

THE BATES LINAC CONTROL SYSTEM

Tomas Russ, Zdenek Radouch
 MIT Bates Linear Accelerator Center
 P.O. Box 846, Middleton, MA 01949

Abstract

The Bates linac control system (LCS), a distributed processing architecture, is described. Due to the historic evolution of the system, a mix of different hardware, operating systems and programming languages are used throughout. However, a standardized interface at the network level enables a smooth system integration. In particular, a "multicasting" scheme for data transmission over the network permits simultaneous database updates on more than one workstation. This allows for true distribution of data processing power.

Introduction

The Bates linac is a 1 GeV, 10 mA peak, 50 μ A average current, 1% duty cycle, electron machine which operates over a wide range of energies, currents and pulse rates. At energies above 500 MeV, the beam is made to pass a second time through 20 of the 22 accelerating cavities by way of an isochronous recirculator. The switchyard delivers beam to 5 experimental areas (S,F, Z,B and C lines).

A total of about 200 magnets, 7 slits, 6 transmitters (2 klystrons each) and 2 magnet moving mechanisms are used to control the machine. Beam instrumentation is composed of toroids (current monitors), traveling wave beam position monitors, profile monitors, and phase/amplitude detectors. Other magnitudes measured are magnet current, transmitter parameters, klystron phasing and slit positions.

Due to the incremental upgrade of the control system as the linac evolved from a single experimental area, single ac-

celerating machine, to a multiexperiment, recirculated beam, its design incorporates a mixture of hardware, computers and software (see Fig. 1). As computer access was added to the original "control panel only" mode, the control system retained its redundancy. Therefore, every controllable device can be accessed through a control panel even if the computer system is down, and viceversa.

The hardware controllers (power supply and motor controllers, transmitter controls, etc.) are concentrated in three physical areas -accelerator (ACC), beam switchyard (BSY), and RF gallery- and are accessed from the Central Control Room (CCR). The hardware controllers are all in-house designed, some of them retaining very old relay technology. Because of their specific nature, no details will be given about them.

Control panels and multiplexers are microprocessor (8085) systems which run an EPROM based program. The main function of the processor is to provide a communications protocol between the controllers' hardware and the controlling device, and -in the case of the control panels- to drive the operator interface. An "ad hoc" multitasking, multiuser real time executive was developed for this purpose, which is written in the processor's assembler language.

The computer system is a true distributed architecture composed of μ PDP11s and STD bus computers performing data acquisition and control, and μ VAX II's and another μ PDP11 acting as data processing and user interface computers(hosts). The μ PDP11 data acquisition/control computers interface to the hardware through CAMAC crates. The STD bus computer does not need this additional interface, as it

