Inhouse Computer Network and HOST Computer

Toshiyuki Sakai,* Koh-ichi Tabata,* Hirokazu Ohnishi* and
Shigeyoshi Kitazawa*

Abstract

Computer network is defined as several standing-alone computer systems (Hosts) are combined on an equal footing with each other through high-speed communication network.

The KUIPNET (Kyoto University Information Processing NETwork) is an inhouse-oriented computer network developed in accordance with the above philosophy, while most announced inhouse systems are constructed rather on the basis of computer-complex configuration. KUIPNET is also prepared to join a prospective wider-distributed computer network via 48 kbps lines. In KUIPNET, we can transmit the data at more than 200 kbps (effective continuous transfer rate) between two processes in different Hosts to share the resources of intelligent terminals in real-time. Two Hosts in KUIPNET are minicomputers in which we have been able to implement reasonable-sized Network Control Programs.

In this paper the aspect of Hosts in the KUIPNET is mainly described.

1. Introduction

The first version of the KUIPNET (Kyoto University Information Processing NETwork) constructed in the fall of 1973 is presented (Fig. 1).
KUIPNET is one of typical inhouse-oriented computer networks developed under the orthodox philosophy of computer network. The connection principle almost coincides with that of the ARPANET, then it is possible to connect to a prospective wider network in national range.

Yet, comparing with the ARPANET, we emphasized the following two problems and have obtained positive results:

(1) Enhancement of local network: The high-speed real-time data transfer has been realized between any pair of inhouse Hosts in concurrence with the key-board type of message communication between other Hosts.

(2) Minicomputer Host: A computer network should accept any type of computer. Experiment has done whether it is possible to connect a

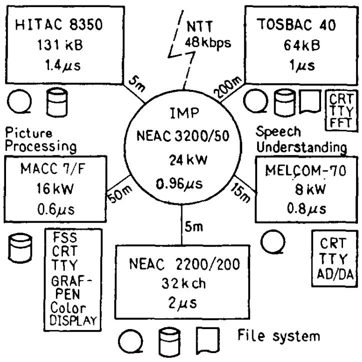

Fig. 1 Organization of KUIPNET

This paper first appeared in Japanese in Joho–Shori (Journal of the Information Processing Society of Japan), Vol. 15, No. 12 (1974), pp. 948~954.

* Faculty of Engineering, Kyoto University

minicomputer system to the network, that is, whether it is possible to implement a reasonable size of a network control program in a minicomputer Host.

In this paper the aspects of Hosts in the KUIPNET are mainly described.

2. Inter-process Communication Method and Protocol

The inter-process communication method should be unified over the network, and this agreement on communication is called "protocol". There may be two alternatives in the inter-process communication protocol:

(1) concentrated protocol or distributed protocol,

(2) connection-oriented protocol or message-oriented protocol.

Concerning (1), in the case of concentrated protocol, a central NCP center which is placed somewhere in the network controls communication in co-operation with NCPs of the individual Hosts. The NCP in each Host has the advantage of simplicity, but the disadvantage of ill-extensibility and troublesomeness. In the case of distributed protocol, an NCP itself does not exist somewhere in the network, but entire NCPs are in every Host, and the advantage and disadvantage of this system are opposite to those of the former case. The KUIPNET employed the latter protocol similar to that of the ARPANET.

Concerning (2), in the case of connection-oriented protocol, before the substantial data transfer occurs a connection (link) is to be established between processes in two Hosts. After the connection is established, several messages transferred over the link, and the link is closed when the transmission has finished. In the case of message-oriented protocol, one link corresponds to one message. A new link is established for a coming message, then after transferring a message the link is immediately closed. This protocol is suitable for the single-message type of communication. Because in the KUIPNET the high-speed and lasting data transfer is dominant, the former connection-oriented protocol (ARPA-type) is adopted, and if necessary it is possible to allocate priority to any one of links established by this protocol.

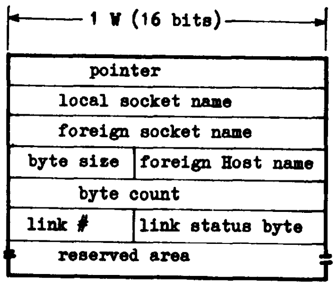

Fig. 2 Configuration of LCB

3. Network Control Program for Host

The establishment of logical connection for inter-process communication and the message flow control is supported by a network control program resident as a part of Host OS.

