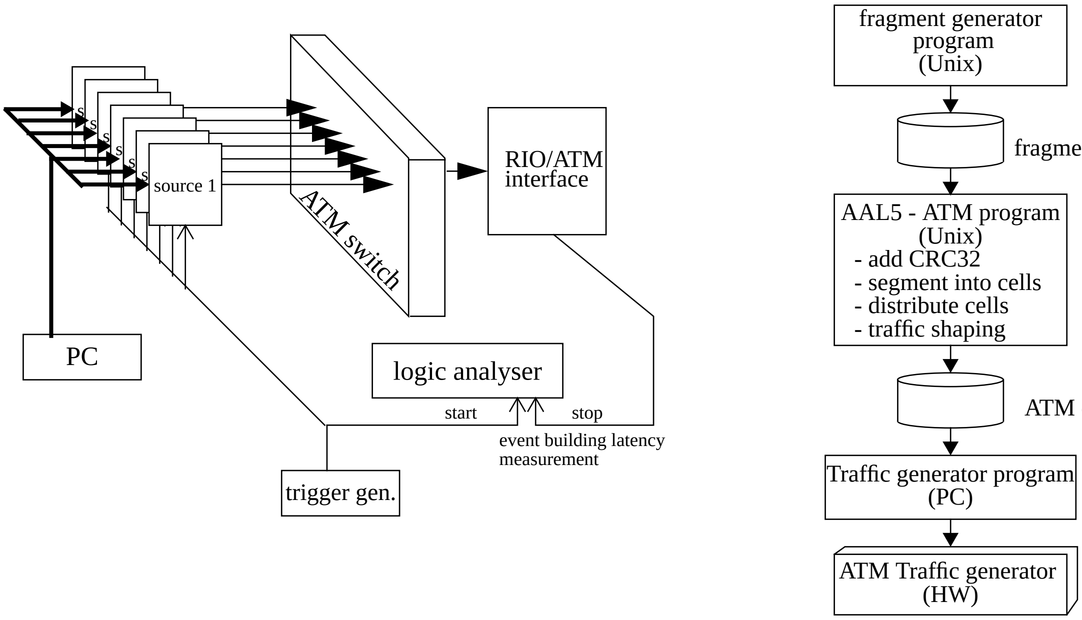


Fig. 20 ATM event builder using a traffic generator

ATM cells containing event fragment data are stored in the traffic generator memory. The traffic generators send cells to an ATM switch, which routes data to dummy destinations and to one or more RIO/ATM interfaces which run the event building software (see section 5.1). The traffic generator controller program running on a PC loads the memories of the generators with data read from a file. This file is generated following a two step procedure (Figure 20, right): a fragment generator program generates AAL5 packets correspondent to a certain sequence of events; an AAL5 to ATM program simulates the work performed by a SAR chip and optionally can simulate the true barrel shifter or randomizer traffic shaping mechanisms.

7.1.1 Performance measurements

At the moment only one ATM interface is available so that hardware and software performance tests have been carried out with an optical loop-back and using the interface both as a source and as a destination. Figure 21(a) shows the software and hardware overheads for sending an Event Fragment PDU (protocol data unit). Most of the software overhead is due to the SAR chip control data structure initialization and network error checking. As the processor of the interface and the SAR chip work in parallel, a new packet can be initialized and submitted while the former packet is being transmitted.

Using the HP Broadband Test System, it has been possible to perform some measurements of the Alcatel switch. In figure 21(b) the cell delay introduced by the Alcatel switch is shown. When the speed gets close to the maximum ATM speed (149.7Mbit/s) the switch starts losing cells and the delay increases.

