Chapter 13 Data Acquisition

13.1 Introduction

The role of the DAQ system is to read and buffer data from the front-end electronics following a Level-1 trigger, to assemble complete events, and to provide storage facilities for event data and for calibration and monitoring information. The architecture of the LHCb Trigger and Data acquisition system is shown schematically in Figure 3.3, and the main parameters are summarised in Table 3.1.

The main functional components of the DAQ system are as follows:

- The Front-End Multiplexers (FEM) are responsible for multiplexing the data from many detector channels onto the front-end links. The multiplexing factor will be determined by the capacity of the links, and by the physical characteristics of the front-end electronics, i.e. the convenience with which channels can be grouped together. For the design of the readout protocols, we assume that the output of the FEMs carry zero-suppressed data in a detector independent format.

- The number of Front-End Links (FEL) has been chosen to match the expected data rate with the capabilities of the link technology, such that at least one link is allocated for each detector segment.This boundary condition is needed to take into account the segmentation and physical layout of the detectors.

- The Readout Units (RU) receive event fragments from several front-end links and assemble them into larger sub-events. Once a sub-event is assembled, the RU transfers it to the next stage for further event building. The multiplexing factor in the RU is chosen to match the bandwiths on its inputs with the output bandwidth towards the readout network.

- The Readout Network has the task of providing the connectivity and the required data transfer bandwidth such that all sub-event data belonging to a given trigger, distributed amongst all the RU's, arrive at a single destination. This destination will vary from trigger to trigger. It will thus support building of full events from the sub-events assembled previously in the RU's.

- The Sub-Farm Controller (SFC) has to assemble the sub-events arriving via the readout network into complete events. Once this is complete, it sends the event to one of the CPUs it manages. After processing by the high level triggers, the SFC has the role of dispatching accepted events, via the Readout Network, to the storage sub-system. It will also monitor its buffer usage and raise a request to slow down the trigger if the free space becomes low.

The detailed requirements [1] have been used as the basic input for design and implementation studies.

13.2 Data traffic patterns

The sizes, distributions and rates of data coming into the DAQ system on the Front-End links have been analysed using simulation data [1]. The results are summarised in Table 13.1 in terms of the average event sizes for each subdetector. The channel occupancies shown assume detector dependent processing for clustering and zero suppression. In the case of the tracking detectors an extra 30% has been added to the estimate for the average event size, in order to allow for the fact that data from two beam crossings are read for these sub-detectors. The number of segments shows the natural physical segmentation of a sub-detector. This is the number of stations in the case of tracking devices, and the lateral segmentation (inner, middle, outer) in the case of calorimeters.

The allocation of front-end links to a sub-detector will require optimisation according to the natural segmentation of the sub-detector, and the distribution of data in the segments. Table 13.1 shows a first assignment of front-end links, which have been calculated under the assumption of a 1

Table 13.1: Average event data sizes for sub-detectors and their possible segmentation

Sub-detector	Channels (10^3)	Occ (%)	Data (kB)	Segs	FE-links
Vertex	220	0.5	4	17	17
Inner Tracker	220	1.0	10	11	44
Outer Tracker	110	4.5	24	10	40
RICH 1	140	2.0	11	2	16
RICH 2	200	0.5	4	2	16
Preshower	6	2.5	1	3	3
ECAL	6	25.0	6	3	12
HCAL	3	10.0	1	2	2
Muon	45	0.5	1	5	5
Trigger			5	1	10
Total	950		67	56	165

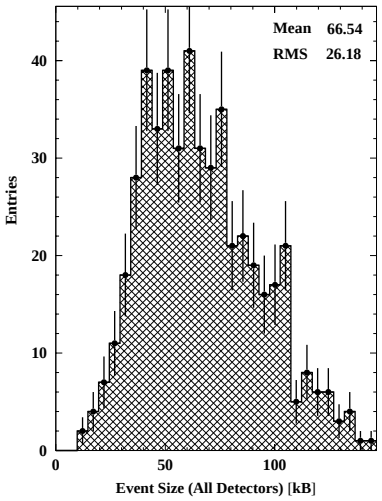

Figure 13.1: Total Event Size distribution.

