Technical Design Report
For an
Ethernet-Based Data Acquisition System
For the
D∅ Experiment

Technical Design Report
For an
Ethernet-Based Data Acquisition System
For the
D∅ Experiment

1.0 Introduction

An Ethernet-based data acquisition system, for which all active components are commercially available, is described. The VME crates are read out using Single Board Computers (SBC) running Linux, and the data is transmitted through network switches to the Level 3 filter nodes. A commodities based system has manageable schedule risk since the resources and personnel required to support the project are widely available. Such a system also presents the advantage of being very flexible in the light of possible future upgrades.

This document is organized as follows: in Section 2, the D∅ readout architecture and the existing implementations are explained. Section 3 is a detailed description of both hardware and software for the ethernet-based system, and Section 4 proposes a method for a smooth transition between the existing and the new system. The Work Breakdown Structure (WBS) is given in Section 5, with the methods used to determine the effort needed to complete the various tasks explained in Section 6. Section 7 describes the derivation of the cost estimate.

2.0 DØ Readout Specifics

2.1 General Architecture

In this document, a readout crate is understood to be a crate, which provides raw event data from the detector. To accommodate a 7.5 MHz collision rate, DØ uses a buffered data acquisition system, in which the Trigger Framework (TFW) drives synchronization of event fragments from different readout crates. Therefore, each of the readout crates is required to have a crate controller, which needs to be able to receive the TFW synchronization information. The latter is distributed over the Serial Command Link (SCL), and in addition to various timing and control signals, contains a 16-bit monotonically increasing Level 3 transfer number for events, which were accepted by Level 2. It should be noted that the SCL does not transmit information indicating which trigger(s) passed for an event.

In the Brown-ZRL or custom design, all crates are read out by VME Buffer Drivers (VBD) which have two 256 kB buffers, and take over the VME bus when the crate controller requests that the crate be read out. In principle, any alternate readout mechanism that can be made to emulate one of the two VBD modes of operation could be used to read out the DØ detector.

2.2 Implementations

There are four different versions of VME crate readout used in the experiment: calorimeter; muon detector; Level 2 trigger; tracking and Level 1 (the last two use a common crate design).

The calorimeter crates all have a module that receives the SCL signal, and ADC modules. Unlike the other systems, the ADC readout does not proceed by direct memory access (DMA): each module explicitly specifies the length of the data to be read. There are no processors in the calorimeter readout crates: the crate controllers are located in a different crate. The calorimeter occupancy measured in Run II a is between 5 and 7 %. The occupancy will rise with luminosity. A conservative factor of 2 to 3 increase in event size will be assumed for Run II b. Calorimeter occupancy of 20 % corresponds to 4 kB per event per crate, or 4 MBps at 1 kHz.

The slave modules in the muon detector are Muon Readout Cards (MRC). The Muon Fanout Card (MFC) receives the SCL signals. A VME processor located in the crate and resident on the J2 bus controls the MFC. The event size measured per muon crate is about 0.5 kB + 0.1 kB per minimum bias interaction. Even at very high luminosities, this remains small (< 2-3 kB/event).

In the Level 2 system, both crate controllers and many of the slaves have DEC alpha processors. The average event size in Level 2 crates is of the order of 0.5 kB per event. Projections estimate this could increase by up to 40 % in Run II b, but this remains small compared to other systems.

Both the tracking detectors and the Level 1 crates use VME Readout Buffers (VRB) to store the data while waiting for a Level 2 decision. Each readout crate has a VRB Controller (VRBC), which receives the SCL signal and communicates with the VRBs and VBD. The average event size in tracking crates can reach 5 kB per event, and grows linearly with occupancy. It should be noted, however, that dead time increases with the event size, forcing the latter to remain well below the 10 kB mark.

System	Calorimeter	Muon	Level 2	Tracking and L1
Run II a event size	< 2 kB	< 1 kB	< 1 kB	< 6 kB
Run II b event size	< 5 kB	< 3 kB	< 2 kB	< 10 kB

Summary of event sizes per crate in the various systems, both for Run II a and Run II b.

3.0 A Commodity Readout System

A system composed entirely of commercially available components (except for a passive extender board required to physically attach components to the existing VME system) can be used to read out the DØ detector and build events directly into the memory of an appropriate Level 3 node. The main components are Single Board Computers (SBC), network switches and the existing Level 3 filter nodes. The following figure is a schematic illustration of the system.

