CERN - Data Handling Division
DD/80/17
October 1980

Distributed Computing, Links and Networks

D. M. Sendall
CERN, Geneva, Switzerland

(Presented at the International Conference on Experimentation
at LEP
Uppsala, Sweden, 16-20 June 1980)

DD/arh

Abstract

Distributed Computing, Links, and Networks

D. M. Sendall

(DD Division, CERN, CH-1211 Geneva 23, Switzerland)

Physica Scripta (Sweden)

In many large high-energy physics experiments, the online data-processing (data acquisition, monitoring, control etc.) is distributed between several computers. This can result in a more powerful and flexible system. In addition, links to large computers and computing centres can be used to extend the resources available locally at the experiment. This brings greater computing power for event analysis, and opens up better facilities for program development and software support, as well as access to specialised facilities (such as graphics stations). Links to other laboratories allow the sharing of facilities to go still further. All this presupposes a well-developed network for computer-computer communication. This talk presents a brief survey of present practices and trends, with some discussion of the impact of standardisation in this field.

1. Introduction

This talk will be mainly concerned with online computing for 'electronic' experiments in high-energy physics; that is, with data-acquisition, equipment monitoring, experiment control, and sometimes analysis (full or partial), in real-time. The definition of real-time will vary according to the activity and the experiment, ranging from the 'immediate' response of the data-acquisition computer to an event trigger, to summaries of chamber performance available every half hour or so. I shall not be talking about the levels of processing (event filter, software trigger) which typically go on in processors intimately connected to the readout system; these topics are discussed elsewhere [1].

The questions I shall be dealing with are these: why is it sometimes useful to split these functions between several minicomputers, or between minis and larger computers, and how is this done? I shall try to give

an account of some of the ways distributed computing is used in high-energy physics, and how this may be relevant to LEP experiments. This is not a scholarly survey of the field, and the examples will mainly be drawn from systems with which I am familiar. Similarly the references are not exhaustive, but rather suggestions for further reading.

2. Clusters of Minicomputers

Let us start with the question of why the 'online' functions have sometimes been split between several minis. These functions are as follows: i) the readout of the data, usually via CAMAC; ii) the buffering and organisation of the data in the computer; iii) (usually) recording the data on magnetic tape; iv) the treatment of the data by monitoring programs in order to keep a check on the functioning of the apparatus; v) the control of the runs, and such activities as error logging. These jobs may be done by a single mini, but in many cases they have been shared between more than one. The advantages of this approach can be summarised as follows.

2.1 Separation of functions

The different functions mentioned make different kinds of demands on the system. It may therefore be possible to split the work up between subsystems, each of which is better adapted to perform a particular type of work. In fact it is difficult to design a single system (especially the software) which is really well adapted to all these different kinds of jobs. There has been a trend towards more computing power and flexibility at the experiment as the complexity of the apparatus increases; this has led to the use of sophisticated operating systems with corresponding heavy overheads and poor response to real-time events. Also, the complexity of such systems makes debugging difficult, especially at the hardware/software interface, where the most intractable problems usually occur.

One useful kind of split, therefore, is to have a relatively simple 'front-end' processor taking care of time-critical jobs such as reading and recording the data, whilst a bigger computer with more facilities but poorer real-time response is used for the actual data-processing. (Another possibility is to add to a mini a more powerful autonomous processor, which can perform more complex calculations on demand, but is not concerned with input-output or with the organisation of resources.)

2.2 Increased power

If there are two or more processors, they can run at the same time, so the overall computing power is increased. Also it is not economical to tie up a powerful processor (even

a big mini) doing a very simple task. For instance, a computer which spends a large fraction of its time organising input and output cannot at the same time be making good use of a fast floating-point processor: adding even a cheap front-end may greatly increase the power of the combined system. Another gain is in memory; in some computer architectures a single processor may have a maximum memory size which prevents the efficient running of all the necessary programs.

2.3 Increased modularity

Even if the two (or more) computers sharing the work are not specially adapted for different tasks, there may still be an advantage in sharing the work. The system may be safer; the more stable data-acquisition functions can be taken care of by one computer, whilst user monitoring programs run on another. So if something goes wrong with the monitoring computer, the system can be 'gracefully degraded', allowing data-acquisition to continue. Valuable independence is thus gained for subsystems with critically important work to do.

These arguments for distributing work between several computers are not specific to the experimental high-energy physics environment. For instance computer manufacturers themselves make use of this technique in their standard products. Some examples are: the CDC's CYBER series, where a powerful central processor has simpler peripheral processors (like minis) to take care of input and output; DEC's VAX11, which has a small LSI-11 computer associated with the system console to control diagnostics, bootstrapping, etc.; and ND's NORD-100 system, where a conventional mini running a general operating system can have extra computing power added in the form of an independent NORD-500 processor, sharing memory and controlled by the NORD-100. There is, however, a further reason for distributed processing in high-energy physics, which will probably be of particular importance in LEP experiments.

