ATLAS DAQ Note 2000-011
10 March 2000

An integrated system for the ATLAS High Level Triggers: Concept, General Conclusions on Architecture Studies, Final Results of Prototyping with ATM

J. Bystricky, D. Calvet, O. Gachelin,
M. Huet, P. Le Dû, I. Mandjavidze
CEA Saclay DAPNIA, 91191 Gif-sur-Yvette Cedex, France

Abstract

This paper recalls the concept of a proposed integrated system for the “ATLAS High Level Triggers”[1] that has been investigated over the last 5 years of our involvement in the ATLAS collaboration. We give the rationale that justifies our proposed model and list a number of architecture principles that could be a basis for the final system. We detail the major results of implementation studies made during the “LVL2 demonstrator program” and the “LVL2 Pilot Project”. We present the final results of our investigations on ATM networking technology.

I. Introduction

This paper is organized as follows. Section II describes the concept of the proposed scheme for ATLAS High Level Triggers (HLT) and give some elements of its justification. Section III describes our proposal for some aspects of the HLT architecture and different implementation options. Section IV introduces the various ATM testbeds. The operation and performance of individual testbed components, Read-Out Buffers, the RoI Builder/ supervisors and the processor nodes are detailed in sections V, VI and VII respectively. Section VIII describes the operation of complete testbeds. Section IX gives the final conclusions of investigations on ATM technology.

II. Toward an Integrated system for the ATLAS HLT

In this section, we first recall the principles of the scheme for the LVL2 and LVL3 triggers described in the ATLAS Technical Proposal published in 1994 [1]. We then introduce our alternative view and the main elements of its justification that come from some of the knowledge acquired by the Trigger/DAQ community over the last few years [2].

A. Principles of the ‘94 ATLAS Trigger/DAQ Model

A detailed view of the overall Trigger/DAQ architecture proposed in 1994 for ATLAS is given in [1]. The system consists of two physically and logically independent parts: the LVL3/DAQ and the LVL2 trigger which is not on the main dataflow. The LVL2 trigger aims to achieve a rejection factor of 100 using the so-called “local/global” scheme. For each event, data from ~5 RoIs are pushed via LVL2 links toward “Feature Extractors” (data-driven hardware or farm of processors) working in parallel. Extracted “features” are sent to a processor within a farm of “global processors” that combines them and issues the LVL2 trigger decision. These are distributed to all “DAQ crates”. Rejected events are discarded. For accepted events, the full event data is sent via LVL3 links to the event builder that assembles full events. These are passed to one of the processors within the Event Filter (EF) farm that achieves an event rate reduction of 10 using complicated algorithms similar to those used off-line.

B. Evolutions (accepted, debated or just proposed)

B.1. LVL2 trigger

In the ‘94 scheme for the LVL2 trigger, shortening decision latency to save on event buffer memories was a major goal, justifying the use of a “push” dataflow, the introduction of parallel processing and the motivation for low latency networks (such as SCI). Hardware based “feature extraction” was considered as well as DSPs or general purpose processors (excluding the use of a standard operating system). Segmenting the systems between LVL2 and LVL3 (networks and processors) as well as segmenting networks inside LVL2 (local network per detector, global network, network for RoI distribution...) was envisaged given the prospects of general purpose processors and networking products for the scheduled date of construction of the system. The ‘94 architecture was mainly defined for high luminosity RoI guided operation, and the way to implement a B-physics trigger compatible with that scheme was being studied.

[1] formerly ATLAS Level 2 and Level 3 Triggers.

A first evolution comes from physics studies indicating that primary and secondary RoIs can be distinguished, and that rejection is possible after processing only primary RoI(s) [2]. Because there are only one or two primary RoI(s) per event, parallel processing at the RoI level cannot bring much advantage. In addition, rejection is also possible after processing RoIs in only one detector, therefore parallel processing at the sub-detector level consumes more network and computing resources than a sequential scheme. A possible sequential scheme for LVL2 was introduced in [3].

Since 94, technology has evolved rapidly and several important factors are now apparent:

- Memory at the Read-Out Buffer (ROB) level can be several tens of MByte so that cutting on latency is not any more a major design constraint.
- General purpose processors running an operating system (e.g. PCs) are now sufficiently powerful to be considered as the main computing engine for the LVL2 trigger. PCs provide a flexible and cost effective platform for data processing and their prospects of evolution are excellent.
- Large switching fabrics with a few 100 ports are available now, and link speed is moving from 100 Mbit/s range to 1 Gbit/s.

These factors are also responsible for the evolution of the LVL2 trigger. The use of parallel processing for RoIs is being abandoned because of its relative complexity and because the latency issue is not the most stringent constraint. The “push” data flow for LVL2 was not pursued because it does not allow for sequential data collection and processing in an easy way. Although early prototypes of the LVL2 trigger used different physical networks and processors for local feature extraction and global processing [4], merging these networks in the same physical network was found more practical and became a realistic option given the evolution of networking products. Similarly, using a separate network for the distribution of RoIs to all ROBs brings significant extra complication to the system and offers no real advantage. Hence the “pull” protocol described in [5] has been adopted for data collection at LVL2. All processors in LVL2 play the same role; the control of the processing for a particular event is entirely given to a single processor.

Another factor of evolution is the understanding of the B-physics trigger and the fact that additional algorithms not originally foreseen at the second level trigger are now envisaged at LVL2 (e.g. inner detector Full Scan, b-jet tagging). The current scheme envisaged by the community for B-physics is based on the analysis of the complete TRT data to search for low p_T tracks after an initial LVL1 RoI guided step. The track candidates identified at LVL2 define new RoIs that are analyzed using the silicon and pixel detector data. In average, ~2 particles candidates are identified at this stage. By the analysis of the corresponding data in the calorimeters, an event rejection of ~10 is expected for these events. This scheme for B-physics has several major impacts on the architecture:

- Sequential processing is needed. This scheme does not fit easily in the original ‘94 local/global scheme but can be mapped on a revised version where local and global networks are merged; and local and global processing are combined in the same physical processor. Sequential data transfer using the “pull” mechanism now considered for RoI data collection is adequate. The same architecture can, in principle, be used for low and for high luminosity running.
- The subset of event data defined by RoIs at LVL1 is not sufficient for LVL2. This system needs also to gather data from one or even several complete detectors (e.g. the TRT, or the complete inner detector). The LVL2 system can define new RoIs (not identified at LVL1) and need a mechanism to obtain the relevant data from the ROBs. A physical copy (triggered by LVL1 RoI information) of data corresponding to LVL1 RoIs from the main memory of the ROBs into a secondary buffer for use by the LVL2 trigger is not any more sufficient to preserve the integrity of the main DAQ/EF dataflow chain because the ROBs should support asynchronous data request coming from LVL2. We think that the proposal to copy data of LVL1 RoIs into a secondary buffer brings an unnecessary complication to the system without preserving what is claimed to be its advantage.
- Running an inner detector scan at LVL2 requires to transfer data from all the corresponding ROBs toward one of the processors affected to LVL2. The traffic pattern generated is very similar to that of event building. It is foreseen that this traffic will be handled in the same network that transports RoI data because using a custom made network for that purpose is an option that has now been abandoned. Although the concept of merging different types of traffic on the same network (RoI data collection and event building like) was rather controversial when introduced in [5], this principle is progressively becoming a requirement for a B-physics trigger at LVL2. For an inner detector scan, ~100 kB of data per event have to be transferred at a rate that increased from ~4 kHz to ~10 kHz (today’s estimate). This leads to the necessity of handling 1 GByte/s of additional event building type of traffic in the LVL2 network. This requirement is identical to that of the event builder following ‘94 estimates (1 MByte events to build at 1 kHz).
- the TRT scan is a compute intensive algorithm. Recent estimates show that ~60 and ~600 processors would be required for LVL2 without B-physics and with B-physics respectively (low-luminosity running; 40 kHz LVL1 rate) [6]. For, high luminosity running, ~60 processors would be required. FPGA accelerators could bring a

significant speed-up when integrated in each processor running LVL2. A proof of principle of using a FPGA accelerator in the context of a testbed has been shown, and quantitative measurements that could demonstrate the real benefits of the approach are starting to appear [7].

