THE NEW DISTRIBUTED CONTROL SYSTEM OF THE NAC

H F Weehuizen, R K Fisch, E J S Franklin, M E Hogan, I H Kohler, J P Rogers, P J Theron
National Accelerator Centre, P O Box 72, Faure, South Africa

ABSTRACT

A microprocessor based distributed control system is being developed for a 200 MeV cyclotron and two injector cyclotrons. The system will be based on at least 25 IBM compatible PC/AT microcomputers grouped into functions of operator consoles, instrumentation controllers and database nodes communicating over an Ethernet LAN. The aims of the system design are flexibility of use, ease of configuration allowing for changing hardware and user requirements, and ready availability of data for performance analysis.

1. INTRODUCTION

The National Accelerator Centre (NAC) operates a 200 MeV Separated Sector Cyclotron together with two injector cyclotrons. One of the latter is a light ion injector while the other which is still under construction will be used to accelerate both heavy and light ions.[1] The control system presently used to control these cyclotrons consists of the classical computer control architecture of a centralized computer controlling remote hardware which has little or no intelligence. Failure of the computer results in failure of the complete system. The present design is also such that the addition of extra operator consoles significantly increases the load on the system and thus impacts its performance. The most important limitation of the present system is that the number of processes that can simultaneously be handled by the minicomputers has a fixed upper limit and this is less than that required to control the cyclotron facility as it is finally envisaged.

There is thus a need to develop a new system that can cope with all the future requirements of the NAC control system. These are as follows:

a) It must be readily expandable to at least double the capacity of the present system.

b) It must be sufficiently flexible to cope with the changing requirements of a partly experimental facility such as at the NAC.

c) Hardware failures should have a minimal effect on the system.

A distributed system inherently offers these advantages. New processes and functions required of the system are usually handled by new nodes which can be optimally configured to handle the new requirements. Expansion of existing facilities by the addition of new nodes means increased parallelism in the operation of the system thus providing minimal impact on the performance of the latter.

Failure of a node results only in the loss of access to, and control of hardware related to the node, allowing the rest of the system to function normally. The only node whose failure could affect the viability of the system is the database node. In this instance a second node acts as the backup and mirrors the image of the active database node.

2. THE NEW SYSTEM

The new system, in keeping with control strategy, is designed to be as homogeneous as possible. It can be divided into four major functions, namely network, instrumentation nodes, console nodes, and database nodes. Figure 1 illustrates the system in block diagram form.

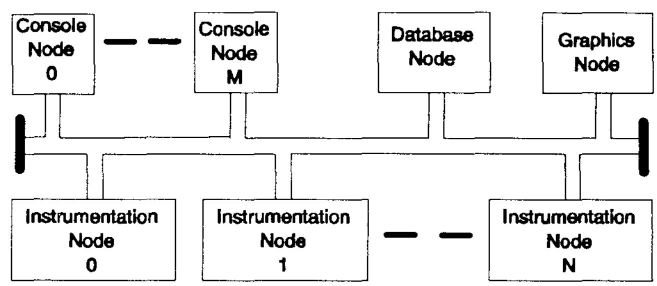

Fig. 1 A block diagram of the new control system.

2.1. The Network

There are three major types of network in use today. Two of these are bussed networks, namely CSMA/CD [2] characterized by Ethernet, and the token bus[3] characterized by MAP or mini-MAP. The third is the token ring[4] and is characterized by the IBM ring. The latter is the least suited to the present application because of the complexities of wiring a ring structure and because the node interface is of the active type, reducing its reliability.

A frequently chosen network for the control environment is that of the token bus, because of the bus structure and because the transmission delay is predictable.[5] However the hardware is costly and the transmission delay is significant because of the long circulation time of the token.[6,7]

Ethernet is attractive because it is a mature standard and the hardware is relatively inexpensive. The transmission delay is short in a lightly loaded network and is in fact shorter than that of the equivalent token bus up to a network loading of 50%. The major criticism of this network is the probabilistic nature of the transmission delay arising from the randomized contention resolution mechanism that it uses. This can be excessive in a heavily loaded network, but provided that the network loading is carefully controlled it gives a performance superior to that of the token bus. Thus the latter network was chosen.

Because the transmission delay of the network can be affected by network loading, it will be worthwhile at this point to see how the behaviour of the control system can affect the network load. Typically a single processor, real time system is event driven, i.e. every time an external status changes, an interrupt is generated in the computer which demands a response. It is up to the designer to ensure that the processor is not overwhelmed by external events in periods of high activity. The situation is aggravated in a distributed system communicating over a shared network where a high proportion of the actions will result in network activity. This will result in high loading of the network resulting in indeterminate delays which will impact both the behaviour of the system and its speed of operation. An alternative which is currently favoured by some of the real time community[8,9] is to use a state driven system. Here the status of node parameters is broadcast over the network at regular intervals resulting in a constant load on the network, thus making transmission delays predictable.

