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'). Specialized functionality tests of the whole DAQ chain as well as the experience gained from using it for real data taking in the H8 testbeam[1] are also presented. Computer models have been used for analysing measurement results from the 10% testbed, and then to validate the models of components which were calibrated using measurement results from small test setups. Full system models have provided insight into 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.

14.2 High-Level Trigger Prototypes

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.

14.2.1 Scope of measurement and validation studies

The scope of the work reported here is limited to a system with full event selection and minimal Data Flow capability, 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

1. Large scale and performance tests of the Online Software are discussed in Chapter 10. The aim of these tests was to verify the overall functionality of the Online Software system on a scale similar to that of the final ATLAS installation and to study the system performance aspects in detail.

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 event-building 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.

14.2.2 Event selection in a LVL2 prototype

14.2.2.1 Prototype and software configuration

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 in-

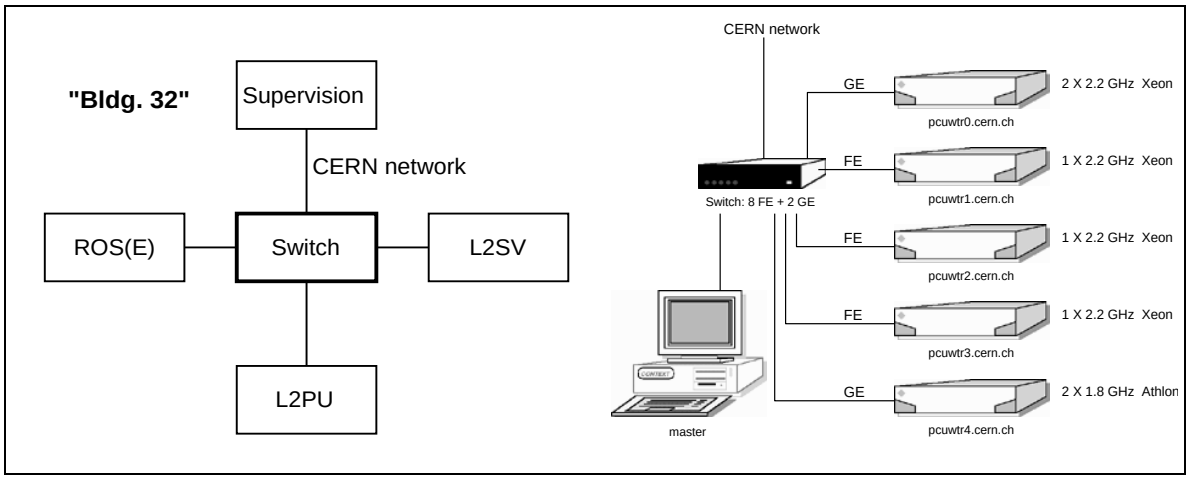

Figure 14-1 The setup for the LVL2 vertical slice testbed. The left figure shows the components in a typical three node configuration. The figure on the right shows the hardware configuration of the five-node LVL2 testbed.

cluded 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 Dual-processor Xeon (2.2 GHz), Athlon machines (1.8 GHz) or single processor Xeons (2.4 GHz). A detailed description of the set-up can be found in [14-1].

The L2PU application 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 interfaces the control aspect of the Data Flow and the ESS. The selection software used in the testbed comprised 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 difficulties and unresolved issues, can also be found in [14-2].

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. Files containing di-jet events at a luminosity of 2×10^{33} cm^{-2} s^{-1} were used (see Chapter 13) for the measurements. The events in the file were pre-selected to contain events that pass the LVL1 trigger. Each event contains an average of 1.06 electromagnetic RoIs. 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.

14.2.2.2 Measurements

After a first cycle of design (see 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.

14.2.2.2.1 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 event-by-event profiling 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 application 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).

14.2.2.2.2 Initial performance of the ESS in the LVL2 prototype

Initial measurements revealed that the LVL2 Calorimeter ESS alone required considerably more processing time than the ~10 ms per event average budget for the LVL2 trigger. Almost all of the processing time was consumed in the data conversion and preparation steps. This software converts the byte-stream data into objects that the downstream algorithms can process and applies calibration constants. It had only recently been made available and had not yet been optimized.

These first measurements also used the initial 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 3.7 ms. Given that a typical electromagnetic RoI spans 13 to 18 ROBs, this latency measurement agrees with the measurements presented in Chapter 8, where each ROB request with comparable payload introduces a latency of ~220 µs.

The LVL2 calorimeter algorithm mean execution time is 1.5 ms per event, with an additional ~1 ms consumed by framework and service overheads. The physics performance of the LVL2 calorimeter algorithm itself has already been documented elsewhere [14-4].

14.2.2.2.3 First performance improvements

After the above measurements were completed, an initial optimization of the ESS was made for a few critical components. Because the above measurements identified that data transfer and conversion dominated the processing time, work was performed to reduce these contributions. These studies are the first to investigate in detail data access performance.

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. After this change, the network access latency was reduced from 3.7 ms to 650 µs per event for the system configuration shown in Figure 14-1, where one ROS delivers all ROB fragments.

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 1.3 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 RoI in the prototype of 3.4 ms and 5.9 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

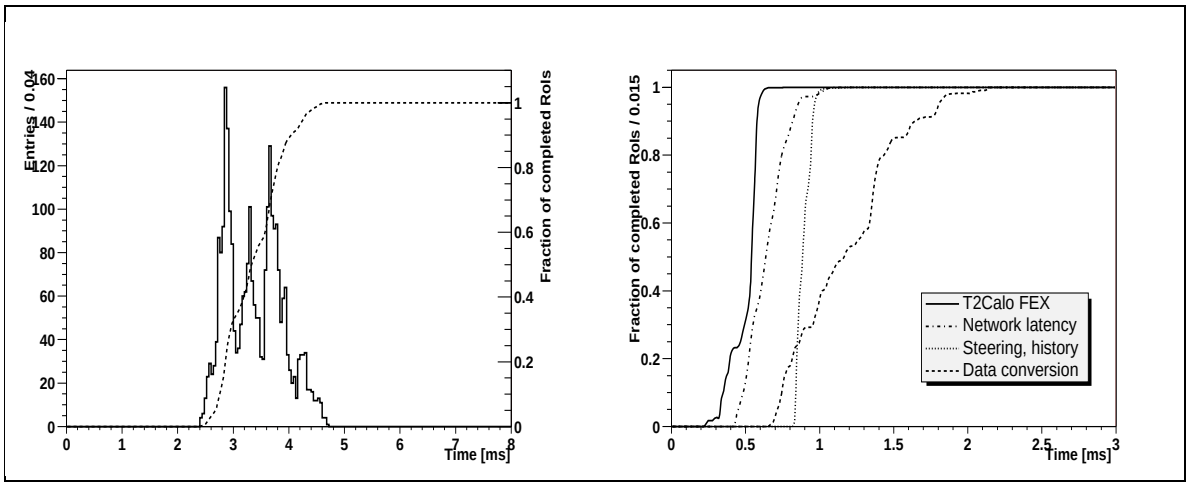

Figure 14-2 Total latency for RoI processing (shown at left) in the LVL2 LAr trigger for di-jet events at low luminosity. The dotted curve is the integral of the distribution, showing that 95% of the events are processed within 5 ms. The four main contributions to the latency are shown (right) as curves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

$\Delta\eta \times \Delta\phi = 0.3 \times 0.3$ 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 1.8 ms for 95% of the events.

The use of zero-suppression in the calorimeter [14-4] 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 310 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. The data conversion time could also be reduced if zero-suppression could be applied upstream of the L2PU (*e.g.*, in the RODs). More details on these measurements can be found in [14-1] and [14-5].

These results demonstrate that the system performance required can be achieved with current software, but that some of the offline components need optimization 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 already been achieved there. Thus, feedback from the re-use of offline software in the LVL2 trigger is mutually beneficial. These LVL2-motivated optimizations will benefit not only the LVL2, but the Event Filter and the offline software itself, reducing the overall computing resource needs of ATLAS.

14.2.2.2.4 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 266 Hz to 323 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). By partially repairing the inefficient parts of the software, an increase in the event throughput has been observed. 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 since 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 470 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 resolved.

All LVL2 testbed measurements quoted above have been done for the LAr calorimeter. At the time of writing, the initial implementation of the SCT / Pixel converters based on the Chapter 9 design was only just becoming available and had not been optimized for LVL2. In previous reference 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 [14-5]. This prototype included all software components necessary to map ROBs to an (η,ϕ) region, request the data [14-6], 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 result is significantly less than the ~30 ms required by the first implementation of the new SCT and Pixel converters. However, by implementing improvements similar to those applied to the LAr converter, the SCT and Pixel decoding software can be expected to achieve a similar level of performance to that of the early prototype.

Because it is executed at the LVL1 muon rate, the processing time of the LVL2 muon trigger, like the LVL2 calorimeter trigger, is also critical. Measurements [14-8] of LVL2 muon trigger 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 in the barrel at high luminosity yielded a mean total processing time of ~10 ms on a 2.4 GHz machine. This processing time been calculated taking into account the cavern background simulation in the muon spectrometer for high luminosity ($L = 1 \times 10^{34}$ cm^{-2} s^{-1}) with a safety factor of two (see Section 13.4.2). It is dominated by data preparation – 1 ms per RoI for the RPCs [14-9] and 7.9 ms per RoI for the MDTs [14-10]. The cavern background increases the MDT occupancy and the data preparation time scales linearly with the occupancy. The remaining 1.1 ms is 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 data flow and network overheads for the muon trigger are similar to those for the LAr trigger, the overall latency for the LVL2 muon trigger is well under control.

14.2.2.3 Conclusions and outlook

As seen in the previous section, detailed studies and first measurements with the HLT event selection software show that there is great potential for optimization of 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. 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 are expected to be available at LHC turn-on, gives total processing times that are compatible with the 10 ms average event processing time budget of LVL2. In all cases, the data preparation is proving to be the major fraction [14-3] of the LVL2 processing time. Thus detailed studies have been made, and will continue, of the data preparation process.

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. First, the implementation of the internal data access including conversion, calibration and data organization need to be optimally designed for execution speed. Secondly, the 'on-demand' nature of many of the offline configuration and data access services, particularly at initialization time, should be adapted to a deterministic model, more suitable for a trigger environment. Thirdly, fragments required for a given RoI must be requested in parallel. Implementing these optimizations brings very substantial performance improvements as was demonstrated above.

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

In conclusion, 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 offline 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.

14.2.3 Event selection in an Event Filter prototype

In the Event Filter, the event selection process is carried out by the Processing Task (PT), described in Section 9.3**.** The PT hosts the HLT selection software, which is fully implemented using ATHENA (the ATLAS offline software framework [14-11]) 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 data flow (EFD) and the PT running the event selection software (ESS) in a single system (see section 9.3.2.3 and [14-12]).

The performance of the EFD running with dummy PTs is described in Section 9.3.2.4. Here we concentrate on the performance involving real events processed by the ESS.

14.2.3.1 Integration of EFD with ATHENA data access services

The PTs are independent processes running on each EF node. In order to access event data, the event selection software running in the PTs are interfaced to the EFD which supplies them with complete events from the Event Builder. This interface has been implemented by adapting the ATHENA conversion services that carry out the conversion between different data types.

The integration of the ESS in the Event Filter consisted then of developing dedicated implementations of some of these data conversion services. These EF-specific implementations read, handle, and exchange event data with the EFD using a shared memory mechanism. When a PT requests an event from the EFD, a pointer to the event in this shared memory is returned to the PT (see Section 9.3.2). After the processing is completed, a trigger decision is sent to the EFD. If the event is accepted, the EFD appends to the original event, data generated during the EF processing.

14.2.3.2 Prototype and software configuration

Tests were performed by running the Event Filter with simulated events. The data set used was a sample of di-jet events that was pre-selected to pass the LVL1 electromagnetic trigger. The data sample is the same as that used for the LVL2 tests and is described in Section 14.2.2. The event size in this sample was ~3 Mbytes. The data were pre-loaded in an SFI emulator. At the time these tests were made, the EF algorithms were not yet available and so a LVL2 calorimeter algorithm was used. Tests using more sophisticated algorithms are planned in the next three months.

For the purposes of these tests, the EF result (data generated during processing) contained a single data fragment produced by a converter dedicated to serialising the results of EF reconstruction, which are in object form. No additional reconstructed objects were serialised in the prototype, however, the same mechanism will be used for multiple object serialization in the future when more complete EF algorithms are available for use in the testbed. For accepted events, this serialized fragment was written into the shared memory from where it was appended to the original event by the EFD and then sent to the SFO.

Three different hardware configurations were used to carry out the measurements:

1. A single-host dual-processor: one Intel Xeon 2.2 GHz processor with 1 Gbyte of RAM
2. A multiple host set-up: two Intel Xeon 2.2 GHz processors with 1 Gbyte of RAM interconnected by Fast Ethernet
3. A multiple host set-up: two Intel Xeon 2.2 GHz processors with 1 Gbyte of RAM interconnected by Gigabit Ethernet

14.2.3.3 Measurements

A series of tests was performed on each of the testbed configurations described above. The tests were:

1. Validation of the exchange of data between the EFD and the ATHENA PT hosting the ESS
2. Throughput measurements.

Validating the event input procedure consisted of checking the integrity of the data after passing it from the EFD to ATHENA, by checking that the ESS produced the same results as when running offline. In order to validate the output procedure, accepted events were sent by the EFD to the SFO and from there to a local disk file. The integrity of these data was then verified by reading, unpacking, and re-processing them offline. The throughput was measured as the average processing time per event (i.e., the inverse of the event rate).

Measurements were conducted with a dual-processor machine hosting all processes, SFI, SFO, EFD, and PTs, (configuration 1 above), and also on a multiple-host setup with the SFI and SFO running on one host and the EFD and PTs running on the other (configurations 2 and 3 above).

The ATHENA PT ran the calorimeter algorithm. The EF selection was configured so that all events were accepted. The total user time per event was on average 180 ms for the di-jet event sample described above, and the virtual memory size was typically 260 Mbytes. Table 14-1 summarises the results.