Gbit/s link capacity. The allocation shown is generous with respect to bandwidth, since average link occupancies range from 10 to 30%.

Table 13.1 shows an average size of 67 kByte. There is a large variation from event to event as shown in Figure 13.1. In the readout protocol and implementation studies, an average event size of 100 kByte is assumed in order to allow for already foreseen factors, such as noise in detector electronics. Although the event building rate for LHCb is comparable to that of ATLAS and CMS (i.e. 40kHz), the average event size is a factor of ten smaller, which reduces the scale of the event building requirements by the same factor.

An average of 4 GB/sec comes from the front-end links into the DAQ. All event data goes to a single processor, but the event building requirement depends on whether all the data are immediately sent, or whether they are buffered and dispatched in several phases according to the trigger strategy. In the 'phased' approach, the event building requirement can be less than 2 GB/s.

The target Level-3 accept rate is 200 Hz. Thus the basic requirement for writing raw data is ~20 MB/s.

13.3 Design criteria

The principal criteria for the design are simplicity and ease of maintenance, scalability to meet new requirements, and the ability to follow developments in technology. A scalable architecture for the DAQ system is essential in order to allow for changes in the running conditions of the experiment that will affect trigger rates, event sizes and trigger algorithms and result in increasing demands on data bandwidth and processing requirements.

Commercial components will be used for front-end links, the readout network and the processing farm, and interfaces suitably designed to prevent dependence on a particular technology. It is important that each component can be tested and validated individually, since this greatly simplifies the integration and commissioning of the complete DAQ system.

Data transmission errors must be detected at all stages where communication links are used. Automatic detection and recovery procedures are needed to ensure the integrity of the data and the event assembly mechanism.

It is important that tests can be run in parallel on different sub-detectors, particularly during calibration and commissioning periods. The concept of partitioning will be used to permit this, and this has important consequences for the design of the DAQ system. A partition is defined to be a subset of the DAQ system that has been configured to function independently of the rest of the system. More than one partition may exist at any time thus permitting parallel data streams. Each stream will have its own set of Readout Units (RU's), Sub-Farm Controllers (SFC's) and its own independent trigger source. The facilities for distribution of trigger and timing information, event building and event storage, must support the partitioning concept.

