EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH

CERN LEP-DI/83-52

DISTRIBUTED DIGITAL CONTROL OF ACCELERATORS

by

M.C. Crowley-Milling

Presented at the
EPS Conference on Computing in Accelerator Design and Operation
Berlin, September 20-23, 1983

Geneva, Switzerland
September 1983

DISTRIBUTED DIGITAL CONTROL OF ACCELERATORS

M.C. Crowley-Milling
CERN

Geneva, Switzerland

1. INTRODUCTION

I must start by defining what I mean by distributed digital control, since the word "distributed" can be used in many different ways. I am going to consider the case of distributed intelligence - a system made up of a number of separate computers, each performing a significant number of tasks in a semi-autonomous way. This definition excludes the master-satellite type of system, if the satellites are just multiplexing data and control signals between them and the master, even if they are geographically distributed; on the other hand, it includes systems where the intelligence is distributed over a number of computers, even if they are gathered together in one room. The distributed intelligence also implies at least some distribution of the system data-base, if the inter-computer transactions are to be kept to a reasonable level.

I would also like to point out that there is a significant difference between a generalized distributed computer system and a distributed control system. In most of the distributed computer systems discussed in the literature, the aim has been to provide the user with a system where computations can be carried out on his behalf whenever spare capacity exists, and he has no interest in where this is, as long as the result is output to his local screen or printer. However, for distributed control systems, the hardware to be controlled is connected to specific computers, and so at least part of his programs must be executed at fixed locations.

I think the earliest example of a distributed control system, in the sense I have defined above, was that for the AGS at Brookhaven in the middle 1970's. Previously, a number of PDP-8 computers had been added to the control system to carry out specific and unconnected duties, but in about 1974 these were connected to a PDP-10, to coordinate their activities¹). In contrast, the SLAC accelerator had a single PDP-9 computer, to which was added eight PDP-8 computers, distributed along the machine, but these only acted as remote multiplexors, and so this system, as it was then²), does not come into my definition of a distributed control system.

Distribution can be geographical, systematic, functional or mixed. Geographical distribution is used when an accelerator (and I include storage rings in the term accelerator) exceeds a certain size and the equipment to be controlled is concentrated at a limited number of points round the ring where access can be obtained, or, in

the case of a Linac like SLAC, where the equipment repeats periodically along the accelerator. Then, a computer placed at each of these points can control a portion of each of the systems that extend all round the ring, power supplies, vacuum, beam monitoring, etc. Such general purpose computers can have almost identical software, which has to include the high level drivers for all the different types of equipment connected. One advantage of geographical distribution is that it can allow local operation of equipment consisting of a number of sub-systems, without connection to the rest of the network. For example, running the RF equipment can involve the vacuum system and power supply system.

Systematic distribution is more usual for smaller accelerators, where all the components of a given system, RF, vacuum, beam monitoring, etc., can be joined to and controlled by a single computer. In this case the software is different in each computer, but needs the high level drivers for only a restricted number of equipments. In a systematic distribution the systems can be subsystems of an accelerator, as in the examples given above, or complete accelerators in a complex of accelerators, as in the CPS, where the booster, the main ring, the antiproton accumulator, etc., each have a system computer³).

Functional distribution is used to describe the situation where the duties carried out by a central computer are split up between a number of separate computers. An example of this is in the SPS system⁴), where separate computers are used to drive the consoles, to provide a system library, to drive displays, to analyse alarms, for program development, etc.

The way control systems for past and present major accelerators have been distributed is shown in Table I. This has been compiled from published articles, and may not reflect the present situation, as control systems, like accelerators, evolve continuously.

In recent years, the development of microprocessors has had a considerable impact on distributed control systems. Up till now, these have mostly been incorporated in the interface system, usually in CAMAC modules, to carry out some task autonomously⁵). Now they are beginning to invade the equipment itself, taking over the duties of sequencing, function generation, surveillance, local servo loops, etc., which were previously carried out by the process computer or specialized hardware⁶).

This tendency eases the process computers of some of their load, but it also introduces difficulties, as will be shown later.

The essential parts of a distributed control system are the computers, the data network to connect them together, the interface to the equipment, the software system

to make them operate together, and the timing system to ensure synchronism. Let us look at each of these in turn.