Table 14-1 Results of throughput measurements obtained with the various test configurations (see text for details)

Configuration	Number of ATHENA PTs	Time per event (1/Rate)	Data volume throughput	Remark
1	1	190 ms	16 Mbytes/s	
	2	110 ms	27 Mbytes/s	
	3	timeout		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	

In the first configuration, adding a second PT profits from the dual-CPU and increases the throughput significantly, though not by a factor of two as the other resident processes (SFI, SFO, and EFD) require some fraction of the CPU. Adding a third PT saturates the memory; consequently, swapping slows down the process considerably. Under these conditions, the PTs were blocked by the timeout mechanism of the EFD. In the second configuration, with multiple hosts interconnected by Fast Ethernet, the throughput was limited by the network bandwidth even with only one PT (in normal EF running conditions, with an event size of the order of 1 Mbyte and a latency of the order of 1s, Fast Ethernet bandwidth should be adequate). In the third configuration (multiple hosts interconnected by Gigabit Ethernet), there was no bandwidth limitation and a single PT used close to 100% of a CPU. Adding a second PT increased the throughput. Here, each PT used about 80% of a CPU, the remainder being used by the EFD.

14.2.3.4 Conclusions

The ATHENA-based event selection software was successfully integrated with the EFD, demonstrating that the EF selection can be built using offline components. As in the case of the LVL2, this approach minimizes development and maintenance costs, while providing a unified event selection chain.

Although the tests described here were not conducted with a full EF selection suite, the measurements already highlight some of the key performance issues. The behaviour of the ATHENA PT running the selection software in terms of processing time, memory usage and event sizes will be further evaluated in order to define an optimal Event Filter hardware configuration. These tests demonstrate the correlation of these parameters.

14.2.4 The HLT vertical slice

The LVL2 trigger and the EF were integrated in a single testbed as shown in Figure 14-3. The

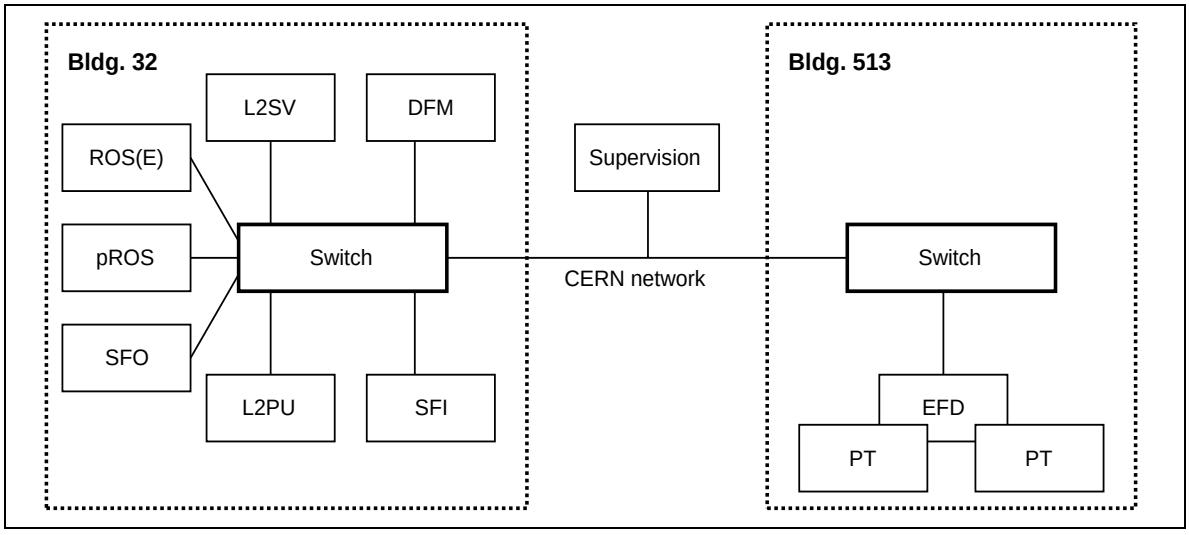

Figure 14-3 Setup for the combined LVL2 and EF vertical slice testbeds.

LVL2 slice (described in Section 14.2.2) and the EF slice (described in Section 14.2.3) were connected to form an 'HLT vertical slice' using the CERN network infrastructure. The DFM and the Event Builder were located geographically close to the event fragment sources. In order to pass the LVL2 result to the EF, one of the ROSs was configured as a pROS (described in Section 8.1.3.4). The entire system, consisting of 11 nodes, was configured and controlled by the Online Software through the CERN network.

During the test, one ROSE was pre-loaded with 120 LVL1-preselected di-jet events with electronic noise as used in the LVL2 slice tests. The energy threshold cut for the LVL2 calorimeter algorithm was set low (20 GeV) in order to accept approximately two thirds of the events. The LVL2 used the T2calo algorithm with optimised data conversion for the LAr Calorimeter (as described in Section 14.2.2). The LVL2 system was configured to execute the ESS in two L2PUs running in parallel. The EF used a non-optimised calorimeter algorithm (which included Tile Calorimeter handling) running in a sub-farm with three nodes, each with one EFD and two PTs. Events accepted by the LVL2 were assembled by the Event Builder (2-3 MBytes/event) and

passed on to the ATHENA-based ESS running in the EF. Finally, accepted events were recorded in byte-stream format by the SFO.

This functional integration was validated by verifying that the event recorded by the SFO was the same as the simulated event selected by the LVL2 trigger. Now that this HLT slice has been functionally demonstrated, it is planned to integrate the HLT slice with the 10% Data Flow testbed (see Section 14.4) in order to study performance aspects of the complete HLT/DAQ system.

14.3 HLT CPU Requirements

The HLT/DAQ architecture has been designed to handle a maximum LVL1 trigger rate of 100 kHz. The estimated CPU requirement of the HLT system to handle this LVL1 rate is summarized in this section.

For LVL2, there are several ingredients to this estimate:

- Examples of the feature extraction and reconstruction algorithm performance are presented in Chapter 13 and its associated references, and in Section 14.2.2. Typical timing numbers on a 2 GHz CPU are:
    - Calorimeter: ~2 ms
    - Muon: ~1 ms
    - SCT/Pixel: ~3 ms
    - TRT: ~9 ms
- The frequency of use of each of the algorithms — calculated from the trigger rates and acceptance factors (see Chapter 13 and Appendix A) and the number of RoIs per event for each RoI type
- The CPU requirement for the preparation of the data to be used by the algorithms

  Initial measurements have indicated that this will be a significant fraction of the total required CPU time. However, as discussed in Section 14.2.2.2, the measured data-preparation times are preliminary and significant improvements can be expected in most cases.
- The time to access data from the ROBs using the RoI mechanism in the LVL2 trigger

  This has been studied in detail and results are presented in Section 8.3.2.2. The data-access time is very small compared both to the algorithm and data-preparation times.
- The CPU overhead of the overall software framework in which the selection algorithms run

  This is estimated to be a few percent of the overall processing time per event.
- Additional, currently ill-defined CPU requirements, such as access to calibration data residing in the conditions databases and monitoring

  It is not foreseen for the LVL2 trigger to access the conditions databases during a run. However, the EF will need this ability. As the conditions database is currently in its design phase, we do not attempt any estimate of data-access times at this stage. This will be addressed further in the Computing TDR. System and local trigger monitoring procedures will certainly increase the overall CPU requirements of the HLT.

The above considerations and the sequential data processing steps (see Chapter 2 and Appendix A) have been used as input parameters to detailed modelling of the LVL2 system. In the model, typical timing numbers have been scaled to 8 GHz CPUs, the speed of CPUs expected to be available in 2007. The result of the model is that for a LVL1 rate of 25 kHz, ~250 CPUs at 8 GHz will be required for LVL2. Scaling this number to the maximum LVL1 rate of 100 kHz gives a total LVL2 CPU requirement of 500 dual-processor machines.

For the EF, performance and timing studies are still in progress. We therefore use a target figure of 1s / event for the average global EF processing time, assuming 8 GHz CPUs. Assuming an event-building rate of 3.2 kHz (corresponding to a LVL1 rate of 100 kHz) this gives an estimate for the EF of 1600 (8 GHz) dual-CPUs. These estimates for LVL2 and the EF are used in the overall HLT / DAQ-system costing presented in Chapter 16.

The resource estimates clearly are subject to large uncertainties coming from various sources:

- At the start-up luminosity of $2 \times 10^{33}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$, for the physics selections presented in Chapters 4 and 13, simulations give a LVL1 trigger rate of ~25 kHz without any safety factor and with a limited physics menu. Clearly, many possible selection channels have not been simulated which would add to the LVL1 rate. There are also very large uncertainties in the underlying cross-sections and the background conditions which will affect this estimate. Designing the system for 100 kHz gives an effective ‘safety factor’ of four compared to the above LVL1 rate.
- In extrapolating the LVL1 rate to 100 kHz, we have made the assumption that the mixture of trigger types remains constant.
- We have made extensive studies of trigger-algorithm data preparation as discussed in Chapter 13 and in Section 14.2.2. These results are still preliminary for several reasons since the detector data formats themselves are preliminary, and studies of data converters in the ESS have been done only with initial prototype-software not yet fully optimized for the trigger

  However we can already conclude that the data-preparation CPU requirement will be significant, and in some cases possibly dominant, compared to the algorithm execution time.
- The ESS is a first implementation of the design presented in this TDR. Work both on the trigger testbeds and in offline studies of the trigger algorithms shows that there are many improvements and optimizations which can be made.
- The ATLAS offline software, upon which much of the ESS is based, has undergone a complete re-design and implementation in the last three years and this process will continue, culminating in the Computing TDR in two years time. Many improvements and optimizations are expected in the offline software which will be directly transferred to the associated HLT components. The estimates presented above reflect the present performance of the ESS and the offline.

14.4 The 10% testbed

In order to study the combined performance of the components and subsystems of the ATLAS Data Flow system, a testbed with full Data Flow functionality and a size of approximately 10% of the final system has been assembled. Although the testbed is necessarily a scaled down ver-

sion of the final system, individual components are operated at rates similar to those expected in the final system.

The primary aim of the 10% testbed is to demonstrate the full and concurrent execution of the RoI data collection and Event Building functionality to check for possible interference between them, for instance, in the form of reduced performance. Measurements will also be performed on the test bed to study outstanding design and implementation issues, for example bus-based and switch-based readout and of the number of central switches. In addition measurements performed on the testbed are being used to validate and calibrate computer models of the system. The subsequent reproduction of the performance and scaling behaviour of the 10% testbed by modelling will strengthen conclusions drawn from modelling studies of full-size HLT/DAQ system.

As the 10% testbed has only been assembled and commissioned in the weeks preceding the submission of this report, only the results of preliminary measurements are reported here.

14.4.1 Description of the 10% testbed

The 10% testbed presently consists of a set of PCs and custom hardware devices, used to emulate ROSs, inter-connected via a Gigabit Ethernet network. The testbed, shown in Figure 14-4, reflects the architecture of the final system. It implements two, separate, central switches for RoI data collection and event building and allows for additional central switches to be added for the studies of scalability, for example two RoI collection switches and two event building switches. In addition, the testbed is such that two methods of accessing data buffered in ROBs may be studied the: bus-based ROS and switched-based ROS, see Section 5.5.4.

14.4.1.1 Readout subsystems in the 10% testbed

In the testbed the bus-based ROS, as described in Section 8.1.3.3, has been implemented on PCs (numbers 108, 114-116 in Figure 14-4). Each PC is equipped with two Gigabit Ethernet NICs connecting the ROS to each of the central switches. As described in Section 8.1.3.3, this version of the bus-based ROS emulates the interactions with the ROBins as the prototype ROBin (see Section 8.1.3.2) is as yet not installed. Additional bus-based ROSs are emulated in the testbed by sixteen programmable Alteon Gigabit Ethernet NICs (see labels ALTx in Figure 14-4). The Alteon NIC has only a single Gigabit Ethernet port, i.e. they cannot be simultaneously connected to both central switches. To overcome this limitation, in the testbed, a bus-based ROS is emulated by two Alteons, one connected to the LVL2 central switch and the other connected to the event building central switch, allowing the emulation of eight bus-based ROSs. Together the PCs and the Alteons provide twelve bus-based ROSs which is 10% of the number foreseen in the final system.

In the switched-based ROS scenario, each ROBin has its own connection to the central switches via a concentrating switch. Within the testbed two different types of emulators of ROBin are deployed, the FPGA ROBin emulators and the Alteon NICs. The FPGA ROBin emulators (FPGA #1 – FPGA #4 in Figure 14-4) has a single Fast Ethernet link to a concentrating switch (labelled T5C-GF in Figure 14-4). Thirty-two FPGA emulators are connected to a concentrating switch and each concentrating switch has two Gigabit Ethernet links, one to each of the central switches. There are 128 FPGA ROBin emulators. The Alteon NICs are as described above.

