

Norihiko Kamikubota, Kazuro Furukawa, Kazuo Nakahara and Isamu Abe
 National Laboratory for High Energy Physics (KEK)
 Oho, Tsukuba, Ibaraki 305, Japan

Abstract

The network communication libraries for the next control system of the KEK Linac have been developed. They are based on TCP/IP sockets, and show high availability among the different operating systems: UNIX, VAX/VMS, and MS-DOS. They also show high source portability of application programs among the different computer systems provided by various vendors. The performance and problems are presented in detail.

I. INTRODUCTION

The KEK 2.5-GeV electron/positron linac has been controlled with a distributed processor network since its first operation in 1982 [1,2]. Since, however, the system resources have become inadequate for increasing demand, we have introduced several subsystems in order to extend the system capability [3]. Furthermore, we have studied the possibility of system rejuvenation by a complete replacement of the minicomputers and their associated fiber-optic network with new ones. The proposed next control system comprises Unix-based workstations as a man-machine interface, an Ethernet as a high-speed communication network, and VME stations as front-end systems [4].

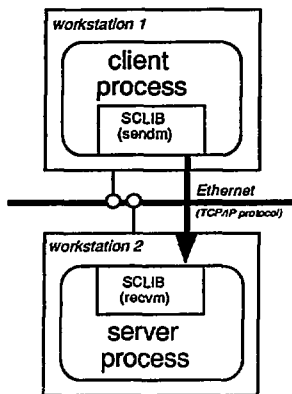
It is apparent that the tools for communication between workstations and VME stations are needed in the proposed system. In addition, some of the subsystems (the operator's console subsystem which comprises DOS-based personal computers [5], a diagnostic expert system for the linac injector developed in a Unix workstation [6], a beam-current monitor developed in a VAXstation [7], and so on) are expected to be used with the next control system. Thus, communication availability between different operating systems is important in our case.

We have developed a network communication library called "SCLIB" for the media of Ethernet with the TCP/IP protocol. Details are described in section II. Another library used to control the magnet power-supplies in the KEK linac, called "MGLIB", has been developed as an improved version of the SCLIB. The features are demonstrated in the section III. A discussion related to these libraries is presented in section IV. We hope that our present experience will provide good guidance for those who intend to introduce a similar network communication system.

II. NETWORK COMMUNICATION LIBRARY

A. Principles for the Library

The library "SCLIB" has been developed for the TCP/IP protocol. It was designed as a tool for real-time data transfer between processes, which is different from a file-transfer tool (like FTP), a file-sharing tool (like NFS), and an



```
main() /* client process */
{
    int sc_open(), sendm(), sc_errnd(), sc_close(); /* sclib */
    extern int sc_errno; /* error code holder for sclib functions */

    sd = sc_open("service_name@nodename"); /* open connection */
    if( sc_errno < 0 ) sc_errnd(); /* error message & exit */

    rtn = sendm( sd, "message"); /* send a character string */
    if( sc_errno < 0 ) sc_errnd();

    sc_close( sd); /* close connection */
}
```

Figure 1
 Example of transferring a character string from a client process (workstation 1) to a server process (workstation 2). The basic flow of the SCLIB function calls is also shown.

Table 1

Computer systems used in the present performance tests. An operating system and a rough estimate of the CPU power for each computer system is also given.

Computer	Operating System	MIPS
DEC DECstation5000	ULTRIX	24
Sun SPARCstation1	SunOS	12
NeXT NeXT	Mach	10
Mitsubishi MX3000II	OS60/UMX (System-V based)	3
Fujitsu A-60	SXA (System-V based)	3
DEC VAXstationII/GPX	VMS WIN/TCP installed	1
NEC PC9801ES (16MHz,80386)	MS-DOS PCTPC installed	-

interactive communication tool (like telnet). The library provides C-language functions. A typical example is shown in figure 1, together with the basic flow of the SCLIB function calls.

The principles that we considered when we starting to develop the library are the following: The first principle is communication availability between different operating systems. We have adopted an inter-process communication technique called "stream socket", which is based on the TCP/IP protocol. It provides the basic network communication functions with error-handling schemes for C language. The socket is one of the standard inter-process communication methods in Unix-based workstations, and is also available on other computer systems.

Since the socket functions require various kinds of network parameters, it is almost impossible to use by those who are not familiar with this field. Thus, the second principle is to prepare easy-to-use functions for application programmers. As shown in figure 1, the SCLIB functions used in a client process require only two parameters: the destination of data ("service_name@nodename") and the data, itself ("message"). In addition, the existence of an error-handling routine (*sc_errrend*) makes the client program very simple.