2. THE COMPUTERS

The same type of computers can be used for both centralized and distributed systems, but in general distributed systems use simpler computers but more of them, since each computer is performing fewer different tasks. Distribution is proposed to be carried out even further in the control system for LEP, where the usual multi-tasking mini-computers will be replaced by an assembly of micro-computers, each performing a single type of task⁷).

3. THE DATA NETWORK

In many of the early systems, special developments were necessary to interconnect the various processors, since few manufacturers could offer equipment for this purpose and even where such facilities were available, they were too slow for real-time applications. The first large network, at Fermilab, was of the single-master, multiple-slave type, using specially developed high-speed serial links⁸). The next large system, for the SPS, used a store-and-forward packet-switching system which treated all computers as equals, allowing any one to take temporary mastership⁴). More recently, the widespread development of Local Area Network (LAN) systems⁹) and attempts to get standards for them adopted could reduce the necessity for "in-house" development for future distributed control systems.

Data networks are usually classified by the topography; star, ring, highway, etc., and most of the networks for accelerator control so far have been of the star type. In this, each computer is joined by a separate line to a node which is responsible for routing the various messages. Large systems can have interconnected stars with several nodes (the SPS/CPS network presently has 5 nodes and about 60 computers, not all involved in controlling the accelerators). The switching at a node can be either circuit-switching or packet-switching. Circuit-switching operates in a similar manner to a telephone exchange; a connection is established between the two computers that want to communicate and this link is maintained until the transaction is complete. In a packet-switching system, messages are broken up into blocks or packets with headers giving address information and these packets are dynamically routed at the node, interspersed with packets from other sources for other destinations.

In store-and-forward systems, the node has buffer space to store temporarily the packets, so that a computer can send a packet at any time, without having to make certain that the recipient is ready for it.

Although some of the LANs being developed for general computer and terminal interconnection use the star topography, the majority are of the highway or ring type. In the highway LAN, all stations are connected to a single highway and when any one transmits, all receive the message, but it is only accepted by the one whose address corresponds to that in the destination header. There must be a mechanism to avoid errors if two stations try to send messages at the same time, and this can be done by allocation of time slots, at fixed or variable intervals, the latter being achieved by the circulation of a permit-to-transmit, known as a token. An alternative requires a station to listen and make sure that no-one else is transmitting before doing so. It must then continue to listen to check that no-one started to transmit simultaneously. In such a case, the station has to stop transmitting and wait a random time before trying again. This mechanism is known as Carrier Sense Multiple Access/Collision Detection (CSMA/CD). The best known system using this is ETHERNET¹⁰), which has a 50-ohm coaxial cable as a highway, carrying signals at 10 Mbits/s. Such a system works well if lightly loaded, but is not very suitable for use in a distributed control system, since a maximum response time cannot be guaranteed, and there is no provision for priority access to the highway.

In the case of the ring type of LAN, the stations are connected to their immediate neighbours to form a closed ring round which messages or packets circulate in one direction. Each station, on receiving a packet, checks the address, copies it if the address coincides with the station address, and then repeats the packet to its downstream neighbour. When the originating station gets the packet back, it removes it from the ring. There are two main methods of providing controlled access to such a ring. In the first of these, packets are continuously circulated round the ring, either "empty" or containing a message. When it has a message to send, a station waits until it receives an empty packet, puts the message or part of it into the packet and sends it on. The Cambridge Ring is an example of this type¹¹). In the second method, a token is circulated and a station must wait until it receives a free token before it can send a message. The latter method seems preferable for an accelerator control network, since it allows the use of a priority scheme for access to the ring¹²).

Another division in LANs is between "baseband" and "broadband" systems. In baseband systems, the data stream is the only signal on the cable, and a bandwidth of not more than twice the data rate is required. A broadband system can be used for many services simultaneously, each being allocated a portion of the bandwidth.

In an attempt to avoid the chaos resulting from all the possible variations in these different types of LAN, an American IEEE working party has been drawing up proposals (P80) for standards in this field¹³). It recognizes that different configurations

are required for different purposes, and so the proposed standards allow both ring and highway in a number of variations which means that their value is somewhat limited.