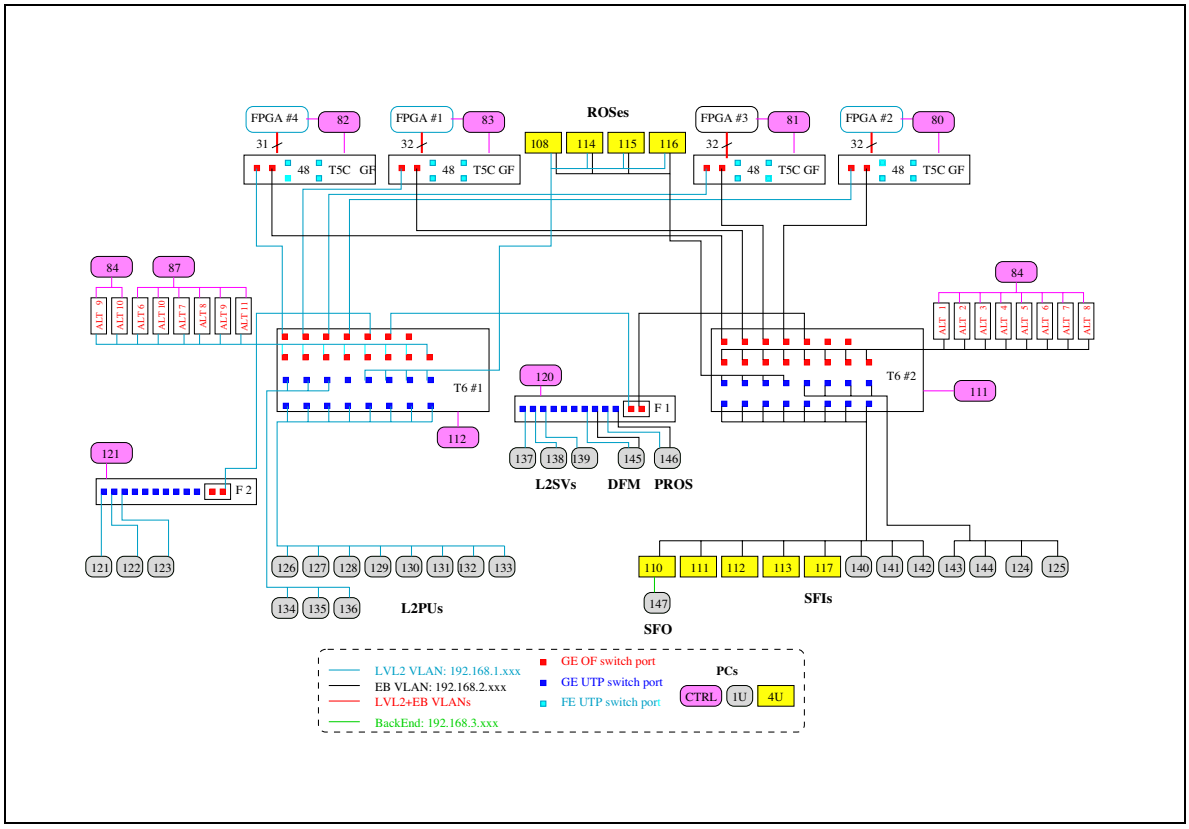

Figure 14-4 Organization of the 10% testbed

Each FPGA ROBin emulator is limited to sending a maximum message size equal to a single Ethernet frame, i.e. they can only emulate a ROBin with a single ROB. However, they will respond to any number of requests for data and used in this way in the testbed they can emulate 1600 ROBins. As the Alteon NIC is programmable, they too can respond to any number of requests for data.

For some system performance measurements, the performance of the Alteon NICs limited the performance of the system, particular, when these devices were used to emulate a ROBin with more than one ROB or a ROS with more than one ROBin. In the case of the FPGA ROBin emulators, the output bandwidth of the concentrating switch (~105 Mbyte/s) limits the obtainable rates particularly when they are used to emulate 1600 ROBins.

14.4.1.2 RoI data collection in the 10% testbed

Referring to Figure 14-4, the LVL2 trigger consists of a central switch (T6#1), fourteen L2Ps, three L2SVs and a pROS. The central switch is a 32-port Gigabit Ethernet switch consisting of sixteen optical ports and sixteen electrical (UTP) ports. The fibre ports are used to connect the ROBin emulators and for connecting to a grouping switch (described below). The UTP ports are used to connect the: PCs used for the bus-based ROS and eleven L2Ps. The L2Ps, of which there are presently eighteen, are 1U rack-mounted 2.2 GHz and 2.4 GHz dual-processor Xeon machines and are connected within the testbed in two ways. Some nodes (PCs 126-136 in Figure 14-4) are connected directly to the LVL2 central switch, while other nodes (PCs 121-123) connect to the LVL2 central switch via a grouping switch (F2 in Figure 14-4). The grouping switch is a Gigabit Ethernet switch with ten UTP ports and two optical ports. The final system has all L2Ps connected to the LVL2 central switch via grouping switches. In the testbed a mix-

ture has been used to understand possible effects of having grouping switches. In the testbed there are fourteen L2Ps this is less than the 10% of the final system but as the L2PUs will not be executing algorithms (at least in initial measurements) each L2PU will collect RoI data at rates beyond what is required of them in the final system, thus the fourteen L2PUs effectively emulate more than 10% of the final L2PUs foreseen in the final system.

14.4.1.3 Event building in the 10% testbed

The event building consists of a central switch, twelve SFIs and a DFM. The central switch (T6 #2 in Figure 14-4) is a 32-port Gigabit Ethernet switch identical to the LVL2 central switch. Twelve of its sixteen optical ports are used to connect the ROBin emulators. Twelve of its sixteen UTP ports are used to connect SFIs (PCs 110-113, 117, 140-144 and 124-125 in Figure 14-4). Other UTP ports are used to connect the four PCs used for the bus-based ROS and to connect the DFM and pROS via a grouping switch (F1 in Figure 14-4). The nine SFIs in the testbed correspond to 10% of the number foreseen in the final system.

14.4.1.4 Simultaneous RoI collection and event building in the 10% testbed

For the measurements consisting of concurrent RoI data collection and event building on operation of the two subsystems is coordinated via back pressure from the DFM to the L2SVs. The L2SVs on the testbed emulates the LVL1 trigger by generating event processing requests to be sent to the L2PU applications on the L2Ps at rates as high as allowed by the back pressure. The DFM maintains a queue of events accepted by LVL2 and awaiting assignment to an SFI. When this queue is full, a message is sent to the L2SV to signal that further sending of requests to L2PUs should be discontinued. When the occupancy of the queue drops below 80%, the DFM sends another message to the L2SVs signalling that sending additional requests can be resumed. If the queues for event processing requests in the L2PUs fill, they also can assert back pressure to throttle the rate with which the L2SV generates events.

The network topology of the final system has Ethernet loops between the two central switches and the ROBin concentrating switches (see Section 8.3.1.2.2) and the Spanning Tree algorithm (STP) or VLANs will be used to ensure a loop free topology. However, the switches used in the testbed do not allow STP to be applied on a VLAN basis and the Message Passing layer does not support VLAN tags. Hence network loops have been avoided by avoid network loops the DFM and pROS (whose communicate across VLANs) were each equipped with two network interface cards, one each to connect to the LVL2 VLAN and the other to the EB VLAN.

14.4.2 Preliminary results of the 10% testbed

This section presents the preliminary results of initial measurements performed on the fully functional testbed, i.e. concurrent RoI data collection and event building. The results of the system performance using a bus-based readout are shown in Figure 14-5 and Figure 14-6, while the system results using a switch-based readout are shown Figure 14-7.

In Figure 14-5, the sustained event building rate is plotted versus the number of L2PUs used in the testbed for different LVL2 accept fractions. In this set of measurements there were eight SFIs and four bus-based ROSs each emulating the support of 12 ROLs. Each L2PU had two worker threads requesting RoI data, for different events, of size 1.5 kbyte from a single ROL per event. The size of the event be built is 48 kbyte. the event building rate is higher than it would be in the

full system because of the smaller number of ROSs (four instead of ~130), even considering the smaller number of SFIs (eight instead of ~90).

For a LVL2 accept fraction of 1% the sustained event building rate increases linearly with the number of L2PUs and an event building rate of 0.5 kHz is achieved. For this accept fraction the rate is limited by the number of the L2PUs in the testbed. As the LVL2 accept fraction increases the event building rate also increases but with increasingly non-linearly with the number of L2PUs. The sustained event building rate reaches a maximum of 5.7 kHz corresponding to each ROS driving its link to the event building at ~68 Mbyte/s. Note that in the final system, LVL2 accept rates of approximately 3% are expected.

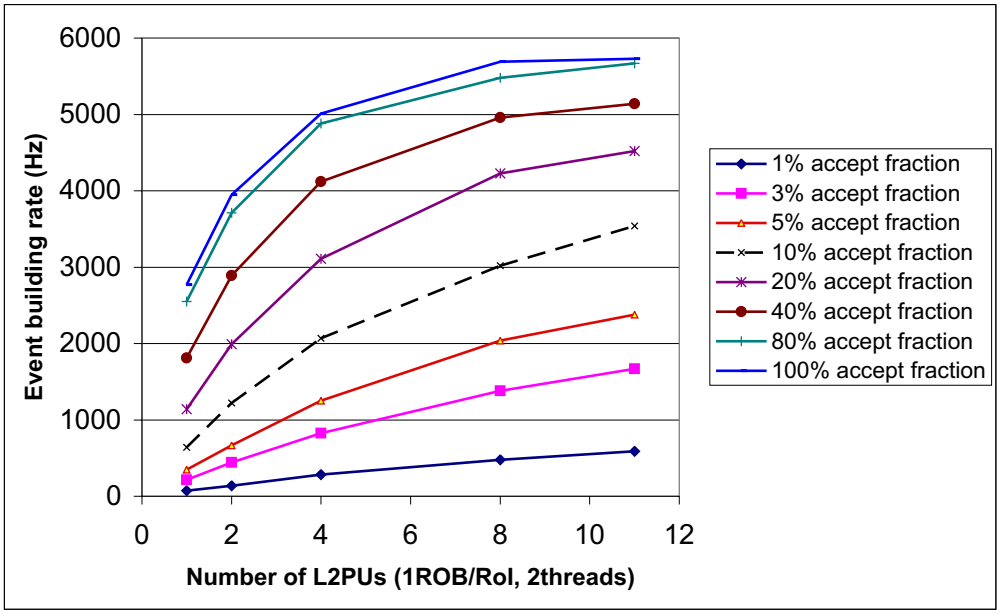

Figure 14-5 Event building rate for simultaneous RoI handling and event building in the 10% testbed. 4 ROS units, each servicing 12 ROLs, were emulated with 4 PCs. Per event the L2PUs requested data as would be received via one of the ROLs associated with one of the emulated ROS units. 8 SFIS were requesting the data from all 12 emulated ROLs of each ROS unit for the accept fractions indicated

The results of similar measurements are shown in Figure 14-6, however, for this set of measurements twelve bus based ROSs were used, four based on the PCs and eight based on the Alteon emulators. The data show that for LVL2 accept fractions of at least 10% the sustained event building rate reaches a 3 kHz limit with only four L2PUs. Similar measurements performed on the testbed but only for event building show that the limit of 3 kHz is due to the limited number of SFIs used and that with ten SFIs a limit of 4 kHz would be reached which is imposed by the limitations of the Alteon emulators. It should be noted however that the 3 kHz event building rate achieved is ~10% lower compared to event building alone. At the time of writing, it is still to be established whether this is due to extra load on the ROS due to the process of RoI data collection. Note that for these measurements, the number of ROSs and SFIs is 10% of those expect-

ed in the final system for a LVL2 accept rate of ~3% and the sustained event building rate achieves the design value of 3 kHz for this accept rate.

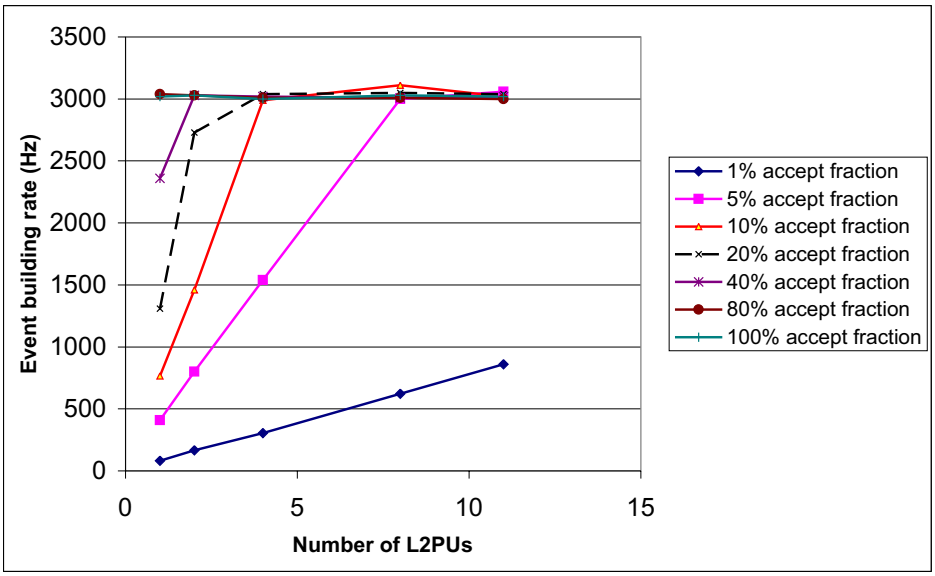

Figure 14-6 Event building rate for simultaneous RoI handling and event building in the 10% testbed. 12 ROS units, each servicing 12 ROLs, were emulated with 4 PCs and 16 Alteon NICs (two per ROS unit, one supplying LVL2 data, the other EB data). Per event the L2PUs requested data as would be received via one of the ROLs associated with one of the emulated ROS units. 8 SFIs were requesting the data from all 12 emulated ROLs of each ROS unit. for the accept fractions indicated

In Figure 14-7 the preliminary results of initial measurements performed on the testbed with a switch-based ROS are shown. The measurements were performed with both types of ROBin emulators (FPGA based and the Alteon). In one set of measurements the 125 FPGA ROBin emulators were used to emulate 1600 ROBs, in a second set of measurement the Alteon NICs were used to emulate 1600 ROBs. In the third set of measurements the 1000 ROBins are emulated by the 125 FPGA ROBin emulators and the Alteon NICs emulate 600 ROBins.

The figure shows the sustained event building rate versus the number of L2PUs, for a LVL2 accept fraction of 3% and eight SFIs performing event building. Ethernet flow control was on. In the measurements where a single type of ROBin emulator was used the event building rate is limited, in the case of the FPGA emulator, the link bandwidth connecting the ROS concentrating switches to the event building central switch. In the case of the measurements performed with the Alteons, the achieved event building rate is limited by the performance of the Alteon NIC. The set of measurements obtained by the joint use of the FPGA and Alteon ROBin emulators reaches the limit imposed by the use of only eight SFIs. Similarly to the results obtained with the bus-based ROS, the sustained event building performance is ~14% lower than in the case of event building alone. This cause of this reduction remains to be understood.