B.2. Event Builder/Event Filter

Over the last six years, the knowledge of rates and data volumes has evolved. The size of full events in ATLAS is now estimated to ~2 MByte [6]. Current LVL2 rejection is estimated to ~20, leading to a 2 kHz output rate after LVL2. Compared to initial figures (1 MByte events at 1 kHz), a four fold increase of the event builder throughput is needed. Doubling the rejection factor of the Event Filter algorithm and halving its average execution time per event are also required if the available CPU power and rate of data recording are unchanged.

As previously explained, the B-physics trigger puts a major additional load on the LVL2 system. The algorithm split between the LVL2 trigger and the event filter is not yet fixed, and the possibility to run part of B-physics trigger algorithms either at LVL2 or in the Event Filter has to be considered. It is therefore possible that the TRT Scan and subsequent steps will be performed by the Event Filter. The output rate of LVL2 would become ~10 kHz. If this hypothesis is correct, it has several major impacts on the architecture of the system:

- If the dataflow of the ‘94 model is kept, the full event data will be pushed to the event builder at the LVL2 accept rate. The requirements for the event builder becomes ~20 GByte/s. This is a 20-fold increase of the requirements compared to the ‘94 estimate. The impact on strategy, technology choices, cost, etc., have to be evaluated.
- If the dataflow is modified so that data transfers for the Event Filter can proceed in several steps, network requirements for the event builder could be significantly reduced. We think that the possibility of sequential data transfers for the EF should not be ruled out until more studies are made. The “pull” protocol proposed for data collection at LVL2 supports all types and combinations of partial, phased, and full event building. Because the “push” scheme does not offer the same level of flexibility, we propose the use of the “pull” scheme for both LVL2 and EF.
- Compared to initial figures (input rate of 1 kHz, rejection of 10), the event rejection done by the Event Filter has to be increased by an order of magnitude, and simultaneously the average processing time per event has to be divided by 10. Although most of the reduction in the average execution time can be expected to come from the additional rejection that will be achieved, we think that selection strategies and algorithm choices may need to be adapted to take into account these considerations.

B.3. System level

Because there are a lot of uncertainties in rates, data volumes, algorithm choices, and selection strategy, flexibility at the global system level is becoming a key point. The frontier between LVL2 and EF can evolve (e.g. all rejection made by LVL2 or part of the rejection also done by EF). The network capacity and processing power needed for LVL2 and EF systems are tightly coupled: any rejection that is not performed at LVL2 has to be done in the EF. Depending on the level of flexibility that is aimed, the possibility to move computing power and network bandwidth between the two systems could be needed to meet the requirements with available resources. If moving part of the B-physics trigger between LVL2 and EF is envisaged, we think that the capability of moving resources is needed (following today’s hypothesis on rates, algorithm execution times, etc.).

III. A Proposal for the Architecture of ATLAS HLT

A. Some Proposed Principles

This section lists a number of principles that we consider important for ATLAS HLT.

- Event selection is sequential. It can be split between LVL2 and EF (flexible boundary). Network bandwidth and computing resources can be exchanged between LVL2 and EF.
- Every ROB is accessible by every processor. A ROB is an event data server: processors can request data from ROBs as many time as needed. This supports a scheme where all data for the EF is obtained in one step and leaves the possibility of having several steps.
- Data collection from the ROBs is initiated by the processors using a request/response protocol (so-called “pull”). The same data collection mechanism is used for LVL2 and EF.
- Collecting partial events for EF is possible. Sequential processing in EF with phased event data collection is possible. This does not exclude that data collection for EF is made in only one step, but leaves more possibilities opened.

B. Dataflow

The principle of the dataflow is shown in Figure 1.

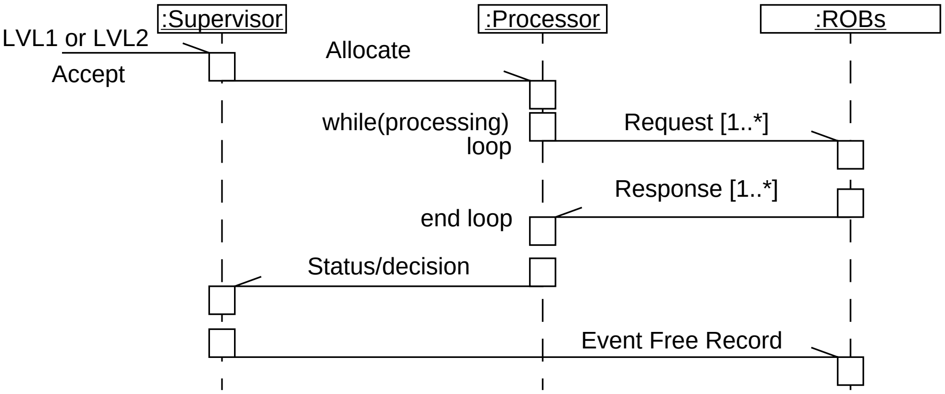


Figure 1. Proposed dataflow for data collection at LVL2 and EF.

The initial guidance for the selection of interesting events will be provided by the Regions of Interest (RoIs) identified at the LVL1 trigger. The selection algorithm consists of a series of steps. A given step is executed only after validation by the previous steps. Each event accepted by the LVL1 trigger is assigned to a processor by a supervisor. The processor “pulls” the information it needs from the sources containing detector data, processes the event fragments received, then decides to fetch more data if needed to make the final decision to accept or reject the event. Accepted events are forwarded to the event filter stage for further selection and analysis while rejected events are discarded. The event filter process may or may not be executed in the same processor that performed LVL2 selection.

C. Implementation

A simplified logical view of the proposed architecture is presented in Figure 2. The main components are:

- the RoI Builder that provides the interface between LVL1 and higher level triggers,
- the supervisors whose main task is to distribute the events accepted by LVL1 to the processors in charge of the selection; these also distribute events accepted by LVL2 if event selection and event filtering are performed in different processors,
- the processors running the selection and/or event filter algorithms. Some of them can include an optional hardware based co-processor to speed-up the execution of some compute-intensive algorithms. Each processor node can be a mono- or a multi-CPU machine, a small cluster of computers or even a remote computing fabric (these two latter implementations would probably require that, for the events assigned to these processors, all the relevant event data is gathered in only one step).
- the Read-Out Buffers (ROBs) grouped into data sources that respond to requests for information from the processors,
- the monitor, configuration and run-control system,
- a common communication network to transfer allocation and protocol messages as well as event data.

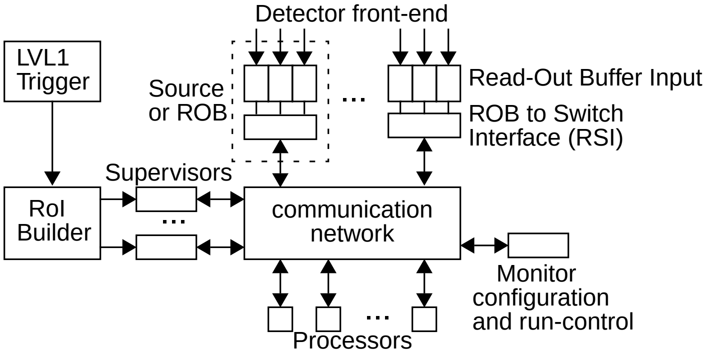


Figure 2: ATLAS LVL2 trigger/EF proposed architecture.