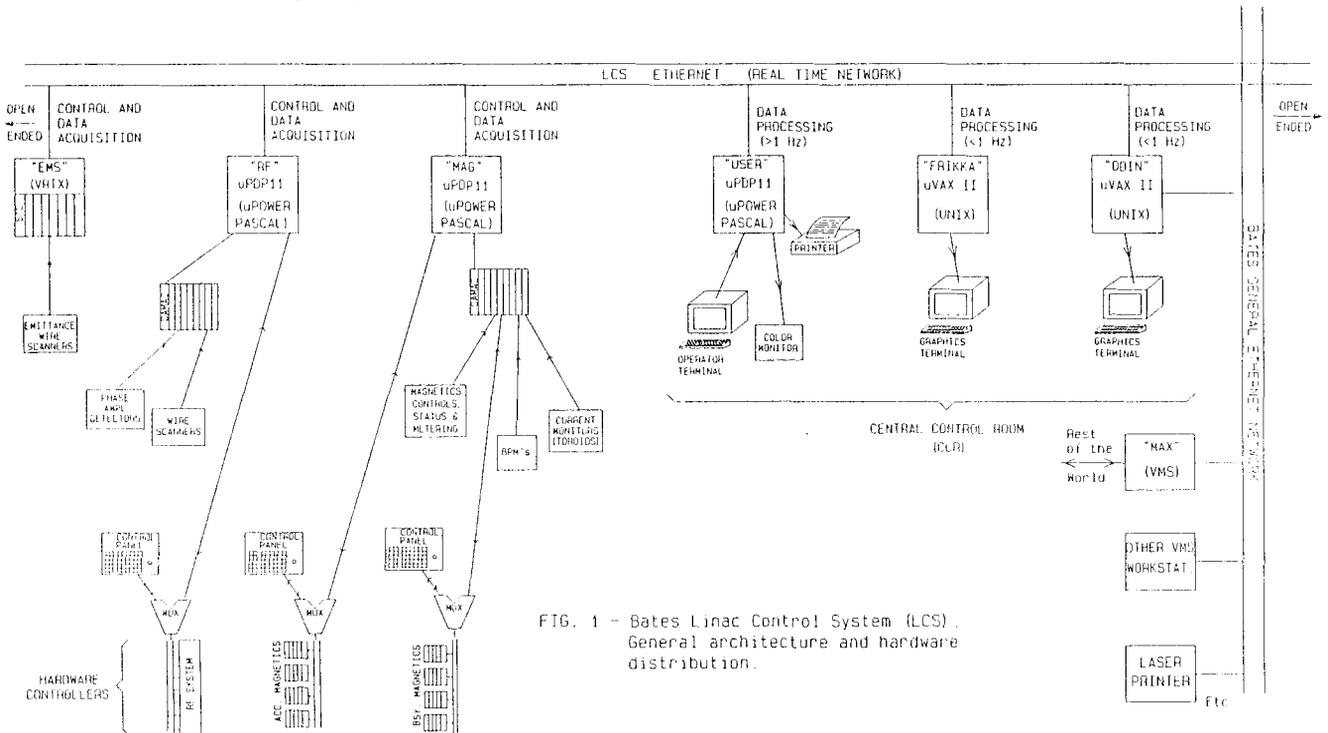


FIG. 1 - Bates Linac Control System (LCS). General architecture and hardware distribution.

can support peripheral cards on the same computer bus. The "true" distribution derives from the fact that there is no "master" computer in the hardware nor software architecture. This allows for easy expandability of the system and (as will be explained later) simultaneous use of instrumentation data by more than one host. An Ethernet local area network is used as the communication link between computers.

All μ PDP11 computers run Micro Power Pascal, a DEC provided real time executive. The STD bus based system runs VRTX, another real time executive marketed by Ready Systems. Both software products are similar in that they provide an environment in which several processes can run concurrently. The kernel then manages the CPU scheduling and provides communication channels between processes. Software development is performed in high level language (Pascal and C respectively) under a host and then downloaded to each target. Because of its lack of full blown operating system, real time response on these systems is optimized. This is important not only at the data acquisition/control level, but also when data processing has to be performed at high I/O rates. For this reason the data processing μ PDP11 (USER) is put in charge of handling all programs that need access to beam data at a rate faster than 1Hz.

The μ VAX II workstations run under the BSD 4.3 version of UNIX. This software product was chosen over the one provided by DEC (Ultrix) due to the easy access to the source code, which was necessary to modify some aspects of the network drivers. Graphics and window management are handled with the X Windows software package.

Computer architecture

Referring to Fig.1, three data acquisition/control computers (labeled MAG, RF and EMS) connect to the machine hardware, in some cases through multiplexers. The two UNIX workstations (ODIN and FRIKKA) and the remaining μ PDP11 (USER) provide data processing and user interface capability. The LCS Ethernet ties all control system computers together. In addition, ODIN incorporates a second network interface to the Bates general network, allowing for file transfer and peripheral utilization by LCS. Devices on the linac are controlled via a request-reply mechanism originating in one of the data processing computers. This mechanism provides fast (< 10 ms) access to the data acquisition/control computers for the purpose of, for example, modifying a device setting. Beam instrumentation data is generally not retrieved, however, utilizing the request/reply feature. Instead, each data acquisition/control computer will place its data on the network at regular intervals, with a "multicast" destination. Multicasting is an addressing mode in Ethernet that enables several nodes to simultaneously receive a packet. Data is placed on the network by the source, and all nodes that want to read that data can do so by instructing its controller to listen to is dictated by the data acquisition rate of the corresponding instrument class. This combination of distributed processing and the dynamic update of the database on the network enables the LCS architecture to be open ended in hardware and software. If a data acquisition/control computer becomes overloaded, part of the hardware can be rerouted to an additional computer. On the other end, if a data processing computer runs out of power to handle all the necessary applications, some of these programs can be installed in another computer. This approach places a heavy burden on the network bandwidth.

However, only the data acquisition rate -which depends on the machine hardware- contributes significantly to network traffic. Data processing software -whose growth is much more unpredictable- does not add to the network overload. Fig.2 shows a traffic calculation for LCS in the present configuration, showing that maximum network utilization is well below 10%, a figure that will keep the number of collisions (main source of delays in Ethernet) at a very low level.