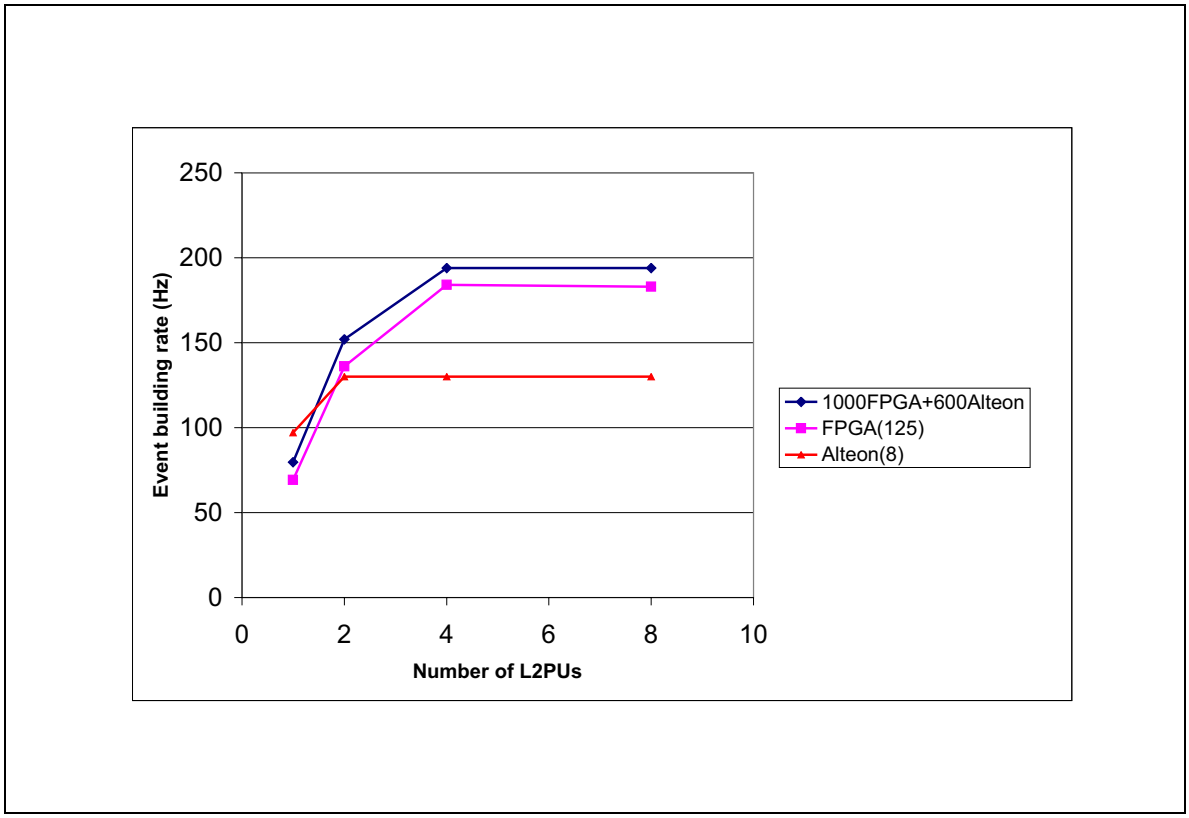

Figure 14-7 Event building rate for simultaneous RoI handling and event building in the 10% testbed for three different ways of emulating 1600 ROBs, as explained in the text. 8 SFIs were requesting the data from all ROBs for an accept fraction of 3%. Ethernet flow control was switched on

The measurements presented here represent only the results from an initial set of studies. Analysis of the measurements presented above, along with modelling of the testbed configurations used for the measurements, are still in progress at the time of submission of this report. Further Studies of the 10% testbed will continue and it is expected to replace the emulators used for the initial set of measurements by the prototype ROBin thus eliminating some of the limitations of the current testbed.

14.5 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 a lower priority than the achievement of the performance requirements during this phase of development. 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, these issues have been addressed.

Four different aspects of the global functionality have been covered: system configuration, stability in cycling through the control states of the system, 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.

14.5.1 System configuration

A data acquisition system must be easily reconfigurable in order to accommodate the substitution of hardware, the change of trigger conditions, etc. Tools for storing/retrieving the configuration parameters into/from a database must be available, and Run Control, Data Flow, and Trigger software must be designed to be dynamically reconfigurable.

In test beams, data taking configurations tend to change very often, because of the addition of new components and sub-systems that need stand-alone as well as integrated debugging in the read-out chain. In order to ease integration of detector read-out chains in the Data Flow system, nested partitions were introduced. Nested partitions allow different groups to develop their databases independently and then to link them together under a common top partition. The success of this technique was recently proven, especially during beam tests of the Muon detectors. For these tests, five different ROD or ROD emulator crates were independently setup and tested, before being connected for further debugging to their corresponding ROS units and finally also to the event building system.

Although the configuration of the HLT/DAQ hardware and of the software applications are specified at the beginning of a data taking session, a number of configuration parameters for various applications can be changed dynamically, e.g. amount of memory to be reserved for the data, length of queues, etc. Some very dynamic parameters which change for every run, e.g. the run number and calibration variables, are not kept in the configuration database and are distributed via the Information Service provided by the Online Software (see Chapter 10) at the start of the run. Mechanisms for dynamic database changes, with corresponding mechanisms to notify the affected applications are being studied.

14.5.2 Finite state machine transitions

When performing a series of measurements with different configuration options, the HLT/DAQ system must be capable of cycling through a finite set of states in a stable fashion. 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.

14.5.3 Monitoring

In a distributed system such as the HLT/DAQ system it is important to monitor the operation of the system continuously. All applications regularly publish statistics on their performance, as well as on the occurrence of errors, via the Information Service. Furthermore, the ROS and the SFI are capable of providing event data for physics monitoring. Operational monitoring has been intensively used to conduct all the performance measurements described in the previous chapters. All aspects of the monitoring facilities have been regularly used by the people on shift during testbeam data taking.

14.5.4 Fault tolerance

The HLT/DAQ system needs to be fault tolerant. Additional work is needed in this area, therefore it was not considered to be appropriate yet to conduct a series of systematic tests in order to

assess the performance of the system in case of errors. In general, an error that 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, for instance when a ROD does not send consecutive LVL1 event identifiers (L1IDs) to the ROS, or when an SFI does not receive the requested data of an event in a time-out period. However, a *FATAL* error in one application, which prevents it from continuing to take data has a potentially serious effect on the overall HLT/DAQ system. For instance, if the system is unable recover from failure of a component that is unique and necessary for data taking, such as the DFM or the RoI Builder. By design, the system can recover from the failure of one or more L2PUs or SFIs, because the L2SV or DFM, respectively, can dynamically mask the failing component. The failure of a ROS, in contrast, requires dynamic re-configuration of the downstream data-taking chain which is not possible at the time of submission of this report. Similarly the mechanism to dynamically mask a single ROL in a ROS is not in place.

14.5.5 Conclusions and outlook

The functional performance of the HLT/DAQ system has been tested during the development of the system and during its exploitation in a testbeds and testbeams. The functionality today is already adequate for successful application of the present prototype implementation in testbeam setups and for carrying out performance measurements. Further development is necessary with respect to the dynamic re-configuration of the system during data taking, in particular in the case where re-configuration is required because of changes in the run conditions on the occurrence of errors. It is essential that components that are not unique can be dynamically excluded from the running system if required. Re-insertion of such components without stopping data taking, may be difficult due to synchronization issues, but the possibility will be studied further.

14.6 Modelling results

14.6.1 Paper model

Estimates of average message frequencies, data volumes and the amount of processing power required for the HLT/DAQ system have been made with the help of a ‘paper model’. The most important results have been presented in Chapter 2. A description of this model and its results can be found in Appendix A.

14.6.2 Computer model

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

To verify that these conditions are met, the dynamic behaviour of the full system needs to be studied with the help of simulation with a 'computer model', as the construction of a full scale testbed is not feasible. Computer models therefore have been developed to obtain information on basic and fundamental properties such as the achievable throughput, distributions of the LVL2 decision time and of the event building time, 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.

The type of simulation used for the computer models is known as 'discrete event simulation'. 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' which has occurred. Only at the time of occurrence of an event is the modelled system 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 at a later time, which are entered at the correct position in the time-ordered 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 HLT/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 Flow applications), or data items. Two simulation programs have been used, the at2sim program [14-15] and the Simdaq program [14-16]. The at2sim program makes use of the general purpose simulation environment of the Ptolemy system[14-17]. 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 being a part of the program.

The component models used in the at2sim program are models of the testbed components described in Section 14.4. They 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 [14-18] and Ethernet switches [14-19] have been developed. 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. For the calibration of the models of the Data Collection applications, time stamps 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 results from 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 possible limitations in the switches which may affect the performance required for the full HLT/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 from measurements usually gave rise to a modification in the model(s) and associated parameters and another calibration phase.

The component models in the Simdaq program are less specific than the models used in at2sim. The current version of the program can be seen as a dynamic version of the paper model. The program makes use of the same information concerning LVL1 trigger menus, mapping of the detector onto the ROLs, sizes of fragments associated with ROLs, execution times of LVL2 processing steps and associated reduction factors, etc. as the paper model. The eta and phi coor-

dinates of the RoIs generated according to the LVL1 trigger menu are chosen at random from the possible RoI positions as defined by the LVL1 trigger (the probability of choosing a certain position is determined by the size of the area in eta-phi space associated with the position). Results for average rates should be the same (within the statistical errors) for both models, while computer model results for the computing resources actually used, taking into account their utilization, should be equal to the paper model results for the computing resource requirements. Good agreement between results from the paper model and Simdaq has been achieved, and the results of both models have been checked for consistency.

14.6.2.1 Results of testbed models

The results of the measurements focusing on the scalability of the Event Builder have been compared to computer model predictions. Three different setups with homo- or heterogeneous sets of ROB emulators have been modelled. In Figure 14-8 a comparison between results from measurements and model predictions for the event building rates as a function of the number of SFIs collecting data is presented. Three setups were investigated: a setup with 125 FPGA ROB emulators (13 ROBs per FPGA), a setup with 8 Alteon ROB emulators (200 ROBs per Alteon) and a setup with a mixture of the two types of emulators: 1000 ROBs were emulated by 125 FPGA emulators (8 ROBs per FPGA) and the remaining 600 ROBs were emulated by 8 Alteons (75 ROBs per Alteon). A set of modelling results is associated to each set of testbed measurement results. The three setups showed different saturation rates due to either the performance of the emulators or due to the arrangement of the setup. For a small number of SFIs, the event building rate increases by 30 Hz each time a new SFI is added to the system (this is the maximal building rate of a single SFI). The lowest maximum event building rate is observed for the setup with 8 Alteons emulating 75 ROBs each. The internal processing time of the Alteon NICs for the incoming messages is 40 μs and as each emulator has to process 200 requests for an event, the upper limit for the rate is 125 Hz. In the setup with the FPGA emulators, the rate limitation is caused by the throughput of the Gigabit Ethernet up-links connecting the concentrating switches to the EB central switch. With 2.2 Mbyte of event data spread uniformly over the FPGA emulators connected via four concentrating switches, each up-link has to deliver 2.2 Mbyte / 4 = 0.55 Mbyte of data. Assuming the maximum payload which can be transferred in 1300 byte packets over the Gigabit Ethernet to be 105 Mbyte/s, the event building rate is limited to 191 Hz. The highest rate can be observed in the setup using a mixture of FPGA and Alteon emulators. The rate limit is determined again by the throughput of the Gigabit up-links between the concentrating switches to which the FPGA emulators connect and the EB central switch. In this setup 1000 out of 1600 ROBs are emulated by the FPGA devices producing 1000/1600 * 2.2 Mbyte = 1.375 Mbyte of event data. This data is spread over the FPGA emulators attached to four concentrating switches and requires that 0.344 Mbyte will be sent per event over the up-link. This limits the rate to 305 Hz. In this setup the limit due to the Alteon NICs is higher, as each emulates 75 ROBs and with 40 μs of processing time for each request the limit would be 333 Hz. In the setup with a mixture of ROB emulators, the rate does not scale linearly with the number of SFIs if this number is larger than four and below saturation of the event building rate. This is due to queuing of packets heading for the up-link in the concentrating switch, which is very sensitive to the traffic shaping (or lack of it). Figure 14-8 shows very good agreement between the results from measurements and predictions from modelling. These results validate the calibration of the model components (switches, SFIs, emulators).

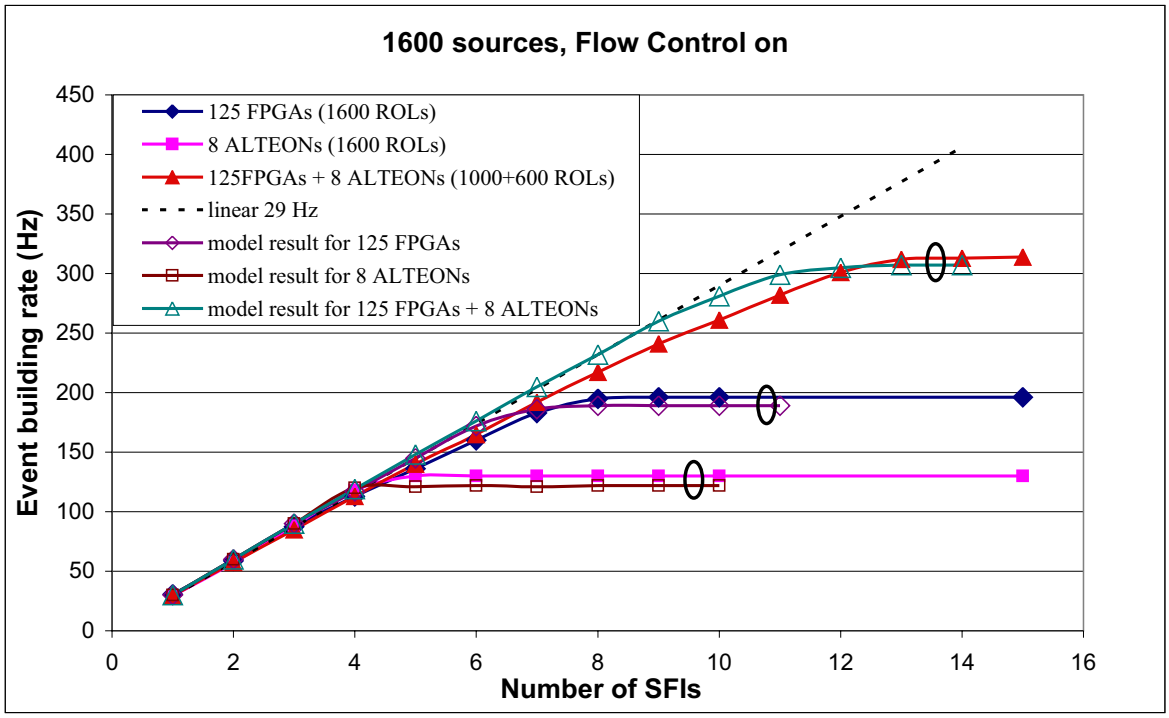

Figure 14-8 Comparison of measurement results and computer model results for event building only, without LVL2 traffic in the testbed. Ethernet flow control was switched on.