3.1 NCP for KUIPNET

The NCP implemented for the KUIPNET registers the connection status of process in the Link Control Block (LCB). Fig. 2 shows a configuration of LCB corresponding to a connection. In Fig. 3 are illustrated the functions of an NCP resident in the main storage of a Host. A user process issues system calls as shown in Table 1. An NCP compiles the information in an LCB into a control command,

Table 1 NCP system calls

l: local socket name
f: foreign socket name
d: data name

Init(l,f)	: request for connection
Listen(l)	: wait for connection
Send(l,d)	: data transmission
Receive(l,d)	: data reception
Close(l)	: refuse/release connection

which makes a portion of text of a control message. The message is sent to the destination Host via the control link with a leader added in front of the text field. The leader indicates the destination Host.

An NCP occupies 1 kW storage for MACC and MELCOM, and 4 k characters for NEAC, in both cases including a buffer of about 8000 bits.

3.2 An example of inter-process communication

A co-operation work can be done by processes scattered in three Hosts. First every Host is loaded with its own NCP and monitor, and waits for a command from an operator. An operator types a command via a console typewriter either of MACC or MELCOM, so as to have all Hosts loaded with their user processes which are in the NEAC's file system. The procedure stated above is shown in Fig. 4. The arrow 1 in the figure is the procedure where, as shown in Table 2, the process in MACC commands the process in NEAC by executing a series of system calls, which consists of Init, Send(task name), and Close, to distribute to every Host the filed programs corresponding to the task name. The process in NEAC which had issued Listen and has been blocked is unblocked and executes program transfer (arrow 2 in Fig. 4). Then by step 3, the same task name is sent to the process in MELCOM, and the things go in step 4 and 5 just in the same way as in step 1 and 2. The step indicated as 6 certifies that all Hosts are loaded with the processes which co-operate the work.

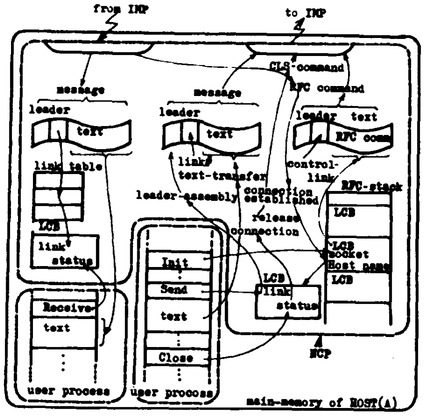

Fig. 3 NCP and user process

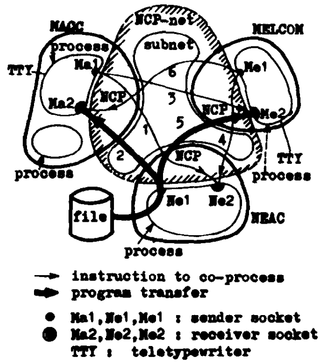

Fig.4 An example of cooperating processes over three Hosts

Table 2 Sequence of system calls corresponding to Fig. 5.

D: cooperating-process name d: program to be transfered

stage	MACC	NEAC	MELCOM
		Listen(Ne 2)	Listen(Me 2)
1	Init(Ma 1, Ne 2)		
	Send(Ma 1, D)	Receive(Ne 2, D′)	
	Close(Ma 1)		
2	Listen(Ma 2)	Init(Ne 1, Ma 2)	
	Receive(Ma 2, d′)	Send(Ne 1, d)	
		Close(Ne 1)	
3	Init(Ma 1, Me 2)	Listen(Ne 2)	
	Send(Ma 1, D)		Receive(Me 2, D′)
	Close(Ma 1)		
4	Listen(Ma 2)		Init(Me 1, Ne 2)
		Receive(Ne 2, D′)	Send(Me 1, D)
			Close(Me 1)
5		Init(Ne 1, Me 2)	Listen(Me 2)
		Send(Ne 1, d)	Receive(Me 2, d′)
		Close(Ne 1)	
6			Init(Me 1, Ma 2)
	Receive(Ma 2, D′)		Send(Me 1, D)
			Close(Me 1)

