14 Overall system performance and validation

14.1 Introduction

In this chapter the system performance of the design presented in this TDR in terms of rate capability and functionality is considered. Results of tests and of modelling, aimed at validating the various aspects of the design, are presented and discussed. The tests concern the performance of event selection, tests of the rate capability of the Data-Flow system in an environment representing about 10% of the final system (the '10% testbed'), and functionality tests of the whole DAQ chain in the context of specialized tests as well as by using it for real data taking with the H8 testbeam. The merits of sequential processing in the LVL2 trigger are discussed on the basis of results from the paper model. Computer models have been used for analyzing measurement results from the 10% testbed, in this way validating the models of components which were calibrated using meaurement results from small test setups. Full system models have provided insight in and strategies for avoiding potential problem areas with respect to rate capability of the full system.

The chapter is concluded with an outlook with respect to anticipated technology developments relevant for the HLT/DAQ system in the near future and with a discussion of the implications of staging.

14.2 High-Level Trigger Prototypes

<u>needs a short paragraph of introduction here</u>

14.2.1 System performance of event selection

The High-Level Trigger will select and classify events based on software largely developed in the offline environment. This approach minimizes duplication of development effort, eases software maintenance, and ensures consistency between the offline and the online event selections. However, given the strict performance requirements of a real-time online environment, it is essential to evaluate the performance of the HLT Event Selection Software (ESS) in a realistic trigger prototype.

The resource utilization characteristics of the HLT software are an important input to the models that predict overall system size and cost. For this reason, a prototyping program was developed to perform dedicated system performance measurements of the event selection software in a testbed environment.

14.2.2 Scope of measurement and validation studies

The scope of the work reported here is limited to a system with full event selection and a minimal Data Flow system, providing full trigger functionality with limited performance. Such a dedicated 'vertical slice test' is sufficient to test the performance of the HLT event selection in a realistic environment. Nevertheless, even in such a limited system, tests and measurements of the data flow aspects relevant to event selection can be performed.

An important aspect of this prototyping work is component integration. Although single components may perform very well in isolated tests, only integration with other system elements may reveal weakness not foreseen in the original design. The integration and testing work described here followed the steps outlined below:

1. Individual component testing and validation (addressed in Chapter 8 for RoI collection and Chapter 13 for the ESS).
2. Functional integration of relevant components (Online Software, Data Flow Software, ESS) in a small testbed, providing feedback to developers.
3. Measurement program, including event throughput and network latencies.

The last two steps were carried out for a LVL2 testbed, an EF testbed, and a combined HLT testbed in the context of validating the HLT/DAQ architecture.

In addition to the testbed measurements, a complementary set of validation tests and measurements can be performed in the offline environment. Although these offline measurements cannot address the full system aspects of the trigger, they help in isolating and understanding the pure algorithmic processing times of the ESS. This is especially relevant for the Event Filter, where events are processed only after all fragments have been assembled, and thus the data flow latencies are completely de-coupled from the ESS latencies.

The following sections summarize the outcome of this integration and measurement program.

14.2.3 Event selection in a LVL2 prototype

14.2.3.1 Prototype and software configuration

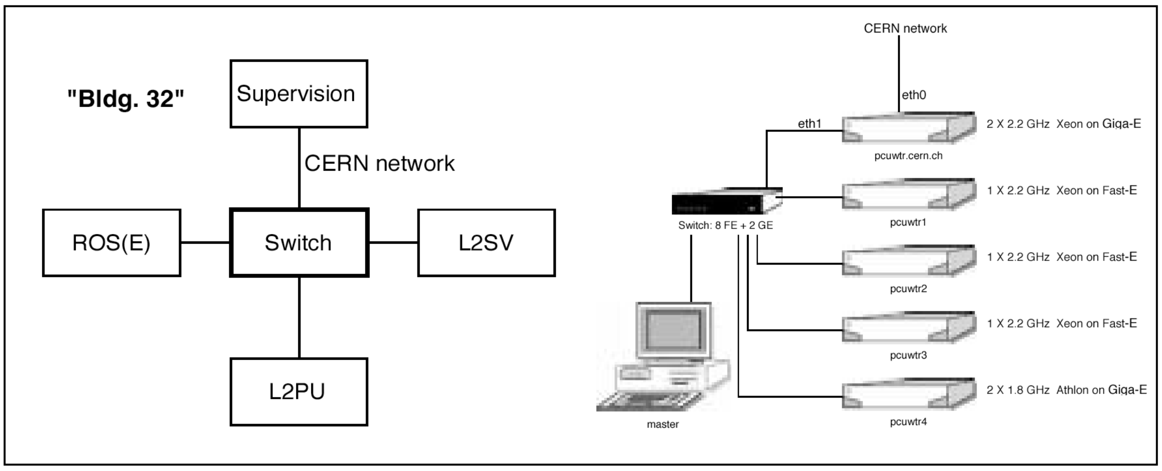

Figure 14-1 The setup for the LVL2 vertical slice testbed. The left figure shows the components in a typical three node configuration. The figure on the right shows the hardware configuration of the five-node LVL2 testbed.

All elements necessary to transport and process event data inside the L2PU were assembled in a LVL2 vertical slice prototype. As shown in Figure 14-1 (left), the following components were included in the prototype:

- L2PU (described in Section 9.2.4)
- ROS or ROS emulator (ROSE, described in Section 8.1.3)
- LVL2 Supervisor (L2SV, described in Section 9.2.3)

The above applications, all controlled by the Online Software (see Chapter 10), ran on a five-node testbed at CERN. Figure 14-1 (right) shows the configuration of the testbed, which was connected to the CERN network through a Fast / Gigabit Ethernet switch. The host machines for the applications were typically either Dual-processor Xeon (2.2 GHz) or Athlon machines (1.8 GHz) or single processor Pentium IV (2.4 GHz). A detailed description of the set-up can be found in [14-1].

The L2PU, the application that effects the event selection, hosts both the Data Flow and the HLT software. In building the vertical slice prototype, the major challenge was achieving the integration of both software frameworks, including the offline components that form part of the ESS. As described in Section 9.2.4.2, the PSC acts as an interface between the control aspect of the Data Flow and the ESS. The selection software comprises most elements described in Chapter 9, including the detector software necessary to assemble and decode the raw data fragments delivered by the ROS. A detailed description of the software integration within the L2PU, including problems and unresolved issues, can also be found in [14-1].

The prototype ran on fully simulated event data. The input data was generated in the offline environment and written in byte-stream format, a serialized version of the raw data format that includes the LVL1 electromagnetic trigger simulation (see Chapter 13). Before starting a run, the detector data fragments were pre-loaded into the ROS (or ROSE) while the LVL1 fragments, corresponding to the RoIs assembled by the RoI Builder, were pre-loaded into the L2SV. Both single electron and di-jet files at a luminosity of 2×10^{33} cm^{-2} s^{-1} were used (see Chapter 13) for the measurements. A suite of trigger algorithms designed to select electromagnetic clusters ran within the L2PU, together with the appropriate detector software to decode the raw data.

The LVL2 calorimeter trigger is the first step after a LVL1 accept of an electromagnetic cluster. Since the calorimeter algorithm executes at the LVL1 accept rate, it is the most critical component when selecting photons or electrons in the LVL2 trigger. For this reason, the LVL2 electromagnetic selection algorithm (T2Calo, described in Chapter 13) was the first algorithm to be integrated in the LVL2 testbed. Unless otherwise noted, all LVL2 trigger prototype measurements shown here are limited to the LAr calorimeter trigger. However, since all data converters and algorithms share the same offline infrastructure, any problems identified in the testbed for the calorimeter (e.g. issues related to the common data flow aspects of the testbed), would most likely arise with other detectors. In addition, by using the calorimeter as a test case for data flow issues, many performance measurements for the other detectors can be first carried out in the offline environment.

14.2.3.2 Measurements

After a first cycle of design (Chapter 9), the HLT event selection software was implemented using mostly offline software components. These components, many of which were in the early stages of development at the time of these tests, had not yet been optimized for performance. Nevertheless, after achieving the integration, it was important to obtain performance measurements with this early software so that any incompatibilities with the LVL2 trigger could be identified. By quantifying the behaviour of the ESS in a realistic trigger environment and identifying any bottlenecks [14-3], a feedback cycle could be established with the original developers of the software.

14.2.3.2.1 Measurement methodology

In order to measure the performance of the LVL2 prototype, various key components of the ESS and Data Flow were instrumented with time stamps [14-1]. The time stamps allowed detailed profiling on an event by event basis of the system behaviour of the prototype. The event throughput, in terms of the mean rate at the L2PU, provided another measure of global performance. All ESS performance measurements quoted here were carried out on a L2PU running in a dual-CPU 2.2 GHz Xeon machine with 1 GB of memory. Unless otherwise noted, the L2PU was configured to run with one worker thread (see Section 9.2.4.1).

14.2.3.2.2 Initial performance of the ESS in the LVL2 prototype

Initial measurements with the Calorimeter ESS ran considerably slower than the ~10 ms per event average processing time budget for the LVL2 trigger. Most of the time was used in the data conversion and preparation steps, which had only recently been made available and had not yet been optimized. This part of the software converts the byte-stream data into objects that the downstream algorithms can process and applies the appropriate calibration.

These first measurements also used the basic form of the so called ‘London scheme’ for data access (described in Chapter 9), where ROB data fragments are requested on demand across the network in a sequential fashion. In this case the total network latencies incurred by the data requests were some **4 ms**. Given that a typical electromagnetic RoI spans 12 to 15 ROBs, this measurement agrees with the data flow measurements of Chapter 8, where a single request with comparable payload introduces a latency of **~220 μs** per ROB request.

The LVL2 calorimeter algorithm mean execution time is **1.5 ms** per event, with the remainder of the time consumed by framework and service overheads.

14.2.3.2.3 First performance improvements

After the above measurements were completed, an initial optimization of the ESS was made for a few critical components. Since the architecture of the LVL2 trigger (via the RoI mechanism) mixes data requests with algorithmic work, it is no surprise that most of the optimization work centred in data transfer and conversion issues. In addition, these studies are the first to investigate in detail data access performance. The performance of the LVL2 calorimeter algorithm itself has already been documented [14-6].

The initial network latency was reduced by implementing the optimization described in Section 9.5.5.2 whereby all of the data fragments of each RoI are pre-fetched across the network in a single operation. Thus adapting to the LVL2 needs, the wholly on-demand data access model implemented in the ESS and so minimizing sequential network transfers. After this change, the network access latency was reduced to **650 μs** per event for the system configuration shown in Figure 14-1.

In order to provide a baseline measurement of the minimum time the data converter function would require using no offline-inherited code, a new LAr converter prototype, was developed. This converter satisfies the time-critical needs of LVL2 but avoids off-line dependencies, and was developed based on a lookup table method for region selection and an optimized raw data conversion scheme. In this prototype converter, the event data is passed directly to the requesting algorithm instead of publishing in a Transient Event Store. In addition, a scheme with calibration after zero-suppression was introduced (details can be found in [14-1]). This optimized converter prototype demonstrated that a data conversion processing time of **3 ms** per event for an RoI size of $\Delta\eta \times \Delta\phi = 0.3 \times 0.3$ and no zero-suppression was possible.

Applying all of the above improvements gives a total execution time per event in the prototype of **4 ms** and **6 ms** for RoI sizes of $\Delta\eta \times \Delta\phi = 0.3 \times 0.3$ and 0.5×0.5, respectively.

Figure 14-2 (left) shows the total latency distribution for di-jet events at low luminosity for a $\Delta\eta \times \Delta\phi = 0.3 \times 0.3$ RoI. As can be seen from the figure, over 95% of the events are processed within 5 ms. Figure 14-2 (right) shows the main contributions to the total latency. Pure feature extraction in the calorimeter is completed within 500 µs, while data access is typically completed within 1.2 ms for 50% of the events.

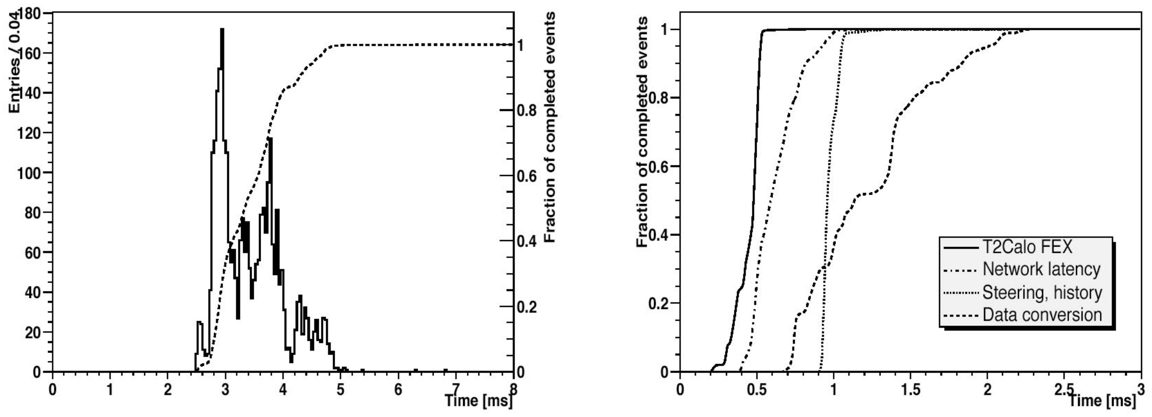

Figure 14-2 Total latency for event processing (shown at left) in the LVL2 LAr trigger for di-jet events at low luminosity. The dotted curve is the integral of the distribution, showing that 95% of the events are processed within 5 ms. The four main contributions to the latency are shown (right) as curves of integrals. The contributions are (in order of importance): data access and conversion, framework overheads, network access time, and algorithmic processing.