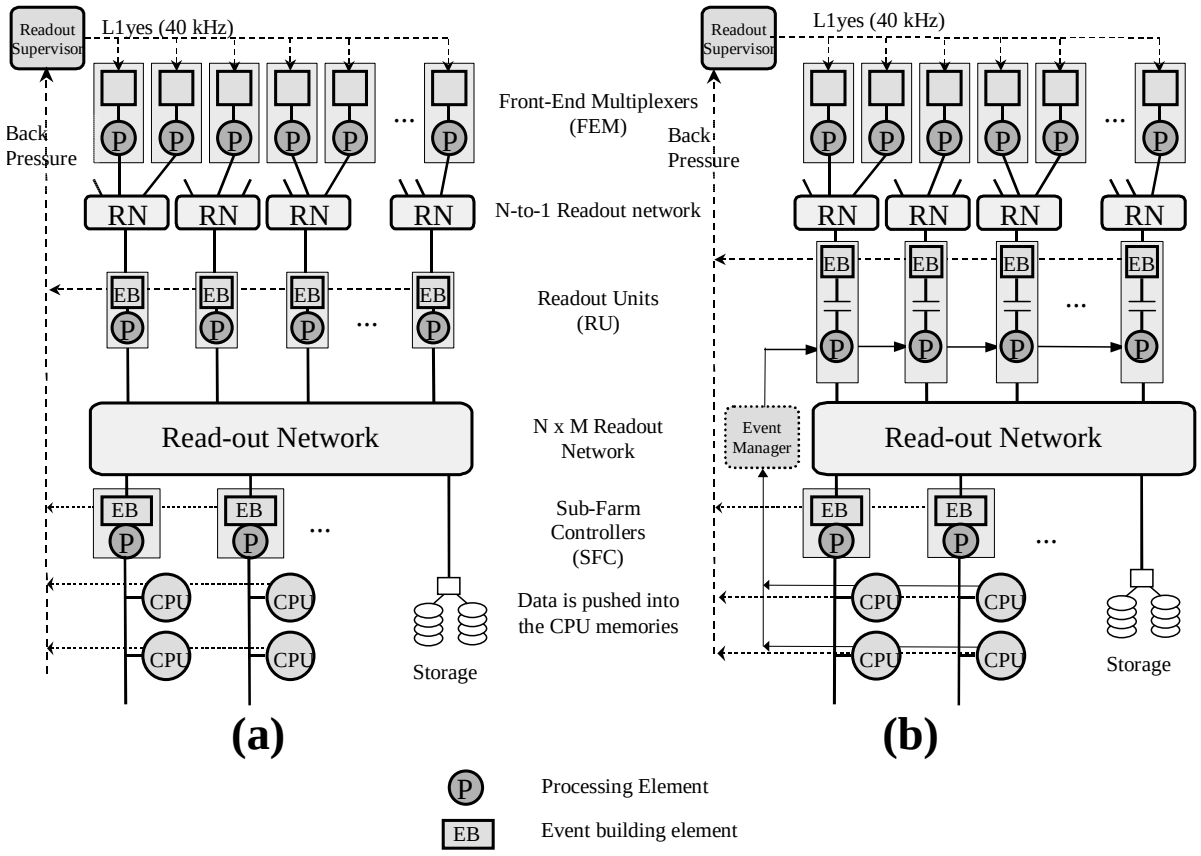

Figure 13.2: Functional elements of (a) the full-readout protocol and (b) the phased-readout protocol. Note the presence of an event manager and the explicit buffering in the readout units to bridge the latency of the Level-2 decision making process in the phased-readout option.

13.4 Readout protocol studies

Within the overall framework of the DAQ architecture, two different protocols have been studied for managing the collection of event data[2]. The first, called the 'full-readout protocol', requires that data are immediately transmitted by all RU's through the readout network, such that a complete event is directly assembled in a destination processor. In the second approach, called the 'phased-readout protocol', data from subsets of the RU's are sent in several phases corresponding to the demands of the high level trigger algorithms. The full-readout approach uses a simpler protocol but places increased bandwidth requirements on the readout network. The functional components of the two protocols are illustrated in Figure 13.2a and Figure 13.2b.

In both approaches the Front-End Multiplexers (FEM's) push data to their appropriate RU after a Level-1 accept decision. There is no further protocol between the input stage of the RU and the FEM. After the RU's the two protocols have important differences, and these are elaborated in the following sections.

13.4.1 Full-readout Protocol

Data from the FEM's are assembled in the RU and are immediately dispatched through the readout network to an SFC. The algorithm for determining the SFC address can be extremely simple, for example using the event number modulo the number of SFC's. The buffering capacities of the SFC's will be designed such that in normal operation there is always sufficient space available to receive data from the RU's. Obviously the system must also be designed to cope with bursts of data. For this a throttle signal is sent to the Readout Supervisor anticipating the buffer overflow situation, and is used to reduce the trigger rate such that the buffer occupancies can be stabilised. The latency of the throttle signal translates into buffering requirements at the different levels of the readout. First simulation results show that tens of millisec-

onds are available for the throttle to be acted upon before the buffers overflow. Hence this feed-back does not have to be a fast signal, and can even be implemented using the control network.

The SFC collects all event fragments, and once the event is complete passes it to a processor running the high level trigger algorithms. Since all data belonging to a trigger is immediately available to the high level trigger algorithms, there is no real need to distinguish between Level-2 and Level-3. Thus the high level trigger algorithms can evolve with complete freedom.

The bandwidth through the readout network required to implement this protocol is of the order of 4 GByte/s, assuming a 40 kHz Level-1 rate and 100 kByte event size.