This architecture covers LVL2 and event builder/EF aspects as described in the original proposal [8]. Issues related to LVL2 have been extensively studied in the LVL2 Pilot Project, and our investigations also covered event building. In particular we note that the network used to transfer data to the second level trigger processors can also be used for event building. We propose to extend to EF aspects the studies on this architecture.

We identify the following options for implementing the proposed architecture.

C.1. Common network, split processor farms option

- A single network for data collection common to LVL2 and EF is used to exchange data and protocol messages between the supervisors, ROBs and processors. A secondary network is used for configuration database access, run control and monitoring...
- Processors are split into two farms: the LVL2 farm performs event selection; the EF farm may perform additional selection before event analysis.
- Transferring computing power and network bandwidth between LVL2 and EF involves re-configuration at the software level only, without hardware changes. Using the same type of processors for both farms would further simplify exchange of computing resources.
- The supervisor is common to LVL2 and EF.
- The EF Farm gets results from the LVL2 farm via the common network.

This model aims to achieve a high level of flexibility while keeping a separation of the processing tasks for LVL2 and EF. The resulting system is uniform at many levels of hardware and software, only the application executed on the various processors differ. We think that this option is a good trade-off between a strict separation of LVL2 and EF into two independent sub-systems and an integrated LVL2/EF system.

This option has been investigated on various testbeds based on ATM [8], [9], [10].

C.2. Common network and common processor farm option

- A single farm of processors is used to perform LVL2 and EF. After accepting an event by running LVL2 algorithms, the same machine executes the EF task. LVL2 and EF tasks may execute in the same or separate processes on one or several CPU's (e.g. SMP's). Network and processor resources sharing between LVL2 and EF is transparent.
- Sending LVL2 results to EF involves only local communication at the processor level.

This model aims to achieve a complete convergence of the LVL2 and EF system at the level of hardware, software and application. It can be implemented on the same hardware as the previous option by software re-configuration. We think that this scheme has the highest potential in terms of flexibility, performance, ease of development and maintenance, as well as future evolution.

This option has been investigated on various testbeds based on ATM [9], [10].

C.3. Split networks, split processor farms option

- Two networks are used for data collection: one to provide data to the LVL2 processor farm, another one for the EF farm. This option would be justified if different networking protocols and/or technologies have to be used for data collection at LVL2 and EF. Exchanging computing resources between LVL2 and EF requires moving hardware. Transferring network resources between the two systems is difficult if not impossible.
- Separate supervisors are used for LVL2 and EF.
- A mechanism has to be defined to provide the EF with results of LVL2. An additional component may be required.

This scheme follows the concept described in the '94 ATLAS Technical Proposal. It cannot be implemented on the hardware configuration suited to the two previous options. We think that this scheme is certainly viable from the technical point of view, but is less attractive than other options in terms of efficiency and flexibility. It leads to a duplication of many hardware and software components, roughly doubles the number of experts and developers needed with implications on global cost, ease of maintenance and possibilities of evolution. Unless a physical separation between LVL2 and EF appears to be mandatory for technical reasons, we think that this option does not compare favorably with the others.

So far, no integrated prototype based on this architecture design have been reported.

IV. Prototype Studies

We have conducted several prototype studies of the common network and split/common processor options, focused on LVL2 aspects and covering aspects related to event building. Because of the importance of addressing all issues relevant to the integration of the LVL2 trigger and DAQ / Event Filter prior to the definition of the final architecture, we have retained the flexible design and functionality of the testbed reported in [8] and, moreover, have extended its capabilities and integrated a number of new components.

A. Modes of operation

The ATM Testbed Software that we designed for the LVL2 Demonstrator Program has been thought for flexibility, performance, modularity and portability. With the same hardware configuration and the same software framework, it demonstrates the operation of a testbed in the following modes of operation:

- A LVL2 like system with a common network for data collection and protocol messages, and a farm of processors emulating a sequential selection process.
- A stand-alone event builder system where full events are assembled.
- A LVL2-EF like system with a common network for data collection for LVL2 and event building; a common supervisor group to allocate events accepted by LVL1 and LVL2; a logical split of the processors into a farm executing a LVL2 like sequential selection, and a farm assembling full events after LVL2 accept and emulating some processing sequence.
- A LVL2-EF like system with a common network for data collection for LVL2 and event building; a common supervisor group to allocate events accepted by LVL1; a single processor farm executing a LVL2-like sequential selection where the last step is the gathering of full event data followed by some processing.

In addition to the ATM Testbed Software, our latest testbed also support the execution of the Reference Software [11] (which covers only LVL2 aspects so far).

B. Description of the latest ATM testbed

Starting from the modest 8-node system assembled during the LVL2 Demonstrator Program, the testbed has grown over the last 4 years and all equipment is now installed at CERN. Putting in common the resources of several of the groups involved in the LVL2 Pilot Project, a system composed of up to 38 commodity PCs, 10 VME and 1 CompactPCI single board computers is being operated. The PCs are a mix of 4 generations of Pentiums with clock speeds from 200 MHz to 450 MHz (mono and dual CPU's). Almost all PCs are dual boot WindowsNT / Linux. The single board computers are PowerPC based (100 MHz, 200 MHz, 300 MHz) and run LynxOS. These different nodes are connected to an ATM switch equipped with 48 ports at 155 Mbit/s (three quarters of the total port capacity of that model of switch). Any machine can act as a processor, data source or supervisor emulator. A typical testbed configuration is a system with 22 sources, 22 destinations, 3 supervisors and 1 monitor.

We implemented an Ethernet based start-up program to start all nodes from a single workstation. For simplicity, instead of a configuration database, we use a shared parameter file that is accessed by each node at start-up. In our implementation, all communications for the run-control and monitoring are handled via ATM once the system has been started. The Reference Software uses a more conservative approach based on a low speed Ethernet path separated from the main data path. We use a simple ASCII terminal based user interface to monitor the operation of the system from a central point. A few basic commands allow to start, suspend, resume or terminate the operation of individual nodes or group of nodes, modify some run parameters dynamically, clear, collect then save online statistics and histograms, etc.

The ATM Testbed Software has been successfully tested in heterogeneous systems composed of LynxOS machines, WindowsNT and Linux PCs, Digital Unix workstations and Symmetric Multi-Processor servers. The system can run on legacy LANs using the standard UDP/IP stack (e.g. for development purposes), and on ATM using the socket interface or optimized true zero-copy device drivers and libraries [12]. The Reference Software runs on all machines excepts the LynxOS platforms and relies on the same optimized ATM libraries and drivers that are used by the Testbed Software.

V. The Read-out Buffers Input and Sources

After a LVL1 trigger accept, all event data are digitized and sent from each of the ~1600 detector front-end cards to Read-Out Buffers via high speed (~1 Gbit/s) point-to-point links. The ROB input card (ROBin) buffers data during the event selection process; only a small fraction of the total event data (few percent) is used by the selection algorithms. In order to reduce the number of connections to the communication network and better match the output bandwidth of ROBins to that of network links, ROBins are grouped in so-called "ROB complexes" or simply ROBs or sources.