The use of zero-suppression in the calorimeter [14-6] has been shown to be very effective in reducing algorithm execution time. Applying a 20 MeV energy threshold cut in the L2PU during the data conversion step reduced the mean algorithm execution time by **290 µs** and **1.4 ms** per event for RoI sizes of $\Delta\eta \times \Delta\phi = (0.3 \times 0.3)$ and (0.5×0.5), respectively. The bulk of this reduction is due to the reduced number of calorimeter cells that the feature extraction algorithm has to process. This reduction could be extended to the data converter if the zero-suppression could be applied upstream of the L2PU (e.g. in the ROS or in the RODs). More details on these measurements can be found in [14-1] and [14-4].

These results demonstrate that the system performance required can be achieved with current software, but that some of the offline components need further optimization in order to provide a better match to the LVL2 trigger. The optimizations described above are now being studied for implementation in the LAr data conversion software (for use in both the trigger ESS and the LAr offline itself) and substantial improvements have now been achieved there, showing that the re-use of offline software in the LVL2 trigger is mutually beneficial. These improvements will benefit not only the LVL2, but the Event Filter and the offline software itself, reducing the overall computing resource needs of ATLAS.

14.2.3.2.4 Other measurements

Since the L2PU will run multiple worker threads hosting the ESS, it is important to test the LVL2 event selection software in a multi-threaded configuration. After applying the changes necessary to make the ESS thread-safe [14-2], the prototype ran with two worker threads in the L2PU. The event throughput increased from **279** Hz to **304** Hz, although the CPU utilization was only **~50%** per CPU. The non-scaling effect is due to an inefficient use of memory allocation locks [14-1] in the Standard Template Library (STL). This will be improved in the next round of software optimization.

The LVL2 trigger can be configured to run multiple L2PU applications in a single host machine. In this configuration, each L2PU runs different instances of the ESS, each processing events in parallel. This scheme increases the resource requirements on the host machine since memory is not shared between the L2PUs and, in addition, the number of external connections increases. The three-node testbed was configured to run with two L2PUs in a dual-CPU host. The event throughput rate was measured to be **XXX** Hz. In order to draw any conclusions from this study, a careful analysis of the resource implications of using multi-process versus multi-threaded applications must be made. This study will be performed when the multi-threading issues outlined above are solved.

All LVL2 testbed measurements quoted so far have been for the LAr calorimeter. At the time of writing, the initial implementation of the SCT/Pixel converters based on the Chapter 9 design were only just becoming available and and had not yet been optimized for LVL2. In order to have some baseline measurements for the SCT/Pixel data conversion process, an early SCT and pixel prototype of the trigger software, implemented before the design described in Chapter 9 was available, was developed and tested in the LVL2 testbed [14-4] setup described above. This prototype included all software components necessary to map ROBs to an (η,ϕ) region, request the data [14-5], and decode the SCT/pixel byte-stream data and the LVL1 information. In addition, a LVL2 tracking algorithm, SiTree [14-7], reconstructed high-p_T tracks within each RoI. The prototype yielded a mean total processing time of 2.5 ms per event for track reconstruction of single electrons with no pile-up. This compares favourably with the ~30 ms necessary to access and convert the SCT and pixel detector data with the first implementation of the new SCT/Pixel converters. However, after implementing the same improvements which have been described above, it is reasonable to expect that the LVL2 SCT and pixel detector decoding software can also be significantly optimized so as to achieve the same level of performance as that achieved by the early prototype.

The muon trigger, like the LVL2 calorimeter trigger, is also critical at LVL2 since it is executed at the LVL1 muon rate. Measurements [14-8] of system performance were carried out in the HLT offline environment (i.e. running the ESS in the offline environment rather than in the testbed). Running a full selection chain for p_T = 100 GeV muons at high luminosity yielded a mean total processing time of 10 ms on a 2.4 GHz machine. This processing time is dominated by data access and conversion, 0.7 ms per RoI for the RPCs [14-9] and 7.9 ms per RoI for the MDTs [14-10]. The remaining 1.4 ms are consumed by the muFast pattern recognition and tracking algorithm itself (algorithm description can be found in Chapter 13). Although at this time the LVL2 muon trigger software has not been integrated in a testbed environment, these measurements indicate that the processing times, including data access, are well understood. Assuming that the same data flow and network overheads that affect the LAr trigger also affect the muon trigger, the total LVL2 latency is well under control.

14.2.3.3 Conclusions and outlook

As seen in the previous section, very detailed studies and first measurements with the HLT event selection software show that there is great potential for optimization in the present initial implementation. Some components of the ESS already perform well, e.g., the selection algorithms. Other components, in particular the detector data converters, need further optimization. Detailed studies have been made of the data preparation process. Dedicated test prototypes have shown that this optimization is possible. In addition, the muon trigger software already performs in the offline HLT environment at a level that is compatible with the LVL2 processing budget. Extrapolating the present performance figures of these prototypes to 8 GHz CPUs which will probably be available at LHC turn-on, gives total processing times that are compatible with the 10 ms average event processing time budget of LVL2.

There are a few open issues that need to be addressed for the LVL2 trigger. The multi-threading inefficiencies encountered with the STL need to be resolved and the implications on the current working model need to be fully understood.

The data access mechanism of the ESS needs to be optimized for the LVL2 trigger. In particular, only the data fragments necessary to take a trigger decision should be transferred, and fragments required for a given RoI must be requested in one block. Implementing these optimizations brings very substantial performance improvements as was demonstrated above.

Since the LVL2 is a performance-critical component, the 'on-demand' nature of many of the offline configuration and data access services should be adapted to a deterministic model, which is more suitable for a trigger environment.

The ESS will be processing events in the LVL2 environment at the LVL1 output rate. This means that any offline components used in the trigger will be executed nearly 10^3 times more frequently in the LVL2 than in the offline environment. This imposes severe constraints of stability and robustness on the software. Increasing the modularity of the software in such a way that the components needed to build the LVL2 form a restricted highly-robust software 'core' will help to address these constraints. This core would still form the basis for the higher-level offline reconstruction.

Running the ESS in a LVL2 vertical slice prototype serves two purposes: it provides performance measurements for estimating resource requirements, and it provides a platform for validating the ESS and the trigger architectural choices. Building the event selection code with offline components, not only reduces the development and maintenance costs in the LVL2, but it helps in optimizing the performance of these components. However, in order for this model to work, the LVL2 constraints must be taken into account in the core offline software. The LVL2 tests shown here demonstrate that this is not only possible, but desirable and mutually beneficial for both the trigger and offline software systems.

14.2.4 Event selection in an Event Filter prototype

In the Event Filter, the event selection process is carried out by the processing task, described in Section 9.3**.** The processing task hosts the HLT selection software, which is fully implemented using Athena and offline reconstruction components. In order to validate the event selection in the Event Filter, a prototype was developed that brings together the DAQ Data Flow sub-system (SFI and SFO), the EF dataflow (EFD) and the processing Task (PT) running the HLT selection software in a single system (***see section 9.3.2.3 and the note "Event Filter infrastructure validation tests" ATL-DH-TR-0004 EDMS Id 391248***).

The processing tasks are independent processes running on every node. The integration of the selection software in the Event Filter consisted mainly in developing concrete implementations of some of the Athena services. In particular, the ByteStreamCnvSvc, the conversion service in charge of reading the raw data in Byte Stream format and convert them into their transient representation (Raw Data Objects) recorded in the Transient Event Store. An interface class, the ByteStreamInputSvc, defines the methods used by the conversion service for reading ByteStream data from a persistency source. The converse process is handled by the ByteStreamOutputSvc. The concrete classes that implement the exchange of event data with the Event Filter dataflow are the EFHandlerInputSvc and EFHandlerOutputSvc. They make use a shared memory based mechanism. The EFHandlerInputSvc issues a request to the EFD for the next event and receives a pointer to the Event in the Shared Heap. When the processing is completed, the EFHandlerOutputSvc returns the answer to the EFD and if the event is accepted, the 'EF pseudo detector' fragment is serialised directly in the Shared Heap in a location previously requested to the EFD.

The Event Filter prototype was tested in various set-ups. The performances of the EF dataflow itself running with dummy PTs are described in Section 9.3.2.4. Here we concentrated on the performance involving real events processed by the HLT selection software.

Tests were performed by running the Event Filter with simulated events. The data set that was used is a sample of Dijet events that have passed the LvL1 electron trigger. These data constitute the main background for the electron trigger. They included noise and pile-up events. The event size was of the order of 3 Mbyte. The data are preloaded in an SFI emulator. The Event Filter was configured to run the HLT suite of electron identification programs. A the time these tests were made, the EF algorithms were not yet available and the T2calo algorithm was used instead. The data files were generated offline and included the result of the LVL1 selection in the form of a LVL1 pseudo-detector fragment. It is needed to provide ROI guidance to T2calo.

The EFHandlerOutputSvc was implemented in the prototype. The set of decisions of the selection process implemented were: 'Accept', 'Reject', or 'Error'. The EFresult pseudo-detector fragment contained a single ROD fragment generated by the converter associated to the EFresult object. No additional reconstructed objects were serialised in the prototype, but the same mechanism will be used when they become available. For accepted events, the fragment was written in the Shared Heap. It is then appended to the original event by the EFD and sent to the SFO.

A series of validation tests were carried out:

1. Validation of the exchange of data between EFD and the Athena PT hosting the Selection Software: validating the event input procedure consisted of checking the integrity of the data after passing it from the EFD to Athena. This is simply done by checking that the Trigger Selection Software produces the same results as when running in the offline mode. In order to validate the Athena output procedure, the SFO wrote the events to a local file. The integrity of these data has been verified by reading, unpacking and processing the event with 'offline' Athena.
2. Throughput measurements.

Measurements have been carried out in different set-ups: a dual-processor machine hosting all processes SFI, SFO, EFD and PTs and a multiple host set-up with the SFI and SFO running in one machine and the EFD and PT in another one.

The hardware used was the following:

1. a single-host dual-processor: Intel XEON 2.20 GHz.with 1 Gbyte of RAM
2. a multiple host set-up: with two Intel XEON 2.20 GHz with 1 Gbyte of RAM with Fast Ethernet connection
3. a multiple host set-up: with two Intel XEON 2.20 GHz with 1 Gbyte of RAM with Gigabit Ethernet connection.

The AthenaPT ran the T2calo algorithm from the debug release in verbose mode. All events were forced accept. The total user time per event was on average 180 ms for the dijets events described above and the virtual memory size was 260 Mbytes. The event size was approximately 3 Mbytes. Table 14-1 summarises the results.

In the first set-up, adding a second PT profits from the Dual CPU and increases the throughput

Table 14-1 Summary of results of ***TO BE ADDED***

Set-up	Number of Athena PTs	Time per event	Data volume throughput	Remark
1	1	190 ms	16 Mbytes/s	
	2	110 ms	27 Mbytes/s	
	3	stopped by time-out		limited by memory swap
2	1	380 ms	8 Mbytes/s	limited by network
3	1	190 ms	16 Mbytes/s	
	2	120 ms	25 Mbytes/s	

although not quite by a factor 2. The other processes SFI, SFO, and EFD require some fraction of the CPU. Adding a third PT saturates the RAM and swapping slows down the process considerably. The PTs were blocked by the timeout mechanism of the EFD. In the second set-up, multi-host with 100 Mbit/s Ethernet connection, the throughput is limited by the network bandwidth already with one PT. In the third set-up with Gigabit Ethernet, there is no such limitation, a single PT can use close to 100% of a CPU. Adding a second PT, increases the throughput. Both PT use about 80% of a CPU, the rest being taken by the EFD. The characteristics of the Athena PT running the selection software in terms of latency, memory and event sizes have to be evaluated in order to define the optimal configuration of the EF. In this test, we have observed the interplay between these parameters.

14.2.5 The HLT vertical slice

The LVL2 trigger and the Event Filter were integrated in a single testbed as shown in Figure 14-3. The LVL2 slice (described in Section 14.2.3) and the EF slice (described in Section 14.2.4) were connected to form an 'HLT vertical slice' using the CERN network infrastructure. The DFM and the Event Building were located geographically close to the event fragment sources. In order to pass the LVL2 result to the EF, one of the ROSes was configured as a pROS (described in Section 8.1.3.4). The entire system was configured and controlled by the Online Software using the CERN network. During the tests, the ROSes were pre-loaded with LVL1-preselected events.

.

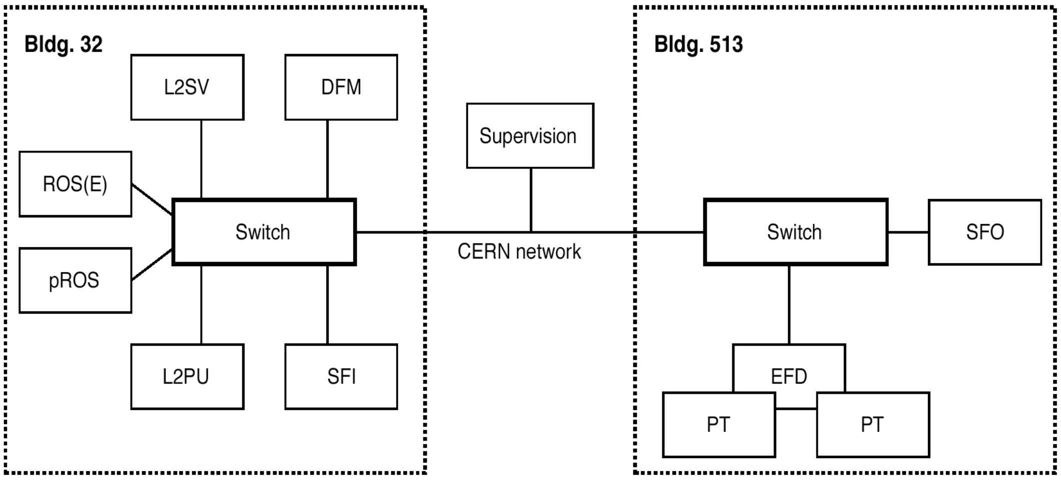

Figure 14-3 Setup for the LVL2 (left) and EF (right) vertical slice testbeds. The combined HLT testbed consisted of both LVL2 and EF testbeds connected via the CERN network infrastructure.