The method adopted at the NAC is to combine the two techniques. System status parameters are communicated at regular intervals of once per second as long as events are occurring at a rate greater than or equal to this. At lower event rates, an update occurs at the occurrence of an event or once every 10 seconds if events occur less frequently. The latter determines the maximum time interval that the state of health of each node is checked.

For the transmission of operator commands, the virtual circuit facility of the transport layer is used to ensure the integrity of the command data. This is an event driven operation which is an acceptable compromise because the rate at which a human operator can generate inputs to the system is low and easily controllable.

2.2. The System Nodes

The IBM PC compatible computer is being used in a variety of applications for which it was never envisaged purely because it offers reasonable performance at a low price. Its reliability is good because of the extensive use of LSI technology. While its architecture is rather limiting, it is still economic to use more or bigger systems to offset its shortcomings.

In an application such as is described here, a multitasking, pre-emptive scheduling operating system is essential. The networked environment together with a good graphical user interface and access to good database facilities are further requirements of the operating system. Currently there are several operating systems to choose from, such as a real time Unix, RMX for Windows, OS/2 or possibly Windows NT when it becomes available. OS/2 has been available for some time and currently provides the most effective solution to our requirements. However its future development in relation to our needs is not yet clear and a migration to one of the alternative systems is still a possibility.

The PC systems used to run this software are 25 MHz 386 AT systems with 8 Mbytes of memory and 16 bit peripheral cards for the instrumentation and console nodes while 33 MHz 486 AT machines will be used for the graphical interfaces and those nodes that require more performance.

2.3. The Instrumentation Nodes

These consist of the system as described above and use an expansion I/O bus that was originally conceived as an 8 bit microcomputer bus. The main processors of this bussed system rapidly outgrew the performance of the bus to become the standard PC as described above. Concurrently the I/O cards increased in sophistication and became based on the 8051 micro controller. Thus the original bus has evolved into a byte wide parallel communication medium between the host and intelligent I/O interfaces.

The function of the instrumentation node is centred around a variable table which holds all the current values of the parameters controlled by the node. There are several values for each parameter. The first group of values consist of a set point reference value and its associated status. The second group would hold the actual value of a variable and its status, including an error status if relevant. Further fields are used for housekeeping and link requests to the variable in the table. Application tasks are activated when any change to the reference group occurs and they perform the appropriate actions on the related hardware. Similar tasks continuously update the actual value and status group of the table from the hardware status.

A communication task transfers information between the network and the variable table. At the request of a console node, this task can continuously communicate the contents of the variable table over the network to update the actual values displayed at the consoles. It was originally intended that this communication would take the form of multicasting to the console nodes, but unfortunately the network software in use does not support this facility, so data broadcasting is currently being used. Unfortunately this loads the communication processors of all the nodes in the network, so point to point communication is going to be tried as an alternative. This will only increase the load of the transmitting node if more than one console selects the same page of variables, an infrequent occurrence during normal operation of the system.

The communication task also receives commands from the console nodes to control the reference values of the variables. Before the process of reference value change can begin, a request must be made to allow control of the variable. Requests can come from several sources in the system but only one may control it, so any contention for access must be resolved at this point. Thus the source whose request is received first is granted access and all subsequent requests are refused until the source granted access releases control, or a communications failure occurs. The request/grant action is uninterruptible, thus avoiding ambiguity in assigning access.

Alarm conditions requiring operator action can be spontaneously broadcast by this task as well. These would be repetitively sent at one second intervals until a response is received.

Functions such as acquisition of dynamic data would be handled by a separate process as these would be application specific, although the data communication format is standard.

2.4. The Console Nodes

The present control system controls approximately 1000 variables. These are logically divided into pages of approximately 20 variables each which can be displayed on a video screen. The display consists of the variable name, its reference value, actual value and status. Four analogue meters are available to provide an analogue display of the actual values of variables selected by the operator. In a similar manner the operator can link two set point units and a touch panel to the reference value fields of the displayed variables. The set point units are used to provide infinitely variable control of the selected variables while the touch panel is used to select predefined states. While the present console is effective it is cumbersome and expensive, thus obviating the use of mobile consoles.

The new system offers similar facilities but presented in a different way. The facilities of the Presentation Manager are used to emulate the functions of the present system. Because the graphical user interface is used, it is easy to configure the displayed page and to include bar and line graphics. Graphical instruments are easily included, substituting for the analogue displays of the present system with no loss in useability.

Initial trials of the system using a mouse as the input device produced mixed reactions. Some operators liked the use of the mouse while others complained that interaction with the system was too slow as repetitive sequences of operations required considerable movements of the mouse. If instruments other than the main video screen were being monitored while the cyclotrons were being adjusted, it was difficult to keep the mouse within the correct adjustment field. Simultaneous use of the set point function and the touch panel emulated on the screen was also not possible.