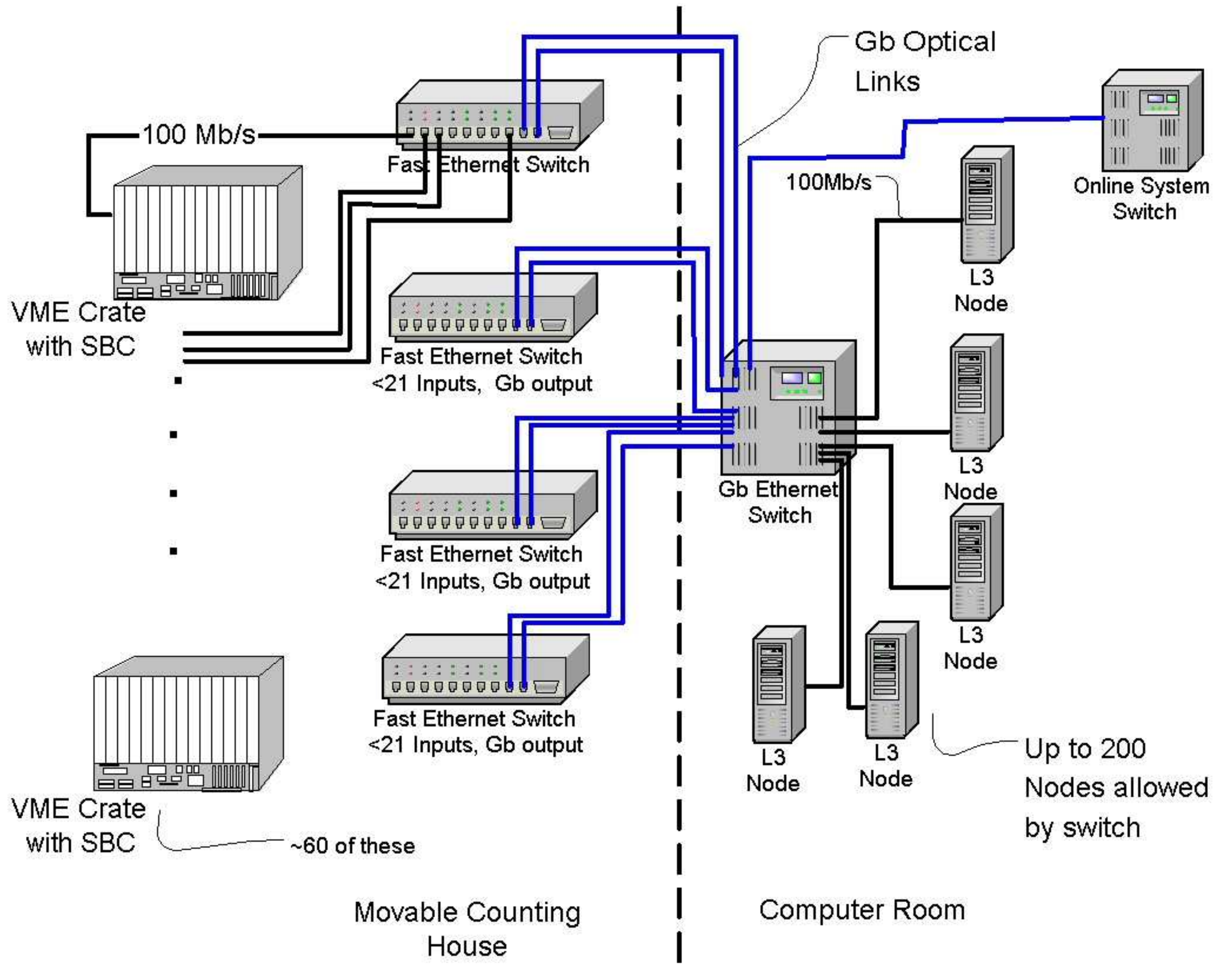

Figure 1: An Ethernet based DAQ system

Event data are read from each VME bus by a SBC, and sent to a Level 3 node according to routing instructions received from the Routing Master (RM) program, which runs on the SBC reading out the Trigger Framework (TFW). The Routing Master chooses a node for event processing based on the Level 1 and Level 2 triggers satisfied, and a list of available buffers for each event type. Event building is done in the Level 3 node before passing the event through the Level 3 filtering process.

Detailed descriptions of the hardware and software components follow.

3.1 Hardware

3.1.1 Single Board Computers

Single board computers have two applications in the system: As readout controllers for event data (playing the role of the VBD in the custom system), and as a host for the routing master.