2.4 Integration of independent subsystems

Two kinds of problem may occur in the high-energy physics environment which may make it desirable to build up the online system from several independent subsystems. The first is the case of a shared experimental facility or multi-user environment. A classic example is the Omega spectrometer at CERN (see below); this has a central detector which can be used by several experiments, who set up in addition their own counters and detectors. Even if the experiments do not run at the same time, one experiment must be able to set up and test its apparatus whilst another is running. This is done by providing separate 'User' computers, which can

operate more or less independently, but still have access to common data and facilities.

The second is the case of a large single experiment with a number of detectors, which have been built and tested independently, often at different institutes. This is likely to be the case at LEP, where there will be a few large experiments involving many laboratories. The online systems used at the stage of testing detectors can be quite sophisticated, involving data-acquisition in test beams as well as test and debugging programs for the electronics. They nearly always include a minicomputer or equivalent processor, and have reached an advanced state of development by the time the detectors are integrated into the final experimental configuration. The people working on these subsystems often use different hardware and software, for many reasons; technical, financial, political, practical (availability, ease of maintenance), or just chance or differences of taste, especially in the absence of clearly defined international standards.

After the initial commissioning phase, when the time comes to integrate the different parts of the apparatus, there is a choice. Either we abandon the individual subsystems and do everything on the main data-acquisition computer, or we try to keep some of the subsystems with their own hardware and software. The first solution may involve a lot of work, especially in rewriting the test software to run on the main online computer. (It may even not be possible if the original subsystem had special features lacking on the main machine.) At present it is made difficult by the absence of standards. Even if everyone uses ANSI FORTRAN and ESONE CAMAC calls [2], many problems remain: the structure of libraries (histogramming etc.), access to data from the data-acquisition system, and so on, are likely to be different on the various systems used. Effort is needed to promote standard practices in this area [3], but the problems are likely to remain with us for some time.

It is worth noting that the integration of test processors into the main experiment requires some foresight and standardisation. The processors must have access to data and equipment along with the main online computer, and will probably want to communicate with it in other ways. Therefore a standard way of sharing access to CAMAC and other data must be worked out beforehand and followed by the different groups involved. In real life, a mixture of the two approaches is usually used: some programs will be rewritten to run in the main computer, whilst some independent subsystems may continue to run in parallel during production.

2.5 Disadvantages of distribution

In deciding whether to use distributed computing, certain disadvantages must also be taken into account. Extra hardware and software is needed to deal with shared access to the apparatus, and with communication between the computers. This may involve special development, and it brings a certain cost, adds its own overhead, and may make the system less reliable. We must be careful that communication channels do not become a bottleneck. In addition, it may be more expensive to add a second computer than to upgrade an existing one, and problems of sharing or duplicating peripherals usually arise.

Another drawback is that the system becomes more complicated and difficult to understand. This is especially the case if it is built up ad hoc out of existing subsystems. The result may be a mess of inconsistent facilities, very difficult for the physicist on shift to use, especially when things go wrong. To avoid this, attention must be given to the overall design, and in particular to interaction with the user. Again, more widespread use of standards would make it easier to put together subsystems to form a coherent whole.

2.6 Interconnection techniques

If the online work is to be split up, the different machines may require access to the same experimental data, and they may require access to the apparatus for other purposes. Usually they also need to communicate with one another; this allows them to synchronise their activities (by event, burst, run, etc.), to use common parameters, to store results or log errors in one place, and so on. It also allows the physicist to control the experiment from a single terminal.

Two kinds of technique have mainly been used for these purposes. The first is to provide common access to the event data by shared CAMAC or some equivalent method. The second is to provide direct communication between the computers, usually using a serial link. Typical link speeds for local transmission are 1 to 5 Mbit/s; however, software overheads usually substantially reduce the average transmission speed. The first method can provide easy access to data without serious loss of speed, but it is not usually flexible enough to be used alone: the two techniques are often combined. Other methods include shared memories and shared input-output buses.

3. Examples

3.1 VAX with front-end PDP-11

In a system in use at SLAC [4], the main online computer is a VAX11/780. This is a powerful machine, but with heavy operating system overheads. A small front-end PDP-11 was therefore added, sharing the Unibus in such a way that the event trigger interrupts the PDP, which then reads out the CAMAC (Fig.1). This avoids the interrupt overhead of the VAX operating system, and also avoids tying up the powerful processor of the VAX with simple routine activities. The PDP has direct access to the VAX memory, so no extra data-transmission is involved. This tightly-coupled system is a good example of separation of functions (Sect.2.1); a single task has been taken out of the VAX and put into a dedicated processor. Similar systems incorporating intelligent microprocessor-based CAMAC controllers have also been developed [5].