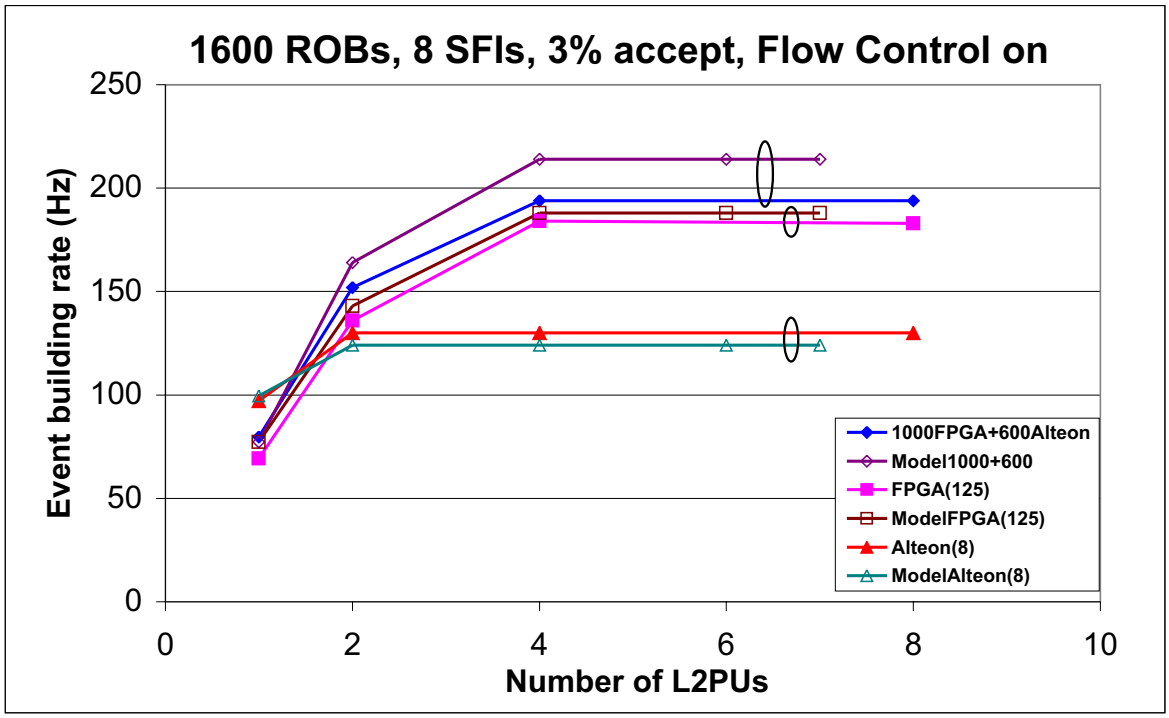

Figure 14-9 Comparison of measurement and modelling results for the event building rate for simultaneous RoI handling and event building in the 10% testbed for three different ways of emulating 1600 ROBs. Ethernet flow control was switched on.

In Figure 14-9 a comparison is made between measurement results obtained with the testbed for simultaneous RoI handling and event building and modelling results. Again the three setups were used for obtaining the results presented in Figure 14-8. 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 event building rate saturates at the same level as for the EB scalability tests. Less good agreement, however still within 10% tolerance, has been reached for the setup with a heterogeneous set of ROB emulators. For this setup the value at which the event building rate saturates is determined by the performance of the 8 SFIs.

14.6.2.2 Results of extrapolation of the at2sim testbed model

The full system model chosen to be implemented in at2sim has 1654 ROBins, each servicing a single ROL and each having an individual network connection [14-13]. The LVL2 subsystem has been assumed to be 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 L2SV DFM, and pROS are connected via a dedicated small Gigabit Ethernet switch to the four central switches. The ROBins are connected in groups to concentrating switches, each having four up-links to the central switches. The number of concentrating switches per sub-detector depends on the average size of the event fragments from a given sub-detector (for results of calculations see Ref. [14-13]). In total there were 46 concentrating switches. Results were obtained for the following configurations:

1. ***Individual ROBin Fast Ethernet***: 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 Gigabit Ethernet***: each ROBin has a single Gigabit Ethernet connection to a concentrating switch (thus the concentrating switch becomes an all-Gigabit switch),
3. ***Two, four or six 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 75 kHz 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 processing step — for each event, data from ten randomly chosen ECAL ROBins were requested and a decision was produced and sent to the L2SV. The L2PUs were not calibrated — they were used only to generate the LVL2 traffic in order to obtain a more realistic environment. The SFIs were requesting data in random order from all ROBins or ‘ROBin aggregates’.

The effect of a credit based event building traffic shaping on the event building latency and queue build-up has been investigated with the model. The left plot in Figure 14-10 shows that increasing the number of credits per SFI above ten does not improve the latency for event building except for the ‘individual ROBin Fast Ethernet’ configuration. In the latter configuration, the

latency is still related to the transfer speed of the Fast Ethernet links (12.5 Mbyte/s). Therefore queuing of fragments during their transfer to the SFIs is unlikely and the time needed for building a complete event will depend on the time needed to transfer an event fragment via a Fast Ethernet link. The shorter latency for setups with 'ROBin aggregates' with respect to those with individually accessed ROBins is due to 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 however scales with the number of ROBins aggregated. As the EB rate is limited by the SFI performance (the network provides sufficient bandwidth), the smaller amount of time needed for generating requests allows an SFI to process events faster. More quantitatively: the SFI rate when sending requests to the individual ROBins is 30 Hz, i.e. per event 33 ms is spent (the SFIs run at 99% CPU utilization). The 33 ms is spent on the generation of 1600 requests and the reception of 1600 replies. It has been measured that for the reception of a packet 14 µs of CPU time is needed. The reception of 1600 packets will take more than 22 ms. The remaining 11 ms is used to produce 1600 requests. In case of an aggregation factor of 2, 5 ms will be saved, for an aggregation factor of 4 or 6 this will be 8 or 9 ms. Interrupt coalescence, for which the default time-out is 65 µs, also plays a role in the speed-up. During 65 µs only a few requests can be generated and consequently, only a few replies will arrive. Thus the gain in processing time due to the use of interrupt coalescence is limited - one interrupt is generated for a few packets received. In the case of aggregation a single request will produce 2 - 6 replies, depending on the aggregation factor. Therefore considerably more packets can be handled per interrupt and the time spent per packet received is smaller. This in turn leads to a reduction of the time needed for event building.

A possible consequence of the aggregation of ROBins consists of overflow of the queues in the switches. The right plot in Figure 14-10 shows the maximum length of the queues associated with the ports in the EB central switches connecting to the SFIs. In case of aggregation, up to six packets of replies (in the six ROBin aggregate scenario) can be returned for a single request, so the maximum queue length (measured in number of packets queued) can be expected to be equal to 6 times the number of credits. Therefore the number of credits may have to be reduced with respect to the scenarios where each ROBin has an individual network connection, otherwise the queue length may reach a switch buffer limit and give rise to packet loss.

In Figure 14-11 a prediction is shown for the event building capability of the full HLT/DAQ system as a function of the number of SFIs of the type used in the testbed. The maximum number of credits was set to 30. Flow control was switched on. The buffer size for the output ports in the central switches was set to 160 packet slots, the flow control was not (and should not have been) activated in the central switches. It can be seen that the maximum event building rate scales linearly with the number of SFIs to a rate of 3 kHz for 110 SFIs.

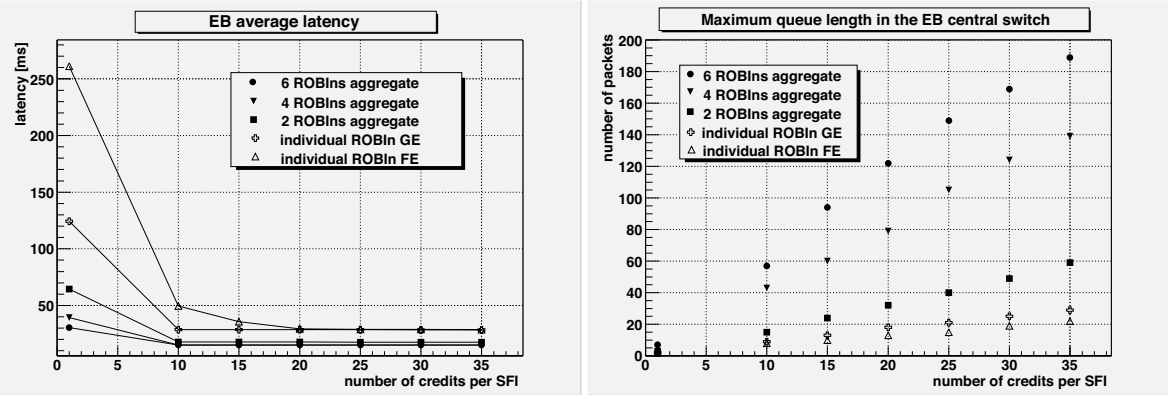

Figure 14-10 Average event building latency and maximum queue length in the EB central switches for different ROBin configurations obtained with the at2sim full system model. Flow control was switched off.

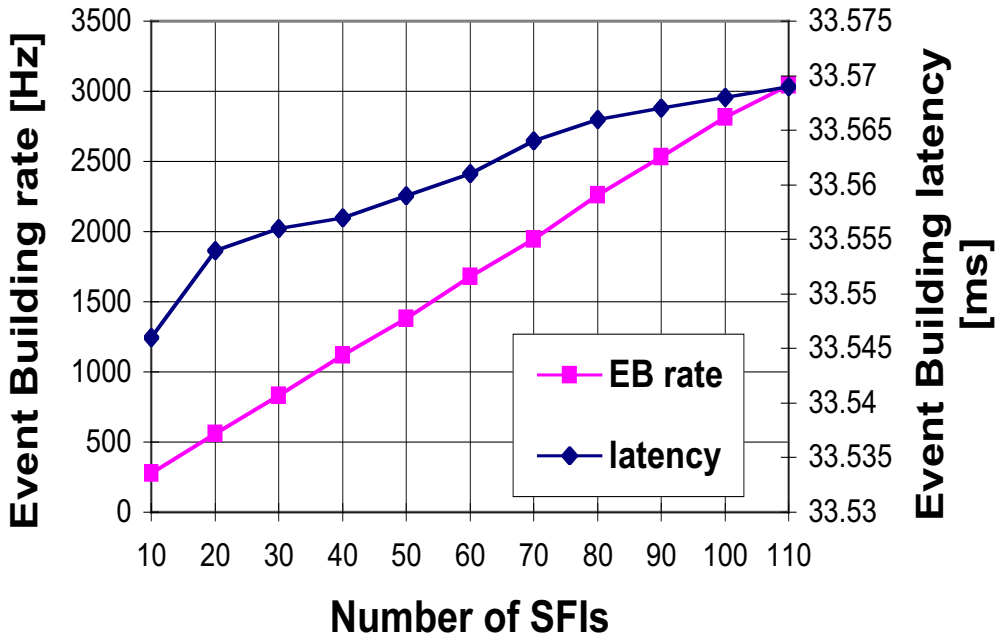

Figure 14-11 Model prediction for the Event Building capability using the calibrated models of the testbed components. The LVL2 accept fraction is 3%.

14.6.2.3 Results of the full system Simdaq model

The results presented in this report for the full system model obtained with Simdaq are all for design luminosity and a LVL1 accept rate of 100 kHz. Although the distributions for decision times and queue sizes are sensitive for the details of the models the general trends are not. The effect of various strategies for minimization of decision times and of queue lengths therefore can be studied with the model implemented in Simdaq, and also apply for low luminosity and lower LVL1 accept rates.

Two different configurations (see Section 5.5.4) have been studied, a ‘bus-based system’, i.e. a system with a bus-based ROS, see Figure 14-12 and a ‘switch-based system’, i.e. a system with a switch-based ROS, see Figure 14-13. In the bus-based system, ROBins are grouped together in groups of three (four ROLs each – 12 ROBs in total) in ROS units, each with two Gigabit Ethernet connections, one to a central LVL2 switch and one to a central EB switch. In the switch-based system ROBins are assumed each to service four ROLs and to be connected via a Gigabit Ethernet connection to a ‘concentrating switch’. In the bus-based system the L2PUs and SFIs can request data from several or all ROLs of a ROS unit, respectively with a single request. The response consists of a single, usually multi-frame, message. The addition of a header by the ROS has not been taken into account in the results presented. A ROB is associated with each ROL in the model of the ROS unit, the maximum numbers of fragments that need to be buffered are output by the simulation program. A single processor in the ROS unit takes care of distributing requests to the ROBins and of collecting and concatenating the responses. In the switch-based system event fragment data associated with different ROLs have to be requested separately. For example, the SFIs have to send four request messages per event to each ROB, and each will respond with four separate response messages (one per request). Again the maximum numbers of fragments that need to be buffered are output by the simulation program.

The L2PUs are dual-CPU machines, each running four processing threads.The model for the L2PU is built around an object managing the scheduling and de-scheduling of the threads on the two CPUs and objects representing the threads. The model allows an arbitrary choice for the number of threads and the number of CPUs. Four L2SV processors each manage a group of 125 L2PUs and four DFM processor each manage a group of 16 SFIs. Each L2PU receives RoI information from and sends decisions to the L2SV controlling the group to which the L2PU belongs. The L2PUs also send LVL2 trigger result data to the pROS (not shown in the figures). One of the DFM processors is associated with each L2SV, the L2SV sends blocks of decisions to it. The DFM processors collect LVL2 rejects and multi-cast them in blocks of 300 clears to the ROS units or ROBins via the central Event Builder switches (and concentrating switches in case of the switch-based system). The DFM translates LVL2 accepts into build requests for the SFIs, in the current model these requests are sent to the SFIs according to a round-robin scheme. Each SFI sends an ‘End-of-Event” message to the DFM controlling the group to which the SFI belongs, after building an event. This is converted to a clear and sent as part of a block of 300 clears to the ROBins or ROS units. The SFIs in this model are single processor machines. Transfer of complete events to the processors of the Event Filter has not been modelled, the events built are discarded by the SFIs.