When used as a readout controller the functions of a SBC are:

1)Using VME, read event data into its main memory.
2)Using a TTL output level and a TTL input level, handshake with the readout modules.
3)Buffer the event data in memory while waiting for routing information from the routing master.
4)Receive routing data from the routing master via Ethernet
5)Transmit event data to the appropriate Level 3 node via Ethernet.

When used as the routing master the functions of the SBC are:

1)Using VME, read the trigger framework data into its main memory.
2)Using an Ethernet interface, receive "buffer free" messages from the Level 3 farm.
3)Run the routing master algorithm which decides which events are to go to which Level 3 nodes
4)Transmit the routing information to all the Single Board Computers functioning as readout controllers.

These functions require that the SBC's have ample memory, fast VME access, very good processor speed, 100 mbps Ethernet, and two channels of digital I/O.

Particular care has been taken in surveying the VME interfaces available. Most new designs use the Tundra Universe II chip for the VME-PCI interface. The Universe II is a fully featured chip - it is compliant with 32 or 64-bit PCI bus architecture, has a programmable DMA controller, offers flexible interrupt logic and allows address pipelining, required for calorimeter readout. Internally, the chip has independent FIFOs for inbound, outbound, and DMA traffic. Evaluations of the chipset on a VMIC 7750 show a slower handshake but faster data transfer than the existing VBD readout controllers. Thus we will use a SBC with this chip.

Performance tests for VME readout have shown that the data transfer from the VME bus is likely to be the limiting bandwidth factor for the board. Using 32 bit VME transfers, the rate achieved for calorimeter crates is 4 bytes per 300 ns, or 13.3 Mbps. For all other crates, this goes up to 36 MB/s because transfers are made using block transfers, and 64 bit data transfers are used. However, only the tracking detectors produce a data volume which could approach 10 MB/sec, so that one 100 mbps (> 11 MB/sec) Ethernet output will be able to handle the data flow. If at a future date, higher readout rates are required, then the second fast Ethernet link on the board can be used, or a gigabit Ethernet card can be added. This means that a safety factor larger than 4 is available using current, readily available Ethernet technology.

An integrated performance test has been performed by VME readout of a video memory, and output of all event data over Ethernet. This test has been performed using an older SBC with a 200 MHz processor, and a data transfer rate of 7 MB/sec was achieved at full CPU load.

The VMIC 7750 single board computer has the features described above. When this board is combined with an Acromag PMC470 digital I/O card, the necessary digital I/O functions can be performed. This equipment is currently being used for a feasibility test of the design principles in this TDR. The card has a 933 MHz Pentium III processor, and a second, built-in 100 mbps Ethernet port available if a second control network is needed.

3.1.2 Extender Board for the SBCs

This is the sole custom component in the system. Its main function is to mechanically support the 6U commercial single board computers in the 9U readout crates, and to propagate signals. All electrical components are passive. The boards have already been designed and a preproduction run manufactured.

3.1.3 Cisco 2948 G

This is an Ethernet switch with 48 100 mbps ports and two Gigabit Ethernet ports manufactured by Cisco. Four of these switches will reside in the moving counting house. Their function in the system is to convert the copper 100 mbps Ethernet media connected to the SBC's to optical Gigabit Ethernet media. This allows the system to meet the requirement that all data signal leave the moving counting house via optical technologies, and reduces the number of lines between the moving and fixed counting houses to eight.

Using 100 mbps Ethernet also allows for greater flexibility in choosing the SBC, as 100 mbps Ethernet is a readily available feature on these types of computers. The switch is also economical and easily maintained: The switch adds less than $200 per port to the system, a substantial saving compared to a pure Gigabit Ethernet implementation. It also eases system diagnostics as any single Ethernet port can be monitored with modest 100 mbps monitoring equipment.

These intermediate switches must add only trivial congestion or packet loss to the system, lest performance be affected. Each switch has two optical Gigabit Ethernet ports available to transmit data to the fixed counting house. Since we desire no congestion in the event building network, we must limit the bandwidth into each 2948G to no more than 2 Gigabits per second. An implementation guaranteeing this is to partition the switch into two, each partition having a Gigabit output Ethernet, and no more than ten 100 mbps input Ethernet ports. In this configuration, the switch is non-blocking and congestion cannot occur in the transmit buffers of the Gigabit Ethernets, since the rates are matched. Therefore packet loss should be reduced to levels that do not affect performance.

The design has five 2948G switches for 65 crates, which should easily accommodate the load. For further safety, contingency on these switches has been set to 100%, so that more can be added in case needed.

3.1.4 Cisco 6509

The function of the Cisco 6509 switch is to move Ethernet packets containing event fragments to the Level 3 nodes without significant packet loss. The salient specifications of the switch are its bandwidth, buffering capability, and port count.