NEED TO ADD RESULTS HERE.

14.3 The 10% prototype

14.3.0.1 Description of the 10% testbed

In Chapter 8, "Data-flow", the performance of the individual components in the ATLAS Data-Flow system has been presented. These components, when operated together, carry out the functions of RoI collection and Event Building. The performance for these two functions using independent subsystems has also been presented in that chapter. The final ATLAS Data Collection system requires simultaneous operation of RoI collection and event building. This section describes results obtained from a testbed with a size of approximately 10% of the final system, with full data collection functionality. Although the testbed necessarily is a scaled down version of the final system, individual components have been operated at rates similar to those expected for the final system. The primary aims of the 10% testbed are to demonstrate full functionality of the data collection in both the LVL2 and the EB subsystems simultaneously and to check for possible interference between the subsystems. The latter is especially important with respect to the choice to be made between a switch or bus based ROS. The testbed results have also been used to calibrate and validate computer models of components and systems. The approach taken is to assure that the measured performance of 10% systems can be successfully reproduced prior to drawing conclusions from modelling full size systems. Finally, the testbed results have been used to study possible ways to achieve a staged approach to a full size system by the progressive installation of components.

The first testbed studies have been done with a single Data Collection application and associated test environment. Next, small setups with one instance of each Data Collection component, aimed at demonstrating functionality, were used. Studies with more complex setups of either EB or LVL2 subsystems followed. The largest testbed inherits from the previous ones and represents about 10% of the final system. It's organization, as presented in Figure 14-4, reflects the architecture of the final system.

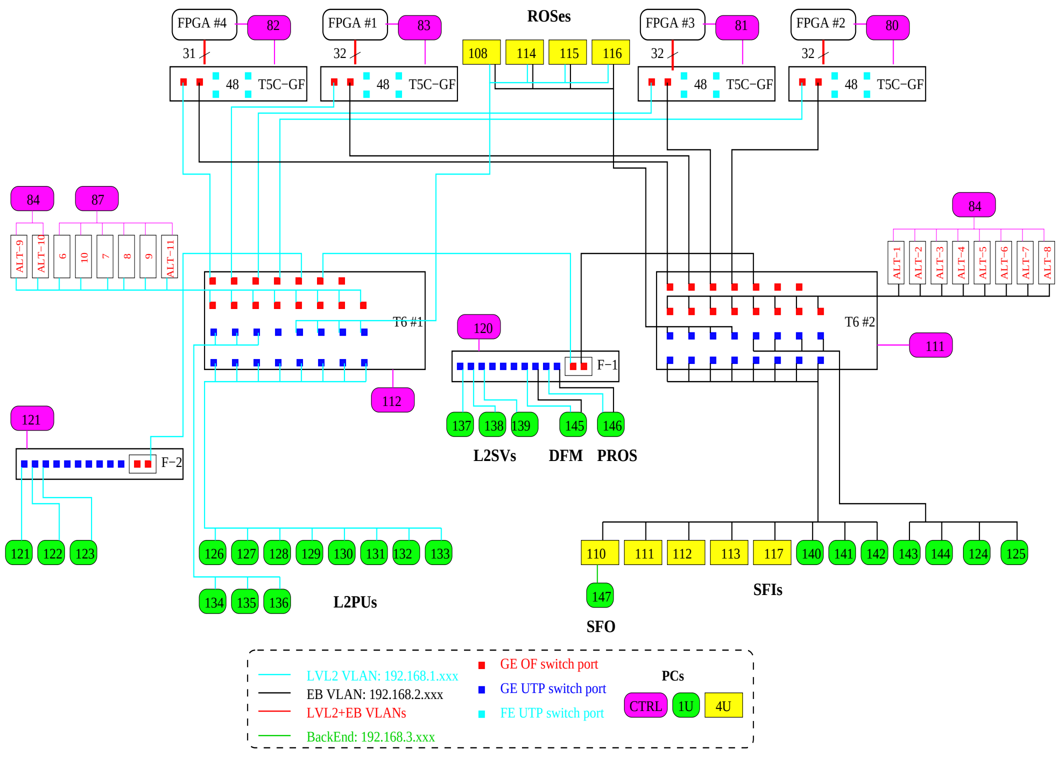

Figure 14-4 Organization of the 10% testbed **COLOURS OF FIGURE TO BE CHANGED OR FIGURE TO BE REPLACED BY SIMPLER FIGURE**

The central part of the 10% testbed consists of two central switches. The central switch T6 #1 has been assigned to the LVL2 subsystem whereas the central switch T6 #2 plays the central role in the EB subsystem. Both central switches are 32 port all-Gigabit Ethernet switches (fibre/UTP ports). For the final system the possibility of installing more central switches is foreseen. For example the number of central switches can be four when ROB concentrating switches with four uplinks are used. The consequences of using more than one central switch per subsystem have been studied in the testbed and the performance has been measured. This could be achieved in a straightforward way as the functionality of the processing nodes is defined by the type of application executed: changing it from SFI to L2PU, or vice-versa is possible by reloading appropriate software into a PC.

In the LVL2 subsystem (T6 #1) some of the processing nodes are attached directly to the central switch (nodes 126–138) and some via a LVL2 grouping switch (F-2 in Figure 14-4). For the final system it is planned to connect LVL2 processing nodes via LVL2 grouping switches, but for the testbed not enough LVL2 grouping switches (like F-2) were available. Therefore PCs running the L2PU application were connected directly to free ports in the central switch in order to produce more traffic in the LVL2 subsystem.

In the EB subsystem PCs 110-113, 117, 124, 125 and 140–145 act as SFIs directly connected to the EB central switch. The current paper model predictions (***REFERENCE TO CHAPTER 2 AND***

APPENDIX A) assume ~90 SFIs for the final system, the number of SFIs installed in the testbed amounts to 10% of the final system.

Three options have been explored with respect to how the detector data stored in the ROBs are accessed. The FPGA ROB emulators (FPGA #1 - FPGA #4) allow to study the option of using ROBs with individual network access via Fast Ethernet connections. This option is described in more detail in the [***ATLAS TDAQ Switch-based architecture note reference***]. The ROB emulators are connected to concentrating switches (4 * T5C). Each concentrating switch has two uplinks, connected to the central switches. The second option studied consists of readout of the data via bus-based ROS PCs. In the testbed PCs act as bus-based ROS emulators (108,114–116).(***perhaps someone could provide more details, or at least a reference to the bus-based ROS..?***). In the third option two or more ROBIns share a single network connection, as in the prototype ROBIn. For this option ROB/ROS emulators are used based on programmed Gigabit Ethernet NICs from Alteon [***REFERENCE TO ALTEON NICs]***. Depending on the software loaded, the ROB/ROS emulator can reply with either individual ROBIn data, or produce reply messages with data from a number of ROBIns. The hardware emulators (FPGA and Alteon NICs) allow to output the data from a number of ROBIns. Therefore the data from more than 1600 ROBIns can be provided with only a limited number of emulators (128 FPGA channels and tens of Alteons).

The architecture with the two central switches and the ROB concentrating switches creates Ethernet loops. The Spanning Tree algorithm is used to switch off the redundant links and VLANs are used to avoid changing the configuration of the testbed. As at the time of writing the Data Collection applications did not support VLANs on a single network interface, the DFM and the LVL2 Supervisor (the two applications which communicate across VLANs) were equipped with two network interface cards and connect via the F-1 switch in Figure 14-4 to both theLVL2 and EB subsystems.

14.3.0.2 Results

Measurement results for the maximum event building rate as a function of the number of credits in the SFIs have been presented in [***REFERENCE TO CHAPTER 8]***. . The measurements were done with the testbed configured for event building only (i.e. without LVL2 traffic) using both central switches and with each of the three different types of emulators separately and with all types simultaneously active. Also measurement results for the maximum RoI handling rate as a function of the number of threads in the L2PUs have been presented in [***REFERENCE TO CHAPTER 8].*** These measurements were done with the testbed configured for LVL2 triggering only (i.e. without EB traffic) using both central switches. The maximum event building rate was found to scale with the number of SFIs until the limit imposed by the total output bandwidth of the FPGA and Alteon emulators is reached, see also [***SECTION ON COMPUTER MODEL RESULTS***]. Also the total RoI handling rate scales with the number of L2PUs until the rate is limited by the data sources. In Figure 14-5, Figure 14-6 and Figure 14-7 results for simultaneous RoI traffic and event building are presented.

Further text needed

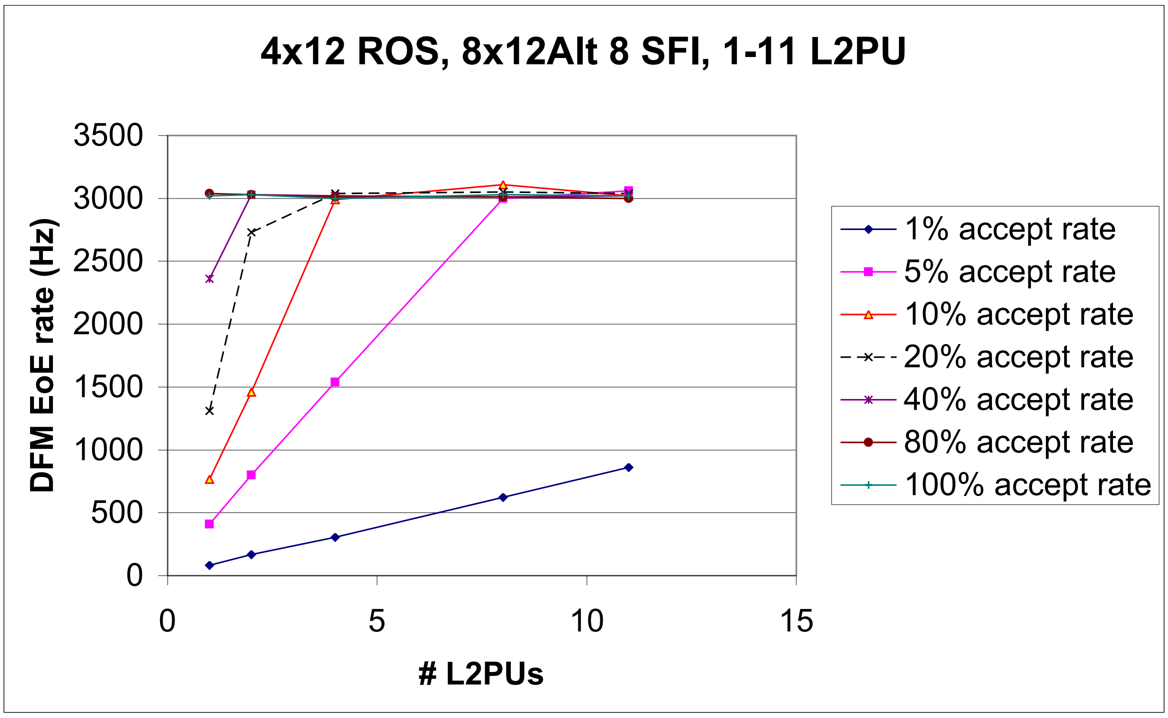

Figure 14-5 Rate for simultaneous RoI handling and event building in the 10% testbed. 12 ROS units, each servicing 12 ROLs, were emulated with 4 PCs and 8 Alteon NICs. Per event the L2PUs requested the data from all ROLs from one of the emulated ROS units. Event building was done by requesting the data from all 12 ROLs of each ROS

.