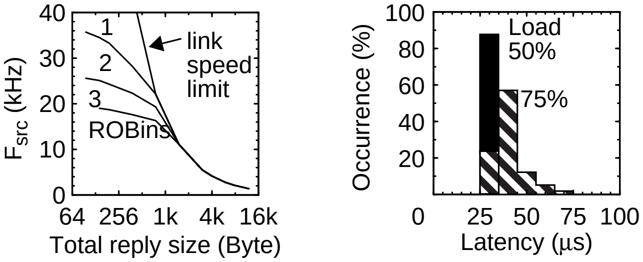
A. Description of the hardware

Each source module comprises a variable number of ROBins (typically 1-8) connected via a backplane bus to a RoB to Switch Interface (RSI) attached to the communication network. We developed a prototype of the ROBin on a PCI Mezzanine Card (PMC) [13]. This form factor gives a lot of flexibility for plugging ROBins in VME single board computers equipped with PMC sites, or inside desktop PCs or CompactPCI crates using the appropriate passive adaptor card. The present version of the ROBin includes a PCI bridge, a 33 MHz Intel I960 processor, its program memory and some glue logic. A more elaborate version of the ROBin that includes a 100 MHz Intel I960 processor, 8 MB of event data memory, a high speed parallel input port and a global control logic has been assembled and is being tested. The ROB to Switch Interface (RSI) is either a standard PC, VME or CompactPCI single board computer equipped with an ATM interface. With these three types of platforms, we have assembled and tested sources comprising 1, 2 and 3 ROBins.

B. Operation and performance

The RSI is in charge of servicing requests coming via the ATM network and distributing to the RoBs the trigger decisions received from the supervisors. When a request for event data is processed by the RSI, it posts a request to each of the RoBs concerned. The memory of the RSI and that of each ROBin is made visible on the PCI bus. The DMA engine of the ATM card uses its chaining capability to send in a single true-zero copy operation event data replies composed of a message header (stored in the RSI memory) and detector data (stored in each of the ROBIN concerned). This scheme minimizes data movement between the ROBins and the RSI, leading to high performance. Alternatively, it is possible that the ROBins will first copy their data in the RSI memory prior to sending them over the network. Though less efficient, this scheme allows the possibility of performing some data manipulation at the RSI level while the zero-copy scheme permits only preprocessing to be done by the ROBins. The trade-off between these two options is being investigated and no choice has been made yet. Both modes of operation are supported in our implementation and can be used simultaneously.

We present in Figure 3a the maximum rate of data requests, F_{src}, that can be serviced by a source versus the total data fragment size of the response message (using the zero-copy scheme).



a) F_{src} vs. fragment size b) request/response latency

Figure 3: Performance of a multi ROBin source.

The RSI is a 300 MHz PowerPC VME single board computer equipped with a PMC extender so that up to 3 ROBins and 1 ATM PMC can be attached to it. The limitation due to the saturation of the output link is reached for packets larger than 1.5 kB. For short packets, we derive the formula:

$$F_{src} = 1 / (20\ \mu s + 10\ \mu s * nb\_of\_RoB) \tag{1}$$

The amount of time needed to process a request in a source composed of 1 RoB (measured from the time of reception of the request message via the network until the reply message is posted to the ATM interface) is presented in Figure 3b. The data request rate is 18 and 27 kHz (i.e. 50% and 75% of source capability) and the reply message size is 128 bytes. The typical request servicing average latency is ~40 µs. The performance figures that were measured on CompactPCI and on a PC are close to those presented.

Final system requirements mandate that sources shall be able to service requests at a maximum rate of ~10-20 kHz depending on sub-detectors. Current results give confidence that this goal can be met although more work is needed to reach a conclusion (operation of the ROBin with its input link, more ROBins per source, data preprocessing in the RSI...).

VI. The RoI Builder and the Supervisors

A. Description of the hardware

A complete description of the Region of Interest Builder (RoI Builder) designed, built, and tested by our collaborators from Argonne National Laboratory and Michigan State University is given in [14]. The RoI Builder is the interface between the LVL1 trigger and higher level triggers. For events accepted by LVL1, it receives information on regions of interest involving muons, jets, electrons/gammas as well as total and missing transverse energies. This information arrives over 7 separate links, which has to be combined event-by-event to form an event summary to be used by the level-2 trigger processors in refining the choice made at LVL1.

The hardware is based on large Field Programmable Gate Arrays (FPGAs) deployed on 12U cards. Each RoI Builder card accepts input from 3 LVL1 trigger partitions. The RoI Builder for the testbed consists of 2 input cards corresponding to 6 of the 7 partitions in the final system. The output cards are mounted on the rear of the 12U crate from where they are connected to Supervisor Processors (1, 2, 4 or 8) by flat cables. The RoI records from each partition may be distributed to the supervisors either in a round-robin fashion or in a more complex manner determined event by event.

B. Operation and performance

From the stream of records it gets from the RoI Builder, each supervisor forms a summary record and sends it over ATM to a destination processor. A combination of a static schedule table and credit based flow-control mechanism is used to distribute events to the processors according to their relative CPU power and instantaneous load. Another task of the supervisors is to collect trigger decisions (over ATM), pack and multi-cast them (also over ATM) to all ROBs. In the mode of operation where the LVL2 trigger and event filter use different processors, the supervisors also schedule a processor of the event filter farm for each event accepted by the LVL2 trigger. The supervisors can run in emulation mode without the RoI Builder, generating events internally at the desired rate.

The maximum event handling rate, F_{sup}, that can be achieved in our setup is presented in Table 1.

Table 1. Supervisor and RoI Builder performance

F_{sup} (kHz)	no RoI Builder		with RoI Builder		
# of supervisors	1	2	1	2	4
PowerPC 200 MHz	35	70	24	48	96
PowerPC 300 MHz	40	80	28	56	
Pentium II 400 MHz	48	96	-		

For the ATLAS experiment, the maximum LVL1 trigger rate will be 75 kHz (upgradable to 100 kHz). These results show that the desired performance is achievable.

Tests of the RoI Builder with the Reference Software have been made to demonstrate functionality. Although current performance figures are lower than that obtained with the Testbed Software, more investigations are needed and various optimizations are possible.

VII. Destination Processor

A. Data collection and algorithms

The destination processor is in charge of making the selection of the events assigned by the supervisors. At each step of the selection, the processor issues data requests to the relevant sources to get the data it needs for the execution of the algorithm. While waiting for the data of a given event to be delivered, it can handle previous events, issue data requests for other events or become idle if no task can be executed.

The local CPU usage to perform this data collection task on a 400 MHz Pentium II PC running WindowsNT is:

$$T_{dst} = 110\ \mu s + 22\ \mu s * nb\_of\_request \tag{2}$$

On the same platform running Linux it is:

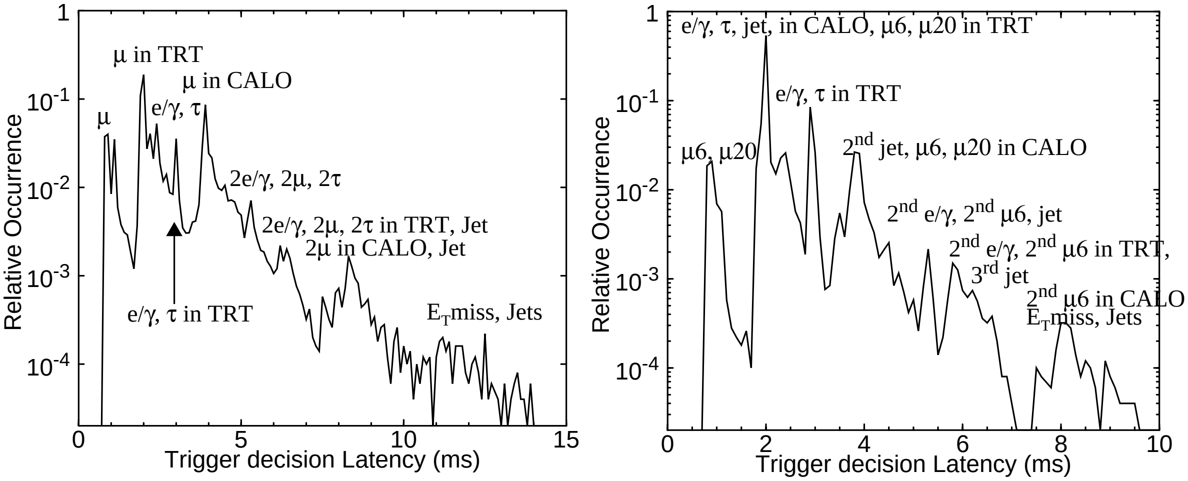
$$T_{dst} = 85\ \mu s + 11\ \mu s * nb\_of\_request \tag{3}$$