<u>Bandwidth:</u> The manufacturer's specified bandwidth for the Cisco 6509 switch fabric is 256 Gigabits/sec. It is an industry convention to compute this number by counting both "takeoffs and landings", so it is perhaps fairer to state the effective bandwidth as 128 gigabits/sec or 16 GB/sec. The D∅ L3 data flow requirement for Run II b is < 1GB/sec (~ 250 KB events @ very few x 1000 Hz). Thus the chosen switch has a safety margin of a factor of 16 in bandwidth over the D∅ requirement

and the switch will not saturate. In terms of Ethernet packets, the 6509 can transmit 100 Mpps, which means the switch is capable of achieving full bandwidth for packets that are only 10% of the standard maximal size (1580 bytes). Since over 50% of the packets transmitted in this design are fully loaded, the safety factor here is well over a factor of 20.

Buffering: The surplus of bandwidth in the switch leads to the transmission of Ethernet packets to the appropriate 100 Mbps output port without loss. At the output port, the packets are either buffered for transmission, or discarded should buffering be unavailable. We will use the "Version 4" blade that has 112 MB of output buffering for 48 100 Mbps ports, or over 2 MB per port, which is far greater than the event size. This allows for all SBC's to transmit their data simultaneously without traffic control shaping.

Switch performance has been studied in earlier versions of the 48-port, 100 Mbps blade. These versions had 56 kB and 112 kB of dedicated transmit buffering, and similarly modest amounts of shared buffering. The tests confirmed that congestion could be controlled with the amount of data present in a blade's transmit buffers.

Port Count: The bandwidth presented to the Level 3 system constrains the bandwidth needed in the switch and the number of input gigabit Ethernet ports. The 6509 switch is extended by the addition of "blades", with a nine blade maximum, the proposed system uses just three blades, meaning that the system can be scaled up by an additional 288 additional Level 3 filter nodes. Cisco also manufactures a chassis in the 6500 series switches capable of holding thirteen blades.

3.1.5 Level 3 Nodes

Forty-eight Level 3 nodes were purchased in summer 2001. These are dual 1 GHz Pentium III PCs with GB RAM and two 100 mbps Ethernet ports. Event building is a very simple process of concatenation of data blocks, and is not expected to consume more than 5 to 10% CPU time on these nodes. Should event building turn out to require more resources, then additional Level 3 nodes can be purchased to provide the same CPU power to the filtering process. The natural number of nodes to purchase is 16, corresponding to one rack, which corresponds to a 33% increase in processing power.

3.2 Data and Control Flow

This section describes information flow through the system. A detailed description of each of the software components follows in the next section. The flow of information through the system is shown in the following figure:

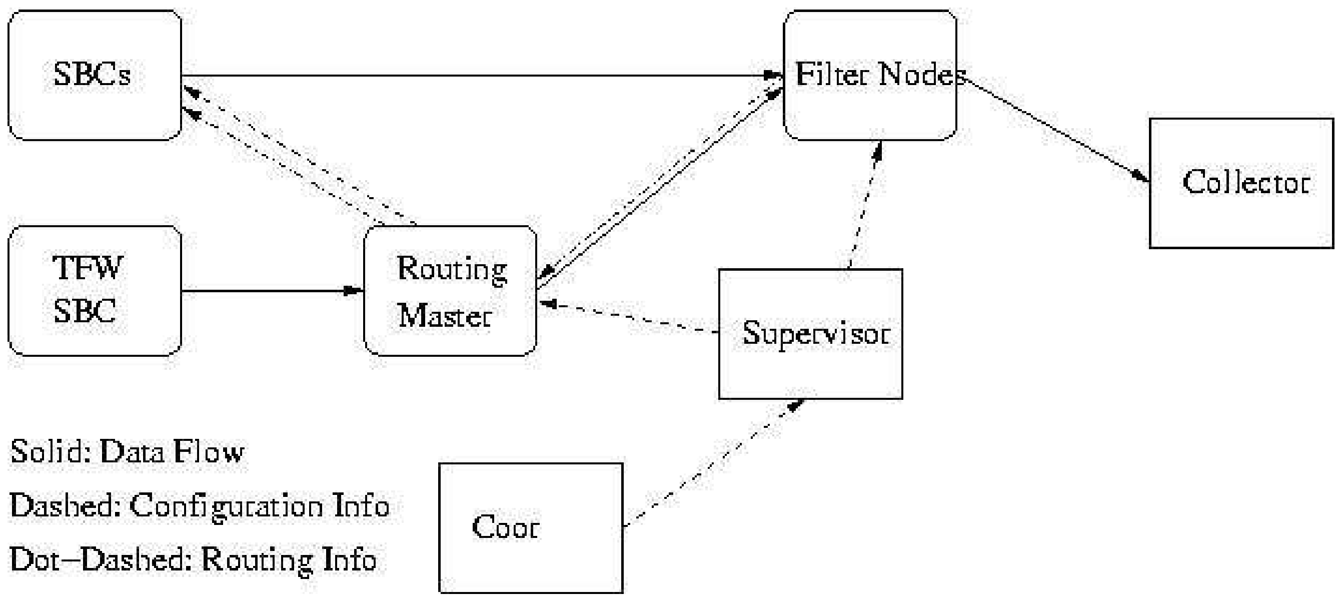

Figure 2: Data flow diagram

Solid lines represent data flowing from the SBCs to the Level 3 nodes. The routing master, shown in a separate block, is a process running on the SBC reading out the TFW crate. Dashed lines show the configuration and control information flow, from COOR (the run control software) to the supervisor, and from the supervisor on to the Level 3 nodes on one hand and to the routing master on the other hand. The routing master further configures the SBCs. Routing information is shown in dot-dashed lines: after making a decision, the routing master sends the relevant information to the SBCs, and at regular intervals it receives the number of available buffers from each Level 3 node. For clarity, monitoring information is omitted from the drawing.

3.2.1 Configuration

At the start of a run, COOR sends messages to the Level 3 supervisor indicating the run type (i.e. the code that needs to be run on the nodes for the run), the number of nodes, the crates to be read out in the run, and trigger bit information.

The supervisor then initializes the appropriate number of nodes, and passes on all the trigger information, including which crates should be present for each Level 1 and Level 2 trigger. The supervisor also informs the routing master which nodes are available for this trigger bit mask.

The routing master then adds these nodes addresses to the node array and communicates that information to the SBCs (except for the TFW SBC). Another option here would be to send the full list of node addresses at the start of the first run.

Similarly, at the end of a run, when COOR informs the supervisor a run is over, the supervisor deallocates the relevant nodes, and informs the routing master that that trigger bit mask is no longer active and the corresponding nodes are not available. The routing master will either instruct the SBCs to remove these nodes from their internal list, or will stop sending routing information using the nodes to the SBCs.

3.2.2 Event Data Flow

Event Data is read from the VME bus by the SBCs and sent to the filter nodes according to the routing information received. All SBCs communicate with all filter nodes, and the TFW SBC is special in that it also receives data on available event buffers from the filter nodes.

Based on this data, and the triggers fired, the routing master chooses an available node, sends this routing information to the SBCs, and sends the TFW data on to the appropriate filter node. This scheme offers the advantage of providing a natural way for the routing master to determine whether a node is busy or dead.

3.3 Software Components

3.3.1 COOR

For the purposes of this document, it is sufficient to mention that COOR, the run control software, transmits all its messages using ITC, a TCP-based communications protocol developed by the Fermilab Computing Division.
Information about which crates are to be read out in a particular run is already transmitted from COOR to Level 3, so no additional development is needed here.

3.3.2 Supervisor

The supervisor acts as the interface between COOR and the Level 3 system. At the start of a run, it receives configuration information from COOR, which specifies the number of nodes to allocate to the run, the type of code to run on the nodes, the crates that should read out for events in the run, and the trigger list. The latter includes the Level 1 and Level 2 trigger bit mask, and the Level 3 trigger list. To start the run, the supervisor allocates and initializes the appropriate nodes, and transmits the trigger information, including the crates to be read out for each trigger bit. It transmits the trigger bit mask to the routing master along with the addresses of the nodes configured for the events coming in on one of those triggers.

The supervisor is also responsible for deallocation of nodes at the end of a run, and transmission of additional trigger information during the course of a run.

The supervisor emulation, which has been developed for the linux filtering farm, can communicate with COOR, allocate and deallocate nodes, and transmit the relevant trigger information to the nodes. Essentially no error handling exists, and multi-run capability is being developed with low priority since this is for testing purposes only. Additional work is required for implementation of communications with the routing master (one week FTE), the implementation of multi-run capability (four weeks FTE), testing in a real environment, and the implementation of error recovery. While the latter is likely to continue at a low level for a long time, it seems reasonable to estimate that reliable operation could be reached within 1 month FTE. The total effort required for the supervisor is then nine weeks FTE. Alternatively, the existing NT supervisor could be modified to provide the functionality with a similar time estimate to completion.