13.4.2 Phased-readout Protocol

Using this protocol, the transfer of data from the RU's to the high level trigger processors takes place in two or more phases[1]. In the first phase, the subset of the event data that is needed by the Level-2 algorithm is transferred from the appropriate RU's to a processor. The data from the remaining RU's must be buffered for the duration of the Level-2 latency, i.e. $\sim$ 10 ms. The Level-2 decision must be transferred to all of these RU's. On reception of a Level-2 "No" decision the data are discarded. On reception of a Level-2 "Yes" decision the data must be sent to the processor that ran the Level-2 algorithm to execute further filtering algorithms (Level-3) on the complete event.

As in the case of the full-readout protocol the RU's will push the data to the SFC's in each phase, and again a throttle mechanism is required to prevent buffer overflows.

The reduction factor of the bandwidth required of the readout network depends on the rejection power of the Level-2 algorithm and the fraction of the complete event needed to execute the algorithm. At a retention of 1 event in 8, and assuming only 30% of the event data is needed by the Level-2 trigger, the required bandwidth is of order of 35% of the value for the full-readout option, i.e. $\sim$1.4 GByte/s.

The phased-readout protocol requires a new functional element, called an 'Event Manager'. Its function is to collect the Level-2 decisions and distribute them to the appropriate RU's. The presence of an event manager would permit the dynamic load balancing across the processors of the high-level trigger farm.

[1] More than two phases could be implemented. However with the current performance of the Level-2 algorithms this would result in little gain. A large number of phases would eventually result in a "data on demand" scheme.

13.4.3 Comparison of the two protocols

The full-readout protocol requires a larger scale readout network, both in terms of bandwidth and in number of RU's and SFC modules. However the complexity of these modules is less than in the phased-readout case, since the protocol is much simpler. It also has to be borne in mind that the smaller the number of RU's, the more ambitious are the requirements on the multiplexing of the data from the FEM's into the RU's.

From the point of view of simplicity of the protocols, and the flexibility it allows for the high-level trigger algorithms, the full-readout protocol is to be preferred. It is the availability of affordable high bandwidth network technologies that will determine which protocol will be implemented. The decision can be delayed for several years, during which time the high level trigger strategy will evolve, and the technology used for the readout network can be evaluated in terms of cost and market share.

In the short term, effort will be concentrated on making simulations ("virtual prototypes") of the entire readout system. Many design issues will be studied, such as the impact of transmission errors on the performance of the system and the impact of latency of the throttle on the buffering capabilities at the different levels of the readout.

13.5 Timing and trigger distribution

LHCb is the only experiment at LHC that has to distribute more than one level of trigger decision to the front-end electronics. The partitioning scheme also places specific requirements on the distribution of different trigger signals to different parts of the DAQ that may be running concurrently, asynchronously and under control of different operators.

In the R&D phase for the LHC experiments considerable work was done in the RD-12 project [3] to develop a Trigger, Timing and Control distribution system (TTC) for the LHC experiments. The TTC system comprises a transmitter that distributes timing and trigger signals over a passive optical fibre network, and a special receiver chip is under development for decoding this information and distributing it to the front-end electronics. One basic feature of this system is that it offers two

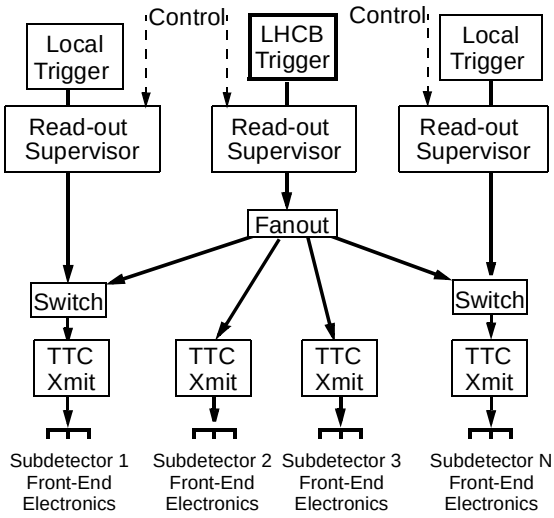

Figure 13.3: Architecture of the LHCb Timing and Trigger Distribution system. The figure shows two subdetectors with their own local trigger and two which only run with the LHCb trigger.

transmission channels (A and B) of 40 MHz bandwidth each. Channel A is reserved to transmit a clock synchronous to the LHC clock and one level of trigger decision, whereas channel B allows commands to be sent to individual receiver chips or groups of chips (broadcasts).