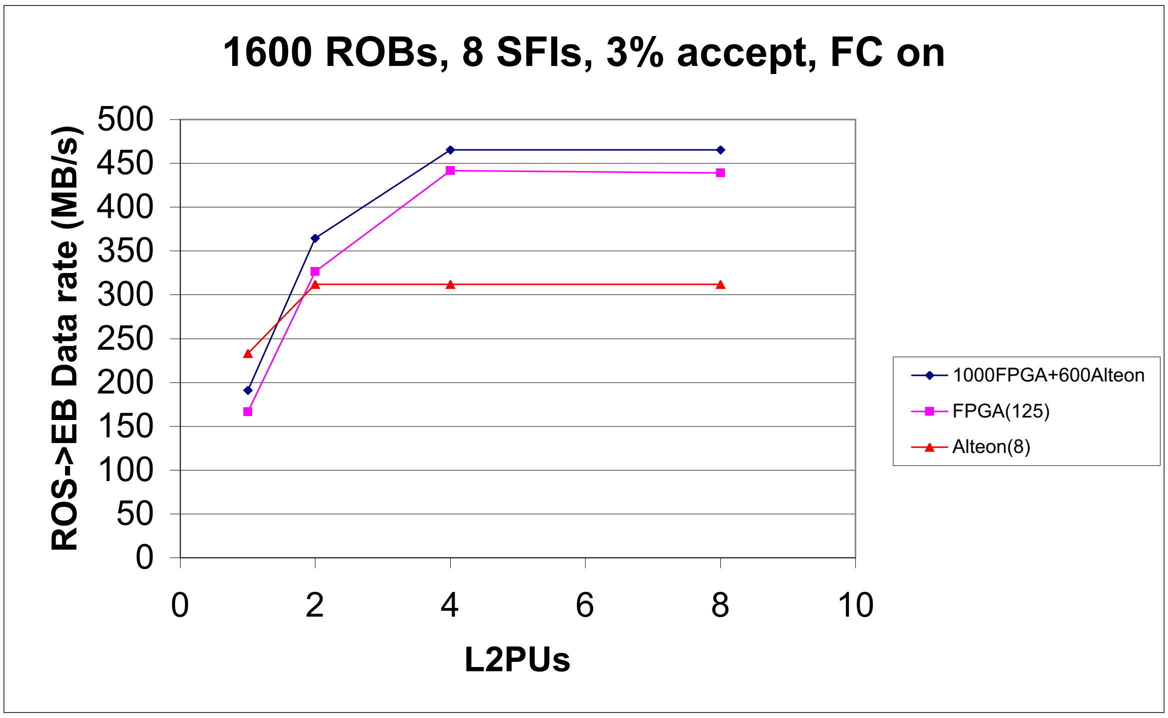

Figure 14-6 Rate for simultaneous RoI handling and event building in the 10% testbed

Figure 14-6 three different setups for switch-based read-out: 1000 ROBs emulated by FPGAs and 600 ROBs emulated by Alteon NICs; 1600 ROBs emulated by the Alteon NICs only and 1600 ROBs emulated by the FPGAs only. The EB subsystem was receiving 3% of processed events with positive decision and for those events 8 SFIs were used to collect data from ROBs and form an event. The L2PUs requested data from ***XX*** ROBs

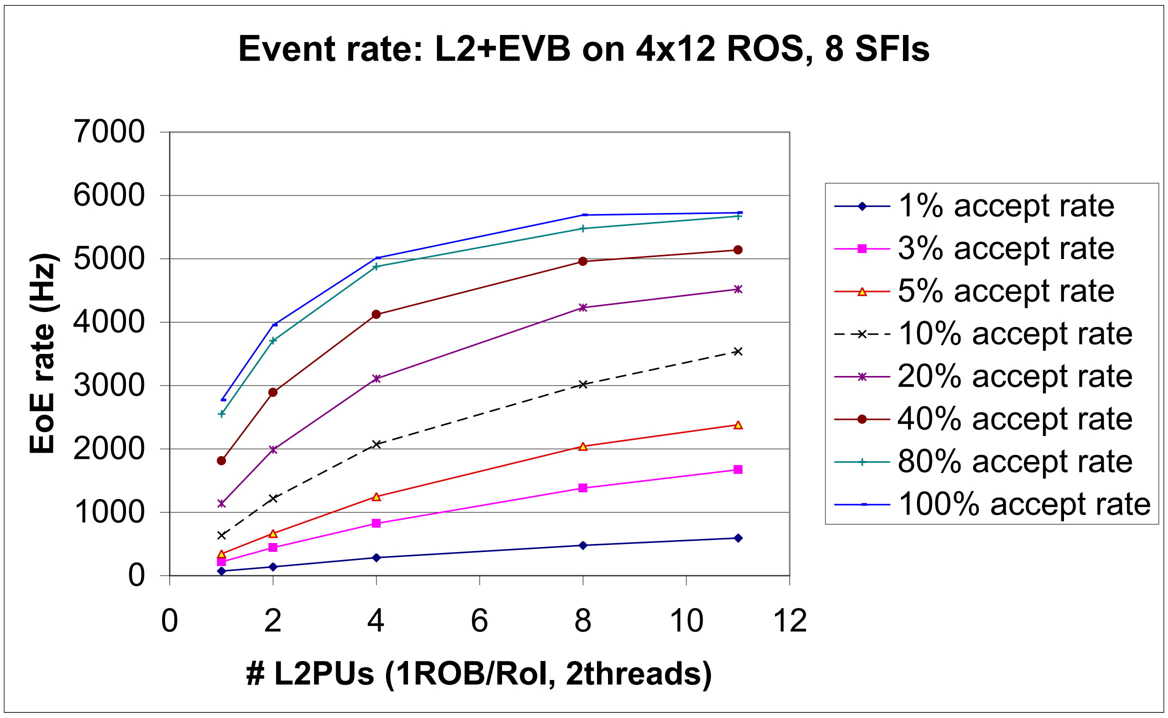

Figure 14-7 ***NEED DESCRIPTION***

Figure 14-7 12 ROS units, each servicing 12 ROLs, were emulated with 4 PCs. Per event the L2PUs requested the data from a single ROLs from one of the emulated ROS units. Event building was done by requesting the data from all 12 ROLs of each ROS.

Conclusion: ??

14.4 Functional tests and test beam

Often, during prototyping, more weight is put on the performance of a system than on its stability and maintainability. Functional user requirements tend to have in this phase of development a lower priority than the achievement of the performance requirements. This is to some extent true also for the ATLAS HLT/DAQ system. Nevertheless, by carrying out a series of functional tests and exposing the system to non expert users at the ATLAS test beam sites, also these aspects have been addressed.

Four different aspects of the global functionality have been covered: system configuration, stability in cycling through TDAQ finite states, monitoring and fault tolerance. These aspects have first been tested in dedicated laboratory setups and then verified in a 'real' environment, during test beam data taking.

14.4.1 System configuration

A TDAQ system has to be easily reconfigurable in order to accommodate the substitution of hardware, the change of trigger conditions, etc. This means that on one side all the tools for storing/retrieving the configuration parameters into/from a database have to be available and on the other side that the Run Control, DataFlow, and Trigger software has to be designed to be dynamically reconfigurable.

In a testbeam setup the data taking configurations tend to change very often, because of the addition of the read-out chain of new parts, which need stand alone as well as integrated debugging. In order to ease the integration of the detector read-out chains in the DataFlow system, nesting of partitions was introduced. Nested partitions allow the different groups to develop their databases independently and then to link them together under a common top partition. The success of this technique has been proven especially during the testbeam period dedicated to data taking with the Muon detectors: five different ROD or ROD emulator crates have been setup independently and tested, before they were connected for further debugging to their corresponding ROS units and finally also to the event building system.

While the configuration of the TDAQ hardware and of the software applications are specified at the beginning of a session, there are a number of configuration parameters for the various applications which can be changed dynamically, e.g. amount of memory to be reserved for the data, length of queues, etc. There is also the possibility to have very dynamic parameters which change for every run, e.g. the run number and calibration variables. These values are not kept in the database and are at the moment distributed via the Information Service. For the future, mechanisms of dynamic changes in the database with the corresponding notification mechanisms to the affected applications are being studied.

14.4.2 Finite state machine transitions

When performing a series of measurements with different configuration options, the TDAQ system must be capable of cycling through a finite set of states stably. This capability has been checked with the help of automated scripts cycling repeatedly through the finite state machine associated with the states. The absence of problems has been verified during all testbeam periods.

14.4.3 Monitoring

In a distributed system such as the TDAQ system it is important to monitor the operation of the system continuously. All applications publish regularly statistics on their performance, as well as on the occurrence of errors via the Information Service provided by the Online Software. Furthermore, the ROS and the SFI are capable of providing event data for physics monitoring. The operational monitoring has been intensively used to carry out all the performance measurements described in the previous chapters. Also, all monitoring aspects have been regularly used by the persons on shift during testbeam data taking.

14.4.4 Fault tolerance

Fault tolerance is a fundamental aspect of the functionality of the TDAQ system. In this area several improvements are still to be achieved, therefore it was not considered to be appropriate yet to carry out a series of systematic tests in order to assess the performance of the system in case of errors. In general an error which is classified as WARNING, does not cause any disruption in the system, except for the possible loss of some event data. Examples of such errors have been observed in the testbeam setup, in case that a ROD does not sent consecutive L1IDs to the ROS, or that an SFI does not receive the requested data of an event on time. On the other hand a FATAL error in one application, which prevents it to continue taking data has potentially a fatal effect on the overall TDAQ. There are components which are unique in the system, like the DFM or the ROI Builder, which are necessary to take data and whose failure is considered not recoverable for the whole TDAQ. By design the failure of one or more L2PUs and SFIs can be recovered, since these components can be dynamically masked off by the L2SV and the DFM respectively. The failure of a ROS requires dynamic reconfiguration of the downstream data taking chain which is not possible at the moment. Similarly the mechanism to dynamically mask off a single ROL in a ROS is not in place.

14.4.5 Conclusions and outlook

The functional performance of the TDAQ system has been tested during the development of the system and during its exploitation in a testbeam set-up. Several aspects of the functionality are already mature enough to allow the present prototype implementation to be used in testbeam setups and for carrying out performance measurements. The aspects which require further analysis are related to the dynamic reconfiguration of the TDAQ during data taking, where this reconfiguration can be required due to changes in the run conditions or due to the occurrence of errors. In particular it is essential that those components that are not unique in the system can be dynamically excluded from a running system. The reinsertion of such components without stopping data taking may not be possible due to synchronization issues, but should also be studied carefully.

14.5 Modelling results

14.5.1 Paper model

Estimates of average message frequencies, data volumes and the amount of processing power required have been made with the help of the “paper model”. The most important results have been presented in ***Chapter 2. In Appendix A*** a description of the model and further results can be found.

With the help of the paper model a direct comparison may be made between sequential and non-sequential processing in terms of the quantity of data to be transported and of the processing resources needed to execute the respective trigger algorithms (non-sequential processing refers to the scenario in which all data that possibly could be needed is requested and processed). In particular the time consuming analysis of the inner tracker data (see Appendix A) is less frequently required for sequential processing. This results in sequential processing being considerably faster than non-sequential for the current processing time estimates, see Table 14-2. The table also shows the increases in RoI request rates and amount of data transport if non-sequential processing is assumed. Another benefit of sequential processing is a shorter average decision time than that resulting from non-sequential processing.

Table 14-2 The relative increase in request rates, Lvl2 data volume and size of the LVL2 farm if non-sequential instead of sequential processing is used.

	Low luminosity	Design Luminosity
LVL2 total request rate (for data from all ROLs)	1.8	2.0
LVL2 total data volume	1.4	2.1
Number of L2PUs	3.2	4.3

14.5.2 Computer model

Because a full scale testbed is not feasible only simulation with a 'computer model' can provide information on the dynamic behaviour of the full system. Computer models have been developed to get answers to very basic and fundamental questions like throughput and latency distribution of the LVL2 and EB subsystems when operating together, queue development in various places in the system (switches and end-nodes) and to study the impact of various traffic shaping and load balancing schemes. Also the interaction between multiple instances of the LVL2 subsystems and between multiple instances of the EB subsystems has been studied.

Computer models of small test set-ups have been developed and have been used for characterizing the behaviour of system components. Also models of testbeds and of the full system have been developed. The type of simulation used for the computer models is known as 'discrete event simulation'. Basically the simulation program maintains a time-ordered list of 'events', i.e. points in time at which the simulated system changes state in a way implied by the type of 'event' occurring. Only at the time of occurrence of an event the modelled system is allowed to change its state, in most cases only a small part of the state of the simulated system needs to be updated. The state change can result in the generation of new events for a later time, which are entered at the correct position in the list. The simulation program executes a loop in which the earliest event is fetched from the event list and subsequently handled.

The model of the trigger/DAQ system implemented in the simulation programs is an object-oriented model, in which most objects represent hardware (e.g. switches, computer links, processing nodes), software (e.g. the operating system, Data Collection applications) or data items. The models of the hardware and software items were kept as simple as possible, but sufficiently detailed to reproduce the aspects of their behaviour relevant for the issues studied. Parameterized models of all Data Collection applications [*reference to the DC note*] and Ethernet switches [*reference to DC note on parameterization of switches*] have been developed. The calibration of the models of the Data Collection applications was determined by analysing time stamps. These were obtained with the help of code added to the Data Collection software (for this purpose a library based on access to the CPU registers was developed). The time stamps provided estimates on the time spent in various parts of the applications. The calibration obtained in this way was cross-checked with measurements performed in specialized setups with the application tested running at maximum rate. Parameterized models of the switches were obtained with the help of dedicated setups. In these setups use was made of hardware traffic generators. The aim was to find any possible limitations in the switches which may affect the performance required for the full ATLAS trigger/DAQ system. The process of identification of appropriate models and a corresponding set of parameters and of collection of the parameter values with the help of dedicated setups was iterative and interleaved with validation phases. In the validation phases larger setups were modelled. Discrepancies between results from modelling and measurements usually led to a modification in the model(s) and associated parameters and another calibration phase.

Two simulation programs have been used, the at2sim program and the Simdaq program. The at2sim program makes use of the general purpose simulation environment of the Ptolemy system. Ptolemy offers support for discrete event simulation and allows the implementation of object-oriented models. The Simdaq program is a dedicated C++ program, with the discrete event simulation mechanism a part of the program. The component models used in the at2sim program are based on testbed components and calibrated as described above. The component models in the Simdaq program are less specific.

14.5.3 Results of testbed models

Text needs further work

The measurements focused on the EB scalability with various setups with homo or heterogeneous sets of ROB emulators have been modeled. In Figure 14-8 a comparison between results from measurements and model predictions for EB rates as a function of number of SFIs collecting data is presented. Three setups were investigated: with only 125 FPGA ROB emulators (each FPGA ROB emulator substituted 13 ROBs), with only 8 Alteon ROB emulators (each Alteon ROB emulator substituted 75 ROBs) and the setup with mixture of the two types of emulators: 1000 ROBs were emulated by 125 FPGA emulators (each FPGA emulator substituted 8 ROBs) and remaining 600 ROBs were emulated by 8 Alteons (each Alteon emulator substituted 75 ROBs). With each measurement line there is a correlated line obtained from modeling. The three setups show different saturation rates as a result of either emulators' performance or the setup arrangement. For a small number of SFIs the EB rate increases by 30 Hz each time a new SFI is added to the system (the maximal throughput of a single. The lowest EB maximal rate is observed for setup with 8 Alteons emulating 75 ROBs each. The Alteons internal processing time of incoming message is 40 µs and as each emulator has to process 200 requests for an event, this sets up the rate upper limit to 125 Hz. In the setup with FPGA emulators, the rate limitation is caused by the throughput of Gigabit Ethernet uplink connecting FPGA switch concentrated switch with the EB central switch. With 2.2 Mbyte of event data spread uniformly between FPGA emulators attached to the system via 4 concentrating switches, each concentrating switch's uplink has to deliver 2.2 Mbyte / 4 = 0.55 kbyte of data.