3.2 Linked HP2100s

A system in use at some CERN experiments (R110, WA18 [6]) consists of two HP2100 series computers which communicate via a serial link system (Fig.2). The CAMAC interface and magnetic tapes are on the 'front-end', which is responsible for data-acquisition and recording. It runs simple, fast software with no elaborate multi-tasking facilities [7]. Requests are sent over the link from the second HP, and events or other information transmitted back. This second computer runs HP's RTE operating system, a full multi-tasking system with many facilities. Some essential monitoring is done in the front-end, but most of it is done on the second machine.

This more loosely coupled system also illustrates the principles of Sect.2.1. The front-end runs a simple system with good real-time response; the second machine runs a more flexible system with better support for user programs (eg. disc file system) but poorer response. The main data-flow is through the front-end, and only a sample is sent to the other machine to be more thoroughly treated; thus neither the overheads of RTE nor the link reduce the throughput of the raw data. It also illustrates the use of already available subsystems; when the experiment was set up, the data-acquisition system existed, but without some of the facilities desirable for monitoring programs; the RTE operating system was available from the manufacturer with these facilities, and so was the hardware and software for the serial link. It was thus easier to couple the two systems than to redesign the software to incorporate all the features in one computer. This example also illustrates 'graceful degradation', since data-acquisition on the front-end can continue even if the link or monitoring computer is down. Of course, there

are limits to the amount of time that the physicists would accept to run with this degraded system. Still, it offers the choice of continuing to take data during a critical period.

3.3 Linked PDP-11s; the Omega spectrometer at CERN

This consists of a large-volume magnetic field in which multi-wire chambers are mounted, with a number of downstream lever-arm chambers. Physics groups set up their own detectors in addition, and the typical experiment requires data both from the central detectors and the group's own detectors. The solution to the online problem [8] consists of a group of four PDP-11s (Fig.3). A user machine such as U1 runs a data-acquisition system to read the user's own data and send it down a CAMAC-CAMAC serial link to the DAC computer. The DAC reads the central detector information and the relayed private data, and records both. The U1 can run monitoring programs which access the private data directly. In addition, there is a second system of serial links [9] by which the user computers can communicate directly with the DAC (in fact via a small message-switching computer). Using these, a program in the U1 can request a complete event from the DAC.

The advantages of this rather complicated scheme are as follows. Although the DAC runs the same operating system (DEC's RSX11M) as the U1, it can still be optimised to some extent for data-acquisition, since monitoring functions are removed to the user computers. (For example, it can have large data-buffers.) In addition, the software on the DAC is rather stable, and is the same for all experiments; all experiment-dependent programs run on the user machines, which also deal with the details of user-specific readout. Users have a measure of independence, and cannot easily interfere with each other or with the DAC. In particular whilst U1, say, is running, U2 can be used to set up the apparatus for the next experiment. When the time comes for U2 to be the main user, nothing has to be physically changed.

This configuration is the result of experience with an earlier one in which experiments were less independent of each other. The U0 computer, with no direct access to CAMAC itself, allows the team responsible for the central detector to investigate problems without interrupting data-taking, by analysing events passed over the link from the DAC. This was also the result of earlier experience where trouble-shooting and data-taking were in conflict.

3.4 Other multi-PDP systems; shared CAMAC

A monitoring computer which gets its data via a serial link may still need access to CAMAC for other purposes

(checking power supplies, calibration of detectors, setting up programmable trigger logic etc.). This can be achieved by sharing CAMAC between the computers, either with a 'multi-source' CAMAC interface such as a System Crate [10] (Fig.4), or with a branch mixer [11]. This approach is used at the NA2 experiment at CERN, which has a configuration similar to that of Omega, to give U0 access to CAMAC. Sharing CAMAC in this way gives rise to problems of contention, and mechanisms must be provided to ensure that the computers do not interfere with each other.

In the design of the software for multi-PDP systems at CERN, largely developed for experiment NA2 [12], emphasis is put on flexibility. The physicist can control the run and get access to any information he needs without having to talk to several different computers. Also, it is easy to move monitoring tasks from one machine to another as the needs of the experiment evolve. This approach requires the full power of a link system; shared access to the data is not enough.