Careful study has shown that this TTC system can be used in the LHCb environment to distribute both Level-0 and Level-1 trigger decisions to the front-end electronics[4]. Channel A will be used to transmit the LHC clock and the Level-0 trigger decision, and channel B to transmit the Level-1 trigger decision. The scheme depicted in Figure 13.3 will be used to support partitioning. This would enable a sub-detector to run either within LHCb, or in stand-alone mode. Both the TTC transmitter and receiver chip could be used without modification. All other components, i.e. programmable switch, fan-out and readout supervisor, will be built specifically for LHCb. At least one transmitter will be deployed for each sub-detector, and a total of a few thousand receivers will be required.

13.6 Error handling

There is a non-negligible probability that data transmission errors will occur, both in the collection of the data and in the distribution of control and status information[2]. The error rate depends strongly on characteristics of the technology used to transfer the data, e.g. the frame sizes, transmission media etc. One essential requirement on the data transmission technology is the ability to detect that an error has occurred.

Transmission errors in the acquisition of data can have two types of consequence:

- Errors leading to a violation of the readout protocol through corruption of protocol information, e.g. the event number. These will result in incomplete events being assembled. The protocol has to be designed to handle these cases, usually manifesting themselves as timeouts.
- Errors leading to corruption of the contents of the event data. This will result in missing or wrong information in small pieces of data.

Since the two readout protocols are very similar, the effect of data transmission errors on their performance is also very similar. Neither of the high-level protocols described above implements a re-send mechanism in case the receiver detects an error [2]. If the expected error rate of the chosen technology is unacceptable, error correction codes can be implemented to overcome the problem. Error correction codes have a minimal impact on the total data volume ($\sim 2\%$), and allow correction of single bit errors and detection of double bit errors [2].

The most serious transmission errors are those in the TTC system, since they cannot be detected immediately but only at a later stage in the readout of the data. For example, a missing clock signal or a wrong Level-0 decision can cause a desynchronisation of the pipelines in the front-end electronics. An appropriate scheme will be devised to recover from these errors once their expected rate is measured. These measurements are currently being performed in the context of the RD-12 project.

13.7 Implementation studies

The DAQ implementation studies aim at investigating the feasibility of the proposed DAQ architecture and protocols, identifying the critical components and suggesting strategies for their implementation. Different technologies have been con-

[2] There are technologies, e.g. SCI, that implement an automatic re-transmission of data at the physical level when an error is detected.

sidered : ATM and SCI which are well advanced standards, the Ethernet family for which Gigabit Ethernet is an emerging standard, and Myrinet a single company solution. A detailed account of the implementation studies and of an ATM implementation is given in [5]. A scheme based on SCI is proposed in [6].

13.7.1 Requirements

The following assumptions on data traffic patterns have been used in simulation studies of the implementation. The data distribution by detector and front-end link is obtained by normalizing the values from Table 13.1 to a total event size of 100 KByte. The Level-1 trigger frequency is taken to be 40 kHz. The inter trigger delay is variable, but is not of great importance, since the DAQ system provides several levels of de-randomizing buffers. The latency of Level-2 is assumed to be 10 msec, and for Level-3 200 msec. For phased event building, the hypothesis is ∼40% of data required for the Level-2 algorithm and a rejection factor of 8.

Safety Factor: The figures above characterize the working point of the DAQ system. To provide a safety factor, it is required that the load at the working point should represent 50% of the maximum load that the DAQ system is able to sustain. A load of 100% would be realised by increasing the trigger frequency by k_F and the event size by k_E with $k_E \times k_F = 2$.

13.7.2 Implementation model

The event building system (Figure 13.4) consists of sources (i.e. RU's) and destinations (i.e. SFC's) connected by a single network. RU's are all identical but allow for a variable number of Front-end Link receivers. The destination modules are all identical as well. The functionality of each component is summarised below.