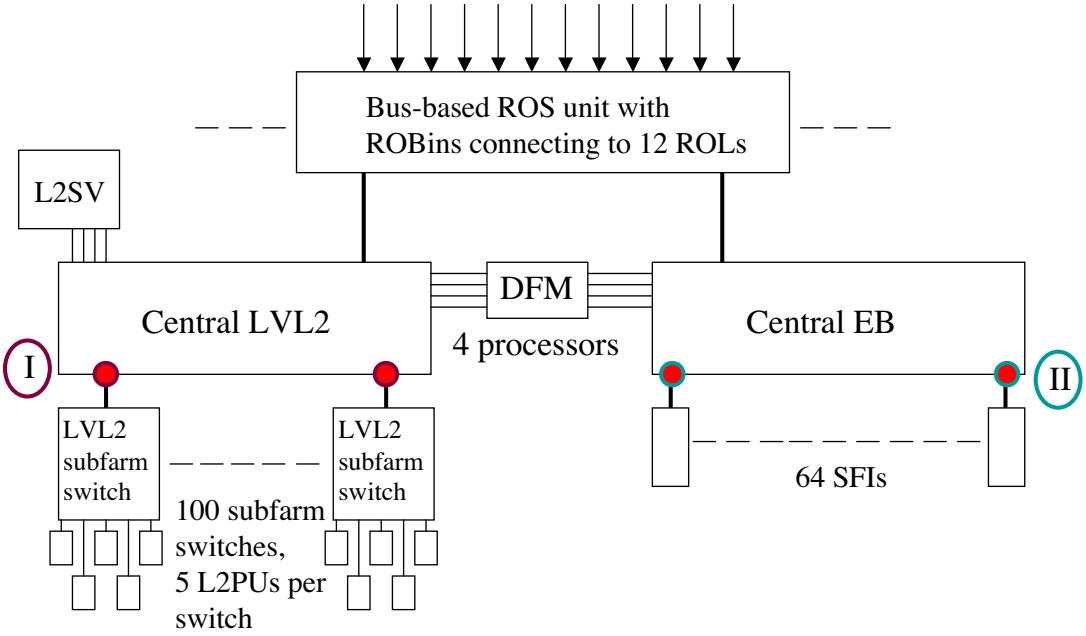

Figure 14-12 Schematic representation of the bus-based system. Possible 'hot spots' are indicated.

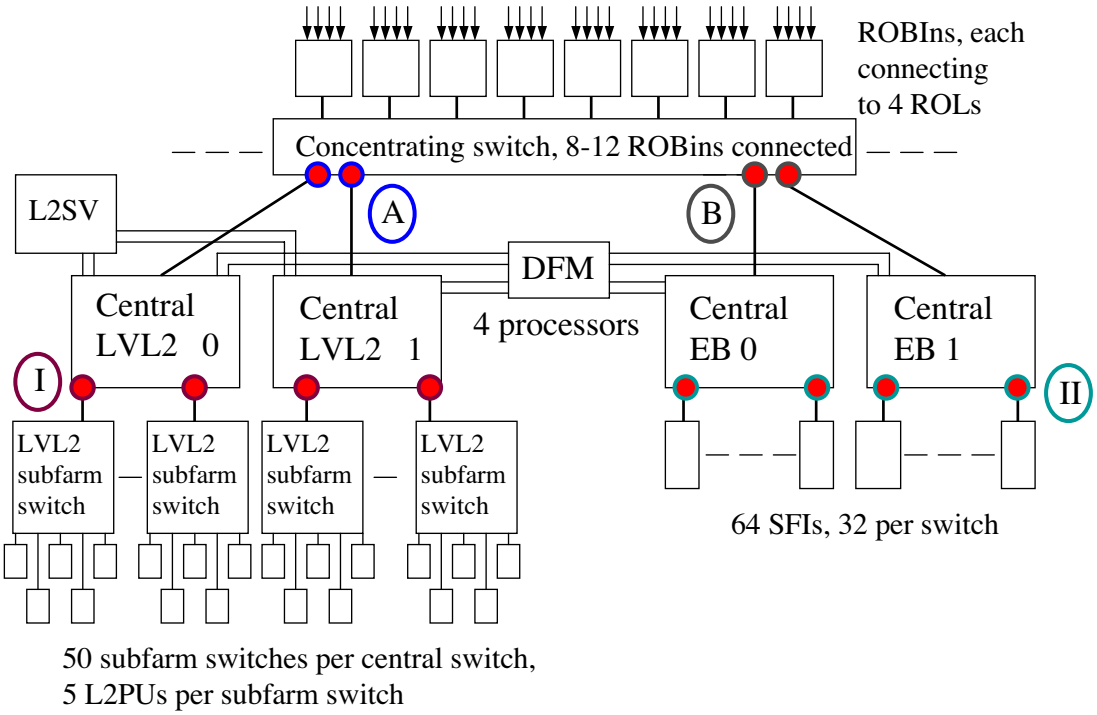

Figure 14-13 Schematic representation of the switch-based system in which ROBins are directly connected to the network. Due to the bandwidth requirements the number of ROBins connected to a single concentrating switch has been set to 8 for the ROBins receiving data from the Pixels and SCT sub-detectors and to 10 for the ROBins receiving data from the TRT and from the LVL1 RODs, for all other sub-detectors 12 ROBins are connected to a single concentrating switch. Possible "hot spots" are indicated.

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 can be implemented with the help of 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 such a record can be maintained without an additional exchange of messages. A new event can then be assigned to the L2PU or SFI with the smallest number of events to process ('least-queued assignment'). This is an effective strategy, which makes high average loads of the L2PUs possible. In Figure 14-14, results for the LVL2 decision time (the interval between the time of occurrence of a LVL1 accept and the arrival of the LVL2 decision in the L2SV) are presented for this type of event assignment. Results are also presented for round-robin assignments in combination with assignment of a maximum of four events to the same processor, as well as for a round-robin-only assignment scheme. The peaks in the distribution for least-queued assignment are caused by the various processing steps. For each step a fixed processing time has been assumed, in reality algorithm processing times will depend on the properties of the input data, this will result in less pronounced peaks than found with the model. The average utilization of the L2PUs is here 77%. The tail of the distribution for round-robin assignment does become much longer for a smaller number of L2PUs, i.e. for higher average utilization. Above a utilization of 85 - 90% stable operation of the system is no longer possible, as the amount of processing resources requested is not evenly distributed over the L2PUs. With least-queued assignment stable operation is still possible at this level of utilization.

After each new assignment by the L2SV of an event to one of the L2PUs, the number of events assigned to that L2PU is entered in a histogram. Distributions obtained in this way are presented in Figure 14-15, again for least-queued assignment, round-robin assignment with a maximum of four events assigned to the same L2PU and round-robin assignment only. The least-queued strategy clearly results in a minimum number of events assigned simultaneously to the same L2PU. From the distributions it can be inferred that the choice of four threads per L2PU is appropriate for the system modelled, two threads probably would not be enough, in particular for round-robin assignment.

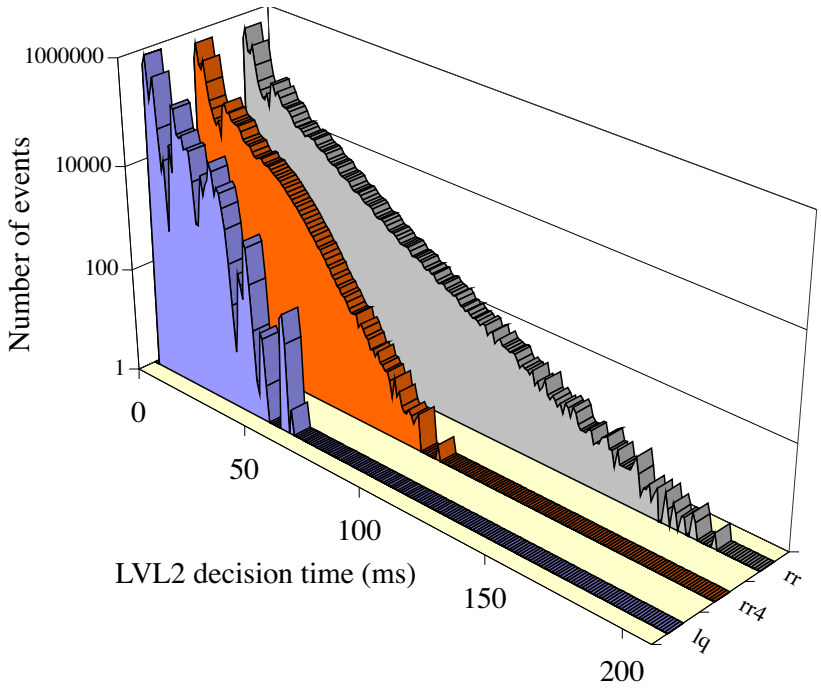

Figure 14-14 LVL2 decision time for the bus-based system, for round-robin assignment (rr), round-robin assignment of at maximum 4 events to the same L2PU (rr4) and least-queued assignment (lq). The results for switch-based read-out are almost identical. The average decision times are 11.9 ms (rr), 10.7 ms (rr4) and 8.8 ms (lq). The maximum number of fragments to be buffered per ROL is about 3100 (rr), 3000(rr4) and 2600(lq), taking into account that deletion of events accepted by the LV2 trigger requires clears from the event building system.

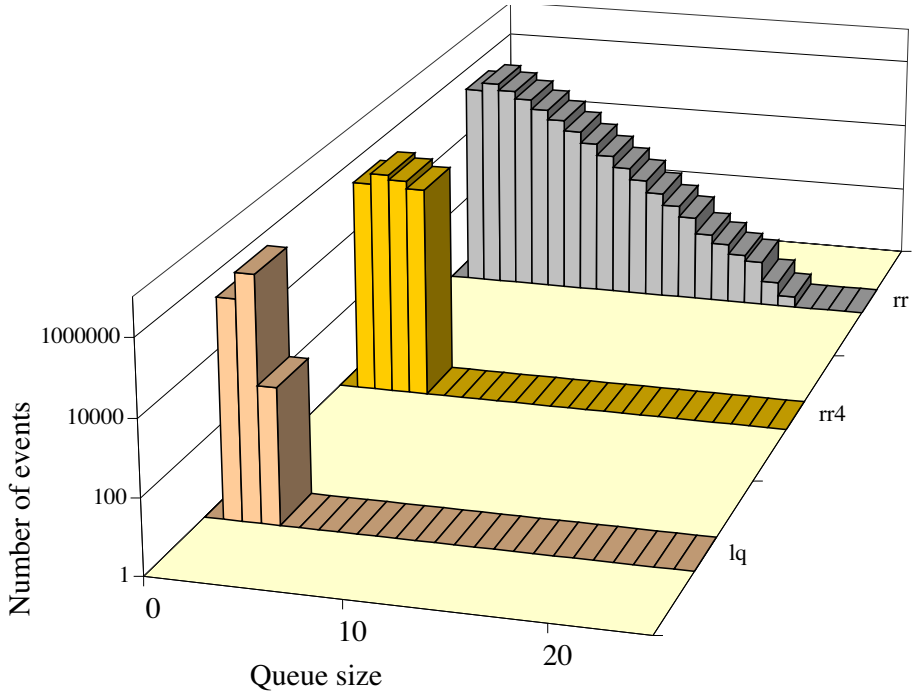

Figure 14-15 L2SV queues for round-robin assignment (rr), round-robin assignment of at maximum 4 events to the same L2PU (rr4) and least-queued assignment. The results for switch-based read-out are almost identical. The averages of the distributions are: 1.8 (lq), 2.4 (rr4) and 2.8 (rr)

With the model the building up of queues can be studied. It is important that the predicted queue lengths do not exceed the available buffer capacity in the switches, as otherwise in reality either packet loss will occur or flow control will be activated. The latter prevents packet loss, but may also cause temporarily blocking of other data transfers. Therefore it is to be preferred to keep the queues short and to prevent long tails in the queue length distributions by using simple, but effective measures. These can consist of requiring the number of outstanding requests in the L2PUs and the SFIs to be smaller than a certain maximum and, for the switch-based system, of choosing a suitable pattern for requesting data by the SFIs, as will be illustrated with a few model results. Less important is the assignment pattern (not to be confused with the assignment strategy) of events to the L2PUs. It should be noted that the results presented in this section are for the case of flow control switched off.

In both the bus-based and switch-based systems, queues tend to form in the output ports of the central switches connecting to the LVL2 subfarm switches (“point I”) and to the SFIs (“point II”). Figure 14-16 shows distributions for the sizes of queues in point I for four different scenarios for assigning events to the L2PUs for the bus-based system. In Figure 14-17 the same results are presented for the switch-based system. The size of a queue is equal to the number of Ethernet frames stored in the queue. For each message the number of frames in the queue is entered in a histogram at the time of arrival of the last frame of that message in the queue. These histograms are displayed in the figures. The distributions have shorter tails for the switch-based system. This is due to the fact that one concentrating switch deals with the data from 32 - 48 ROLs, while in the bus-based system one ROS unit deals with the data from 12. ROLs. On average therefore somewhat more data are flowing for the switch-based system via a single port into the central LVL2 switch than for the bus-based system. This leads to less queuing in point I as the incoming frames arrive one after the other with the same speed as with which they can also be output again. The figures also show that for the bus-based system the tail of the distribution for least-queued assignment becomes somewhat smaller if subsequent events are assigned as much as possible to L2PUs connected to different LVL2 subfarm switches. However, the tail can only effectively be suppressed by limiting the number of outstanding requests in the L2PUs. Again the distribution for the bus-based system has a longer tail than for the switch-based system, this is now mainly due to the fact that in the first case there is a maximum to the outstanding number of (in most cases) multi-frame messages requested, while for the bus-based system in most cases single frame messages are requested. The distribution for the LVL2 decision time is for both cases almost identical to the distribution for least-queued assignment presented in Figure 14-14.