Assuming maximal payload which can be transferred in 1300 byte packets over the Gigabit Ethernet, to 105 Mbyte/s, it limits the EB rate to 191 Hz. The highest rate can be observed in setup with mixture of FPGA and Alteon emulators. The rate limit is set again by the throughput of the Gigabit uplinks between FPGA concentrating switches and the EB central switch. In this setup 1000 out of 1600 ROB is emulated by the FPGA devices thus they produce 1000/1600 * 2.2 Mbyte = 1.375 Mbyte of event data. This data is spread across FPGA emulators attached to 4 concentrating switches and requires that 0.344 Mbyte will be sent over the uplink. With useful payload on the Gigabit Ethernet of 105 Mbyte this sets up the limit rate to 305 Hz. In this setup the limit due to Alteons is higher, as each of the emulators should substitute 75 ROBs and with 40us of processing time of each request would limit at 333 Hz. In the setup with mixture of ROB emulators, the rate does not scale linearly for the number of SFI bigger than 4 and below saturation. This is result of temporary congestions of packets in the concentrating switch heading for uplink and is very sensitive to the traffic shape (or lack of it). The plot shows very good agreement between results from measurements and predictions from modeling. These results validate calibration of the model components (switches, SFIs, emulators).

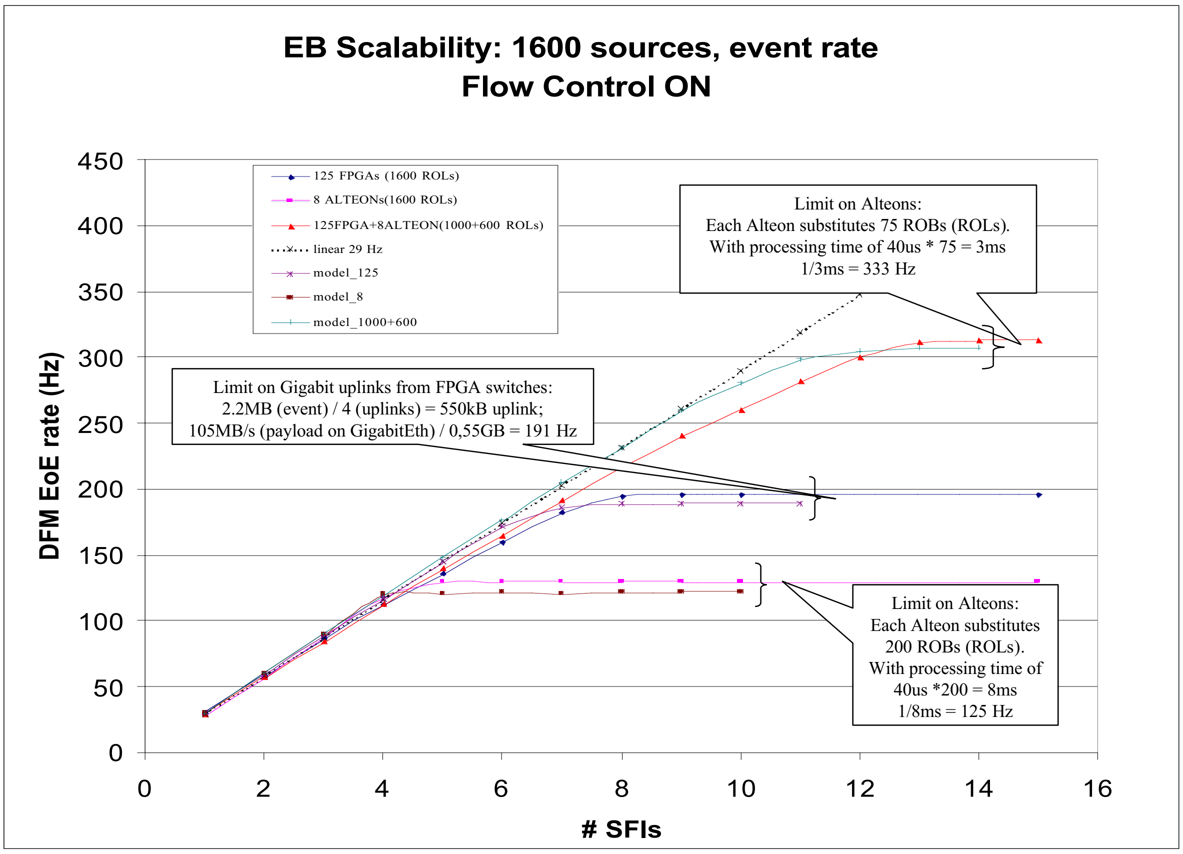

Figure 14-8 Comparison of measurement results and computer model results

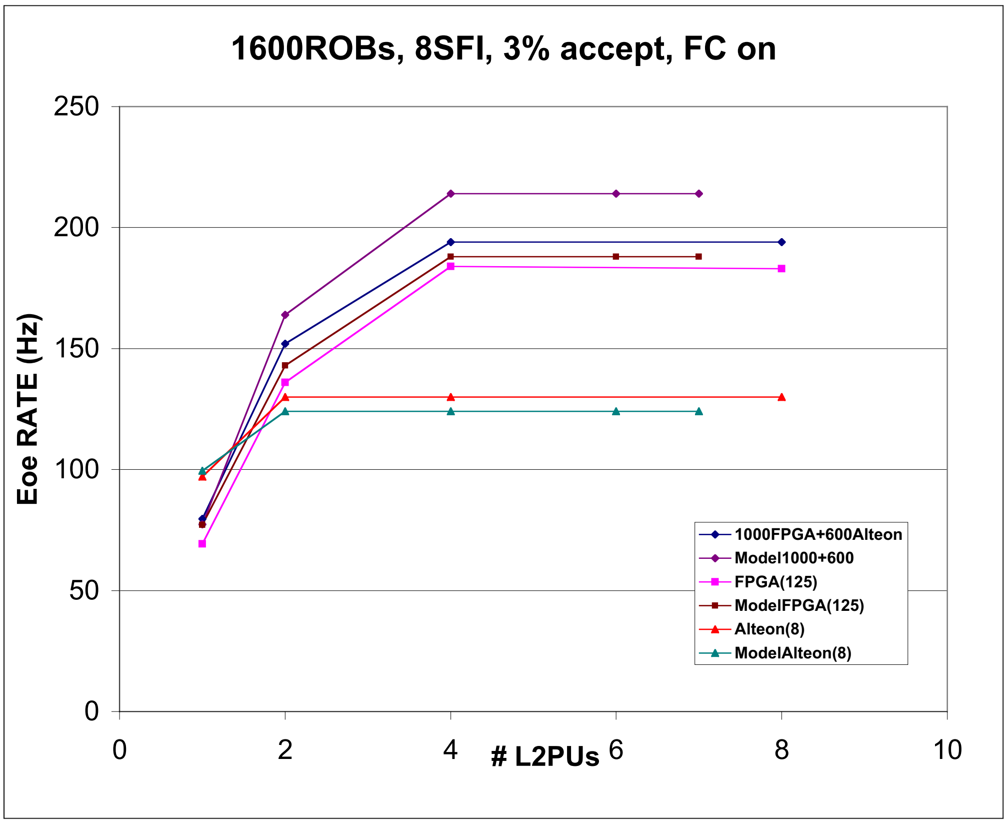

Figure 14-9

The testbed configurations aimed at demonstration of the full system operation with emulators of 1600 ROBs with an individual network access have been modelled, the results are presented in Figure 14-9. Models of the three different setups: 1000 ROBs emulated by FPGA and 600 ROBs emulated by Alteons; 1600 ROBs emulated by the Alteons only and 1600 ROBs emulated by the FPGAs only have been run to compare model predictions against the measurements. The model of the LVL2 system have been scaled by increasing the number of L2PUs. The EB subsystem was receiving 3% of the processed events accepted by the L2PUs and for those events 8 SFIs were used to collect data from the ROBs and to build events. Very good agreement has been obtained for the setups with either Alteons or FPGAs ROB emulators: . In both cases, the EoE rate saturates at the same level as for the EB scalability tests: either because of Alteon emulators internal processing or because of the concentrating switches uplinks where FPGA emulators are connected (see figure for the EB scaling). Less good agreement, however still within 10% tolerance, has been reached for the setup with heterogeneous set of ROB emulators. For this setup the EoE rate saturates due to the limited number of SFIs (8) and their performance.

14.5.4 Results of extrapolation of testbed model and identification of problem areas

The availability of network connections and switches with sufficient bandwidth and of a sufficient amount of computing resources in the DAQ and HLT systems is not sufficient to guarantee that the performance requirements are met. Also necessary are:

1. an even distribution of the computing load over the available computing resources,
2. minimal congestion and large enough data buffers in switches,
3. sufficient spare processor capacity and network bandwidth to cope with fluctuations.

The computer models of the full system can be used to identify possible problem areas.

====> *next part only partially valid*

The full system model implemented in at2sim is the ATLAS network-based architecture with 1564 ROBIns with individual network connections (ROBIns of which the data is not used in the LVL2 trigger have not been taken into account). The LVL2 subsystem is composed of 180 L2PUs connected to two Gigabit Ethernet LVL2 central switches in two groups of 90. The L2PUs are connected to the central switches via Gigabit Ethernet L2PU grouping switches with 7 L2PUs attached to the same switch. The EB subsystem is composed of 80 SFIs connected in two groups of 40 to two Gigabit Ethernet EB central switches. The SFIs are connected directly to the EB central switches. The L2Super, DFM, pROS are connected via a dedicated small Gigabit Ethernet switch to the 4 central switches. The ROBIns are connected via concentrating switches with 4 links to the central switches. The ROBIns were grouped as in the full size ATLAS: groups of ROBIns from a subdetector connected to a number of concentrating switches. The number of concentrating switches per subdetector depends on the average fragment size produced by ROBs from a given subdetector (calculations are presented in the network-based architecture note). In total there were 46 concentrating switches. Results were obtained for the following configurations:

1. ‘individual ROBIn FE’: each ROBIn has a single Fast Ethernet connection to a concentrating switch, this reflects the configuration in the 10% testbed with the BATM T5Compact switch
2. ‘individual ROBIn GE’: each ROBIn has a single Gigabit Ethernet connection to a concentrating switch,
3. ‘2 ROBIns aggregate’, ‘4 ROBIns aggregate’, ‘6 ROBIns aggregate’: two, four or six ROBIns are assumed to share a single network connection, a single request produces a response with a size two, four or six times larger than the response of a single ROBIn, the number of ROBIns connected to a concentrating switch is a factor of two, four or six smaller than for individually connected ROBIns

The traffic generated in the model resembles the traffic in the final system: the LVL2 subsystem was running at event rate of 75 kHz and the EB subsystem at a rate of 3 kHz. The L2PUs were making only one step: for each event data from 10 randomly chosen ECAL ROBs was requested and a decision was produced and sent to the Supervisor. The acceptance factor chosen resulted in 3 kHz event building rate). The L2PUs were not calibrated — they were used only to provide the LVL2 traffic and get the EB latency in the more realistic environment. The SFIs were requesting data from randomly addressed ROBs.

====> *end of partially valid text*

In the full system model implemented in Simdaq the same LVL1 trigger menus as used for the paper model are used to generate an appropriate number and type of RoIs for each event. In both models the eta and phi coordinates of the RoIs are chosen at random from the possible eta, phi coordinates (as defined by the LVL1 trigger). ROBIns are grouped together in groups of 12 in ROS units, each with two Gigabit Ethernet connections, one to a central LVL2 switch and one to a central EB switch. ROBIns of which the data is not used in the LVL2 trigger have not been taken into account. The mapping of the detector on the ROBIns is the same as for the paper model. Also the same processing times and LVL2 processing sequences are used. Average message rates and volumes and total CPU power utilized obtained from the paper and the computer model of the full system therefore should be equal within the statistical errors. The component models (processors, switches) however have not been calibrated as was done for at2sim. The main use of Simdaq therefore is for studying general trends in the behaviour of the full system with respect to building up of queues and of effects of possible choices for processor allocation strategies. ***Also switch-based read-out, system configurations in Figure 14-13 and Figure 14-14, further text to be added***

14.5.4.1 Load balancing

An even distribution of the computing load can be achieved by means of a suitable strategy for assigning events to the L2PUs or SFIs. For example a simple and effective strategy consists of the LVL2 supervisor or DFM maintaining a record of how many events are being handled by each L2PU or SFI. As the supervisor and DFM are notified when processing is finished this should be straightforward to implement. A new event can then be assigned to the L2PU or SFI with the smallest number of events to process (‘least-queued assignment’). Simulations of the LVL2 system have shown this to be a very effective strategy, with which high average loads of the L2PUs are possible. See Figure 14-10 for results for full system with 500 dual-CPU L2PUs utilized at 77 %. ***More text to be added***

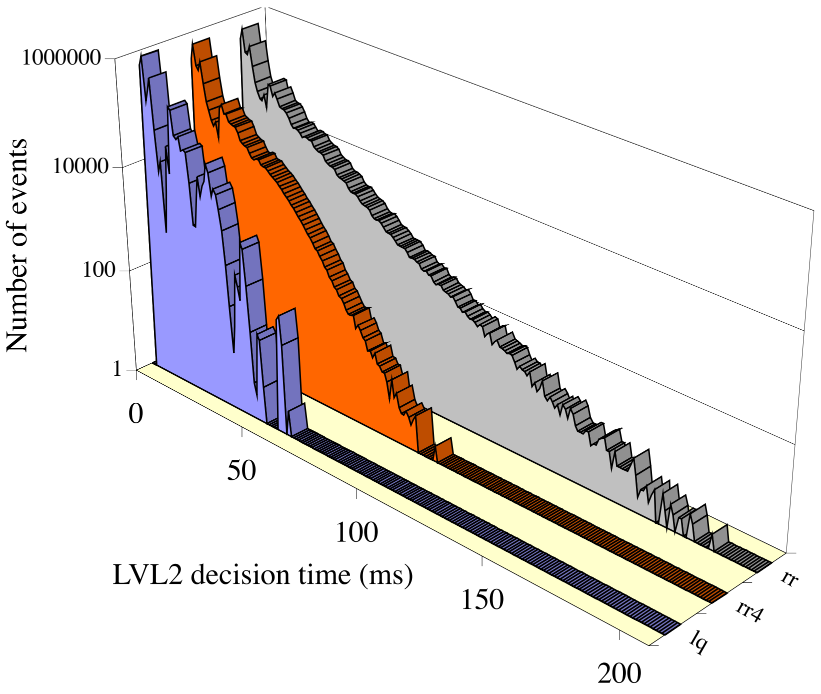

Figure 14-10 LVL2 decision time for bus-based ROS units, for round-robin assignment (rr), round-robin assignment of at max. 4 events to the same L2PU (rr4) and least-queued assignment. The results for switch-based read-out are almost identical.