The third principle is to ensure high source portability of application programs among computer systems provided by various vendors. This leads to an easy replacement of the hardware of our present control system. Actually, we

have found several "small" differences in the socket functions provided by different vendors, and have tried to include vendor-dependent parts within the library as many as possible. The use of the C language is also preferable from this viewpoint.

B. Communication Throughput

In order to study the overall communication throughput, including library overhead, test programs with the SCLIB functions have been prepared. We have carried out measurements of the data-transfer times between two computer

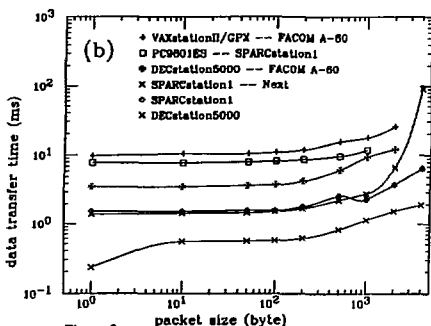
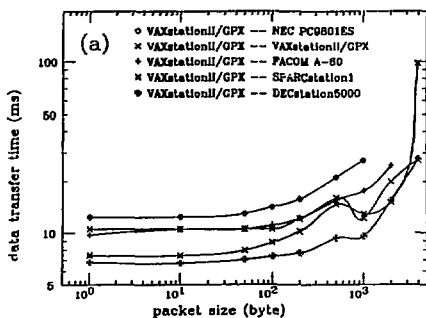


Figure 2
Measured data-transfer times between the computer systems listed in table 1.

(a) Data-transfer times between a VAXstationII/GPX and other computer systems.

(b) Data-transfer times for other combinations of computer systems. Two of them (SPARCstation1 and DECstation5000) are the result of the case that two processes are in the same workstation.

systems (processes) for the seven computer systems listed in table 1. The results for typical cases are shown in figure 2.

The following points can be pointed out.

(a) The data-transfer time takes 0.2-30 ms for 1-1000 bytes; of data, depending on the CPU power of the computer systems used.

(b) The data-transfer time remains almost constant as the size increases from 1 to 100 bytes. It typically becomes twice as the size increases from 100 to 1000 bytes.

(c) The best record among combinations of computers in table 1 was found in the case that the two processes are at the same workstation, DECstation5000. The data-transfer time was 0.2-1 ms for the 1-1000 bytes data.

(d) Assuming a data size of 1 kB, the overall throughput is evaluated to be 400-1000 kB/s for data transfer between typical workstations (10 or more MIPS).

In addition, the dips observed at around 1000 bytes are considered to be caused by the buffering scheme of the TCP protocol.

C. Discussion

The round-trip response time between the operator and a local control device was studied in the present control system [1]. It was evaluated to be of the order of 100 ms for our data of 128 bytes. According to the results of our present measurements, one or two orders better throughput can be expected in the next control system. If workstations with greater CPU power (for example 40-100 MIPS) become available in the future, further improvement in the communication throughput can be expected.

The measured communication throughput seems to be sufficiently high for most of the applications required to control the accelerator. However, it should be noted that the time required to open network connection, which usually takes 0.1-5 seconds and is necessary each time to start inter-process communication, is not included in the values given here.

III. MAGNET CONTROL LIBRARY

A. Introduction

The development of applications used to control the magnet power-supplies of the KEK linac was possible only with the minicomputers used in the present control system (Mitsubishi MELCOM 70/30) and its backup workstation (Mitsubishi MX3000II). In order to realize a better environment for software development, we first tuned the workstation serving as a gateway between an Ethernet and the present control system [3]. We then developed a library called "MGLIB" in order to make it possible to control the power-supplies at any of workstations connected with the Ethernet.

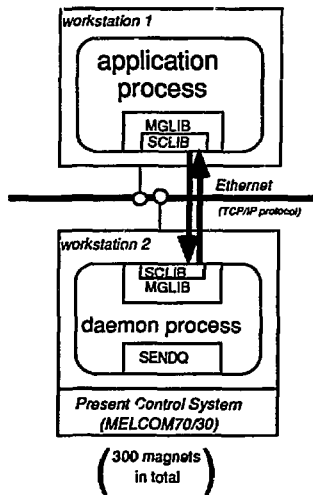
The mechanism of how a power-supply is controlled is shown in figure 3. When an application process calls a MGLIB function, an information message is sent to the

daemon process in the gateway. The daemon process controls the power-supplies according to the received message. It then returns a reply message which includes the result of the controlled power-supply. The SCLIB functions are used at the message interchange between two processes.

An example of MGLIB function calls is also shown in figure 3. Here, an initializing function (*sc_open* for the SCLIB) is not necessary, since initialization is carried out automatically at the first call of the MGLIB functions.