Readout Units (RU)

A Readout Unit collects data from one or more front-end links, trying to achieve a load of its network link of ∼50%. The determination of the multiplexing factor depends on the front-end link data rate, on the bandwidth connecting the RU to the network, and on the requirement of partitioning that forbids links from different sub-detectors to be mixed. These constraints result in an average load somewhat lower than 50%.

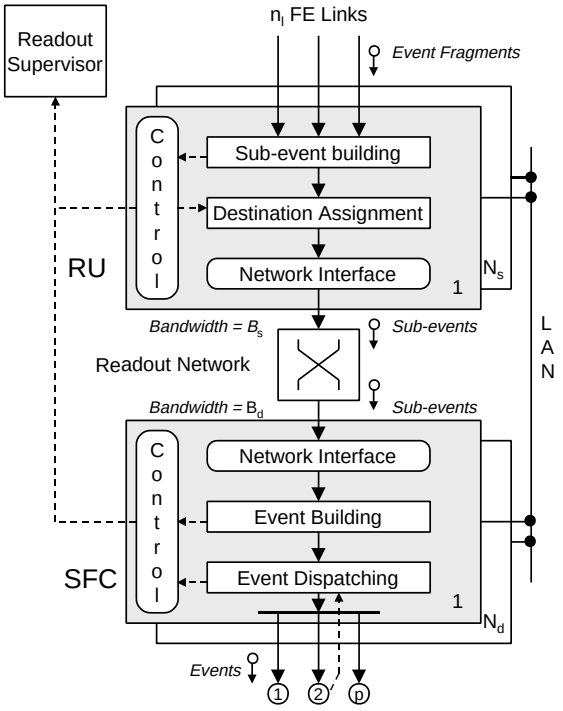

Figure 13.4: Implementation and functional model of the event building system.

The RU's have to cope with a total frequency of event fragments several times higher than the Level-1 rate. For example, a network providing a bandwidth of 1 Gb^{-1} can require a multiplexing factor of 4 in some sources, thus leading to a rate of event fragments at 160 kHz. Higher bandwidths lead to higher multiplexing factors. The implementation of RU's must rely on hardware solutions, or at least on dedicated processors, to sustain such rates. A detailed table giving the assignment of Front-end Links by detector can be found in [5].

The assignment of a sub-event to a destination can be determined locally on the basis of a look-up table or by a dedicated controller that collects information on the status of the SFC's and event processing units. Although the first solution is distributed, it requires a "slow" backpressure control from the destination to the readout supervisor in order to control the Level-1 trigger rate thus avoiding buffer overflow in a destination. Another solution based on a central controller allows the load to be balanced while keeping an optimal overall throughput. It can easily be implemented if the control messages are carried by the same data network as the main DAQ stream. Possible implementations of the event building control are discussed in detail in [5].

Table 13.2: Number of SFC modules (∼50% load on link)

Network Technology	Full Readout		Phased Readout	
	# of SFC's	CPU/ SFC	# of SFC's	CPU/ SFC
Ethernet 100 Mbs^{-1}	800	2 - 3	400	5
ATM 155 Mbs^{-1}	400	5	250	8
ATM 622 Mbs^{-1}	120	17	60	34
Ethernet 1000 Mbs^{-1}	80	25	35	57
Myrinet 1280 Mbs^{-1}	56	36	27	73
SCI 4096 Mbs^{-1}	132	15 - 16	67	30

If phased event building is implemented, the source modules must implement a buffering scheme where Level-3 data are stored until a Level-2 decision is made. The additional buffer space required is modest: 1–2 MByte per RU (assuming event building latency of a few msec, processing time of 10 ms and the highest rate of 80 kHz); it represents a shift of buffer capacity from the sub-farms to the RU's.

Sub-Farm Controllers (SFC)

The SFC has the task of assembling sub-events to full events and implements the event building protocol that copes with sub-event losses. It must also implement the partial event building scheme required by the partitioning requirements.