3.5 The UA1 and UA2 experiments at the CERN $p\bar{p}$ collider

The online systems under development for the large collider experiments at CERN illustrate many of the ideas we have been discussing. Like many others, they use the Romulus data-acquisition system [13]. CAMAC crates are read out by special crate-controllers, and the data flows down branches to branch-drivers; these elements can be built up into a tree-structure. Since the crate-controllers issue their own CAMAC commands, the only traffic on the branch is the data; this makes it easy to introduce a 'spy' module [14], allowing another computer access to the data.

Fig.5 shows a simplified view of the UA1 online system. Data from each detector flows down a Romulus branch with a spy module, allowing a monitoring computer to access it. The data from all these branches is finally collected in a single branch, and read out by the main data-acquisition computer, a NORD-100. A second NORD-100, intended for control and calibration, also has access to the data via a spy module. In addition, the two NORDs share CAMAC via a System Crate. The main data-acquisition NORD has a Nord-500 processor attached, to add computing power needed for track-reconstruction. The monitoring computers for the individual detectors can run in parallel with data-taking, as discussed in Sect.2.4; at UA1 they are all CAVIAR microcomputers [15]. An exception is one HP2100, used to monitor a large number of ADC channels; the control NORD communicates with this via a serial link, to control its operation and read back accumulated histograms for display.

A similar arrangement will be used for UA2; there the monitoring systems for the individual detectors will not all be the same, but will include computers already in use at the different collaborating groups. The main data-acquisition and control will be done by a single VAX.

4. Links to larger computers

So far we have been considering clusters of minis dedicated to a single experimental setup. Online minis can also be linked to 'big' machines (at a computer centre for instance) to get access to shared facilities. Usually such a link is needed because we want to do things the local mini cannot do. Such links can be used for many purposes. One is the analysis of event samples, needing more power than the local machine can provide. Another is the provision of program development facilities, such as file and edit systems, cross-compilers, documentation, etc. Often all these facilities can be provided by the same 'host' computer.

Once such links exist, the way in which the work is distributed between local minis and the computer centre is open to choice. At one extreme we can reduce to a minimum what is done locally, for instance recording all data at the computer centre instead of on local magnetic tapes [16]. Alternatively, as much computing power as possible can be provided at the experiment. The choice will depend on individual circumstances, and generally falls somewhere between these two extremes. Centralisation is often more cost-effective and operationally attractive, but can lead to difficulties over sharing resources and allocation of priorities. It may also overload the real-time capacity of the host, and of course needs an extremely reliable link system. Often, therefore, experiments prefer some security and independence even at some cost in efficiency. An interesting example of distribution involving multiple local minis and a remote host can be found in [17].

Sample analysis. It is often desirable to pass a sample of events through the complete 'offline' analysis program rather quickly; this can help to detect problems which the online monitoring programs do not pick up. The online computer does not usually have the computing power to do this, and it is very difficult to run large programs on minis with 16-bit word length. In any case, it may not be easy to transport an existing analysis program from the computer centre to an online machine. (Problems arise with FORTRAN dialects, the absence or incompatibility of libraries, etc.) It is usually better to send the events to the computer centre for analysis. It is also necessary to get the results back, and most convenient if they can be made available at the experiment itself.

Program development. It has long been recognised that the typical online computer configuration is not well adapted to the needs of program development, and that poor program development tools waste the physicist's time. Good editors for minis are rare, local disc storage is usually inadequate, poorly organised, difficult to back up, and very expensive per byte compared with the large shared disc systems of central computers. Mini operating systems are often not designed for efficient time-sharing between several terminal users. There is often a lack of terminals, fast printers, etc. In addition, there is no possibility of gaining access to special devices such as laser printers, mass-storage systems, etc. With all these drawbacks, although some program development work is often done locally, the tendency is to provide facilities to do it where it is most convenient.

Software support. The benefits of central storage of subroutine libraries, documentation, etc., are considerable. To keep these things on a large host computer, where they can be easily controlled and updated, allows software support to be much better organised, with corresponding reduction of confusion and easier troubleshooting.

4.1 Implementation; protocols

The distance between the experiment and the computer centre is usually too large to consider any connection except a serial link. The link carries many different kinds of traffic, reflecting the varied uses mentioned above. Usually, therefore, its use is governed by a hierarchy of 'protocols'. At a given level, two processes communicate using the services provided by the level below, without needing to know exactly what goes on underneath.