3.3.3 Routing Master

At initialization time, the routing master transmits a master node index to the SBCs, associating each receiving process on a filter node with a simple number. When it receives run start information from the supervisor, it associates the indicated node addresses with the received trigger bit mask. When an event passes one (or more) of the triggers in that run, the routing master compares the fired trigger bit mask with the trigger bit masks in memory, and chooses an available filter node capable of handling this event for processing. It sends the TFW data to that node and the node index along with the event's Level 3 transfer number (a total of 32 bits) to the other SBCs (each message to the SBCs is likely to contain information for multiple events).

It should be noted that the algorithmic part of the routing master is very similar to the existing Event Tag Generator (ETG) emulator. Code that makes routing decisions based on Trigger Framework (TFW) information exists, and is fast enough.

This component's basic functionality is fairly easy to implement, but it will require some careful tuning since it drives the intrinsic system latency, a key issue that will determine the system's ability to recover from problems. We estimate that it will take three FTE weeks to implement the functionality, with a conservative estimate of one additional FTE month to bring the software to a state of reliable operation with low latency (total: seven weeks FTE).

3.3.4 SBC Readout Software

Given a Level 2 accept, the readout controller software acquires event data from a VME crate, and places this data into shared memory. Within the next 0.5 seconds, the software acquires routing information from the routing master. The routing data is placed into shared memory, and events having routing information are sent to the appropriate Level 3 node. After an event is transmitted the corresponding shared memory buffering resources are freed.

The design requires configurable physical memory buffers. The VMIC 7750 board has 128 MB of memory, most of which is available for this purpose. Under normal running conditions, this constitutes five or more seconds of event buffering.

<u>VME Data Acquisition</u>: The SBC runs LINUX, which is not a true real time operating system. Therefore it is important that event data be acquired efficiently and with deterministic timing, so that no dead time is generated. To achieve this, VME data acquisition is implemented not as a conventional UNIX process, but as a series of interrupt service routines. Therefore, jitter is limited to interrupt latency times, and not LINUX scheduler latency times.

The readout handshake signals are transmitted on the J3 bus. This requires the addition to the SBC of a digital I/O card, wired to this bus. This card and the Universe II chip provide the interrupts driving the readout.

Routing Information: At the beginning of a run, the Routing Information Receiver (RIR) receives the master node index table from the routing master, and places it into user space. Subsequently, it waits for routing information from the routing master, and places it in shared memory.

Event Transmission: The Event Dispatcher (ED) process runs in user space. At the start of a run, after the master node index table has been received, this process opens connections to all the filter nodes. When routing information is available it sends all the event fragments for which it received the destination node, and goes into a waiting mode.

In addition to these, there is an information process keeping track of the SBC status, and exporting it to TBD.

Software development work is supported portably by a network receiver process, which receives event fragments and sends routing information.

Note that the TFW SBC is special since it also runs the routing master and all the associated communication lines.

3.3.5 Level 3 Nodes

Each Level 3 filter node runs event building, event IO, and filtering software. The event building software needs to be developed, while the event IO and filtering software exist, and need no dedicated modifications. The expected event rate to a given Level 3 node is 20 Hz.

On initialization, the event builder receives the trigger bit mask it will process, and for each trigger, the list of SBCs from which it should receive event fragments. The event builder establishes connections to the SBCs, holding these connections open for the duration of the run. At the start of a run and again when its input queue hits the "almost empty" watermark, the event builder notifies the routing master of the number of complete events it can receive simultaneously. This conservative estimate is based on the event processing time.

Event fragments are transmitted using ITC. In the event builder, ITC software receives event data fragments into a managed memory heap. The core event builder process places pointers to these fragments, sorted by crate number, into event fragment lists. It asserts that the event number is within some reasonable range from the "current event" and that the crate number is not a duplicate, and is in the crate list for this type of trigger. By maintaining lists of fragment pointers, the event builder avoids additional intermediate memory copy operations of event data.

After a time-out, if event fragments are missing (i.e. some crates are still missing) then an error message is produced and the event is dropped.

Otherwise, when the event is complete, its fragment list is put on the output complete event queue waiting for space in shared memory. When shared memory is available, a block of memory of exact size is allocated, the event is built, and the space holding the fragments is freed. Finally, the event is made available to the event IO process. Multiple events are processed in this way simultaneously.