Although for supervisor emulation (single thread polling loop application) results are slightly better with WindowsNT (maybe optimizations of Visual C++ compiler are better than that of GNU compiler), shorter context switching and interrupt handling times in Linux make the data collection part more efficient in the processor code (multi-thread interrupt-driven application). Once the desired data fragments have been received, it is necessary to format them in the appropriate structure prior to the execution of the algorithm. Our measurement show that reformatting can be done at ~20 MB/s. So far we have only implemented the algorithm for processing electron/gamma type of RoI, but more complex algorithms and a minimal trigger menu is being studied. This electron/gamma algorithm executes in ~80 µs. In total, a 400 MHz Pentium II, WindowsNT PC is able to handle events composed of 1 electron/gamma RoI at ~2.8 kHz rate (~3 kHz with Linux). For a 14 processor testbed, the total event processing rate is ~22 kHz corresponding effectively to the sum of the capabilities of each processor (not all machines are identical). This type of result provides an input to modeling studies where the combination of expected rates, rejection factors, data volumes, network capabilities and algorithms sequence and timings are used to estimate the number of processors that will be needed to cope with the event rate in the final system.

B. LVL2-like selection sequences

Although no real selection algorithms acting on simulated data were run with the ATM Testbed Software, we tried to run some complex sequences of data collection and dummy processing. Because trying to mimic the complete list of items composing a trigger menu is rather difficult, we made a number of simplifications and approximations. Starting from the trigger menus given in [6], we took the few most significant items and combined the remaining ones in new “trigger items” that do not have any meaning from the point of view of physics, but have a comparable contribution in terms of data movement and processing involved. With these approximations, simplified test trigger menus for low and high luminosity have been derived (see Table 2 and Table 3 in the appendix). The sequences of processing simulated for each type of event are given in Table 4 and Table 5 for the low and high luminosity menu respectively. Data volumes for each detector and RoI types as well as simulated processing times are given in Table 6.

The system is configured with 15 sources, 1 destination (PC 400 MHz), 1 supervisor and 1 monitor. The maximum rate that can be handled by the destination processor is ~950 Hz for both the high luminosity menu and the low luminosity menu without B-physics. When B-physics is added, a rate of 75 Hz is measured. The corresponding execution time is ~13 ms which is close to the sum of the execution time of the low luminosity menu sequence (1 ms) plus the execution time of the inner detector scan performed on one event out of four (45 / 4 =11 ms for algorithm emulation only; the time spent to copy the 120 kB of data received has to be added). With all the hypothesis made, ~80 processors (utilized at 100%) would be needed to cope with a 75 kHz LVL1 rate for the high and low luminosity menus. About ~1000 processors would be needed for the B-physics menu.



a) Low luminosity without B-physics

b) High luminosity

Figure 4. Processor operation for the low and high luminosity test menus.

The system is run at 475 Hz event rate (i.e. half of the capacity of the processor node for the no B-physics case) and the distribution of trigger decision latency is measured (Figure 4a and Figure 4b). The peaks observed correspond to the different trigger menu items. The average decision latency is 2.8 ms and 2.4 ms for the low and high luminosity menu respectively.

VIII. Operation of the System

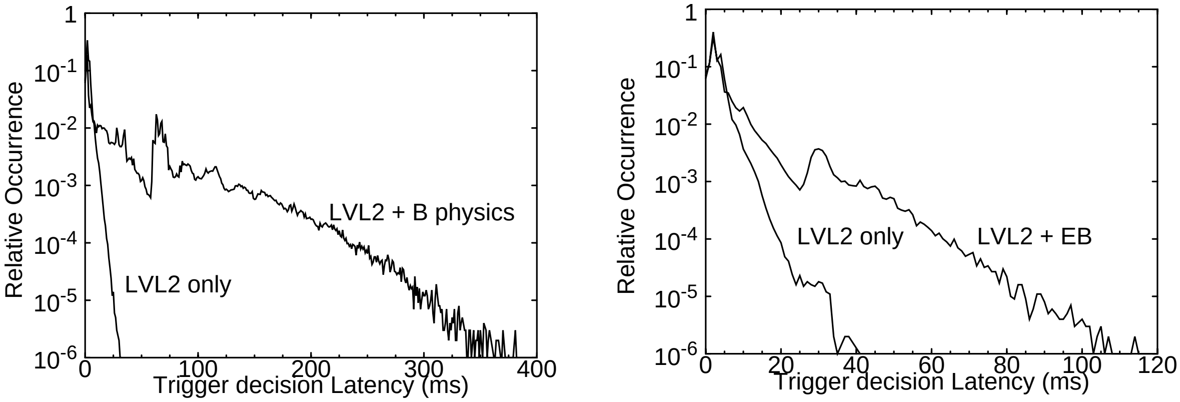
A. LVL2-like selection sequences followed by inner detector scan or event building

The system is now configured with 15 sources, 9 destinations, 1 supervisor and 1 monitor.

The maximum event processing rate for the system is 7.4 kHz and 675 Hz for the low luminosity test menu without and with B-physics respectively. These rates are ~9 times the rate that can be achieved by a single destination processor (9 x 0.95 = 8.55 kHz ≈ 7.4 kHz and 9 x 75 = 675 Hz). The system is run at 50% of its maximum capacity (determined by the processors). The trigger decision latency distribution is shown on Figure 5a. The average is 3.2 ms and 33.4 ms for the low luminosity menu without and with B-physics respectively.

For the low luminosity menu without B-physics, a test is made with executing full event building after a LVL2 accept (in the same processor that accepted the event). The LVL1 event rate is 2.2 kHz when full event building is enabled (this operation is made at 72 Hz given the rejection factor of 30 achieved at LVL2; for our set of parameters the size of each event is 240 kB). The distribution of trigger decision latency is shown in Figure 5b. The average is 3.2 ms and 5.5 ms for the low luminosity menu without and with event building respectively.

A test is made with full event building done in a separate farm of processors after LVL2 accept. No increase of the LVL2 trigger decision latency is observed compared to the case where no event building is performed.



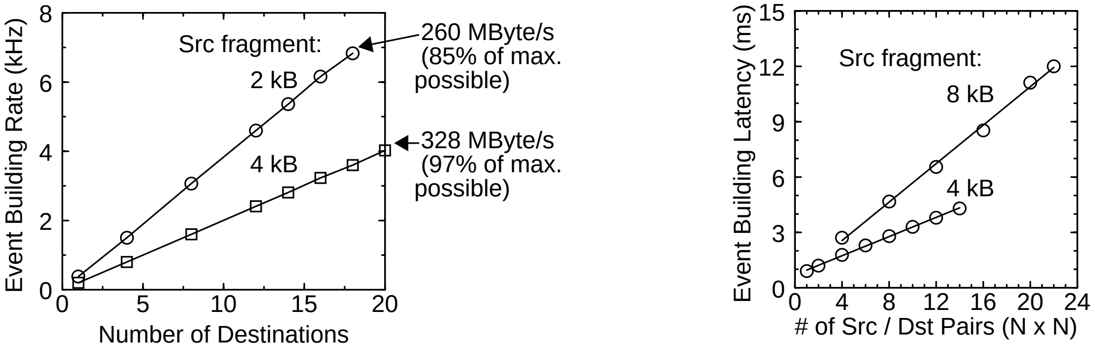
a) Low luminosity with/without B-physics

b) Low Luminosity with/without event building

Figure 5.System operation, low luminosity test menu without/with B-physics or event building.