The lowest level (line protocol) may organise the transmission of a block of data from a buffer in one machine to a buffer in the other, taking care of gross errors in transmission. Above this, an end-to-end protocol will take care of communication between a program on one machine and a program on the other. This involves the establishment of logical links between the programs (analogous to making a telephone call), the splitting up of data into blocks of suitable length for transmission, flow control (not sending data if the other end is not ready), and probably more error detection and correction. This level can be 'seen' directly by a user program, normally via a simple set of calls from a high-level language like FORTRAN. Two programs in the two machines can use these calls to set up a connection and exchange any messages they like. There are usually various protocols provided at the next level up, to do some of the things most commonly needed (and users can define their own higher-level protocols if necessary).

An example is a file-access protocol, which allows a program running on the mini to manipulate files (open, close, read/write records, etc.) on the remote host computer. This is done by an interface package in a subroutine library, accessible from a high-level language, which uses the end-to-end protocol to exchange suitable messages (requests for action, or data) with the file-manager on the remote host. Using this file-access protocol, a file-transfer program can be provided, allowing a user to transfer files conveniently between the file-bases of the two computers.

We can see how these mechanisms are used to implement some of the facilities described in Sect.4. For instance, the file-access protocol can be used to send samples of data for analysis. A monitoring task running on the online mini can request events from the data-acquisition package; then instead of analysing them itself, it can write them to a file it has opened on the remote host. When enough events have accumulated, the file can be closed, and a job submitted on the host to run the analysis program. Many file-access protocols provide job-submission facilities, so that this can be done automatically. Otherwise the user will have to log in to the host via its normal terminal system and run the job himself. Similarly, the results may be sent back to the local mini for printing or storage; otherwise the normal output mechanisms for offline jobs can be used.

A file-transfer program is also needed if the host is being used for program development. For example, the user may keep his sources on the host file-system and edit them using the host's interactive terminal system. He may cross-compile them if suitable facilities exist. The resulting modified sources or compiled object programs can then be transferred to the local computer for compilation or linking. In some cases a host can be used for the whole chain of program preparation, right down to the production of a loadable program image which can be transferred to a local file system for subsequent use, or bootstrapped directly into the online machine via the link. Bootstrapping local minis with no local storage of their own is another service which the host can provide; usually a simplified protocol is needed to do this.

Another example of a high-level protocol is one for organising direct communication between a sampling program running locally and an analysis program running on the host. (This is an alternative to writing events to a file and then submitting a job). The protocol required might then include messages sent from the host to request an event, with replies from the local program indicating valid data, none available, error conditions, etc. Similarly the results could be sent back directly and displayed on demand. Such a system can be powerful, using the remote machine to extend the

scope of a local task. However, it demands real-time response on the part of the host, and so it may be difficult for the host operating system to deal with if this is heavily oriented to batch work. An interesting example of this kind of direct interaction between the two computers is the EDAS system at GSI Darmstadt [18].

4.2 Message switching; topology

So far we have considered direct links between the host and the online minis. (In practice, at the host end the links are dealt with by a front-end processor of some kind, which is used to multiplex them onto an input/output channel. This is because the host is not well suited to the real-time response required; another example of the problem discussed in Sect.2.1). These links can be implemented as a simple radial network (Fig.6). An example is Bisonet at Fermilab [19]. However, a more powerful method is to connect all links to a switching node (Fig.7). This is itself a small computer whose job is to receive messages from one 'subscriber' of the network and forward them to another. The name of the destination is contained in a header at the beginning of the message. With this arrangement other hosts are easily added, and the minis can also communicate with each other. This type of system is usually implemented as a packet-switching network, where a long message or block of data to be transmitted from one machine to another is split up into packets of a certain maximum size, which are transmitted and routed independently. Such a network can be used for many purposes, since new subscribers can easily be added and all subscribers can communicate with each other. The design of such networks and their protocols is a rapidly growing field [20].

In addition to online minis and large hosts, computers providing special services can be attached to the network. Examples existing or under discussion at CERN include: a graphics/plotter station controlled by a mini, to which output can be routed from any of the computers at the centre; an interactive graphics facility with an intermediate size computer and display system, for the study of complex events in collider experiments; a file and edit machine dedicated to running a powerful and efficient editor, to relieve the central computers of this part of their load; minis running information-presenter programs, to allow physicists to look at histograms and other results accumulated either online or offline. In practice the interconnection of the subscribers is usually more complicated (Fig.8), but the basic principle remains the same.

4.3 Integration of networks

So far the problems of communication between 'co-operating' minis at an experiment on the one hand, and between those minis and remote facilities on the other, have been discussed separately. However, most of the problems to be solved are the same, and similar networks have been set up to deal with them. In many cases, existing networks for communication between minis have been specific to one type of mini. It is difficult to adapt these to the more general conditions of a laboratory-wide network with subscriber machines of different types. One way round this is to use some kind of 'gateway'; this is a computer which can talk both kinds of protocol.