Development of the code is estimated to take three weeks FTE, to which an equivalent amount of time has to be added for fine-tuning and debugging in an integrated environment.
It should be noted that on a 400 MHz machine, receiving 5 MBps from 50 sources takes about 30% cpu time when using `sigio()`, (and up to 80 % when using `select()`).

3.4 System Feasibility ("Slice") Test

To demonstrate the system's feasibility, development of a full slice of the system was started in late August 2001. This so-called "slice" test has 10 SBCs, a Cisco 2948 G switch, a Cisco 6509 switch with a version 4 blade, and makes use of some of the existing Level 3 farm nodes. The major risks to be investigated on the hardware side are a) VME integration problems with some of the readout modules, b) insufficient performance by some of the network elements, and c) generation of coherent noise in the calorimeter ADCs by the SBCs. Software-wise, the test is an important factor in the estimation of the effort needed for development and integration, and will expose a) insufficient understanding of the system itself, and b) bugs in both existing and new software.

The first SBCs were received at the end of October, simultaneously with the Cisco 2948G and the version 4 blade, which was installed in a spare 6509. At this stage (November 26th 2001), the following milestones have been achieved:

- Stable readout of a calorimeter crate over a period of days at 500 Hz (limited by the TFW readout).
- Stable readout of a tracking crate over a period of days at 7 kHz (using internal event generation).
- Stable dual readout of the TFW crate over a period of weeks. Here dual readout means that for each event the data is read by both an SBC and the VBD. This mode of operation is essential for a smooth transition between the existing system and the proposed one (see below).
- Coordinated readout of the TFW and a calorimeter crate with routing and event building. This establishes the viability of the software design: both the connection setup and data flow proceed as expected.

Further short-term tests include:

- Readout of a muon crate.
- Event building with more than 2 crates and passing on of the event to the filtering process.
- Examination of calorimeter data for SBC-induced coherent noise.
- Attempting to congest the network between the 2948G inputs and the Level 3 nodes.

The slice test will demonstrate that none of the potential hardware risks are real, and that the occurrence of remaining software problems is reduced to an acceptable frequency for initial running.

4.0 Transition from the Existing System

The transition from the existing system to the commodity system requires special attention because the DAQ downtime needs to be minimized. Detector readout is required not only during physics running, but also during longer shutdowns since these are the times during which sub detector repairs and improvements are done.

To be able to continue development at all times, three solutions to operate the routing master have been considered:

- A TFW-emulation crate exists in which both the Level 3 transfer number and the so-called Level 1 qualifiers are available for readout by the routing master SBC. Each of the 16 Level 1 qualifiers corresponds to a subset of the 128 Level 1/Level 2 trigger bits, so that this system can emulate the TFW with lesser data.
- It is possible to read out the TFW crate twice, once through the VBD, and once through an SBC. The penalty is a factor of two in speed, which is probably not an issue until 500 Hz is reached, and the addition of a few wires to the TFW VRBC
- Both the existing system's Event Tag Generator emulator (ETG) and the routing master can coexist on one PC and share the TFW data.

All of these options will allow continuous system development in a realistic environment without perturbation to the detector readout.

To smooth the transition period, all of the readout crates should be tested individually beforehand, to identify possible glitches due to slow rise times, marginal voltage values, etc. This can be done using any of the three development routing master options during accelerator downtimes. Assuming one day of accelerator studies per week but no other downtime, and successful testing of and average of six crates per day, this would take ten weeks in real time. This schedule has a built-in contingency because the expectation is to try up to ten crates on a day, and assumes excellent accelerator running.

The fastest way to make the switch from the existing system is then a gradual move in which detector subsystems are converted one-by-one. This requires a routing master which operates in conjunction with the existing ETG emulator: the Level 3 node which receives the event will receive event fragments from SBC's as well as through the current path. This requires the addition of some features to the routing master, ETG emulator, event builder, supervisor and NT node code, but all of these are limited in scope and should not require more than a few days to implement. The advantages of this approach include not only faster transition, but also dealing with a smaller number of crates at a time. The disadvantage is that scheduling is difficult since accelerator downtimes longer than a single day are preferred (certainly for the first system). Experience acquired during the testing of individual crates should prove extremely valuable. This process can start as soon as 2 weeks after individual crate testing has started, and is expected to take about 3 months due to the scheduling difficulty.

5.0 WBS Summary

The work is broken down into 4 major components.

The first major component is the development and maintenance of the overall system architecture. This includes promulgating the system concepts to the people doing the work and developing written documentation. The system concepts obviously include the data and control flow, but also the monitoring and specification of the system's qualities, monitoring of the project for completeness and needed changes, overseeing non-functional aspects of the system and the inclusion of the system in Critical Systems plans etc.