ETHERNET HARDWARE CHARACTERISTICS:
 Transmission speed -> 10 Mbits/s.
 Maximum packet -> 1500 bytes (11.5 ms.)
 TRAFFIC CALCULATIONS:

Device	How many	Bytes/dev	Info send every	% utilization
BPMS	14	8	200 ms	0.57
Toroids	10	4	200 ms	0.24
XY current	14	4	200 ms	0.33
Wire scan.	15	3000	1000 ms	4.5
Magnetics	512	10	1000 ms	0.51
Phase/ampl	25	20	1000 ms	0.5
Water contr.	12	20	1000 ms	0.23
RF controls	7	100	1000 ms	0.84
Emission (future)	3	5000	5000 ms	0.56
Synchrotron	3	100	1000 ms	0.36
Total				5.82%

FIG. 2 - Maximum bandwidth utilization on the network.

Software

Because of the concurrent programming features of Micropower Pascal and VRTX, and the standard multitasking capabilities of UNIX, software for LCS is organized as a series of independent processes which communicate among themselves through data packets and semaphores. Fig. 3 shows a top level data flow diagram for the LCS software. In the example shown, all data acquisition/control processes are active acquiring data from the hardware and placing it into the instrumentation database (multicasting). Two of the host processes (BEAM ENVELOPE DISPLAY and XY DISPLAY) use this data to update screens for the operator. The remaining process is adjusting a magnet current by sending request/reply packets to the data acquisition/control process labeled MAGNET.

Each data acquisition/control process is on an infinite loop waiting for requests on its main queue. These requests could be actual data packets sent by another process, or a signal given by an external interrupt (beam pulse interrupt service routine, for example). The process queues are identified by a unique name, and a system wide clearinghouse (name table server) matches these names with port numbers, keeping track of where they are located (network node and process within that node). In this way communication between processes is made network transparent, i.e. a process doesn't need to know that particular multicast address. In this way all the beam instrumentation data is being updated *on the Ethernet cable* all the time. In the LCS implementation, a multicast address is created for each instrument class (phase/amplitude detectors, beam position monitors, magnet currents, etc.); a data acquisition/control computer may transmit on more than one multicast port. In this way, data processing computers don't have to retrieve but the part of data they are interested in.

The collection of multicast data can be viewed as the beam instrumentation database -maintained on the network instead of in a file or computer memory- whose maximum access time

in which node another process resides, just the name of its main queue. The name table services are taken up by one CPU in the system after an arbitration process, and are dynamically reallocated to another CPU if the first one fails.

Multicast ports (used to communicate the beam instrumentation database) are also allocated by the name table server. They are created by the process that will generate the data, and all processes willing to read such data identify the corresponding port by interrogating the table server.

The network software is layered up to the ISO/OSI model level 3. Because of the network support in UNIX systems, the ARPA IP (Internet Protocol) is implemented system wide. Lack of data flow control (normally implemented in higher levels of the ISO/OSI model) within the LCS system is not a problem because of the request/reply nature of all transactions. A simple timeout mechanism enables the requesting computer to detect a failed communication link. For communication outside LCS (transactions to the Bates wide network), the full TCP/IP protocol is used.

Multicast recognition is not a standard feature in the Internet software used by UNIX, so some modifications had to be made in order to enable it. The absence of a memory sharing feature in our version of UNIX also creates limitations in the retrieval of multicast packets, as no buffering mechanism exists for these ports. In our present implementation, multicast packets that arrive while the data processing program is not waiting for them are lost.

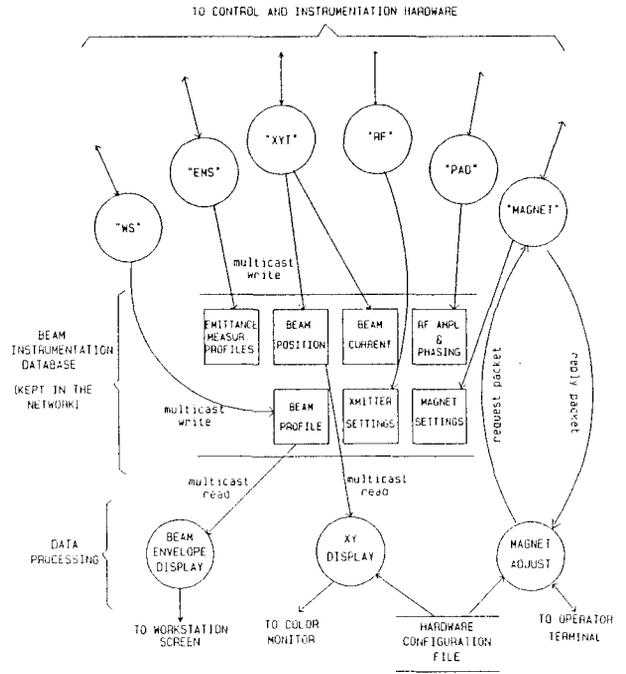


FIG 3 - Top level data flow diagram. Shown are three host processes, two displaying instrumentation data using the multicast read and one modifying a magnet setting through the fast request-reply mechanism.