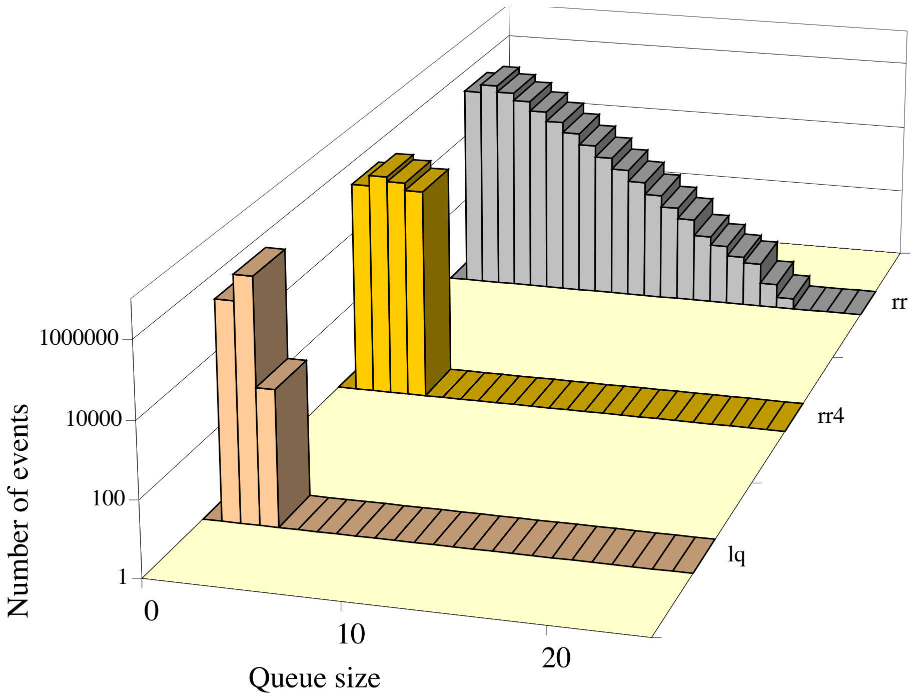

Figure 14-11 L2SV queues for round-robin assignment (rr), round-robin assignment of at max. 4 events to the same L2PU (rr4) and least-queued assignment. The results for switch-based read-out are almost identical.

14.5.4.2 Congestion in switches and buffer sizes

The effect of the credit based event building traffic shaping on the event building latency and queue build-up has been investigated with the configuration modelled in at2sim. The latency plot in Figure 14-12 shows that increasing the number of credits per SFI above 10 does not improve the latency for event building except for the ‘individual ROBIn FE’ configuration. For more than 10 credits the latency for this configuration will still depend on the Fast Ethernet link transfer time, as it is unlikely that all 10 requested ROBIns will reply at the same time and the replies will form a queue in the concentrating switch for the uplink to the central switch. The shorter latency for setups with ROBIn aggregates with respect to individually accessed ROBIns is explained by the smaller number of requests to be generated per event. The CPU time for receiving replies scales with the number of frames received. The latter scales approximately with the number of ROBIns. The CPU time spent on generating requests scales with the number of ROBIn aggregates. Relative to the time required for receiving the replies the SFI can therefore exhaust the credits assigned faster than for individually connected ROBIns (the CPU time is shared between the process receiving replies and the process generating requests). This also causes longer queues in the central switch, as can be seen in Figure 14-12.

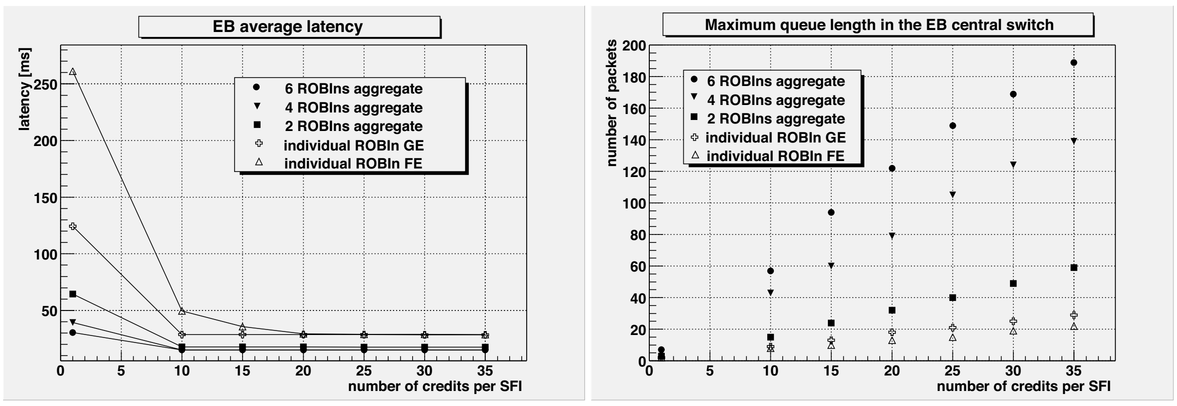

Figure 14-12 Average event building latency and maximum queue length in the EB central switches for different ROBIn configurations obtained with the at2sim full system model

Following figures for simdaq results. Explaining text to be added

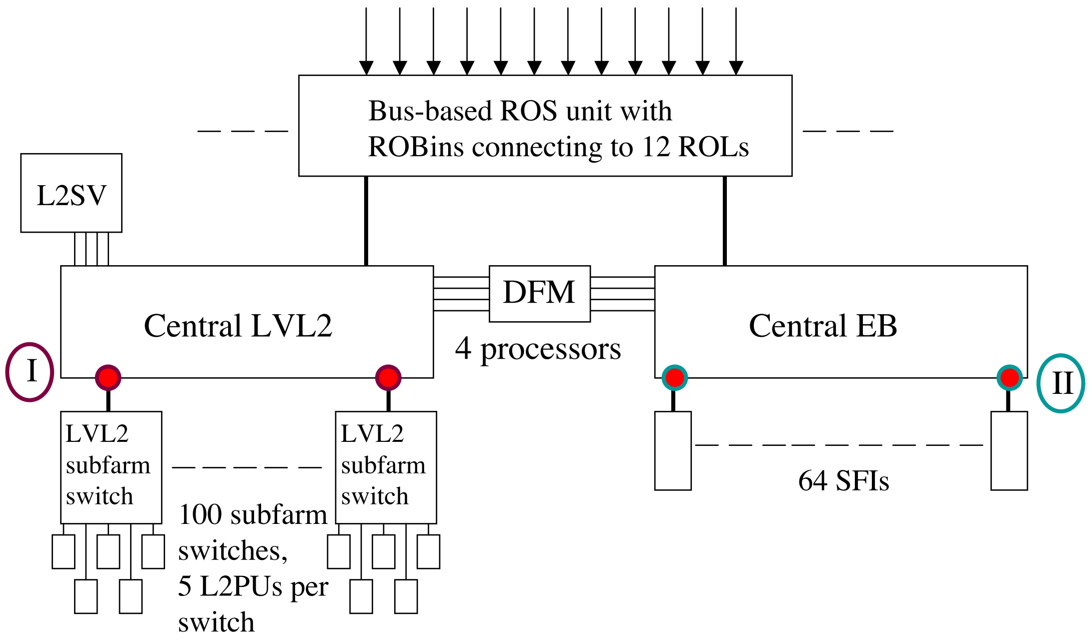

Figure 14-13 Schematic representation for computer model of system with bus-based ROS units. Possible 'hot' spots are indicated.

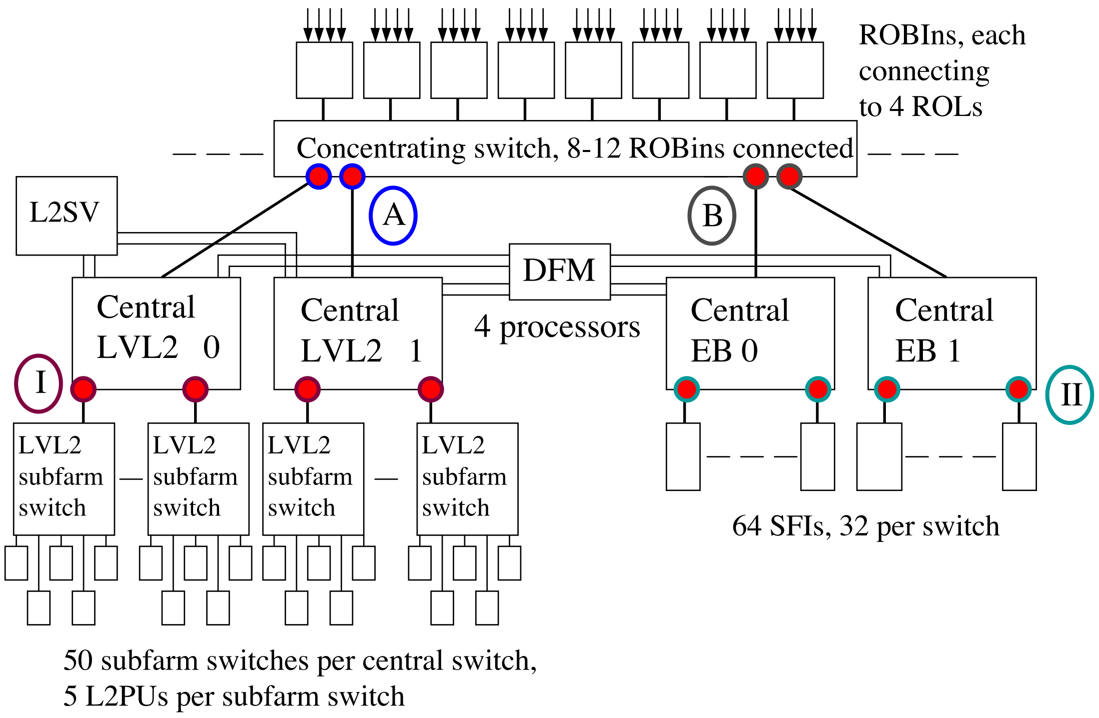

Figure 14-14 Schematic representation for computer model of system with ROS Ins directly connected to network. Possible 'hot' spots are indicated

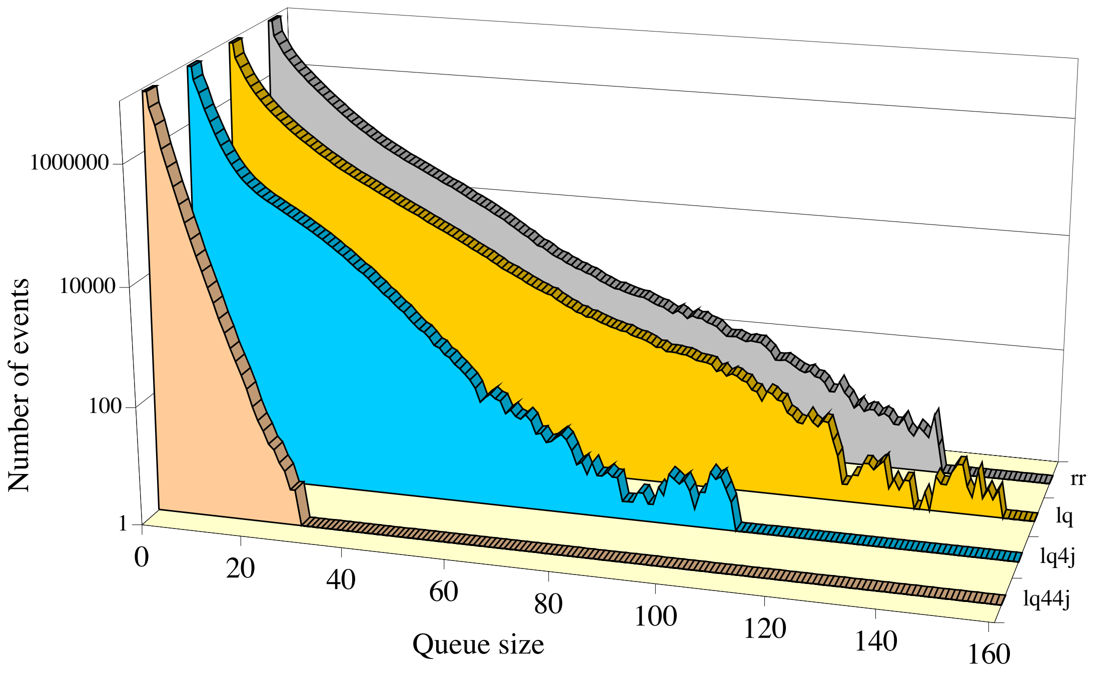

Figure 14-15 Queue sizes at point I, bus-based read-out, ROS units have dedicated link for LVL2 traffic and can send multi-frame messages as response to a single request, one request per ROS unit. Requiring number of outstanding requests to be smaller than 4 (“lq44j”) drastically reduces tail of queue. Assigning subsequent events to L2PUs connected to different subfarm switches (”lq4j”) also has a beneficial effect.