For instance, a serial link system for PDPs was developed at CERN to meet the needs of the Omega spectrometer (Sect.3.3). This grew to handle not only local traffic between the minis in clusters, but also traffic between clusters, and between small PDPs and larger hosts; a full message-switching system (OMNET) was developed, offering a variety of services [9]. At a later date, a more general packet-switching system CERNET [21] was set up, primarily to allow access to the CDC and IBM machines at the Computer Centre. In order to extend the services of this new network to existing OMNET subscribers, one of the OMNET switching nodes was adapted to serve as a gateway between the two networks. Thus a PDP could run the standard CERNET software, the packets being transported via OMNET to the gateway. All OMNET subscribers became CERNET subscribers without needing any new physical links.

Linking two established networks in this way can lead to problems [22], and in any case it may be better to standardise the physical links used by all types of mini, with consequent saving in cost of maintenance and spares. This could be done in the case of CERNET, because the decision was taken to interface all mini subscribers via CAMAC (as had been done elsewhere [16,19]). One could therefore consider replacing all the special-purpose links described in Sect.3.2 to 3.4 by the standard ones independent of mini type. Of course this would involve rewriting existing software and replacing existing hardware; there might be more serious difficulties however. A general-purpose packet-switching system like CERNET has to be designed to ensure operational integrity, and independence of the special features of particular subscribers. This can make the software heavy both in average transmission speed and in memory requirements. A network dedicated to local traffic of a restricted kind between machines of the same type can afford to be simpler and faster. The problem of providing compatible subsets of general facilities to overcome these difficulties is beginning to be tackled; it is hoped that eventually all the com-

puter-computer communication in the laboratory can be brought into a unified scheme.

4.4 Extensions of distributed computing within the laboratory

A problem not obviously related to 'online' programming is access by programmers to the computer centre machines, in order to do program development, submit jobs and get their output. Two methods have commonly been used, work-stations (with card-reader and printer), and terminals connected to some interactive system like Wylbur [23]. In a large laboratory it is not convenient to put all these in the computer centre. Remote work-stations can be installed, communicating with the host via serial links; often these have a small computer to handle the link and the local peripherals. Usually protocols follow the manufacturer's standards. For terminals, circuit switching may be used, exactly as in a telephone exchange, to allow different hosts to be dialled up; alternatively, a small computer may be used as a terminal concentrator, linked by a network to the host.

If the concentrator and host are linked by a general packet-switching network, another high level protocol can be provided, known as a virtual terminal protocol, by which a terminal attached to the mini can be used as a remote terminal of the host. A program running in the mini uses the end-to-end protocol to relay the terminal input/output to a partner program in the host; this is interfaced to the host interactive system in such a way that the latter can behave as if it was dealing with a real terminal. Notice that this technique can be used to allow a physicist to use a terminal of his online mini as a terminal of the host.

Clearly, if good enough communication tools exist, we can approach the ideal of doing all computing where it is most effective. The distinction between on- and off-line becomes less important; if adequate attention is paid to standard practices, a program can easily be moved from one computer to another to meet the needs of changing circumstances. (Examples are the transport of monitoring programs between the PDP-11s of the systems described in Sect.3.4, or between the PDP and the IBM in the EDAS [18] system. Use can also be made of facilities which only exist on other computers; for instance, the user of a small PDP at CERN is linked to other computers as shown in Fig.9. He can use his local terminal to talk to Wylbur on the IBM [24], to edit a program. He can then use similar software [25] to talk to a larger PDP, transfer the source from the IBM using a file-transfer program, compile and prepare a program image there, and bootstrap it into his local PDP.

Networks can also be useful in the computer centre itself; at CERN, CERNET is used for traffic between CDC and IBM, so that programs developed using Wylbur on the IBM can be submitted to the CDC. It is also used to send output to special graphics stations, and to route CDC output to the IBM or vice versa, so as to take advantage of their different output possibilities. More ambitious schemes of organising computer centres around fast bus systems have been proposed [26]; the high-bandwidth short-distance buses appropriate to this kind of traffic will be different from the longer distance packet-switching networks, but the two could perhaps be integrated.

Will all these techniques be required in the LEP laboratory? The short answer is yes. Both multi-mini clusters and links to computer centres are likely to be essential. At LEP, with a small number of large experiments, there may be a strong case for providing a lot of computing power locally at each intersection [27]. At one extreme one might consider eliminating the computer centre altogether. However, this will not remove the need for links. As local computers become more powerful, they become less well adapted to dedicated data-acquisition, as we have seen; in addition, test systems will be needed. So local links will be essential. Also, even in a model in which most analysis is done locally, links between the 'regional centres' would be needed, because they provide the only efficient way of distributing and maintaining programs and support facilities. Finally, access to special facilities will be required. So the LEP scene is likely to follow the trends already observed in present laboratories.