The lengths of the queues associated with point II (output ports of the central EB switches connecting to the SFIs) can be controlled with the maximum number of credits, i.e. outstanding requests, in the SFIs, as already discussed in Section 14.6.2.2. For bus-based read-out again multi-frame messages are requested, so for the same maximum number of credits the queues will become longer, see also Section 14.6.2.2.

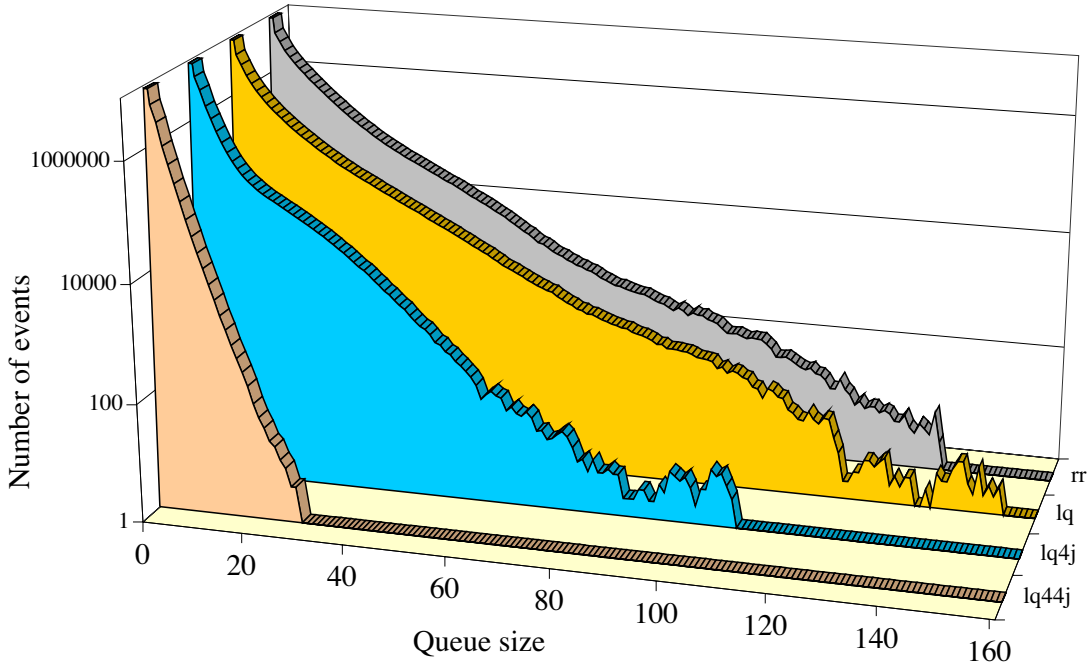

Figure 14-16 Queue sizes at point I, bus-based read-out for round-robin assignment of events to L2PUs (rr), least-queued assignment (lq), least-queued assignment of subsequent events to L2PUs connected to different subfarm switches (lq4j) and for the same strategy, with as additional requirement that the number of outstanding requests is smaller than 4 (lq44j).

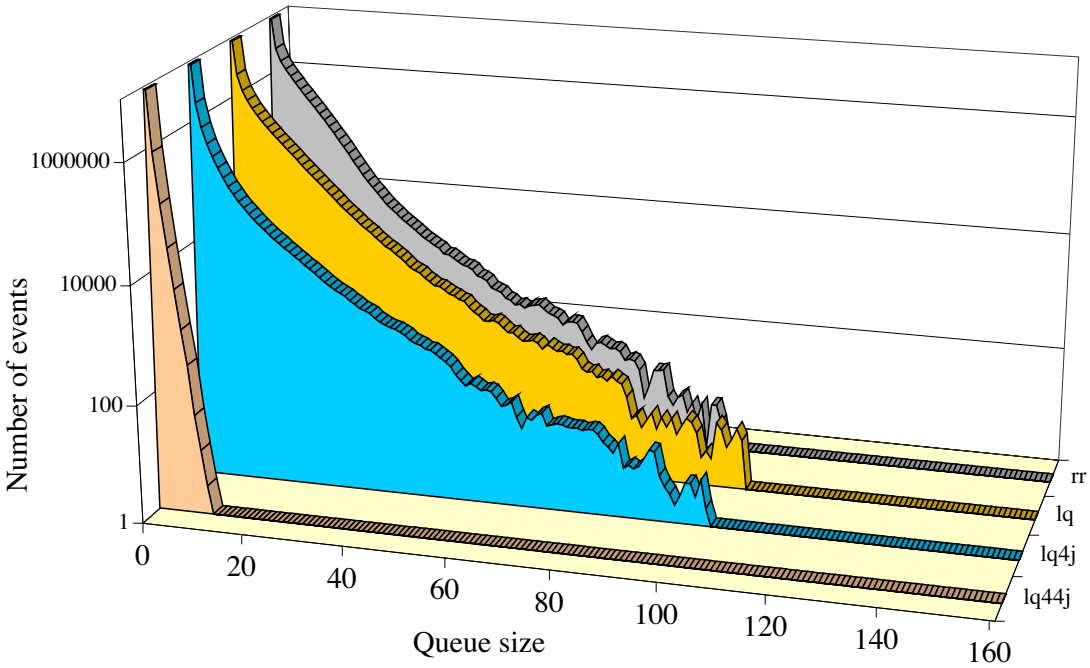

Figure 14-17 Queue sizes at point I, switch-based read-out for round-robin assignment of events to L2PUs (rr), least-queued assignment (lq), least-queued assignment of subsequent events to L2PUs connected to different subfarm switches (lq4j) and for the same strategy, with as additional requirement that the number of outstanding requests is smaller than 4 (lq44j).

Queues associated with point B (output ports of concentrating switches connecting to the central EB switches) in the switch-based system are sensitive for how the SFIs send event fragment requests. For round-robin requesting the requests arrive in all ROBins connected to a single concentrating switch almost at the same time, the responses may then cause contention for access to one of the up-links. It is not excluded that requests from different SFIs follow closely after each other with as consequence more contention. This contention will be alleviated if each SFI requests data from the ROBins in a way which avoids sending subsequent requests to the same concentrating switch, e.g. by selecting the ROBins at random, as done in the testbed measurements and in the at2sim computer model. In the Simdaq model first data are requested from only one of the ROBins associated with each concentrating switch. Only one request message is sent to each ROBin, so the data associated with only a single ROL is requested. This procedure is repeated for the remaining ROLs until all data is requested. In Figure 14-18 the effect of this procedure is shown for the concentrating switches connected to the ROBins receiving data from the Pixels detector. The procedure has also a favourable effect on the sizes of the queues in point A (output ports of the concentrating switches connecting to the central LVL2 switches) for the same detector (see Figure 14-19). However, these queues tend to be short and are less likely to give rise to problems. The maximum number of credits per SFI in the model has been set to 80. This number directly controls the maximum queue size at point II, but queuing at Points B and A is less sensitive to this number.

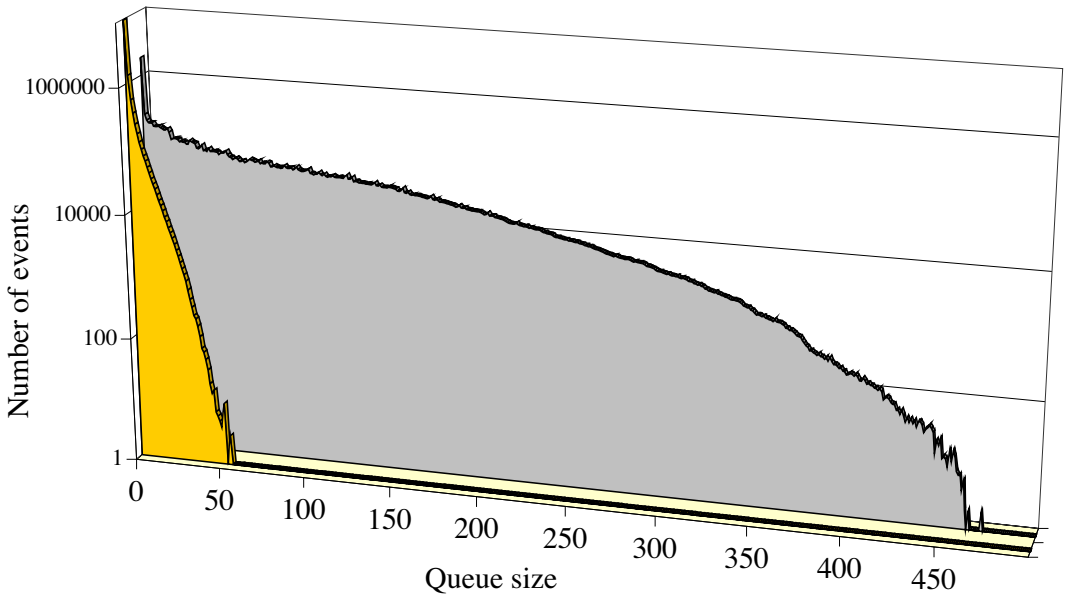

Figure 14-18 Queue sizes at point B for the Pixels. The distribution with the long tail occurs for round-robin requesting of data by the SFIs, the tail disappears if nearly simultaneous requesting via the same switch is avoided with the help of a suitable request pattern, see the text for further explanation.

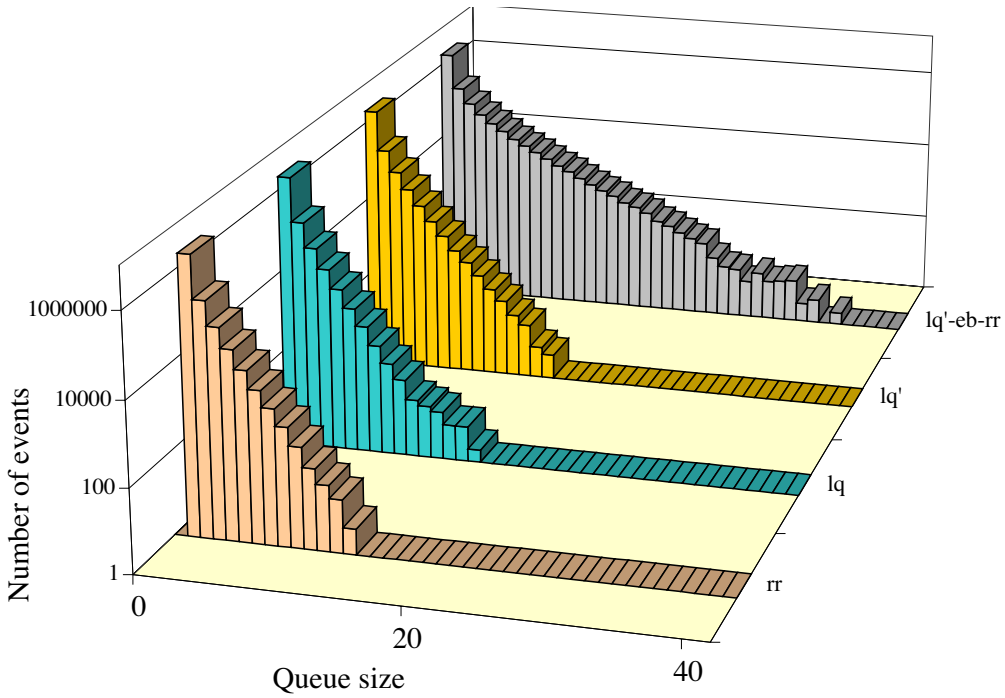

Figure 14-19 Queues sizes at point A for the Pixel detector, for round-robin requesting of event fragments by the SFIs in combination with least-queued assignment of events to L2PUs and a maximum number of outstanding requests of 4 per L2PU (lq'-eb-rr), and for a request pattern avoiding nearly simultaneous requesting by the SFIs via the same switch for the same type of assignment of events to the L2PUs as for lq'-eb-rr, for least-queued assignment of events to the L2PUs without further conditions (lq) and for round-robin assignment (rr).

14.6.2.4 Conclusion

A good understanding of the behaviour of current technology (hardware and software) is available in the form of calibrated component models. An understanding has been developed of how hot spots can be avoided in the full system and how an even distribution of the computing load over the L2PUs and SFIs can be obtained. The modelling results indicate that it is justified to assume that a system based on Gigabit Ethernet and with the current type of components can be operated at the performance level required. A model based on calibrated components for the full system (including the LVL2 system) will make it possible to find good choices for connectivity and operating conditions of the system, but already from the present results it is clear how potential problems can be avoided.

14.7 Technology tracking

14.7.1 Status and prospects

The ATLAS HLT/DAQ system is almost exclusively comprised of off-the-shelf commodity equipment; PCs, and Ethernet links and switches; the exceptions being the RoI Builder and ROBins where custom equipment has had to be developed. The technical evolution of commod-

ity computing, communications equipment, as well as pricing, is therefore an important consideration in the performance, costing and life cycle of the HLT/DAQ system.

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

14.7.1.1 The personal computer market

Moore's Law, the doubling of the number of transistors on a chip every 1.5 years, has been respected over the last three decades, and the trend is expected to continue at least through to 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 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 to hold. The turndown in the world economy and a reduced willingness to invest the large sums of money required to deliver new generations of microprocessors may however change this expectation.

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 8 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 Filter farms where the processing time will be long compared to the I/O time. Components in the system with high I/O requirements will be more bounded by link speed, but will also benefit from improvements in processor performance, as illustrated by the performance of the DFM as a function of processor clock speed shown in Figure 14-20.

14.7.1.2 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. In the long-term, the optimization of the operating system will continue, fuelled by strong support from the academic 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.