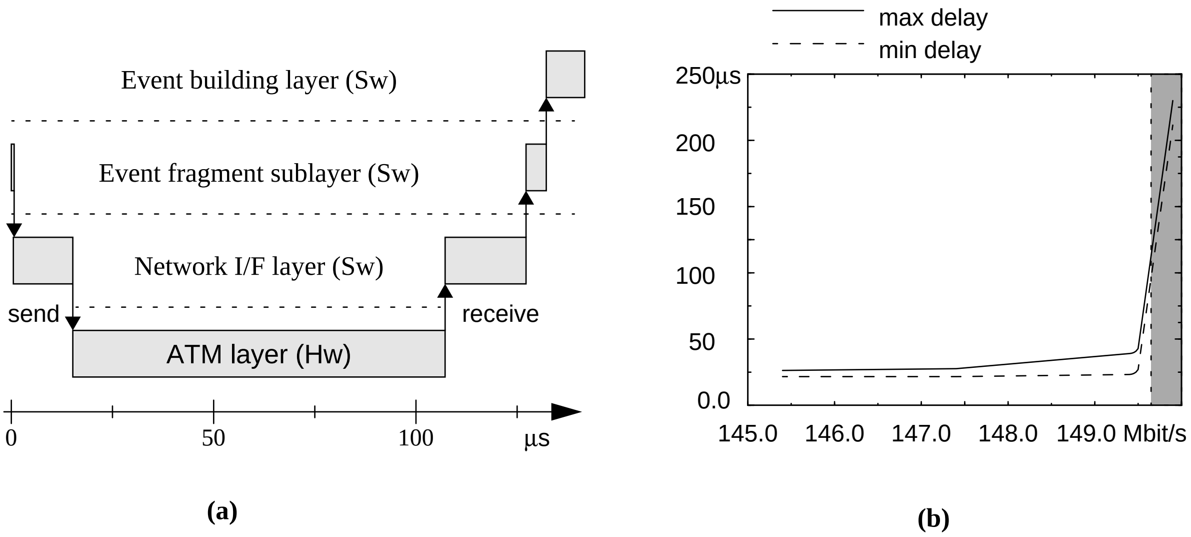


Fig. 21 **(a)** Sending and receiving an Event Fragment PDU of 1 AAL5 packet. Speed=90 Mbit/s; Hardware delay=11µs; packet size=800 bytes. **(b)** Cell delay measured between two Alcatel switch links

7.2 ATLAS AT&T ATM-based Real-Time Demonstrator.

A general purpose demonstrator based on the Phoenix AT&T switch [13] is foreseen at Saclay. The aim of this demonstrator is to validate an ATM-based architecture as a possible level 2 solution for the ATLAS experiment. The foreseen test bench, shown in figure 22, includes:

- Source data generators (developed by RD31 and described in section 6.3),
- An 8-port Phoenix-based ATM switching fabric (purchased from AT&T),
- Destination processors (workstation and/or VME processor board),
- Protocol software, e.g. data flow-control and error recovery mechanisms (currently under development within the RD31 collaboration and described in section 5.1),
- level 2 algorithms software.

This demonstrator will allow us to compare the measurements performed on real hardware against the results predicted by simulation and to refine the models. The test bench can be used to study the architecture of any level 2 subdetector. However, because Saclay is strongly involved in the Atlas calorimetry, in a first step we will consider the electromagnetic and hadronic calorimeter subsystem.

7.2.1 Components of the demonstrator

Source data generator: At present, physics events have been generated by simulation using the ATRECON code [48]. Samples of events accepted by the level 1 selection algorithm are available. Those events will be used to produce traffic patterns for data generators. In order to achieve high sustained event data transmission rates, the event data fragments should be stored in the source modules. Assuming a few kByte event data fragments, the required source buffer size will amount to a few MBytes. The data generator is a VME format board developed within the RD31 collaboration ([36] and sect. 6.3). It can sustain the full 155Mbit/s rate.

Switching fabric: We expect that a 256 port switch running at 155 Mbit/s (or 64 ports at 622 Mbit/s) will be adequate for the Atlas level 2 and level 3 calorimeter subsystem (see section 4.5.2 and 5.2). At present, our system will use an 8x8 AT&T switching fabric with ports running at 155Mbit/s. We can implement a demonstrator which includes four data sources, three destination processors and a controller (figure 22).

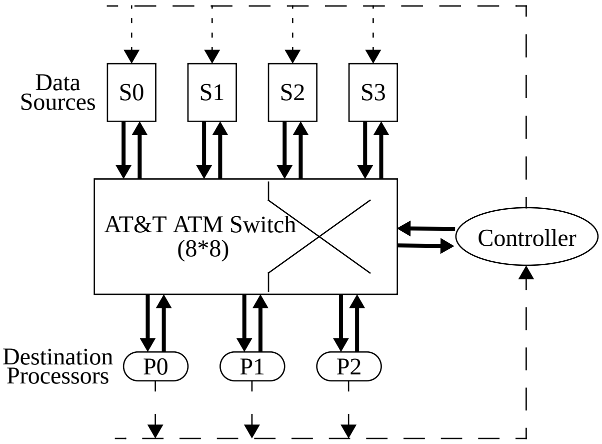


Fig. 22 Schematic view of the Atlas ATM demonstrator based on the AT&T switch.

Destination processors: We already have installed two Sparc-20 workstations equipped with SBus/ATM [49] interfaces and SunOs and Solaris device drivers. We also have a VME/ATM [50] interface card and CES RAID running under LynxOs. We plan to evaluate level 2 local and global selection algorithms on various platforms (e.g. PowerPc, C80). The TI C80 [51] multimedia video processor is currently under evaluation with TI emulation package.