In a simple model, the SFC dispatches full events to one of the processors in the sub-farm that it controls, the individual processors not being "visible" from the RU's. This gives the possibility of implementing a local load balancing that issues an overflow warning only when all the local resources become scarce.

The number of SFC's is not constrained by bandwidth considerations. However the number of processors required to process the flux of incoming events in an SFC is proportional to the readout network link bandwidth and is determined from the average processing time per event and the number of events per second arriving in the destination. Table 13.2 gives the number of destinations for various technologies, assuming a bandwidth occupancy of ∼50%. The table has an entry for both protocols.

Readout Network

The main function of the network is to route all the parts of a given event to one destination. As such it participates in the event building process. The concentration of data towards one destination results in contention for shared resources. Depending on the mechanism used by a particular technology to resolve contention, this may lead to reducing the usage of the available bandwidth, thus limiting the maximum load that the network can tolerate. This is an important issue that can be studied by simulation.

The network can also be used for routing control messages. The problem of contention is not important since most messages travel in the opposite direction to the data transfer.

The main advantage of phased event building is to reduce the network size by a factor 2-3, depending on the data requirements for Level-2 and on the rejection factor. The cost is a higher multiplexing of Front-end Links for Level-3 data, and a more complex control of the Level-3 RU's, the RU having to implement an event retrieval or discard logic.

13.7.3 An ATM implementation

The RU's are connected to the readout network by 622 Mbs^{-1} links. The Level-2 RU's need to multiplex at most 2 Front-end Links. In the phased event building case, a multiplexing factor of up to 4 is required for the RU's providing data for the Level-3 trigger. For the destinations, we have selected links at 155 Mbs^{-1} to have a small number of processors in the sub-farms, possibly using 'commodity' symmetric multiprocessors (note however that phased event building requires 8×1000 MIPS processors per sub-farm). The average load on a destination link is ∼50%, at the normal working point conditions. The network consists of interconnected switches offering 10 Gbs^{-1} aggregate bandwidth each, or the equivalent of 64 ports at 155 Mbs^{-1}. Table 13.3 gives the network characteristics for both event building protocols.

Simulations of roughly equivalent networks, with ports at 155 Mbs^{-1} only, have been run: a 1024 ports switching network for the full event building and 512 ports for the phased event building, equally distributed in both cases between sources and destinations. Results show that no congestion occurs for events of 100 KByte at frequencies up to at least 70 kHz (Figure 13.5). The buffer occupancy inside the network due to contention is much lower than the capacities usually provided by commercial

Table 13.3: Number of RU's and SFC's for the ATM scenario

	Full-readout	Phased-readout
# of L2 RU's (622 Mbs^{-1})	125	50
# of L3 RU's (155 Mbs^{-1})	–	60
# of SFC's (155 Mbs^{-1})	494	244

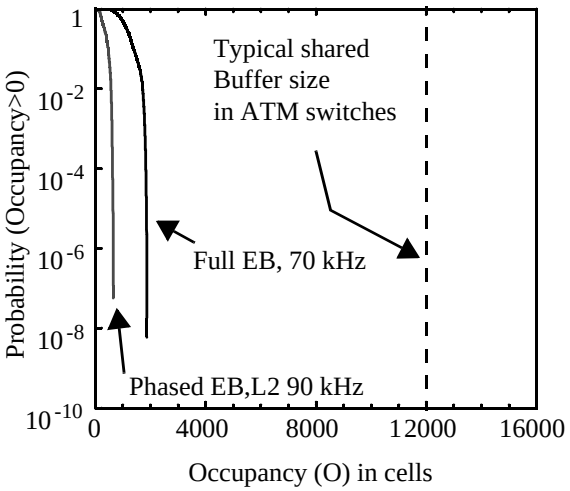

Figure 13.5: Tail distribution of Queue occupancy in switching nodes.

switches.

13.7.4 Outlook

This example of an implementation enhances our confidence that the proposed DAQ architecture is feasible. Alternative or emerging networking technologies might provide equally valuable solutions and will deserve further studies. There are 2 important subjects for further investigation. The first is a detailed study of the feasibility of RU's, mainly from the point of view of sub-event building. Reference [5] suggests a layout that could easily be studied without hardware developments. The second field of investigation is network simulation adapted in particular to technologies other than ATM.