As far as I know, no "standard" commercial LANS are yet being used for accelerator control, but the LAN principles are being applied. For example, SLAC are using a CSMA/CD broad-band highway system of their own design¹⁴⁾ and TRISTAN will use a slotted ring¹⁵⁾. A token ring system is being considered for LEP.

Data transmission is usually carried out on coaxial or twisted pair cables, but the use of optical fibre cables is planned for a number of new machines, e.g. TRISTAN and LEP. One of the disadvantages of optical fibres is their low resistance to radiation, so that normally they cannot be used inside an accelerator tunnel.

4. THE INTERFACE

With the conventional minicomputer systems, there is no fundamental difference in the interface requirements between a centralized and a distributed system, although there may be detailed differences due to the distances involved, etc. The basic requirement is to transfer a number of digital or analogue values between the equipment and the computer, making the necessary transformations (D-A, A-D) on the way. This is normally carried out in a Command/Response manner, using a standard code, e.g. CNAF in CAMAC.

However, the situation changes when microprocessors are incorporated in the equipment. This primary interface is now between the equipment and its internal microprocessor, and a different kind of interface is needed between the equipment microprocessor and the process computer. Since there is intelligence in the equipment, communication does not need to be limited to Command/Response, and messages can be exchanged between the two parties. Instead of a series of CNAFs, the equipment can be told what to do in a message in the form of a string of characters, which can be meaningful to the user. This has the advantage that the equipment can be tested in isolation, using a simple terminal.

The physical links between the equipment microprocessor and the process computer have similar requirements to the intercomputer network, and a single type of LAN could be used for both purposes. However, the differences in distances, number of connections, etc., means that a single type that will cover all requirements will be more expensive than using two separate types, optimized for the application. For LEP, we intend to use the aircraft MIL/STD-1553B multidrop highway system¹⁶⁾ to connect between the process computers and the equipment.

Although there are considerable advantages in putting microprocessors into the equipment, such as off-loading some of the work of the process computers, and making the equipment self-surveying and self-testing, there are also some disadvantages.

The main one is that the response time for simple actions is slowed down. With a conventional system, with equipment connected through CAMAC to a process computer, the latter can perform some action on the equipment in the time it takes to output a few CAMAC commands - of the order of 10-100 μ s even with a slow serial loop.

However, with a microprocessor in the equipment, when the process computer sends a message, the microprocessor has to interrupt its present task, interpret the message, carry out the required action, compose a reply message and send it to the computer. This may take times in the range of 10-100 ms. In some cases, this increase in reaction time can be partly offset, if similar actions have to be taken on a number of equipments, by making provision for them to be taken in parallel, while a conventional system would have to take them sequentially.

A second disadvantage, more organizational than affecting performance, is that, since the primary interface between the equipment and the microprocessor becomes the responsibility of the equipment design group, and not the controls group, it is more difficult to obtain standardization of the hardware used.

5. THE SOFTWARE SYSTEM

The software system for a distributed control system has a number of requirements additional to those needed for a centralized system. They affect the applications language, the operating system and the database.

5.1 The Applications Language

While a distributed system could be programmed in any language, the applications programmers' task is made much easier if the language contains explicit provision for writing a program, different parts of which can be designated to run in different computers, with the necessary provisions for synchronization of returns and for exception handling. As far as is known, NODAL¹⁷⁾ was the first language to provide these facilities, through its EXECUTE, REMIT, WAIT and exception (!) commands.

Only recently have other languages become available with some of these features, the best known being ADA with its separate tasks and rendezvous mechanism. Even newer is OCCAM¹⁸⁾, which tries to provide the same facilities in a simpler package, aimed specifically towards process control.

5.2 The Operating System

A centralized system requires a computer with a very comprehensive operating system, since one computer has to perform so many tasks. Unfortunately, it seems that the more sophisticated an operating system becomes, the slower its response for real-time tasks, unless special tricks are used.

In a distributed system, the process computers controlling the equipment can have relatively simple multi-tasking executives, which can reside entirely in main memory, removing the need for discs, which is important for geographically distributed systems. One or more of the control computers must have an operating system with the full facilities, but this need not be involved in the fast real-time operations.