B. Stand-alone Event Builder

The 48 node testbed was operated in event builder mode with the processors requesting complete event data from up to 22 sources. Because we have only assembled 3 ROBins at present, most sources were not equipped with physical ROBins but ran an emulation of them internally.



a) 20 x N Event Builder Performance

b) Scalability of N x N Event Builder

Figure 6. Stand-alone event builder mode of operation.

As shown in Figure 6a, the aggregate event builder throughput scales linearly with the number of destination processors. With 20 processors and event fragments of 4 kB per source, the total maximum average throughput is 328 MB/s, corresponding to 97% of the theoretical network capacity and ~1/3 of the initial target performance for the ATLAS event builder (1 GB/s). Because the final system will have ~20 times as many nodes, we think that this performance goal is reachable (even with a 2-4 fold increase of the initial requirement). Tests of event builder scalability were made on a N x N event builder, varying N. Results of measurements are shown in Figure 6b. As expected, the event building latency increases linearly with the size of the system.

C. Combined LVL2/ Event Builder tests

The system was operated in the split LVL2 / event builder and single farm modes. A test algorithm is shown in Figure 7a. The first two steps emulate a trigger LVL2 like selection algorithm. The last step is event building followed by some event filter emulation. Event rate, data volumes and processing times are chosen in a way that network links occupancy and processor CPU utilization are ~50%. For each source, the event selection and event builder traffic is ~4 MB/s (each). Assuming a final system with 512 sources and 2 GB/s global throughput, the requirement for each link of the event builder is ~4 MB/s, identical to the traffic load generated in our tests.

In a first setup, the 14 processors are split into two groups: 7 processors run the trigger LVL2 like process, the 7 others are dedicated to the event building of selected events. In this test, slower machines are assigned to the last step of the sequence which is less demanding in terms of rate and real time response. In a second test, the 14 processors form a single farm with each processor running the complete algorithm sequence. We show the distributions of trigger decision latency, T_{dec}, in 7b. The average of T_{dec} is 3 ms and 3.7 ms for the split farm and single farm setup respectively.The slow machines used for the event filter emulation in the split farm setup also perform the high rate part of the sequence in the single farm setup. This explains the difference in the latency profiles. Many other considerations need to be taken into account before choosing the processor farm configuration best suited for ATLAS.

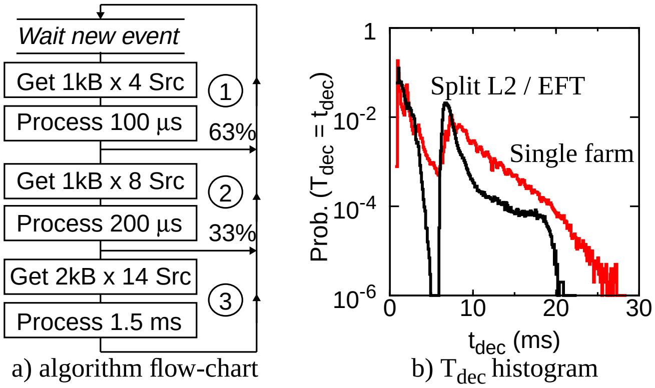


Figure 7: Single farm versus split farm mode of operation.

Considering the flow of data, the various tests suggest that a common network could be used for both the LVL2 trigger (with and without B-physics) and the event builder.

D. System Partitioning

The possibility of partitioning the system in several sub-systems operating independently has been tested. The configuration is shown in Figure 8.

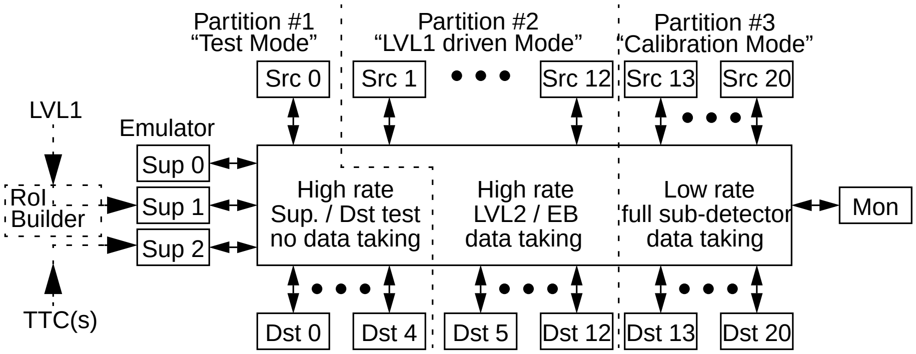


Figure 8. System partition test.

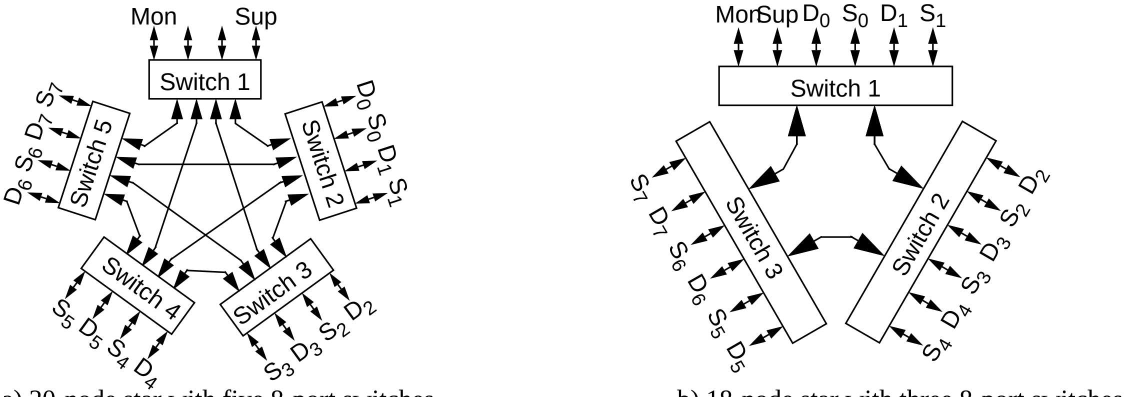
The testbed is split into 3 logical partitions; a supervisor controls the distribution of events for each partition. The operation is as follows:

- partition 1: 1 supervisor, 1 source, 5 destinations. The supervisor distributes internally generated events to the destinations that return a random trigger decision. The supervisor accumulates decisions and sends them to the source. This allows to test communications between these elements.
- partition 2: 1 supervisor, 12 sources, 8 destinations. The supervisor dispatches events to processors that perform a sequential data collection and processing. In a real system, events could come from the RoI Builder. This partition is doing normal event selection and data taking.
- partition 3: 1 supervisor, 8 sources, 8 destinations. The supervisor sends events to processors that collect data from all the sources in that partition. In a real system, the trigger could originate from a Trigger Timing and Control (TTC) domain, the sources could correspond to a particular sub-detector where test/calibration are performed at low rate while the rest of the system is running.

We could start/suspend/resume/stop each partition independently. Although a lot of functionality is not present in our software to have a usable system (it is not the purpose of this development), we think that the concept of logical partitioning is compatible with the architecture that we propose. We have exposed the principle of using multiple supervisors (with different types of input) to achieve it.

E. Switch cascading

One of the critical aspects of networking for LVL2 and EF in ATLAS (and several other experiments) is to build networks with a large number of ports. Switches with thousands of ports may or may not be affordable or available. Cascading switches to build larger networks is likely to be required. On the testbed, we have studied the star topology which seems one of the most attractive topology to reduce the number of switch ports used to interconnect the different switches.