5. Links outside the laboratory

So far we have been discussing distributed computing within the laboratory, but it is possible to extend the principle to include links to other places, especially the home institutes of physicists working at an accelerator laboratory such as CERN or DESY. This is a very interesting possibility for the future, and there is already some experience to build on.

For several reasons it is desirable to spread the 'offline' computing load between different computer centres. National and other centres already exist, and may have spare capacity for this kind of work. More important, perhaps, is to avoid the unnatural concentration of experience and facilities at a few accelerator laboratories. It is also essential to improve the ability of people, even at small institutes, to participate in the work of large experiments. The easy transfer of experimental data, programs, and documentation is thus of great importance [28].

5.1 Present practices

A good example is the Rutherford Laboratory, which has had a link from CERN to its computer centre for some years. This serves as a work-station, and as a terminal concentrator for the Rutherford Electric terminal system. Physicists can therefore use the Rutherford's computing facilities whilst they are at CERN; this is especially useful for those on short visits, and visiting teams can choose to do the bulk of their analysis at the Rutherford instead of at CERN. Since the Rutherford has work-stations of a similar kind in many Universities in the UK, their users can continue to work conveniently wherever they are. The link has also been very useful for exchanging information between physicists in the different laboratories. Recently a new service has been added to allow a connection to the CERN IBM; files can be transferred between the two centres, making co-operation and exchange of software easier. It is also possible to log into the CERN IBM, and to submit jobs, from the Rutherford; however, for the reasons mentioned above, this is not intended to be a common way of working. Similar facilities exist between DESY and the Rutherford, and between CERN and the computing centres at Saclay and LAPP.

5.2 Future developments

Most of these interconnections are of an ad hoc nature, taking advantage of existing facilities on the different machines. However, we can expect to see a trend to the use of more general-purpose networks as these become available. Since the first extensive private network ARPANET was set up in the USA to offer timesharing facilities to its subscribers, the growth of private, public, and international networks has been very fast [20]. Some European countries already offer at least experimental services, such as EPSS in the UK and TRANSPAC in France. If the connections between computer centres and their remote users, and between different centres, can make use of these, a wide range of services could be implemented in a consistent way. The Rutherford has already made use of the private packet-switching SRCNET to implement some of its facilities for remote users.

One difficulty of using these networks has been the lack of standards, not surprising in a field that has grown up so quickly. However, substantial progress has been made, at least up to the level of end-to-end protocols, and there are widespread plans to implement the CCITT's X25 [29] and related standards. This will make life easier for the potential user, and is especially important when interconnection of different national networks is considered; this of course is essential for LEP. The Rutherford is already starting to use X25, and the British public PSS system will

also use it. CERNET should have an X25 gateway next year. Work is also going ahead on standards at other levels, and many of them should be well-defined by the time LEP starts up.

With the standardisation and improvement of the basic communication tools, many interesting new facilities can be thought of, especially in the domain of information exchange and access to remote data-bases. These are likely to be of great value in making the participation of physicists from widely spread and often small institutes more effective.

Most of the applications mentioned require the transfer of only small amounts of data; it is assumed that the bulk of raw data to be processed elsewhere will continue to be transported on magnetic tapes. However, the STELLA experiment [30] is using satellite transmission to send large quantities of data from CERN to the participating laboratories. Whether this can ever be economical enough to replace magnetic tape as a transport medium is not clear, but it opens up interesting possibilities for the exchange of larger quantities of data in real time than is possible at present (a DST could be exchanged, for instance).

6. Conclusions

I have tried to show the different ways in which distributed computing is used at present in the laboratory and might be used in the future. I believe that it will become even more important in the future laboratory. To use these techniques efficiently, the adoption of standards of many kinds is essential. Because of the importance of this area, I believe that one should start now to design an overall communication system which can serve the foreseeable needs of LEP, both within the laboratory and outside.

References

- 1) Dobinson R.W., 'Data Acquisition at LEP', These Proceedings.
- 2) 'Subroutines for CAMAC', ESONE/SR/01 (1978).
- 3) Lillestøl E., 'Data-processing Standards for Experiments at LEP', These Proceedings.