Protocol software: The protocol software will implement both “Push” and “Pull” data flow-control. It will include the event building protocol layers and will address error detection and recovery mechanisms.

7.2.2 Current status

We are evaluating commercial SBus/ATM Adapters in a LAN emulation mode. Two Sparc-20 stations running under SunOs are connected back-to-back via ATM. We measured data transmission performance using different classes of LAN protocols (UDP and TCP based UNIX sockets).

We are currently modifying the device driver software in order to work directly at the level of AAL5 protocol and to achieve higher transmission rates. We evaluate VME/ATM interfaces in real-time environment. The VME/ATM device driver, provided by the manufacturer, has to be ported to LynxOs. We expect the AT&T switching fabric to be delivered within a few months.

8. PLAN OF WORK

The integration of the event builder demonstrators based on the Alcatel switch (at CERN) and the AT&T switch (at Saclay) is planned to continue (as was described in section 7 above). These two demonstrators are complementary in that they investigate fundamentally different ATM switch architectures and congestion control methods. The Alcatel demonstrator is of a generic nature, whereas the AT&T-based demonstrator will be targeted more to the requirements of the Atlas level-2 trigger.

The generic event builder demonstrator at CERN will be used to study performance issues and evaluate various event building protocols and traffic shaping schemes. An important goal is to investigate methods of management and control of the event builder, perhaps based on high-level congestion control techniques developed by the ATM Forum [52], so as to make it a user-friendly system to operate and integrate into the overall DAQ.

The RD31 “Atlas team” will continue the architecture design and simulation studies adapted to the Atlas level-2 trigger system, using the calorimeter sub-system as a realistic model. They will develop their demonstrator in conjunction with the development of an “intelligent”, flexible dual-port source memory that can support the required data flows.

The CMS experiment will be actively investigating various switch-based event building schemes in the near future. Given the importance of this item to the CMS DAQ system, a CMS group is participating in RD31 with the goals of (1) developing architectural concepts and modelling of CMS event builders using more realistic input data flows deduced from physics simulations and from expected detector read-out organization, and (2) investigating Dual-Ported Memory architectures as inputs to ATM-like switches (define, help simulate and design and finally participate in the testing of prototype modules).

9. References

[1] J. Christiansen et al., NEBULAS - A high performance data-driven event building architecture based on an asynchronous self-routing packet-switching network, CERN / DRDC 92-14 and CERN / DRDC 92-47.

[2] J. Christiansen et al., NEBULAS - A high performance data-driven event building architecture based on an asynchronous self-routing packet-switching network, CERN / DRDC 93-55.

[3] ANSI X3T9.3 committee, Fibre Channel draft proposed standard, Rev. 4.2.

[4] The Fibre Channel Association, P.O. Box 9700, Austin, USA, Fibre Channel - Connection to the Future, ISBN 1-878707-19-1 (1994)

[5] J.-P. Dufey, Problem statement for Fibre Channel event builder modelling, RD-31 note 95-02, January 1995.

[6] J-Y. Le Boudec, The asynchronous transfer mode: a tutorial, Computer Networks and ISDN Systems 24 (1992) 279-309.

[7] International Telegraph and Telephone Consultative Committee, ITU, Geneva; recommendations I.150, I.211, I.311, I.321, I.327, I.361, I.362, I.363, I.413, I.432, I.610.

[8] I.Mandjavidze, Modelling and performance evaluation for event builders based on ATM switches, RD-31 internal note 93-06, December 1993.

[9] I. Mandjavidze, A new traffic shaping scheme: the true barrel shifter, RD-31 internal note 94-03, February 1994.

[10] I. Mandjavidze, “Software Protocols for Event Building Switching Networks”, presented at the International Data Acquisition Conference, Fermilab, Oct. 1994.

[11] Fore Systems Inc., Pittsburgh, the ASX family of ATM switches.

[12] Alcatel Data Networks, Alcatel 1100 HSS, private communication.

[13] GlobeView-2000 Broadband System, System description, AT&T network system, Red Bank, New Jersey, 07701

[14] IBM Corp., Nways ATM products.

[15] Buhr, P.A. et al., μC++: Concurrency in the Object-oriented Language C++, Software - Practice and Experience, Vol 22(2) (February 1992), pp 137-172.

[16] A. Marchioro, I. Mandjavidze, Pros and cons of Commercial and Non-Commercial Switching Networks, in Proceedings of the International Data Acquisition Conference, Fermilab, Oct. 1994 (to be published), also available as RD31 note 94-12.

[17] D. Calvet, A MODSIM Model of the AT&T Phoenix switching Fabric, RD-31 Internal Note 94-07, Aug. 1994.