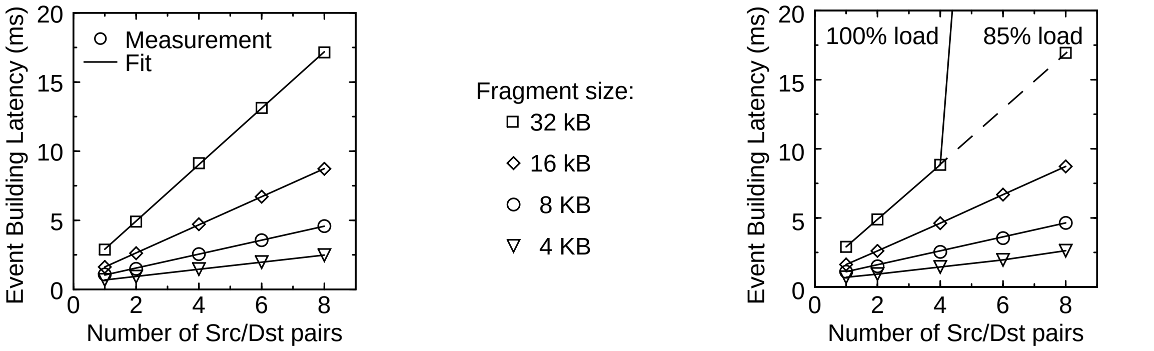
a) 20-node star with five 8-port switches

b) 18-node star with three 8-port switches

Figure 9. Cascaded switch test.

The arrangement on Figure 9a is a configuration where sufficient bandwidth is provided by inter-switch links to cope with a balanced event building type of traffic without reaching the saturation of these links first. The arrangement on Figure 9b is a configuration where inter-switch links reach saturation before other links (with a balanced event building type of traffic).

Performance measurements of event building latency are shown in Figure 10a and Figure 10b.



b) Event building 20-node star network

a) Event building 18-node star network

Figure 10. Measurements on Interconnected switches.

Linear scaling of performance is observed with the 20-node star topology. As expected, for the 18-node star network, internal link switch saturation is reached when the load on external links is ~86%. Using more than one link between any two switches (link aggregation) could be needed to build larger networks economically.

Although this has not been investigated in our demonstrator, we do not foresee major difficulties related to the routing of traffic in larger multi-path networks and difficulties to achieve a correct load balancing in link aggregations because we use only Permanent Virtual Connections (PVCs). Similar network arrangements can be constructed with other technologies (Ethernet) and we think that these should be studied (simulation and prototyping).

IX. Final Conclusions of Studies on ATM Technology

Over the last 5 years, we think that we have sufficiently studied ATM technology in view of the ATLAS T/DAQ to draw the final conclusions on these investigations, stop further research and developments in this field until the final choices for ATLAS are made, maintain the level of competence acquired on ATM and support the existing system as long as the proof of adequacy and cost effectiveness of another technology does not reach a comparable level of maturity.

A. What we achieved with ATM

- Establish and demonstrate methods to avoid network congestion in event builders.

- Demonstrate communication performance boost with low level zero-copy driver/library for ATM network interface cards.
- Validate a “pull” protocol for data collection for LVL2 and EF using point-to-point and point to multi-point communications.
- Show the use of multi-cast of LVL2 decisions in an integrated network for data and protocol messages.
- Prove the principle of data collection for LVL2 on a system with up to 48 nodes.
- Demonstrate full event building on the same system.
- Show the principle of sequential data collection and processing using simplified trigger menus.
- Show the principle of merging protocol traffic with data traffic on the same network.
- Test a combined network for LVL2 and event builder with split/combined processors.
- Test logical partitioning of the system and network.
- Investigate multi-switch arrangements.

B. Pending studies, limitations, unsolved issues with ATM

- Only PVCs have been used. Connection setup and maintenance would be difficult in a large system.
- The limit on the number of connections to set up is an intrinsic limitation of a connection-oriented technology. Networks composed of more than a couple of thousands nodes may hit that limit.
- Link aggregation has not been tested but could be needed to interconnect switches.
- Running multiple protocols over the same network interface was done at Osaka University (running TCP/UDP/IP and ATM AAL 5 simultaneously) but could be investigated in more details.
- Using “standard” device drivers would be preferable to a custom development, though we think that the performance requirement cannot be met with a standard approach.
- End-to-end data transfer integrity is not guaranteed. So far, no attempt is made to recover lost messages. This strategy is acceptable if losses are rare and do not introduce a bias in the trigger.
- Cost of ATM remains high compared to Ethernet.
- Network interface cards at a speed higher than 155 Mbit/s are not there. This bandwidth limit is a problem that Gigabit Ethernet may solve in a more practical and economical way than 622 Mbit/s ATM.

C. ATLAS HLT based on ATM

Today’s proposal for an initial deployment of an ATM network for ATLAS HLT, could be an interconnection of 8 Cabletron SmartSwitch 9500 ATM switches, each with 224 ports at 155 Mbit/s. A network with 1232 usable 155 Mbit/s ports (560 ports would be used for inter-switch connections) could be built for a cost of ~3.6 M$ (based on today’s public price for this product – 1.8 k$ per port). Assuming that about half of the usable ports (i.e. 600 ports) are connected to sources, the raw bandwidth in the source to processor direction is 93 Gbit/s (10 GB/s net after subtracting the overhead of SONET and ATM). Assuming that this type of network is used for full event building of event of 1 MByte (2 MByte) equally spread across all sources, the saturation rate is 10 kHz (5 kHz). If this network is used to transfer LVL2 traffic only, or LVL2 and full event building traffic, limits have to be evaluated for the different scenarios of LVL2 processing sequences, data volumes, etc.

Another network configuration, could be an interconnection of 25 FORE ASX 1000 ATM switches, each with 64 ports at 155 Mbit/s (1600 ports in total). A network with 1000 usable ports (600 ports used for inter-switch links) could be built for ~1.6 M$ (based on today’s public price – 1 k$ per port).

Evolution of technology, emergence of new products, global cost, requirements for ATLAS, results of investigations on Fast/Gigabit Ethernet are among the many factors that will determine the final choices that will be made at the appropriate time.

D. Past, present and Future of ATM

In the early ‘90, ATM was seen as the best candidate to unify the LAN/WAN world, integrate voice and data over the same network and bring multi-media applications to the desktop. Divergence of interests and a slow standardization process eroded the confidence of potential customer’s who found Fast Ethernet available, easy to learn, compatible with legacy LANs and sufficient for their needs. Though seldom deployed up to the desktop, ATM has been mainly used at the core of large corporate and enterprises networks. As large capacity switches are also becoming available in Ethernet and because Gigabit Ethernet links offer a double fold increase of throughput compared to 622 Mbit/s ATM links for half of their cost, the supremacy of the Ethernet family on this market is acquired. For wide area networking, ATM has been more successful and is now a mature and well established technology. Telecom operators and Internet Service providers are deploying multi-million dollars ATM networks for insatiable customers. It is clear that SDH/SONET is today, and will remain in the foreseeable future, the technology of choice to transport information over medium and long distances. Whether ATM cells, IP datagrams or other type of packets are the pieces carried over SONET is still subject to an intense debate. Technology developments, strategic directions of major equipment suppliers and choices of operators are pushing in directions that can equally lead to a stagnation or an expansion of the market share taken by ATM, or even to a progressive abandon of this technology in favor of IP for example.

X. References

[1] ATLAS Collaboration, “ATLAS Technical Proposal for a general purpose pp experiment at the Large Hadron Collider at CERN”, CERN/LHCC 94-43, 15 December 1994.