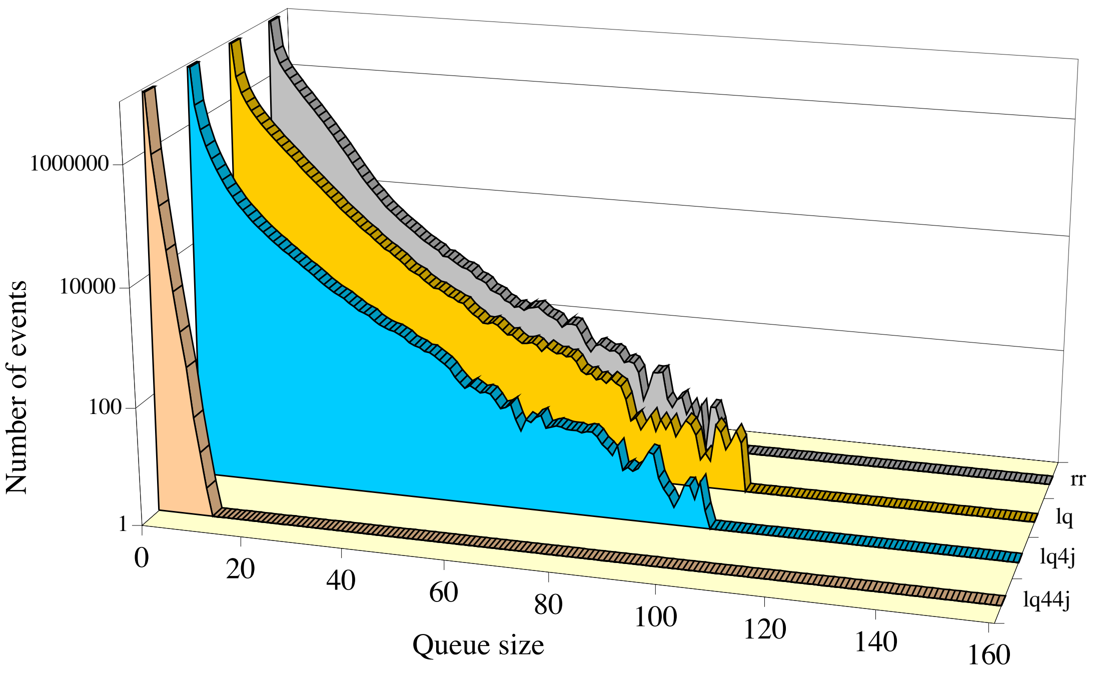

Figure 14-16 Queue size at point I, switch-based read-out, ROBIns servicing 4 ROLs have one network connection for EB and LVL2 traffic, requests only for data from single ROL, response to requests is a single frame message, this explains difference with previous figure.

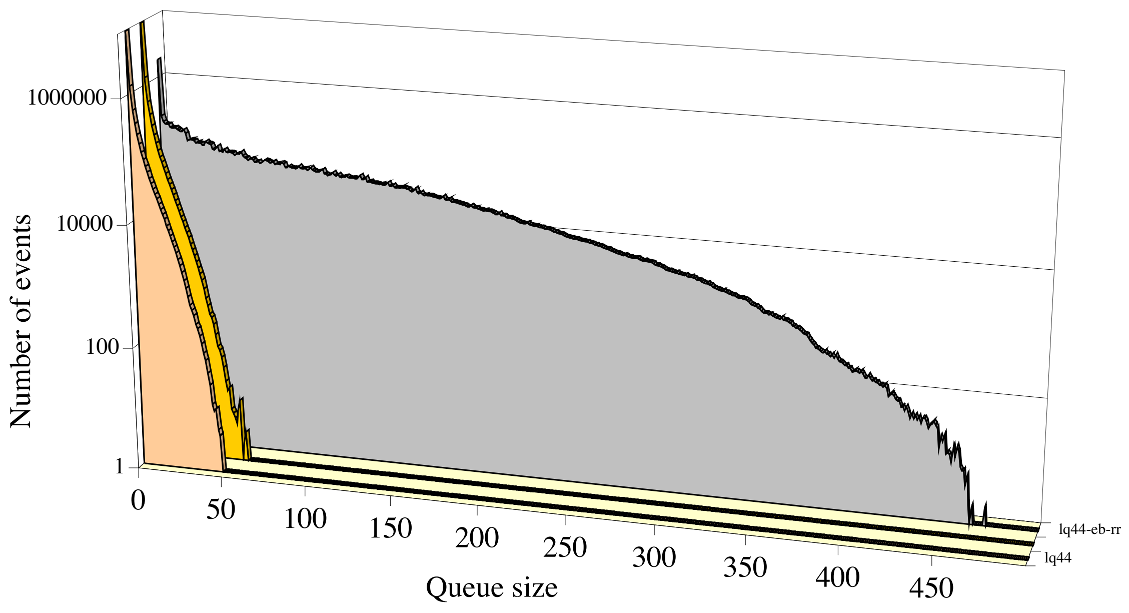

Figure 14-17 Queues in point B for Pixels, switch-based read-out. Disitribution with long tail for round-robin requesting of data by SFIs, tail disappears if nearly simultaneous requesting via same switch is avoided.

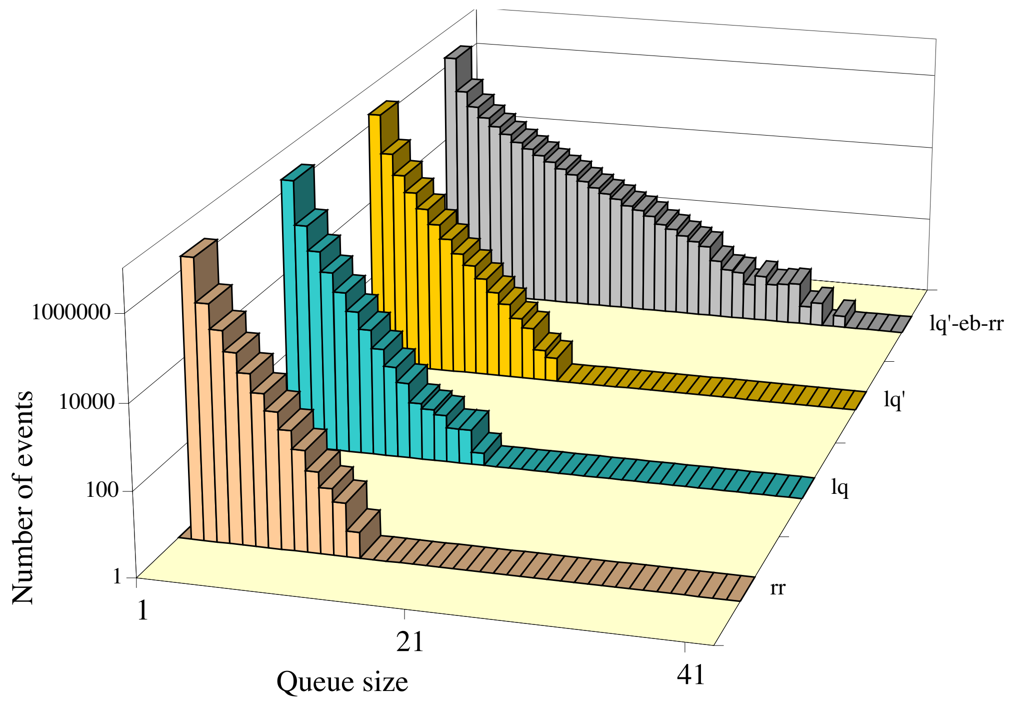

Figure 14-18 Queues in point A, switch-based read-out, Pixels, EB request pattern has some influence on queuing of LVL2 fragments.

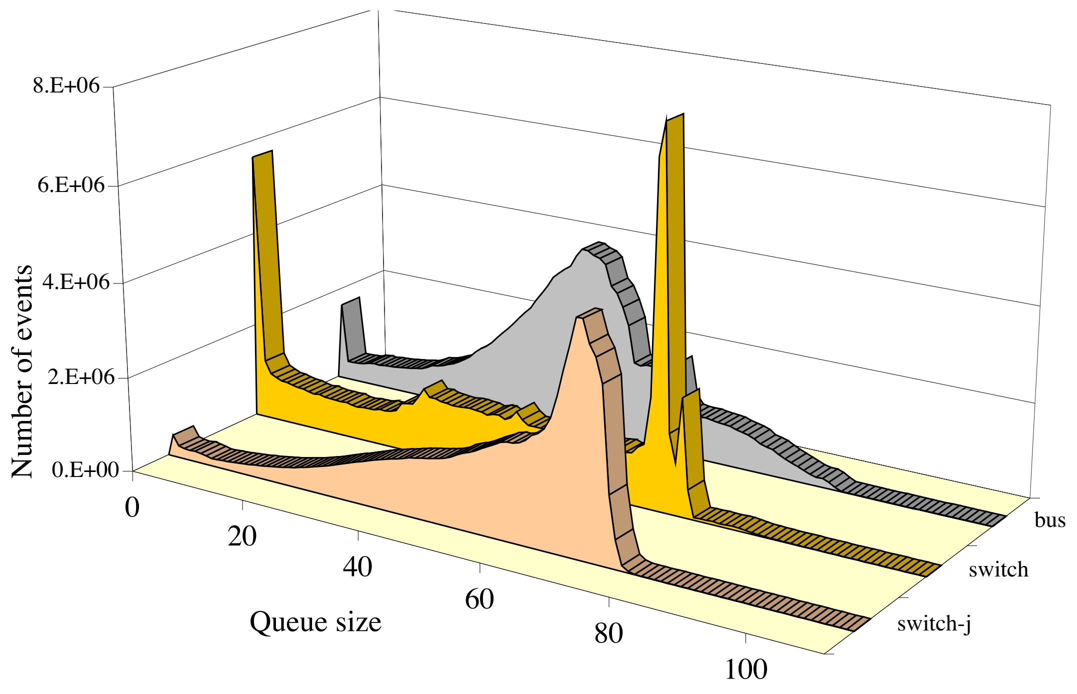

Figure 14-19 Queues in point II, switch and bus-based. “switch-j” refers to requesting data from the first ROBIn connected to the first concentrating switch, then the first ROBin connected to the second switch, etc. and next requesting data from the second ROBin connected to the first concentrating switch etc. Note linear scale.

Conclusion: request patterns and credit based request schemes can control queues.

14.6 Technology tracking

14.6.1 Status and Prospects

The ATLAS TDAQ system is to a large extent comprised of off the shelf commodity equipment; personal computers (PCs) and Ethernet links and switches; the exceptions being the RoI builder and ROBIns where specialized equipment has had to be developed. The technical evolution of commodity computing, communications equipment, as well as pricing, is therefore a significant element in the performance, costing and life cycle of the TDAQ system.

Impressive price and performance improvements have occurred over the last two decades. In this section we consider the prospects over the next decade, a period which covers the run up to the commissioning of ATLAS and the first few years of running.

14.6.1.1 The personal computer market

Moore's Law, the doubling of the number of transistors on a chip every couple of years, has been maintained over three decades, and still holds true today. Intel expects that it will continue at least through the end of this decade. In practice Moore's law has resulted in a doubling of PC performance about every two years, where performance can be quantified in terms of the clock speed of Intel's high end microprocessor chips. The computer industry has offered increasing performance at a more or less constant unit price.

For the future it seems that technically, on the time scale of ATLAS, Moore's law will continue. The only blip on the horizon being economic: the turndown in the world economy and an unwillingness to invest the large sums of money required to deliver new generations of microprocessors.

The current performance of PC based components and systems in the ATLAS TDAQ are based on 2 GHz PCs. In estimating the performance of the system we have assumed the use of 4 GHz PCs. This is a conservative estimate. In practice the processing power needed for the LVL2 and event filter farms will be purchased in stages and will therefore be able to profit from still higher processor clock speeds. This will be particularly true for the event farms where the processing time will be long compared to the I/O time. Components in the system with high I/O requirements will also benefit from improvements in processor performance but will be more bounded by link speed. Figure 14-20 shows the performance of the DFM as a function of processor clock speed.