14.7.1.3 Networking

The ATLAS HLT/DAQ system uses Ethernet network technology for RoI collection and event building, as well as data distribution to the EF. It is also used in other networks associated with control and monitoring. Ethernet is, throughout the world, the dominant local area network

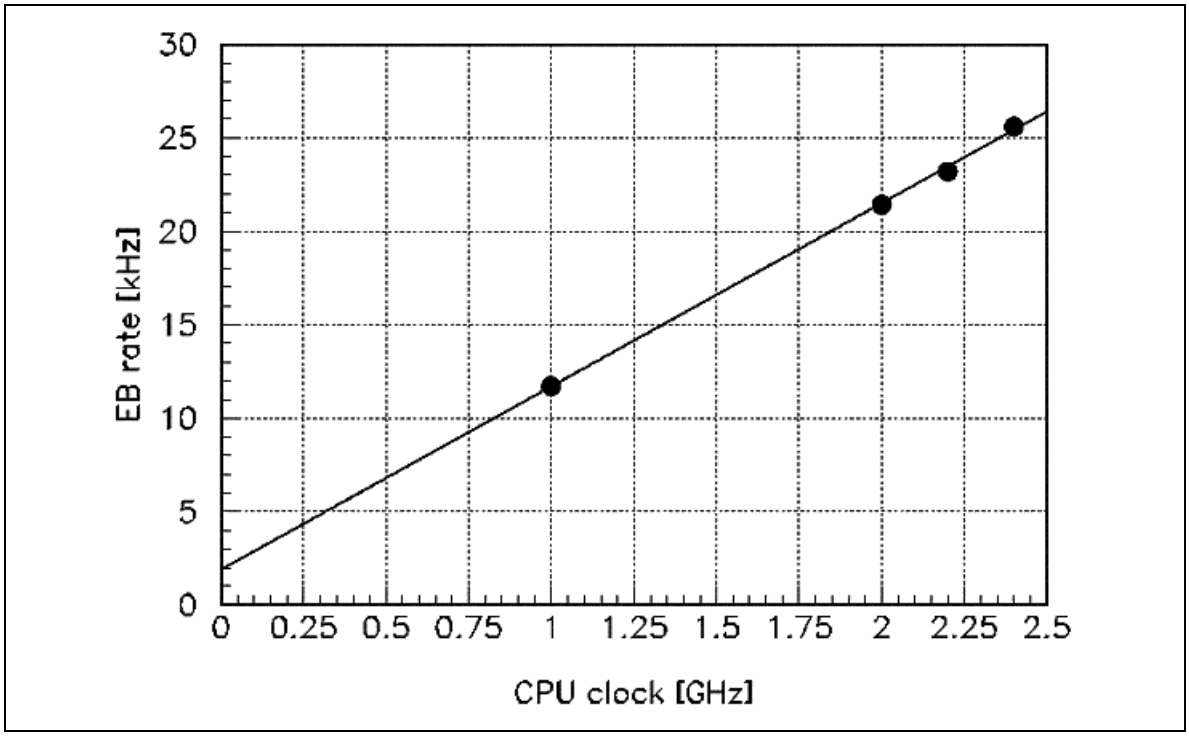

Figure 14-20 The performance of the DFM as a function of processor clock speed.

(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 [14-20].

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.

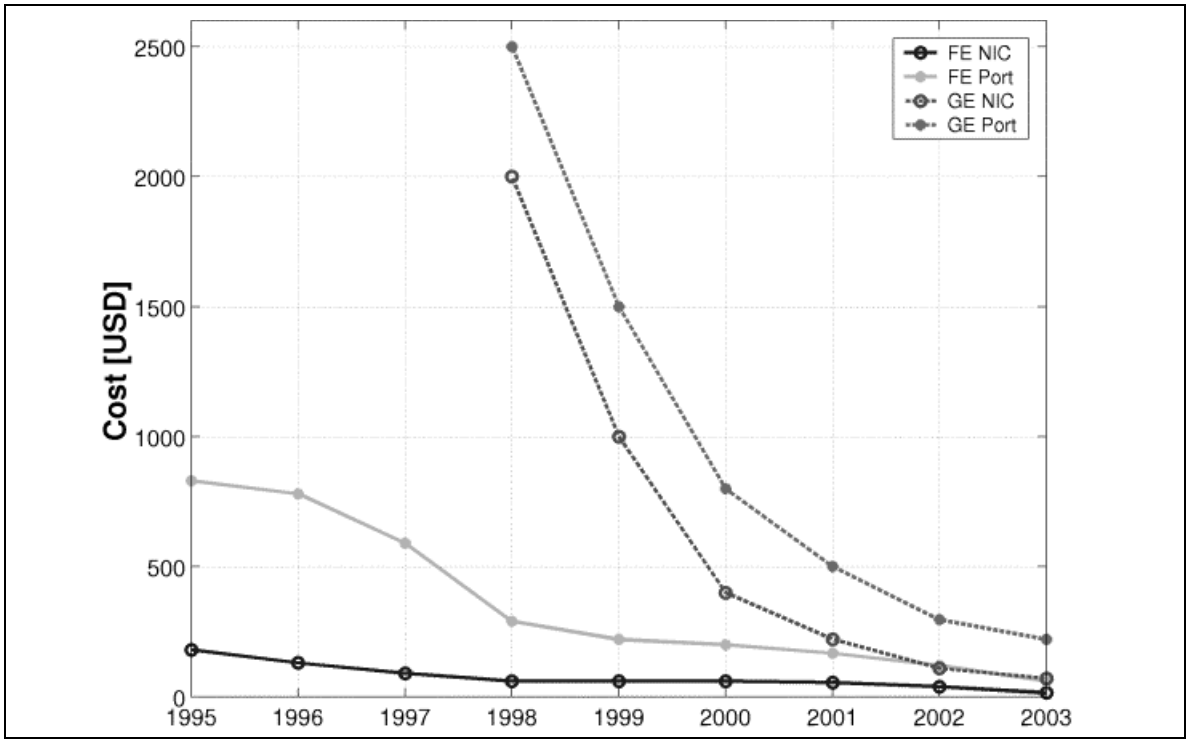

Figure 14-21 The evolution of the cost for Fast and Gigabit Ethernet switch ports and network interface cards.

The HLT/DAQ system described in this TDR can be built using Ethernet switches available today. Even the most demanding components in the system, the large central switches, are comfortably within today's norm. The prognosis for using Ethernet is therefore excellent. 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 Gb/s transmission between CERN and NBI, Copenhagen over the GEANT pan European backbone network [14-21]. Figures 14-22 and 14-23 show the setup used and some of the results obtained.

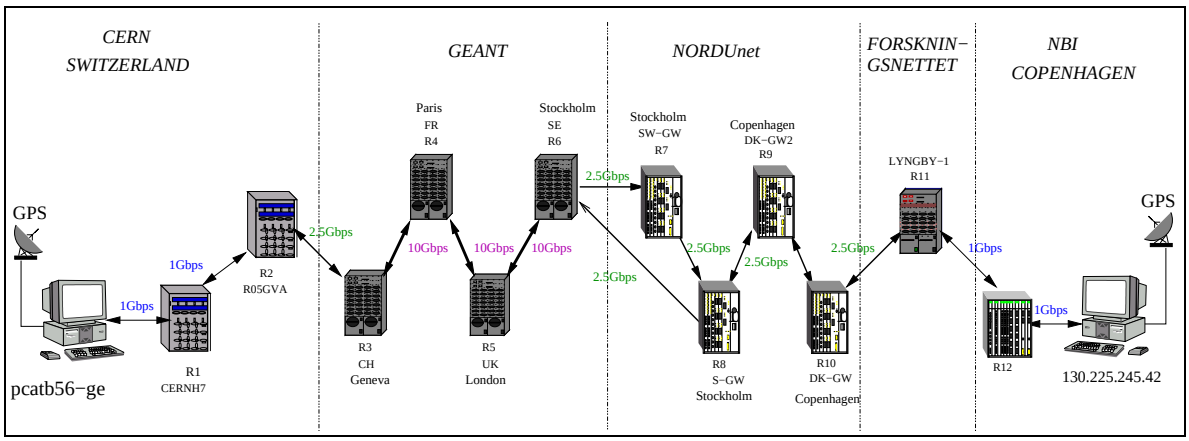

Figure 14-22 The network infrastructure between CERN and NBI (Copenhagen) over which Gb/s tests have been carried out.

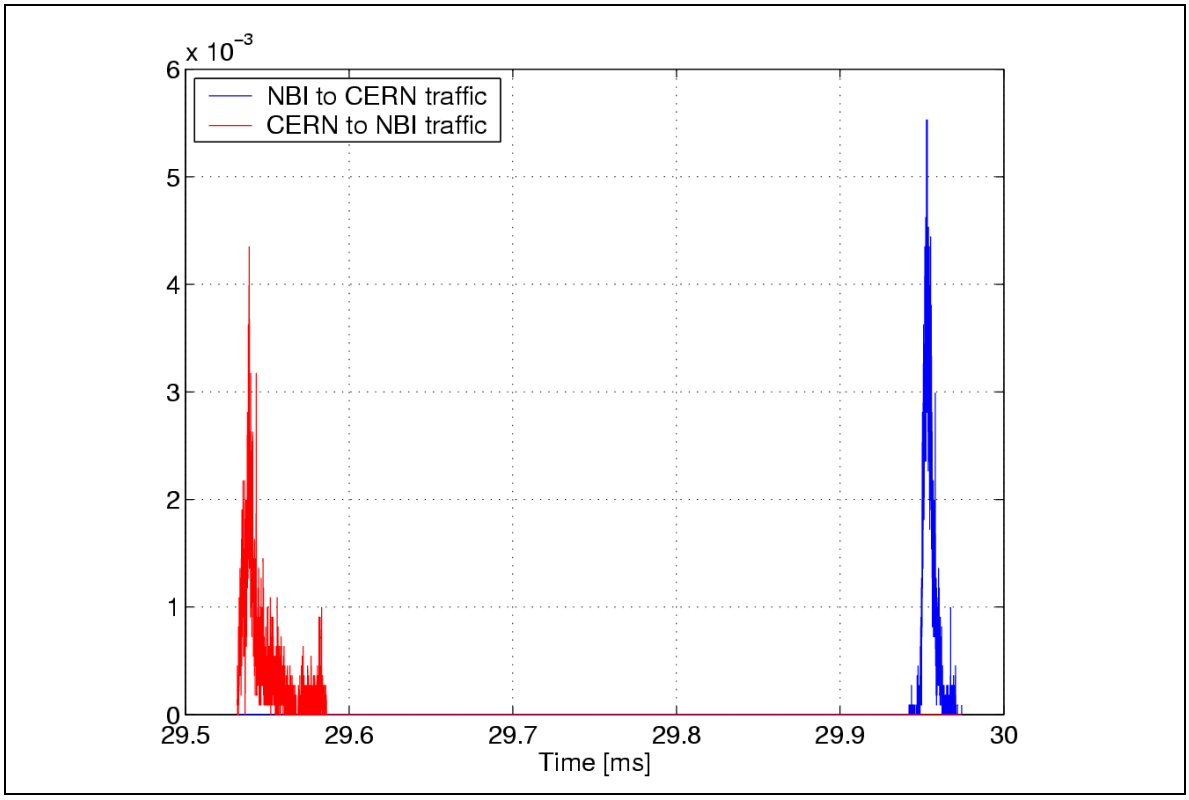

Figure 14-23 Packet latency measured between CERN and NBI (Copenhagen) at a 1 Gbps transmission rate.

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-Gigabit Ethernet may have an important role to play. However, the use of such a scheme will ultimately depend on the economics of long haul telecommunications. This factor, as well as technical considerations and practical testing, are part of our on going program of work.

14.8 References

14-1 Tests of the ATLAS LVL2 Trigger with PESA Selection Algorithms", Andre dos Anjos *et al.*, ATL-DH-TR-0002 (June 2003)

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

14-3 HLT Validation of ATHENA", M.Bosman, K.Karr, C.Bee, S.Gonzalez, W.Wiedenmann, ATL-DAQ-2002-005 (Jan 2002)

14-4 "Further studies and optimization of the LVL2 electron / photon FEX algorithm", S.Gonzalez, T.Shears ATL-DAQ-2000-42 (May 2000)

14-5 "Initial LVL2 tests with the SiTree algorithm", J.T.M.Baines *et al.*, ATL-DH-TN-001 (March 2003)

14-6 "A data manager for the ATLAS HLT", J.T.M.Baines, W.Li, ATL-DAQ-COM-2003-021 (June 2003)

14-7 "Initial LVL2 Tests with the Si Tree Algorithm", Andre dos Anjos *et al.*, ATL-DH-TN-0001 (March 2003)

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 ATLAS ATHENA web site, http://atlas.web.cern.ch/Atlas/GROUPS/SOFTWARE/OO/architecture/General/index.html

14-12 Event Filter infrastructure validation tests ”, C.Bee *et al.*, ATL-DH-TR-0004 (June 2003)

14-13 *ATLAS TDAQ: A Network-based Architecture,* H.P. Beck et al.,ATLAS EDMS Note, ATL-DQ-EN-0014, https://edms.cern.ch/document/391592/

14-14 Tigon/PCI Ethernet Controller, Revision 1.04, August 1997, Alteon Networks, F. Barnes et al, "Ethernet Networks for the ATLAS Data Collection System: Emulation and Testing", Paper presented at RT2001, 12th IEEE-NPSS Real Time Conference, Valencia, June 2001, http://atlas-tdaqtalks.web.cern.ch/ATLAS-TDAQtalks/RT01/meirosu.pdf

14-15 R.Cranfield et al., "Computer modelling the ATLAS Trigger/DAQ system performance", submitted to IEEE Trans. on Nuclear Science

14-16 J.C. Vermeulen et al., "Discrete Event Simulation of the ATLAS Second Level Trigger", IEEE Trans.Nucl.Sci.45:1989-1993,1998

14-17 Ptolemy project, http://ptolemy.eecs.berkeley.edu

14-18 P.Golonka, K.Korcyl, ”Calibration of the ATLAS Data Collection component models.”; ATL-DQ-TP-0001, https://edms.cern.ch/document/391590

14-19 P. Golonka et al., “Modelling large Ethernet networks for the ATLAS High-Level Trigger system using parameterized models of switches and nodes.”, CERN-OPEN-2001-061, poster presented on RT 2001 Conference, Valencia

14-20 Dobinson et al. RT2001 Lyon

14-21 Santiago di Compostella, RT2003