13.8 Detector Control System

The Detector Control System (DCS) will be used to monitor and control the operational state of the LHCb detector and the associated experimental equipment such as gas systems, high voltages, and readout electronics. The system should be able to operate the experiment as a whole from the control room during physics data-taking periods, but it should also allow the operation of the different sub-systems in a stand-alone manner at other times. The DCS will also acquire slowly changing data from the detector (for instance environmental parameters such as temperatures, positions, etc.), and store them permanently to be accessed by the reconstruction and analysis programs.

The number of control and monitoring channels will be of the order of many tens of thousands. To handle this complexity, these channels will need to be grouped into logical devices and devices will be grouped into sub-systems. Each sub-system should offer only summary data and high level commands to other sub-systems, and to the operator.

Robustness and reliability are important requirements for the DCS. The DCS must be fully operational at all times, regardless of the state of the experiment, of the DAQ system or of the LHC accelerator. The DCS should be well integrated with the DAQ system, since both need to exchange information and commands, and to ensure that a coherent interface is provided for control and status display in the control room.

13.8.1 The DCS architecture

The architecture of the DCS inplementation is shown schematically in Figure 13.6. It is a distributed control system consisting of control stations (I/O servers) and general-purpose workstations or servers, interconnected by a local area network. The equipment is connected to the control stations by means of a field bus or by separated analog or digital lines. The control stations will constantly read all I/O channels and report to clients running on the workstations and servers only when changes are detected. Specialized process control stations will be implemented using commercial programmable logical controllers (PLC). They will also be connected to control stations.

A key feature of the software architecture is the central database. This database will contain all the configuration parameters for the complete system including the configuration of the data points, relationships between equipment, data presentation parameters, graphics, etc.

13.8.2 Organization and planning

The DCS will be part of the computing project of LHCb. As for the DAQ, the central DCS will provide the frameworks and the infrastructure to the

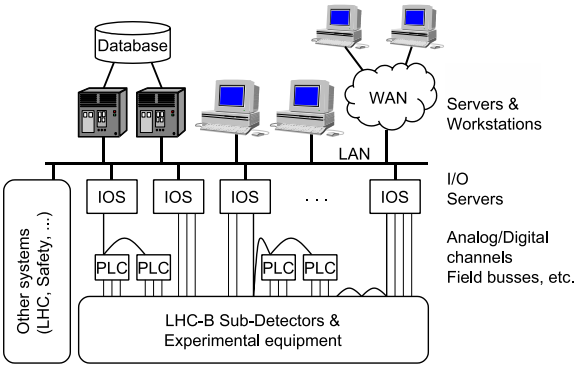

Figure 13.6: Hardware architecture for the Detector Control System.

sub-detectors groups which will need to configure and develop the specific control functions for their sub-detector.

A common project for the DCS for all LHC experiments has recently started. The objective of this project is to explore ways of providing a common DCS for all 4 experiments in a resource effective manner and recommend and support which control system to use. LHCb is participating in the project, and is currently engaged in writing the user requirement document for the DCS kernel [7].

References

[1] F.Harris, M.Frank, "LHCb Data Flow Requirements", LHCb 98-027 DAQ.

[2] B.Jost, "DAQ Architecture and Readout Protocols", LHCb 98-028 DAQ.

[3] O. Villalobos Baille et al., "Timing Trigger and Control Systems for LHC Detectors - Status Report on the RD12 Project", CERN/LHCC 97-29, 1997.

[4] B.Jost, "Trigger and Timing Distribution in LHCb", LHCb 98-031 DAQ.

[5] J.P.Dufey, I. Mandjavitze, "DAQ Implementation Studies", LHCb 98-029 DAQ.

[6] H.Müller, "SCI Implementation Study for LHCb Data Acquisition", LHCb 98-30 DAQ.

[7] P.Mato, "Detector Control System for an LHC experiment", LHCb 98-005 DAQ.