The second component focuses on the hardware aspects, their specification, procurement and installation. The maintenance of hardware drawings is part of this.

The development of the software is the third major component of the WBS. To minimize schedule slippage, the software components are decomposed into elements as much as possible, so that the time estimates to completion can be based on a larger number of small projects.

The final part considers the essential components of system integration. This includes demonstration that every single crate can be read out in standalone mode, integration of the various software components, development of transition software, and the actual transition (i.e. sequential conversion of each of the detector's systems). There is a potential risk here, which cannot be addressed during the slice test: Mediocre performance by one or more crates will require identification and repair or replacement of the faulty component.

5.1 Schedule

An MS Project file of the schedule has been developed. The work to be done is broken down into 4 major components: system architecture, hardware, software and integration. The system architecture involves the system design and documentation and naturally requires contributions from all involved.

The second part of the WBS concerns system hardware specification and procurement. Procurement is a significant factor driving the schedule. Time estimates are based on experience, and the only potential problem is related to the SBC's. The boards available on the market use the latest processors and usually have not yet reached production quantities at the time they will be ordered. It should be noted however that system integration could start with only 15 boards on hand. Assuming a decision to proceed is made on January 2nd, 2002, then all the hardware should be on hand by April 16th. This includes one week to start the process, three months procurement time and one week to prepare the hardware for installation.

For the software part, estimates are based on a fine segmentation of the tasks, previous experience with similar software projects, and developer input (development of most of the software components has already started). It should be noted that software development has started with the "slice" test, which leads to some available slack in this part of the schedule, since the currently expected date of completion is March 19th, 2002. But the software schedule relies on a high level of commitment from the developers, and the available manpower is limited.

The system integration estimate is based on experience. The possibility to read out the trigger framework crate twice, once through the SBC and a second time through the VBD, allows both systems to coexist and make system integration much easier. In this mode, the required DAQ downtime is minimal, such that interference with the experiment's ongoing operation is kept small. Transition software needs to be written to make this possible. The conversion of crates from one system to the other assumes crates will be tested during regularly scheduled accelerator studies, with full subdetector conversion performed when an opportunity arises. Full conversion of the D∅ data acquisition will be completed by July 18th, 2002.

Table 1: Milestones

Begin Testing at D0	10/31/01
All Crate Types Tested	11/30/01
Software Specifications Available	11/30/01
Decision To Implement	1/2/02
Transition Software Complete	1/29/02
Software Integration Complete	3/19/02
SBCs and Switches Available	4/16/02
Full System Operational	7/18/02

5.2 Cost Estimate

An MS Excel file with the cost estimate has been developed, of which the summary is shown in Table 2. The contingency for each item has been evaluated as follows:

- The cost for each type of network component is well known, so a 10 % contingency has been assigned, with the expectation being that the cost will be lower than the present estimate. For some of the network components, the contingency is set at 100 % to allow for a doubling of the numbers should these units under-perform.
- For the SBC's, the 30 % contingency corresponds to the possibility that a different board may be used if tests with the VMIC 7750 or Acromag card are not satisfactory.
- The cost for an additional rack of nodes is somewhat uncertain. While computer costs go down, it might not be possible (or wise) in the near future to purchase nodes identical to the current ones. Therefore, a 30 % contingency has been estimated.
- The item listed as cables, patch panels etc. includes all the smaller components not explicitly listed, leading to a corresponding uncertainty in cost estimated at 50 %.
- The network diagnostic equipment essentially consists of portable PC's with the appropriate hardware and software to examine data at various points in the network.
- A test stand will be needed to debug SBC failures or other problems both during development and running. The major uncertainty there is driven by the lack of knowledge of the most frequent failure modes. It is assumed more expensive equipment like an oscilloscope can be borrowed from the existing pool.

WBS	**1.1**				
WBS 1.1	**ITEM Commodity DAQ**	**M&S TOTAL**	**CONTINGENCY %**	**Cost**	**TOTAL Cost**
1.1.1	*Switches*	95,188	49	46,678	141,866
1.1.2	*SBC's*	278,140	30	83,442	361,582
1.1.3	*Level 3 Nodes*	35,000	100	35,000	70,000
1.1.4	*Cables, patch panels, etc.*	35,000	50	17,500	52,500
1.1.5	*Network diagnostic equipment*	20,000	30	6,000	26,000
1.1.6	*Teststand*	19,000	46	8,650	27,650
1.1	**Total**	**482,328**	**41**	**197,270**	**679,598**

Table 2: Cost Summary