Rimmer E.M., IEEE Trans.Nucl.Sci. NS-27, 631 (1980).
- 4) Browne M.J., Granieri C., Sherden D.J, Weaver L.J, 'A PDP Front-end for a VAX11/780' SLAC-PUB-2494 (1980).
- 5) See for instance Minor M.M., Shlaer S., Thomas R.F, IEEE Trans.Nucl.Sci. NS-23, 459 (1976).
- 6) References to CERN experiments in this paper are by experiment number: see 'Experiments at CERN in 1979' etc. edited by the Secretary of the Research Board, CERN.
- 7) Rimmer E.M., 'HP BCS Data-acquisition System' CERN (1978).
- 8) Bizeau C., Bourgeois F., Clarke D., Davies H., Lauper S., Litt J., Silverman A., 'The Data-acquisition System for SPS Omega', CERN OM/SPS/85 (1974).
- 9) Parkman C.F. and Lee J.G., 'OMNET - High Speed Data Communication for PDP11 Computers' CERN 79-11 (1979).
- 10) For instance the one manufactured by Fisher Controls Ltd., UK.
- 11) See for instance Lang A., 'A Branch Mixer Unit', CERN CAMAC Note 37-01 (1974).
- 12) Botterill D., Charles B., White V., Nucl.Instrum. & Methods 166, 541 (1979).
- 13) Ponting P.J., 'A Guide to Romulus/Remus Data-acquisition Systems' CERN-EP Electronics Note 80-01 (1980).
- 14) Vanuxem J.P., 'A Romulus Routing Unit', CERN-EP (1979).
- 15) Cittolin S. and Taylor B.G., 'CAVIAR: Interactive Microcomputer Control and Monitoring of Multi-crate CAMAC Systems', Proceedings of the Joint Conference on Microprocessors in Automation and Communications, University of Kent, UK (1978).
- 16) Zacharov B. IEEE Trans.Nucl.Sci. NS-23, 436 (1976).

- 17) Schachter M.J., 'CELLO - An Advanced Technology Detector at Petra', These Proceedings.
- 18) Chestnut R., Grasmueck B., Hadsell R., Krieger E., Lebershausen W., Lowski J., Plappert W., Richter M., Rother H., Siart O., Thieme R., IEEE Trans.Nucl.Sci. NS-26, 4417 (1979).
- 19) Brenner A.E., Curtis D.B., Droege T.F., Dunne R.A., Martin R.C., IEEE Trans.Nucl.Sci NS-23, 442 (1976).
- 20) For a short review with many references see: Charles B.J., 'An Introduction to Current Networking Standards', Rutherford Laboratory Report RL80-014 (1980).

Other useful reviews are given in: Proc.IEEE 66 No.11 (1978); and 'Data Communications', Infotech, Maidenhead, UK (1980).
- 21) Gerard J.M., 'Development and Use of a High-speed Packet Switching Network in a High-energy Physics Laboratory', CERN DD/79/2 (1979).
- 22) Piney C., Parkman C.F., Fluckiger F., 'Lessons from an Endpoint Interconnection of High-Speed Local Packet Networks in a Virtual Call Environment', CERN DD/79/12 (1979).
- 23) Fajman R. and Borgelt J., Commun. ACM 16, 314 (1973).
- 24) Watts R.P., 'A Virtual Terminal Protocol for CERNET', CERN-DD SCS Note No.04 (1978); also Nierhaus R., 'Mini-Wylbur', CERN-DD (1979); Hoffmann G.R., 'On the Portability of Interactive Pascal Programs using CERNET', CERN-DD SCS Note NO.16 (1979); Hoffmann G.R., 'WYLBUR Bridge for NORD-10', CERN-DD ND 79-3/1.
- 25) Raffi G., 'Remote Terminal Communication with RSX11M', CERN DD/79/6 (1979).
- 26) For instance the HYPER Channel Network Adaptors, manufactured by Network Systems Corporation, USA.
- 27) Van de Walle R.T. (Chairman), 'Data Analysis', ECFA-LEP Working Group 1979 Progress Report, 249 (1980).
- 28) Mulvey J.H., 'Participation in LEP Experiments by University Groups', These proceedings.

Hine M.G.N. (Chairman), 'Data Transmission', ECFA-LEP Working Group 1979 Progress Report, 237 (1980).
- 29) 'Public Data Networks', CCITT Orange Book, Vol VIII.2, International Telecommunications Union, Geneva (1977).
- 30) 'The STELLA Experiment', Technical Report of the European Space Agency (1978).

Figure 1: VAX with front-end PDP-11 (SLAC)

Figure 2: Linked HP2100s (CERN)

Figure 3: Online System of the CERN Omega Spectrometer

Figure 4: Shared Access to CAMAC via a System Crate

Figure 5: Simplified Diagram of CERN UA1 Online System

Figure 6: Radial Links to a Host Computer

Figure 7: Simple Star Network

Figure 8: General Interconnection of Switching Nodes

Figure 9: CERN Network Facilities