[2] ATLAS Collaboration, “ATLAS DAQ, EF, LVL2 and DCS Technical Progress Report”, CERN/LHCC 98-16, 30 June 1998.

[3] J. Bystricky et al., “A Sequential Processing Strategy for the ATLAS Event Selection”, *IEEE Trans. on Nuclear Science*, vol. 44, No 3, pp. 342-347, June 1997.

[4] P. Clarke et al., “SCI with DSPs and RISC Processors for LHC 2nd Level Triggering”, *ATLAS Internal note DAQ-19*, 1 December 1994.

[5] D. Calvet et al., “A study of Performance Issues of the ATLAS Event Selection System based on an ATM Switching Network”, *IEEE Transactions on Nuclear Science*, vol 43, No 1 February 1996, pp. 90-98.

[6] J. Bystricky and J.C. Vermeulen, “Paper Modeling of the ATLAS LVL2 Trigger System”, *ATLAS Internal Note COM-DAQ-2000-022*, 16 March 2000.

[7] C. Hinkelbein et al., “Prospects of FPGAs for the ATLAS LVL2 Trigger”, *ATLAS Internal Note DAQ-2000-006*, 27 February 2000.

[8] D. Calvet et al., “Operation and Performance of an ATM based Demonstrator for the Sequential Option of the ATLAS Trigger”, *IEEE Transactions on Nuclear Science*, vol 45, pp. 1793-1794, August 1998.

[9] D. Calvet et al, “Emulation of the Sequential Option of the ATLAS Trigger an using the ATM Local Area Network of the RCNP Institute”, *ATLAS Internal Note DAQ-130*, 31 August 1998.

[10] D. Calvet et al, “The ATLAS High Level Trigger ATM Testbed”, *Proc. XIth International IEEE Conference on Real Time Systems*, Santa Fe, USA, 14-18 June 1999, pp. 101-105.

[11] R. Hauser, “The Atlas Level 2 Reference Software”, *ATLAS Internal Note COM-DAQ-2000-032*, 17 March 2000.

[12] D. Calvet et al.,”Performance Analysis of ATM Network Interfaces for Data Acquisition Applications”, in *Proc. Second International Data Acquisition Workshop on Networked Data Acquisition Systems*, Osaka, Japan, 13-15 November 1996, World Scientific Publishing 1997, pp. 73-80.

[13] O. Gachelin et al.,”ROBIN: A Functional Demonstrator for the ATLAS Trigger/DAQ Read-Out Buffer”, *Proc. 2nd Workshop on Electronics for LHC Experiments*, Balatonfüred, Hungary, 23-27 Sept. 1996, pp. 204-207.

[14] R.E. Blair et al.,”A Prototype ROI Builder for the Second Level Trigger of ATLAS Implemented in FPGAs”, *ATLAS Internal Note DAQ-99*-016, 7 December 1999.

XI. APPENDIX

Table 2. Test trigger menu for low luminosity running.

Item(s)# [a]	Description	LVL1 Rate (kHz)	Fraction of total rate (%)
1	1 MU	23	57
9	1 EM	11.5	29
10	2 EM	1.6	4
30	1 TAU + XE	1.34	3
2 to 8	2 MU	1.093	3
11-29	1 EM + 4 JET	0.721	2
31-55	2 TAU + XE + 2 JET	0.894	2
TOTAL	-	40.148	100

[a] item or list of items with numbering used in [6]

Table 3. Test trigger menu for high luminosity running.

Item(s)# [a]	Description	LVL1 Rate (kHz)	Fraction of total rate (%)
4	1 EM	24.3	62
5	2 EM	4.9	12
2-3	2 MU	4.3	11
1	1 MU	3.9	10
22 to 24	1 TAU + XE	0.961	3
6 to 21	3 JET	0.502	1
25-55	1 TAU + XE + 2 JET	0.532	1
TOTAL	-	39.395	100

a. item or list of items with numbering used in [6]

Table 4. Test sequences for low luminosity running (without B-physics).

Event type	Processing	Reject fraction[a] (%)	Accept ratio[b](%)	LVL2 output rate (kHz)
1 MU	Muon	25		
	Tracker	47		
	Calorimeter[c]	90	4	0.92
1 EM	Calorimeter	86		
	Tracker	86	2	0.23
2 EM	repeat 1 EM twice		0.1	0.001
1 TAU + XE	Calorimeter	80		
	Tracker	60		
	Missing Energy	0	8	0.107
2 MU[d]	Muon	0		
	Tracker	0		
	Calorimeter[e]	0	4	
	Muon	25		
	Tracker	47		
	Calorimeter[c]	90	8	0.087
1 EM + 4 JET[f]	Calorimeter	86		
	Tracker	86		
	Calorimeter Jet 1	25		
	Calorimeter Jet 2	25		
	Calorimeter Jet 3	25		
	Calorimeter Jet 4	25	0.2	0.001
2 TAU + 2 J + XE[g]	Calorimeter Tau 1	0		
	Tracker Tau 1	0		
	Calorimeter Tau 2	73		
	Tracker Tau 2	42		
	Calorimeter Jet 1	25		
	Calorimeter Jet 2	25		
	Calorimeter XE	0	0.6	0.005
TOTAL			3.3	1.351

a. expressed as a fraction of what comes from the previous step.
b. fraction of LVL1 events that passes LVL2 selection.
c. for the B-physics trigger, this step is not run. It is replaced by a full scan of the inner detector (no rejection), followed by the analysis of 2 muons in the calorimeters (rejection 90%)
d. event is accepted if either the first or the second muon passes the selection criteria.
e. this step is skipped for the B-physics trigger.
f. event is accepted if EM is confirmed and all jets are confirmed.
g. event is accepted if at least one of the tau is confirmed and the 2 jets are confirmed.

Table 5. Test sequences for high luminosity running.

Event type	Processing	Reject fraction[a] (%)	Accept ratio[b](%)	LVL2 output rate (kHz)
1 EM	Calorimeter	83	0	
	Tracker	82	3	0.729
2 EM	Calorimeter	83	0	
	Tracker	82	0	
	Calorimeter	83	0	
	Tracker	82	0.09	0.044
2 MU	repeat 1 MU twice		0.2	0.008
1 MU	Muon	25		
	Tracker	47		
	Calorimeter	90	4	0.156
1 TAU + XE	Calorimeter	80		
	Tracker	60		
	Calorimeter XE	0	8	0.077
3 JET	Calorimeter Jet 1	25		
	Calorimeter Jet 2	25		
	Calorimeter Jet 3	25	42	0.211
1 TAU + 2 J + XE	Calorimeter Tau	80		
	Tracker Tau 1	60		
	Calorimeter Jet 1	25		
	Calorimeter Jet 2	25		
	Calorimeter XE	0	4.5	0.024
TOTAL			3.3	1.249

a. expressed as a fraction of what comes from the previous step.
b. fraction of LVL1 events that passes LVL2 selection.

Table 6. Data volumes per source (# of ROBins x size in Bytes) / source multiplicity / execution time[a] (μs).

	Sub-detector		
Processing	Calorimeter	Muon	Inner Detector
RoI MU	4x1k / 4 / 100	4x128 / 4 / 100	4x256 / 4 / 300
RoI EM	4x1k / 4 / 100	-	4x256 / 4 / 200
RoI TAU	4x1k / 4 / 100	-	4x256 / 4 / 200
RoI JET	4x1k / 4 / 100	-	-
XE	4x32 / 7 / 100	-	-
Inner Detector Scan	4x2k / all[b] (15) / 45000		
Event Building	4x4k / all (15) / 0		

a. the processor makes a data copy of all the received fragments before spending the specified amount of CPU time.
b. in the test, all sources are participating (not only those of the inner detector).