B. Source portability

Some problems concerning source portability have



```
main() /* application process */
{
  int mg(), mg_errand(); /* mglib functions */
  extern int mg_errno; /* error code holder of mglib */

  rn = mg("CUR", "magnet_name", &curr); /* get magnet current */
  if (mg_errno < 0) mg_errand(); /* error message & exit */
}
```

1

Figure 3

Example of an application process which controls a magnet power-supply of the KEK linac. Inter-process communications between an application process (workstation 1) and a daemon process (workstation 2) are carried out with the SCLIB functions (see text). Workstation 2 serves as a gateway between the Ethernet and the present control system.

arisen during the development of the MGLIB. The most serious one is the difference of byte order in the expression of numerical values. We have described byte-conversion routines for each computer system with C preprocessor statements.

Since the conversion routines are described in the MGLIB library, application sources are expected to show high portability. Actually, sample programs for basic operation of the magnet power-supplies show perfect portability among the computer systems in table 1. In addition, the basic availability of the SCLIB/MGLIB functions is also checked in a Hewlett Packard workstation with HP-UX, and a Force 68030-based VME system with OS9.

C. Discussion

The typical throughput of a single control action is obtained as 100-400 ms. This value is understood to be the round-trip response time in the present control system.

After introducing the next control system, we should re-develop a new daemon (server) process with a VME environment. It would not be easy since the development of such a server process requires deep understanding of the network parameters and workstations, even with the present SCLIB functions. However, the sources of applications developed so far with the MGLIB functions will be available without any modification, even in the next control system.

IV. DISCUSSION

One of the problems concerning an Ethernet is response delays due to network packet collisions, which inevitably occur in an Ethernet when the network traffic rate is considerably high. An easy answer to this is to replace an Ethernet with a FDDI (Fiber Distributed Data Interface) fiber-optic network system. It is a token-ring type network and is more suitable for a real-time purpose. In addition, we would expect one order better communication throughput since it is capable of a communication rate of 100 Mbit/s. Moreover, it is worth noting that a replacement is possible without any software modification.

The present libraries provide a possibility to use an Ethernet as a high-speed data-transfer network with low cost. There exists a plan to use these libraries for the control system of the TENKO-100, a 100-meter long laser interferometer aimed for detecting gravitational waves now under construction in ISAS (The Institute of Space and Astronautical Science) [8,9].

V. ACKNOWLEDGMENTS

We would like to thank Professor A. Asami, head of the Injector-Linac division in KEK, for his support and kind encouragement. We also thank the staff and operators of the linac for their valuable discussions and assistance.

VI. REFERENCES

- [1] K. Nakahara, I. Abe, R.P. Bissonnette, A. Enomoto, Y. Otake, T. Urano and J. Tanaka, *Nucl. Instr. Meth.* A251(1986)327
- [2] K. Nakahara, I. Abe, A. Enomoto, Y. Otake and T. Urano, *Nucl. Instr. Meth.* A247(1986)153
- [3] K. Furukawa, N. Kamikubota, K. Nakahara and I. Abe, *proceedings (Int. Conf. on Accelerator and Large Experimental Control Systems, Vancouver Canada, October 1989)*, *Nucl. Instr. Meth.* A293(1990)16
- [4] K. Furukawa, N. Kamikubota, K. Nakahara and I. Abe, "Upgrade Plan for the Control System of the KEK e-/e+ Linac", this conference
- [5] K. Nakahara, I. Abe, K. Furukawa and N. Kamikubota, *proceedings (Int. Conf. on Accelerator and Large Experimental Control Systems, Vancouver Canada, October 1989)*, *Nucl. Instr. Meth.* A293(1990)446
- [6] I. Abe, M. Kitamura and N. Nakahara, "A Diagnostic Expert System in FF Linac Control", this conference
- [7] K. Furukawa, N. Kamikubota, T. Shidara, K. Nakahara and G. Horikoshi, *proceedings (14th. Int. Conf. on High Energy Accelerators, Tsukuba Japan, August 1989)*, *Particle Accelerators* 29(1990)245
- [8] N. Kamikubota, K. Furukawa, K. Nakahara, Y. Ogawa and A. Asami, "A study of the Control System for Long-Baseline Laser Interferometer", the proceedings of the Sixth Marcel Grossmann Meeting on General Relativity (MG6) in press, Kyoto Japan, June 1991; KEK preprint 91-124
- [9] N. Kawashima, T. Yokota, Y. Ogawa and N. Kamikubota, "Development of 100 m Laser Interferometer Gravitational Antenna (TENKO-100)", the proceedings of the Sixth Marcel Grossmann Meeting on General Relativity (MG6) in press, Kyoto Japan, June 1991