FORINJECTOR VEPP-5 KLYSTRON GALLERY CONTROL SYSTEM

A.N. Aleshaev, I.V. Belousov, I.E. Borunov, R.G. Gromov, K.V. Gubin, A.A. Nikiforov
BINP, Novosibirsk, Russia

1 INTRODUCTION

VEPP-5 forinjector klystron gallery (KG) is a large electrophysical installation that includes main part of forinjector RF-system: RF-power supply, synchrosystem, klystron units. This paper presents the control system (CS) of it. CS is based on the ”standard” 3-level model [1].

Details of its realization reflects KG functional requirements and experience accumulated in BINP, as well as financial and technical abilities.

2 BASIC PROBLEMS, REQUIREMENTS AND STRUCTURE OF CONTROL SYSTEM

As the control object, the klystron gallery consists of 6 logically independent subsystems:

- 4 klystron posts (klystron amplifier with modulator and power supply);
- system of RF-power and synchronization;
- system of RF-phase and amplitude monitoring.

Operating condition of whole installation is pulsed with repetition rate up to 50 pps.

Each subsystem includes a large number of input/output channels. Some of the channels require the service at every impulse, some of them are ”oscilloscopic” in principal. CS hardware is based on CAMAC standard, each subsystem needs 1-2 crate for supply.

Particularities of installation are:

- high level of interference;
- logical independence of subsystems;
- operating condition are static.

CS problems are:

- control and monitoring supply for each input/output channel during 1-2 operation cycles are required;
- it is necessary to receive the whole KG condition data at least for 2-3 operation cycles;
- maintenance of log file;
- ability for organization of local control console to debug a subsystem;
- interaction with CS of other VEPP-5 installations.

To solve these problems and to take into account the installation features:

- each subsystem has a corresponding hard and software unit
- field electronics are concentrated near the corresponding subsystem
- extern connections of subsystems are minimized, only impulse or numerical channels are used
- main stream of information is concentrated on the second level, where control of the field electronics (level 3) and primary data processing are performing.

The first CS level (servers and consoles) is connected with the second level (based on the intellectual crate controllers (ICC)) by Ethernet link with specially developed in INP protocols (like UDP) on this top. CS structure is presented in fig. 1:

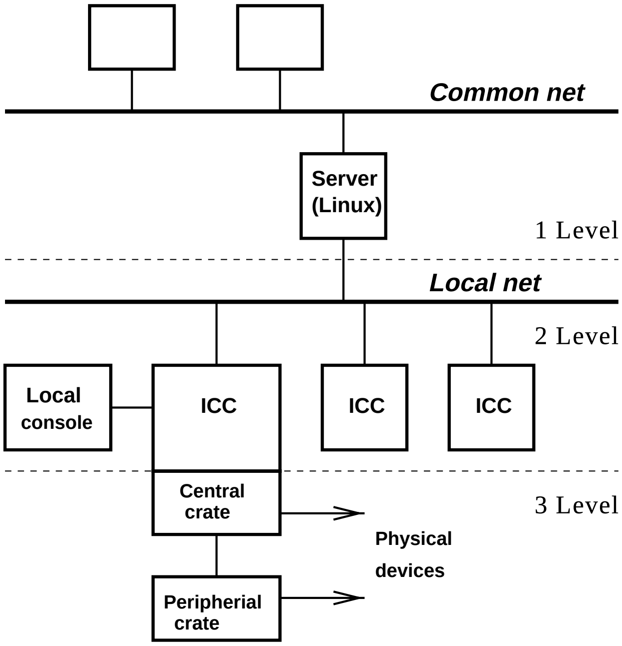

Figure 1: *Control system hardware structure.*

- each ICC supplies personal subsystem

0-7803-5573-3/99/$10.00@1999 IEEE.

- central computer (Server) runs as file server and channel of connection with each ICC
- control programs (operators console, dynamic data base at al.) work in the Server
- high level clients can access Server only
- local control console is organized on the basis of the proper ICC.

3 HARDWARE

As ICC we use Odrenok that is widespread in INP. To connect CAMAC and Ethernet, the special CAMAC-Ethernet adapter was developed. Pentium-166 with OS Linux is used as Server. ICC Odrenok has a ICL 1900 instruction set, added by the commands for CAMAC bus processes and vector operations. Odrenok provides the data transfer through CAMAC bus with the speed up to 1 Mbyte/sec. The speed and the reaction time on the interruption (about 15 mks) may be well used for design of the effective CS. The moderate requirements for exchange speed between subsystems and Server (average data stream is close to 10 kbyte/sec) make Ethernet protocol suitable for the ICC connection. Besides, the use of Ethernet doesn't requires the specialized drivers and interfaces in Server. CAMAC-Ethernet adapter used in our workstations provides the speed of the data exchange up to 400 kbyte/sec. It is sufficient for real-time network design.

4 MID-LEVEL SOFTWARE

The basis of a mid-level software is the specialized operation system ODOS for ICC Odrenok designed in INP [2]. It is multi-task OS with reaction time on the external events 1 ms approximately. ODOS supports a lot of special abilities for real-time control:

- timing of programs with step 100ms and 1s
- synchronization of the programs with LAM signals from CAMAC modules
- synchronization of the programs with signals from interruption register (CAMAC module where external pulse signals are registered with repetition rate up to 1 kHz)
- mechanism of interprogram and interprocessor communication

For interaction with other computers (for example high level) a special protocols based on Ethernet were developed. We can not install the whole TCP/IP (even UDP) due to Odrenok features. But our protocols are sufficiently complete and can be translated into TCP/IP in power servers. These protocols supplies:

- booting and starting of ODOS
- loading of programs during operation
- information exchange between any network processes
- terminal connection with high-level consoles
- transfer of directives from other computers
- access to filesystem.