Of greatest importance is the communications package, which provides the means for the operating systems in the various computers to communicate in a homogeneous way. In most distributed systems in operation today, it has been necessary to write this specially for the system, as manufacturer's implementations, such as DECNET, are usually too slow for real-time control systems.

5.3 The Data Base

A distributed control system usually involves a distributed data base, since a central data base would require a computer to access this data base even when performing a local, autonomous action, which would increase the data network traffic appreciably, and slow the system down. Sometimes this difficulty is overcome by having a central data base containing all data, and then having subsets in the other computers, according to the duties they perform. This brings the usual problem of keeping up-to-date multiple copies of a data base.

In my opinion, a control system data base should be divided into a number of subsets, each subset being located where it is most appropriate, and any program requiring data should retrieve it from the computer holding the subset.

As an example, let us take the case of the extreme distribution, as proposed for LEP, where there are microprocessors incorporated into the equipment. Each of these microprocessors should hold the data table concerning that piece of equipment, such as, in the case of a power supply, demanded and measured values, maxima and minima, tolerances, conversion and calibration factors, etc. Although access to these data should be available from other computers, this is normally only required for fault-finding, etc. For most transactions, the equipment is sent a message to perform some action, or read or set a value in engineering units, the reply informing the requester if it is not able to do so.

At the next higher level, that of the process computer, the sub-data base takes the form of a directory, linking mnemonic names with the equipment addresses on the interface system, and holding information concerning interaction of one equipment with another on the same computer, etc.

Above the process computers, at the central control level, we need a data base that holds what I will call the machine physics parameters, the required values of Q, beta, chromaticity, etc., and the required settings of the various machine elements to give these values. This is the data base with which the control programs will mainly interact. The required settings will probably be obtained through some modeling or simulation programs, which in the past have necessarily been run off-line, but more and more frequently are being run on-line, as the computing power in the control system increases, and then access to a further data base is needed, that holding the machine optics parameters.

Other "central" data bases are those for the alarm system, allowing coded messages from the process computers to be analyzed and appropriate messages displayed to the operator, those containing "help" messages to guide the operator when in doubt, and those containing inventories of the equipment, with information on who to call when there is trouble, and where to find them.

From the above, which by no means includes all the data bases concerned with the operation and maintenance of an accelerator, it can be seen that the design and decomposition of the data base for a distributed control system is extremely important, and a badly designed system can result in continual frustration.

Allied with the subject of data bases is program storage. Some distributed systems use computers with sufficient local storage to hold all or most of the programs needed, but it is more usual to have some computers with a minimum of local storage and a central library from which they can load programs and run them as required.

6. The Timing System

In most accelerator systems there are three precisions of timing that are required; of the order of a second, of a millisecond and of a microsecond or less. For operations involving decisions by an operator, it is satisfactory if the system can respond in a time of the order of a tenth to one second. Most data transmission systems used with distributed control systems will allow responses on this time scale, even when several computers are involved, so no special provisions have to be made.

In the millisecond range, where, for example, synchronism between ramping power supplies controlled by different computers is required, or a closed orbit acquisition at a particular part of the cycle must be made, or a beam extraction started, it is unwise to rely on signals transmitted between computers, as to achieve a precision of even 5 ms requires special provisions. The problem can better be solved by providing a separate network carrying timing signals which can be distributed to the equipment with time critical requirements. Such timing signals can include a series of clock pulses, say at 1 ms intervals, with coded event markers interspersed. Then, rather than try to get a message through the system to say "do something NOW", one can pass a message at leisure, well beforehand, to say "do something on the nth clock pulse after event code X". With this system, it is possible to get synchronization, with a resolution depending on the clock pulse interval, to within the order of a few microseconds between hundreds of pieces of equipment, irrespective of the number of computers in the system.

Synchronism to precisions less than a microsecond usually are limited to the radio frequency or beam monitoring systems, and it is usual to make special separate provision for this, the computer system not being involved, except in setting programmable delays, phase shifts, etc.

7. Conclusion

The special properties, requirements and limitations of distributed control systems have been described. The main advantages of distributed over centralized systems can be summarized below, not necessarily in the order of importance.