14.6.1.2 Operating systems

The Linux operating system has evolved rapidly in the last years. Many commercial companies have invested heavily in improving the operating system (IBM, HP, Sun). Currently the main developments are in the areas of user interfaces and high-performance computing. ATLAS can clearly benefit from the Linux developments in the high-performance computing area. Such improvements include better support for multiple processors, better multi-threading and improved networking support. Practically all the new developments toward high-throughput transmission protocols over long-haul links were first implemented under Linux. Long-term, the optimization of the operating system will continue, fuelled by strong support from the aca-

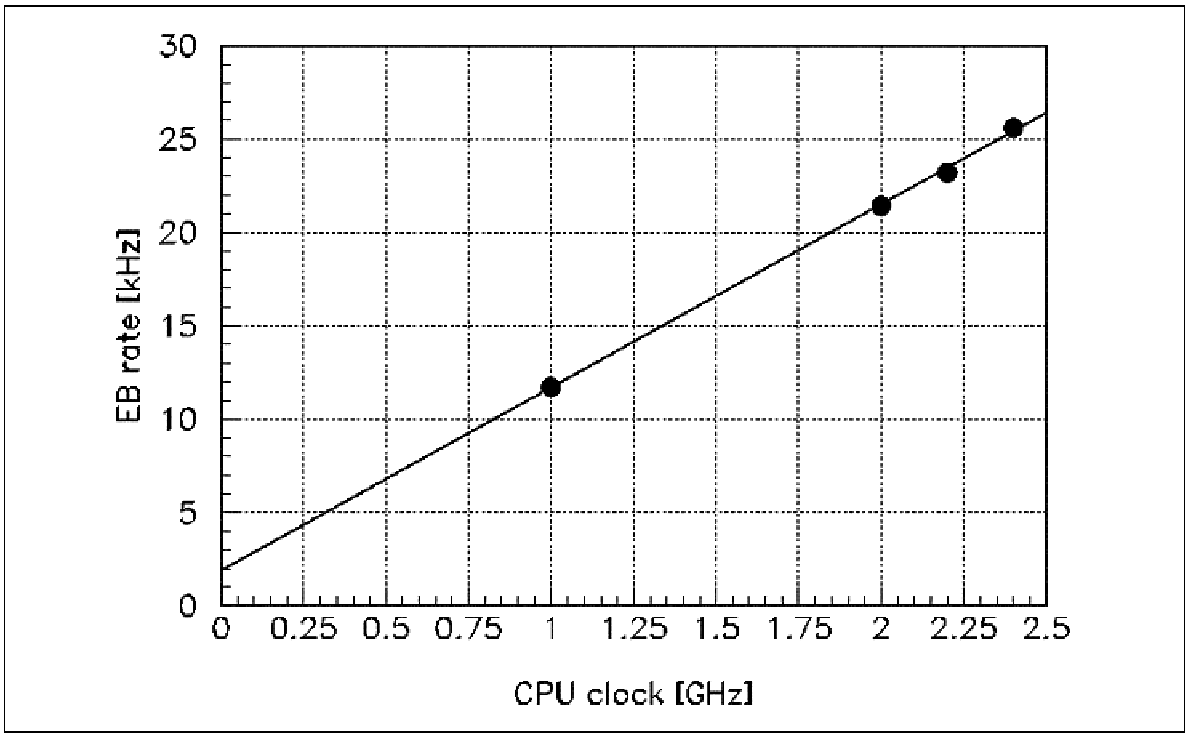

Figure 14-20 The performance of the DFM as a function of processor clock speed.

demic and commercial worlds. The wide-spread usage in universities means that ATLAS will have access to qualified Linux professionals throughout the life of the experiment.

14.6.1.3 PC Buses

I think we have to put in something about the future of PCI bus.

14.6.1.4 Networking

The ATLAS baseline system uses Ethernet network technology for RoI collection and event building, as well as data distribution to the event farms. It is also used in other networks associated with control and monitoring. Ethernet is, throughout the world, the dominant local area network (LAN) technology. It has evolved from the original IEEE standard, based on a 10 Mbit/s shared medium, to today's point to point links running at speeds of up to 10 Gbit/s [14-14].

The price of Ethernet technology has followed a strong downward trend driven by high levels of competition in a mass market. Figure 14-21 shows the price of 100 Mbit/s (FE) and 1 Gbit/s Ethernet (GE) network interface cards and switch ports as a function of time. Most PCs are now delivered with a GE controller integrated on the motherboard, making the connection essentially free. Further price drops will certainly occur for GE switch ports, in line with what has happened earlier with FE. This trend is coupled to the increasing provision of a GE connection in all PCs.

The proposed baseline system can be built using Ethernet switches available today. Even the most demanding components in the system, the large central GE switches in the data collection system, are comfortably within today’s norm. The prognosis for using Ethernet is therefore ex-

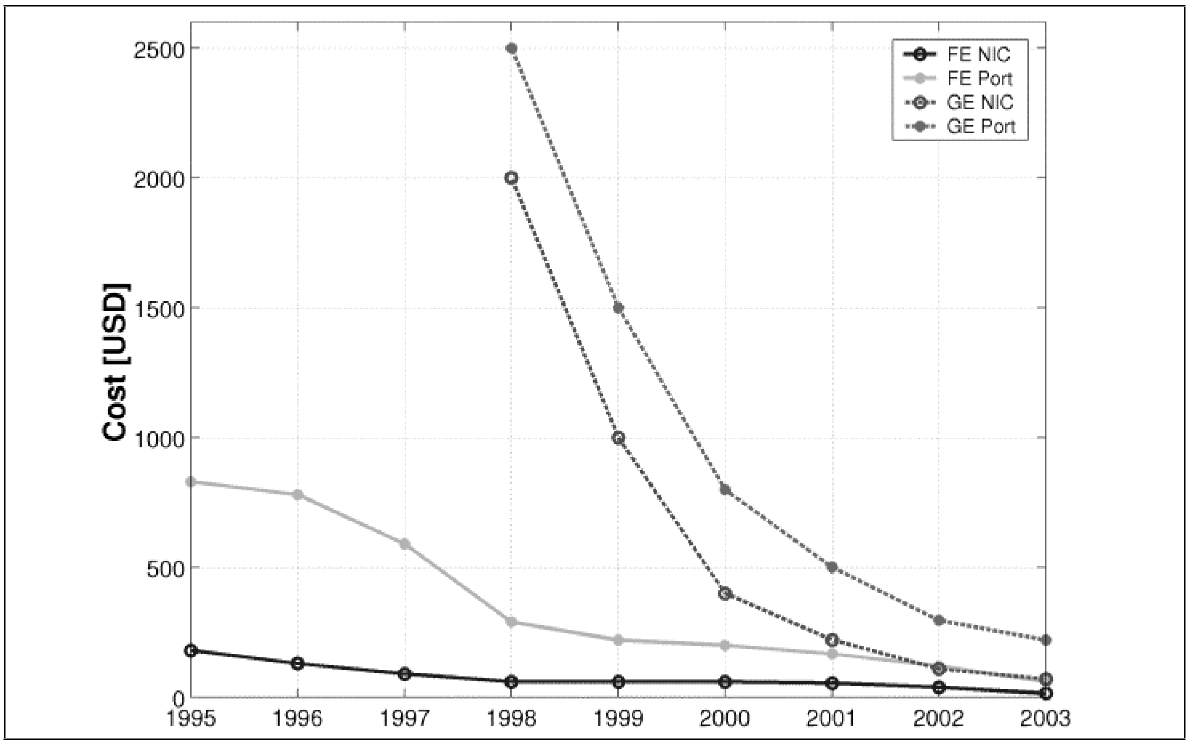

Figure 14-21 The evolution of the cost for Fast and Gigabit Ethernet switch ports and network interface cards.

cellent. It is a very widely supported international standard, which meets and even exceeds our foreseeable need and will certainly have a lifetime surpassing that of ATLAS.

Consideration is being given to the use of off site computing capacity to process ATLAS events in real time. Tests made recently have shown the technical feasibility of error free Gbps transmission between CERN and NBI, Copenhagen over the GEANT pan European backbone network [14-15]. Figures 14-22 and 14-23 show the setup used and some of the results obtained.

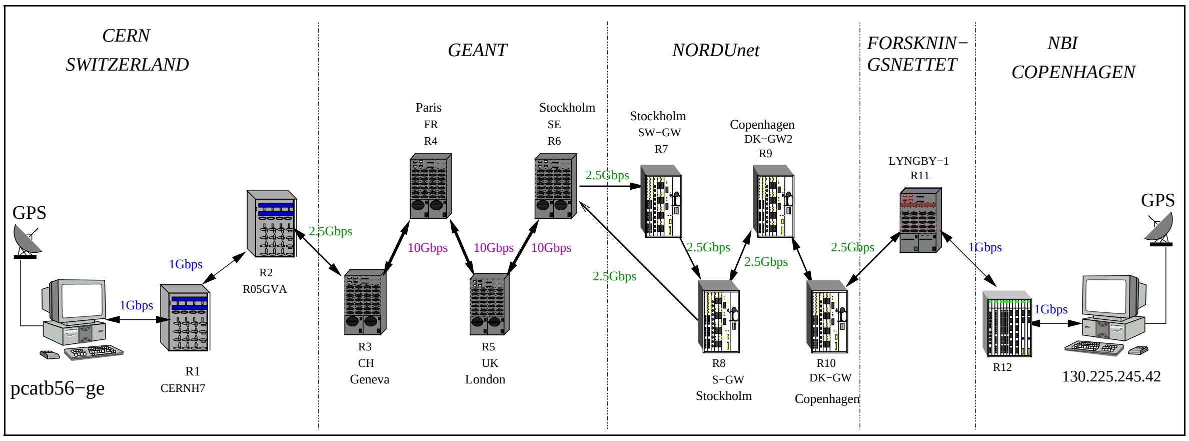

Figure 14-22 The network infrastructure between CERN and NBI (Copenhagen) over which Gbps tests have been carried out.

For the future, it appears technically feasible to export events at high rates from CERN to centres in member states, for processing in real time. It is within this context that 10 GE may have an important role to play. However, the use of such a scheme will ultimately depend on the eco-

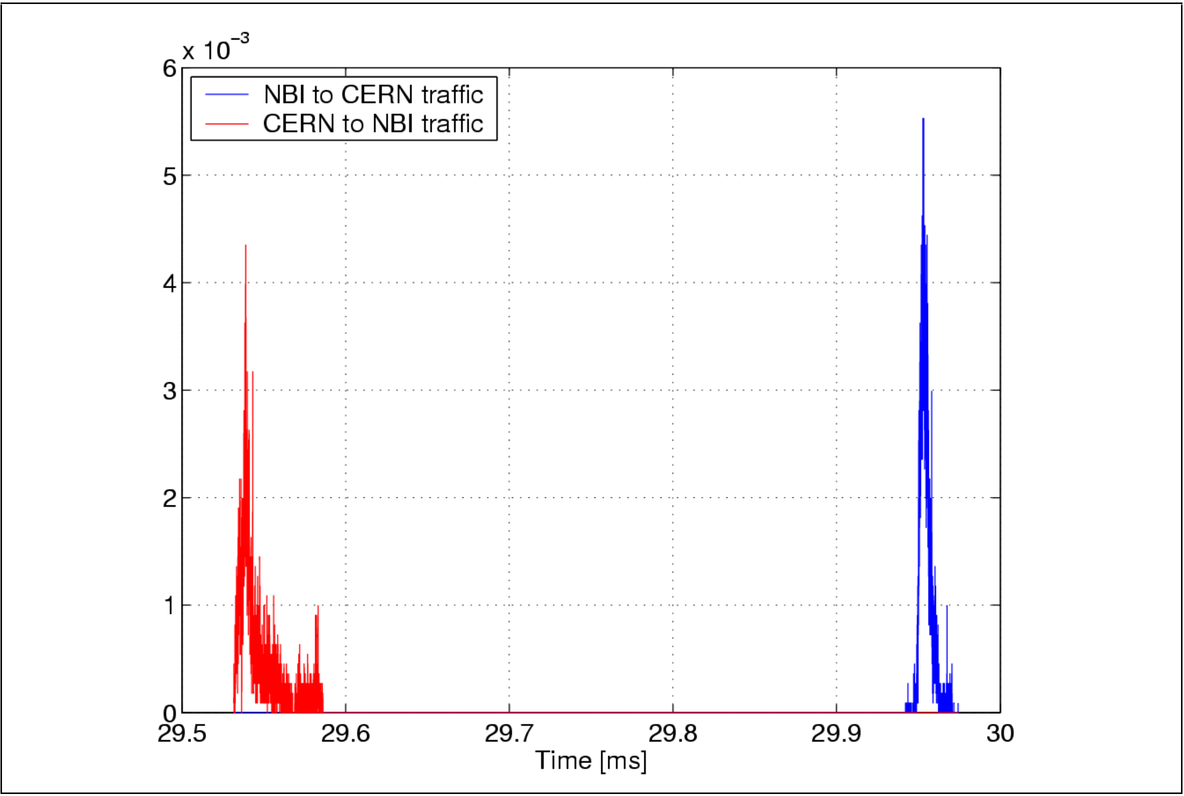

Figure 14-23 Packet latency measured between CERN and NBI (Copenhagen) at a 1 Gbps transmission rate.

nomics of long haul telecommunications. This factor, as well as technical considerations and practical testing, are part of our on going program of work.

14.7 Implication of staging scenarios

Re-interpretation of performance numbers for staging scenarios

14.8 Conclusions

14.9 References

14-1 LVL2 vertical slice results

14-2 “The use of Gaudi in the LVL2 trigger”, S.Gonzalez, A.Radu, W.Wiedenmann, ATL-DAQ-2002-012 (January 2002)

14-3 Athena validation note

14-4 “Initial LVL2 tests with the SiTree algorithm”, J.T.M.Baines *et al.*, ATL-DH-TN-001 (March 2003)

14-5 "A data manager for the ATLAS HLT", J.T.M.Baines, W.Li, ATL-DAQ-COM-2003-021 (June 2003)

14-6 Zero-suppression in calo note

14-7 SiTRee reference

14-8 "The implementation of the muFast algorithm in the new PESA framework", A.Di Mattia, ATL-COM-DAQ-2003-024

14-9 "Definition of Raw Data Objects for the MDT chambers of the muon spectrometer", K.Assamagan *et.al.*, ATL-COM-MUON-020

14-10 "Raw Data Object definition for the RPC chambers of the ATLAS muon spectrometer", K.Assamagan *et.al.*, ATL-COM-MUON-019

14-11 EF vertical slice results

14-12 Web histogram services

14-13 HLT vertical slice results

14-14 Dobinson et al RT2001 Lyon

14-15 Santiago di Compostella, RT2003

14-16