A new interface is being developed which offers facilities somewhere between those of the full emulation just described and the discrete controls of the present system. A reduced function keyboard is being developed. This will provide direct page selection and numeric value entry for setpoint values. Included will be a joystick to allow the adjustment of set points and a LCD graphics display with optical sensors will provide touchpanel entry. These will all interface via the standard PC keyboard input allowing the use of standard software. This provides an economic solution and maximum flexibility in that either the new keyboard or a standard keyboard may be used. Mobile or temporary consoles can thus be standard computers.

Central to the console software is a large variable table holding a copy of the information in the instrumentation nodes together with the names of the nodes from which they originate. This table is updated by the transmission of variable data from the various instrumentation nodes. When a page is displayed on the console the variable information is selected from the local table. Operator actions at the console are transferred to the local table process from where the commands are directed via virtual circuits to the correct instrumentation node.

At startup the page layouts and variable groupings are fetched from the database node and held in memory thus forming a local memory resident database. It is held memory resident because network or disc operations would cause the console response to operator commands to be too slow.

2.5. The Database Nodes

The control system is required to be flexible in that the facilities at the NAC are still expanding and the equipment controlled by the system is subject to alteration. It is thus a requirement of the system that it be easily configurable. The hardware of a distributed system lends itself naturally to reconfiguration, but the software must be designed to be so. The logical way to provide flexibility is to hold all the system variables in a database. The system may then be reconfigured by modifying the database. Commercial database packages for networked systems are based on a central file server serving the workstations over the network. This is not ideal for a control system as the access of a disc based database over a network is too slow for real time response. However the convenience of a single point of entry for configuration information is important.

At startup, subsets of the main database are transferred to each node on the system as required and remain memory resident for access by the local node. The console nodes each have a complete list of all the page configurations and display formats of each variable and its logical address. The instrumentation nodes receive a list of the variables pertinent to their respective nodes.

Another function of the database node is to log the status of the system at regular intervals for collecting trending information and for system status recovery after interruptions in operation or return to operation at a previously used energy level.

The use of the database node is such that it is not crucial to the operation of the control system. If it did fail, operation of the system would continue, but several important facilities would be lost. Thus a mirrored disc is used on a separate node which is able to take over the functions of the main node should the latter fail.

2.6. The Graphics Node

This node has a 20 inch, high resolution colour graphics display which will be used initially to display the beam profiles derived from data collected by harp and scanner devices situated in the various beamlines throughout the facility. Up to six devices will be displayed simultaneously on the screen which will be refreshed at the rate of four to six traces per second. A laser printer will provide hard-copies of the screen information.

The node will run application specific software under the supervision of the control nodes. The former will communicate with the instrumentation nodes that control the diagnostic hardware. These will send data across the network to the graphics node where it will be prepared for display. A virtual circuit connection will be used here to provide flow control over the network as the graphics display task is computation intensive and thus will be slower than the data acquisition task.

3. CONCLUSIONS

The new control system offers several advantages over the old system which has now become obsolete. It is readily expandable with an upper limit beyond that required to run the present cyclotron facility. The data handled by the system is readily accessible to personal computers attached to the network so off line analysis of performance data is easily achievable.

Because of the distributed nature of the system, its integrity is not dependent on any single node in the system. Failure of a node thus results in loss of a facility rather than the whole system. Equipment maintenance is facilitated through local control and through the nearly continuous availability of the control system as there is no need to stop the entire control system for maintenance purposes.

The reliability of the new system is currently equal to that of the present system and is orders of magnitude better than that of the present system in its early stages of operation.

4. REFERENCES

1) Burdzik, G.F. et al, "Decentralisation of the NAC Control System" in **Proceedings of the 11th Int. Conf. on Cyclotrons and Their Applications,** (Ionics Publishing, Tokyo, Japan, October 1986) pp. 422-425.
2) **IEEE Standard for Local Area Networks: Carrier Sense Multiple Access with Collision Detection. Std 802.3-1988 and supplements** (IEEE, New York).
3) **IEEE Standard for Local Area Networks: Token Passing Bus Access Method. Std 802.4-1990** (IEEE, New York).
4) **IEEE Standard for Local Area Networks: Token Ring Access Method. Std 802.5-1989** (IEEE, New York).
5) **MAP: General Motors Manufacturing Automation Protocol, Version 3.0,** (GM Technical Center, Warren, Michigan, 1987).
6) Weehuizen,H.F., Rodd, M.G. and McLaren, S.G.,"The Simulated Performance of Practical Ethernet and MAP Networks" in **Proceedings of 7th IFAC Workshop on Distributed Computer Control Systems** (Pergamon Press, London, 1987) pp.73-77.
7) Damsker, D.J.,"A Critique to MAP and Proway", ibid.
8) Meyer, S.P., McLeod, I.M. and Rodd, M.G.,"The design of Shoptalk, a Distributed Multiprocessor Computer Control System", ibid.
9) Izikowitz, I., Rodd, M.G. and Guo-Feng Zhao, "A Real Time OSI Based Network: Is it Possible?" in **Proceedings of 9th IFAC Workshop on Distributed Computer Control Systems** (Pergamon Press, London, 1990) pp.9-16.