[18] S. Tether, SchedSim: A Tool Kit for Building Scheduled-Event Simulations, private communication.

[19] M. Letheren et al.,”An Asynchronous Data-Driven Event Building Scheme based on ATM Switching Fabrics”, IEEE Trans. on Nuclear Science, vol. 41, No 1, Feb. 1944. Also available as CERN / ECP 93-14.

[20] Nomachi, M., “Event Builder Queue Occupancy”, SDC-93-566, August 1993.

[21] V.P. Kumar et al., Phoenix: A building block for fault tolerant broadband packet switches, Proceedings of the IEEE Global Telecommunications Conference, December 1991, Phoenix, USA

[22] Oechslin, P. et al., ALI: A Versatile Interface Chip for ATM Systems, Proceedings of the IEEE Global Telecommunications Conference’92, Orlando, 6-9 December 1992, pp 1282-1287.

[23] I. Mandjavidze, “Review of ATM, Fibre Channel and Conical Network Simulations”, presented at the International Data Acquisition Conference, Fermilab, Oct. 1994.

[24] H. Rajaei, SIMA, An environment for parallel discrete event simulation. In Proceedings of the 25th Annual Simulation Symposium, Florida, April 1992.

[25] I. Mandjavidze, A data-driven event building scheme based on a self-routing packet-switching Banyan network, RD-31 note 93-07.

[26] A. Marchioro, I. Mandjavidze, A data-driven event building scheme based on a conic self-routing packet-switching Banyan network, RD-31 note 94-06.

[27] J. Gerl and R.M. Lieder, Euroball III, European Gamma-Ray Facility, GSI Darmstadt, 1993.

[28] MODSIM II - The Language for Object-Oriented Programming, CACI Products Company, La Jolla, California, January 1993.

[29] ATLAS SIMDAQ, A. Bogaerts et al., Modelling of the ATLAS data acquisition and trigger system, ATLAS Internal Note, DAQ-NO-18.

[30] M. Costa et al., ATM-based event building, ATLAS Internal Note, DAQ-NO-024, December 1994.

[31] ATLAS Technical Proposal, CERN/LHCC 94-43, LHCC/P2, 15 December 1994.

[32] The Compact Muon Solenoid (CMS), Technical Proposal, CERN/LHCC 94-38, LHCC/P1, 15 December 1994.

[33] I. Mandjavidze, Modelling of an ATM implementation of the CMS Virtual Level 2 Architecture, RD-31 note 95-3, February 1995.

[34] M. Costa, ATM Event Building Software, RD-31 note 94-08, December 1994, revised February 1995.

[35] Creative Electronic Systems SA, Geneva, RIO 8260 and MIO 8261 RISC I/O processors - user’s manual, version 1.1 (March 1993).

[36] C. Paillard, An STS-OC3 SONET/ STM-1 SDH ATM Physical layer implementation and Application to an ATM Data Generator, RD-31 note 95-04 February 1995.

[37] ANSI T1.105-1991, Digital hierarchy - Optical interface rates and formats specifications (SONET).

[38] International Telecommunications Union, Geneva, Switzerland, recommendations G.707, G.708, G.709.

[39] PMC-Sierra Inc., the PMC5345 Saturn user network interface manual (May 1993).

[40] UTOPIA Specification: An ATM PHY data path interface, draft version 1.06 (October 1993), working paper of the ATM Forum, 303 Vintage Park, Foster City CA 94404, USA.

[41] T. Lazraq et al., ATM traffic shaping in event building applications, RD-31 note 94-09

[42] L. Gustafsson et al., A 155 Mbit/s VME to ATM interface with special features for event building applications based on ATM switching fabrics. In Proceedings of the International Data Acquisition Conference, Fermilab, Oct. 1994 (to be published), also available as RD-31 note 94-11.

[43] Henrion, M. 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.

[44] Hewlett Packard, Broadband Series Test System (1994).

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

[46] M. Costa et al., Randomizer Protocol, RD-31 note 95-01, February 1995.

[47] M. Costa, An ATM based Event Building test system using ATM traffic generators, RD-31 note 95-5.

[48] ATLAS software group, ATRECON manual, ATLAS Internal Note SOFT-NO-15 (1994)

[49] S/ATM 4615 Adapter, User's Guide, Interphase corp., June, 1994.

[50] V/ATM 5215 Adapter, User's Guide, Interphase corp., September, 1994

[51] TMS320C80 Multimedia Video Processor, technical brief, TI, 1994

[52] R. Jain, Congestion control and traffic management in ATM networks: recent advances and a survey, draft version January 26, 1995, submitted to Computer Networks and ISDN Systems.