- (a) For geographically distributed systems, complete systems can be operated locally even when links to the centre and the central system are unavailable. Similarly, parts of the equipment can be tested and commissioned separately during the construction period.
- (b) Distribution of tasks between a number of computers gives the possibility of parallel processing with a corresponding gain in speed and reduction in response time.
- (c) Computer configurations and operating systems can be tailored to suit the application. New requirements can be catered for by the addition of one or more computers to the network without disturbance of the existing system.
- (d) Local surveillance and testing can reduce the load at higher levels which only need to be informed if anything goes wrong. Diagnosis of faults can be easier, despite the increase in number of units, as they can be arranged to test each

other. Maintenance can be easier in a modular system, by exchange of modules, and it is easier to provide redundancy in vital parts of the system.

- (e) Different groups can work on different parts of the system with the minimum of interference.
- (f) The needs for bandwidth in the communications system for a large machine are lower for a decentralized than for a centralized system.
- (g) A distributed system can have economic advantages. It may require more hardware, which is becoming cheaper, but the software, which, if anything, is becoming more expensive, is generally simpler, and replicated.

References

1. "Use of a General-Purpose Time-Shared Computer in Accelerator Control", M.Q. Barton et al. Proc. 9th Int. Conf. on High Energy Accelerators, Stanford, May 1974.
2. "Initial Experience with a Multi-processor Control System", K.B. Mallory. Proc. 9th Int. Conf. on High Energy Accelerators, Stanford, May 1974.
3. "The Improvement Project for the CPS Controls", G. Baribaud et al. IEEE Trans. Nuc. Sci., NS-26, 3, June 1979.
4. "The Design of the Control System for the SPS", M.C. Crowley-Milling. CERN 75-20. Dec. 1975.
5. "A Versatile CAMAC Crate Controller and Computer", C. Guillaume et W. Heinze. Nuc. Instr. and Methods 177 (1980) 327-331.
6. "Integrating Local Intelligence into the LEP Power Supplies", J.G. Pett. Proc. 5th Int. Solid-State Power Conversion Conference, Geneva, Sept. 1982 (Intertec).
7. "Replacing Mini-computers by Multi-microprocessors for the LEP Control System", J. Altaber et al. IEEE Trans. Nuc. Sci., NS-30, 4, Aug. 1983.
8. "Intercomputer Communications in Real Time Control Systems", S.R. Smith et al. IEEE Trans. Nuc. Sci. NS-20, 3, June 1973.
9. "Untangling Local Area Networks", R. Parker and S.F. Shapiro. Computer Design, March 1983.
10. "Ethernet : Distributed Packet Switching for Local Computer Networks", R.M. Metcalf and D.R. Boggs. Communications of the ACM, Vol. 19, 7, July 1976.
11. "The Cambridge Digital Communications Ring", M.V. Wilkes and D.J. Wheeler. Proc. of Local Area Communications Networks Symposium, N.B.S. May 1979.
12. "A Token-ring Network for Local Data Communications", R.C. Dixon et al. IBM Systems Journal, Vol. 22 No. 1/2 1983.
13. "Local Area Networks : A Pair of Standards", M. Graube. IEEE Spectrum, June 1982.

14. "Wide-band Cable Systems at SLAC", W. Struven. IEEE Trans. Nuc. Sci. NS-30, 4, Aug. 1983.
15. "Design of the Control System for TRISTAN", H. Ikeda et al. IEEE Trans. Nuc. Sci. NS-28, 3, June 1981.
16. "Serial Digital Bus Heads for Industrial Systems". Electronic Design. 28, 19, Sept. 13. 1980.
17. "The NODAL System for the SPS", M.C. Crowley Milling and G. Shering. CERN 78-07.
18. "Process Oriented Language meets demands of Distributed Processing", R. Taylor and P. Wilson. Electronics. Nov. 1982.

Table

Publication date	Machine	Type of Distribution		
		Geographical	Functional	Systematic
1973	NAL	x		x
1973	SPS	x	x	x
1977	PETRA			x
1979	PEP	x		
1979	CPS		x	x
1979	CESR			x
1980	SNS (RHEL)	x		x
1980	ISABELLE	x		x
1981	SLAC	x		
1981	TEVATRON	x		x
1981	MEA (NIKHEV)	x	x	
1981	FMIT (LAMPF)			x
1981	Stonybrook			x
1981	TRISTAN		x	x
1983	LEP	x	x	x