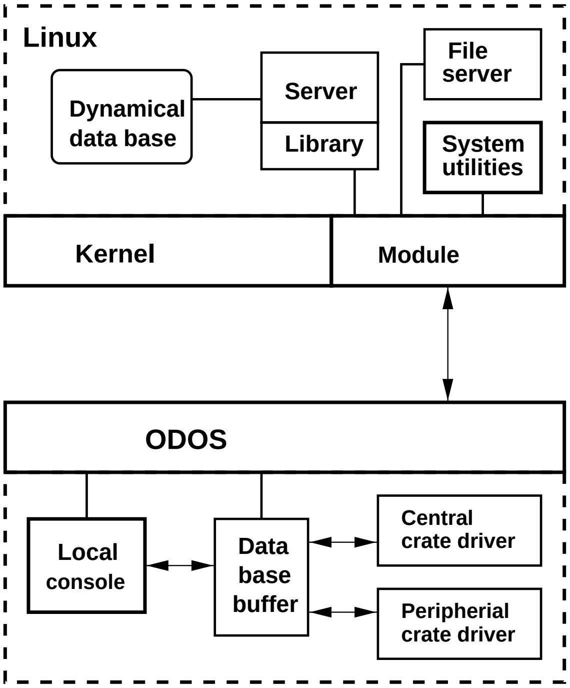

Figure 2: *Control system software structure.*

Protocols provide sufficiently high protection from deformations and losses of data, and guarantee the speed of exchange up to 200 kbyte/sec between two workstations. This OS is developed with the framework of the VEPP-4 and VEPP-2M control system modernization [3]. ICC drives CAMAC modules, collects and performs primary data analysis. ICC software typically consists of 4 programs (fig. 2):

- driver for central crate with ICC and CAMAC devices, necessary service at every cycle of installation operation
- driver for peripheral crate with non-intellectual controller
- ”buffer” - fragment of distributed data base - where accumulation an primary data analysis, and communication between Server and local control console are performing
- local console for organizations of an operative debugging near the devices.

Drivers are synchronized my means of LAM signals with CAMAC devices, and by ODOS interruption system with other programs. Data and ”command” flags exchange are organized through ”buffer” only. Data preparation and transfer to Server (or to local console) as well as the system reconfiguration appears by Server (local console) request only.

5 SERVER SOFTWARE

ODOS protocols use Ethernet packets of non-standard type. Therefore the Server requires input/output facilities for these packets. Support of the work with these packets is integrated into OS Linux as one of the protocols supported by kernel. This method allows to operate with ODOS packets using standard UNIX facilities. It is realized by loaded module supporting primary functions for transfer, receiving and stand by of this packets by standard socket interface and formalized as a client libraries and utilities. For normal interaction between ICC and Server, and for convenient user operation we also need the following minimum of the special software (fig. 2):

- file server for booting ICC through local network and maintenance of ODOS file system
- terminal console ICC client
- client library where main netware functions for ODOS protocols are supported.

Client software consists of control and visualizations programs as well as data base for description of whole installation [4]. Client programs compose the high level of CS and realize next functions:

- form of the control commands from operator to level 2 software
- form of request for receiving data of system condition, settings and measurements
- data visualizations and store

Data base consists of number of text files, where full description of the installation is stored. Special compiler translates data to binary dynamics data base for ICC. Server and ICC use the same data base text files, but binary data bases are different since the ICC an Server has different hardware platforms.

6 PRESENT STATUS

In present the working versions of loaded module, client libraries and utilities are completed and tested. Direct speed of the packets exchange between Server and ICC is up to 350 kbyte/sec (fig. 3) with maximum packet size of 1200 bytes. The Server operation with three concurrent ICC reduces the speed of exchange for not more than 10%. ICC application software and current version of Server software are tested. CS for the first subsystem is completed.

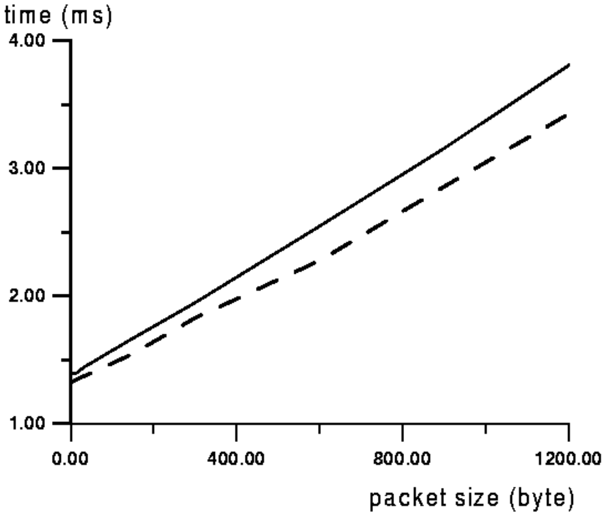

Figure 3: *Dependencies of the packet exchange time versus packet size: solid line – between Server and single ICC, dashed line – between Server and 3 ICC simultaneously.*

7 REFERENCES

[1] Gotz at al. *Experience with a standard model'91 based control system at the ESRF*, Proc. ICALEPCS'93 (Berlin) NIM 352A (1994) p. 22-27

[2] A.N. Aleshaev, Preprint INP 89-67, Novosibirsk 1989 (in Russian)

[3] A.N. Aleshaev at al. *VEPP-4 control system upgrade.* Proc. ICALEPCS'97 (Beijing) p. 34-36

[4] D.Yu. Bolkhovityanov at al. *The project of a control system software for a VEPP-5 complex*, Preprint INP 98-53, Novosibirsk 1998 (in Russian)