4. Functions Realized by KUIPNET

The KUIPNET, which is the inhouse computer network for the purpose of information processing researches, has the following functions:

(i) Function-sharing between a minicomputer and the general-purpose computer: The advantage accomplished by the computer network is as follows; the resource sharing such as color display of speech sound, and the parallel processing as a product of function-sharing between a minicomputer and the general-purpose computer. Input/output control and pre-processing are performed with the high-speed control ability of a minicomputer, on the other hand complicated processing of extracted parameters and filing of data are performed by a general-purpose computer. Consecutive works which were processed fragmentarily in the coustomary way are now possible to be processed continuously and without delay via the online computer network.

(ii) Remote file: Although various types of recent minicomputers are of good performance and have become cheap enough to be installed in every laboratory, the relative expensiveness of peripherals for those minicomputers restricts the extent of applications within specialized works. So it is beneficial to connect a minicomputer with other computer well equiped with peripherals.

(iii) Real-time processing based on high-speed communication: The transmission bit rate is 1.6 Mbps on the line between IMP and Host in the KUIPNET. The effective continuous transfer rate, defined to be the minimum value of the ratio of the total bits in any message to the interval between its arrival and that of the just previous message, is assured to be more than 300 kbps between a pair of processes in any separate Hosts which are designated by the priority specifying circuit. The circuit is made in the IMP of the KUIPNET. With this characteristics, the network mode job which requires a special demand such as real-time one is accepted together with the regular network mode jobs.

5. Operating System for Minicomputer Host

The operating system of minicomputer Hosts must have the following two functions; as a Host in computer network and as an intelligent terminal. The monitor occupies about 2 kW (1 W: 16 bits, buffer: 500 W) including NCP. The organization of monitor is shown in Fig. 5.

5.1 Design principle of monitor for minicomputers

(i) Multi-job and scheduling: In order to make a minicomputer Host still possible to manage the network independent job, the following two kinds of job must be treated together; the minicomputer mode job and the network mode job which is co-operating with other Host. Multi-programming technique has been introduced. Although regular processes are scheduled by the first-in first-out basis, only a single process can be given the top priority. Owing to this priority, it is possible to execute the high-speed processing within real time regardless of existence of many other processes to be processed.

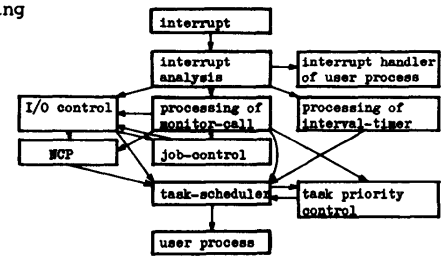

Fig.5 Organization of monitor for minicomputer-Host

(ii) Input/output control: User processes are allowed to treat the special interruptions that have been registered beforehand. Processes directly execute input/output instructions that are not accompanied by interruptions, so as to reduce input/output over-head by monitor and to achieve high-speed processing characteristics.

(iii) Log-in from terminal: User may develop, initiate, and log-in his own software and access the subsystem for information processing and interactive interface program from various types of terminals.

5.2 Interactive processing program

(i) Assembler: In order to allow a larger size of user programs and to make the list-printing time short, a protocol was settled for communication with the file control program in NEAC, and was implemented around an assembler program of a minicomputer.

(ii) Debugging program: Because of insufficiency of protective functions against program errors, a minicomputer Host may violate other Hosts in the network. The debugging program has been developed which is a utility to debug a program by executing instruction by instruction checking effective address boundary and priviledged instructions. During this execution, the program under debugging can actually perform data input/output to/from the network or peripherals, and moreover is served tracing and snapshot functions.

6. The Future

The KUIPNET is scheduled to be soon connected to the remote larger computer systems via the high-speed communication lines or switching network.

7. Acknowledgement

We authors acknowledge the contribution of many persons in our laboratory on the construction, operation, and use of the KUIPNET.

References

1) F.E. Heart et al: The interface message processor for the ARPA computer network, AFIPS, SJCC, Proceedings, p. 551.
2) T. Sakai et al: Inhouse computer network KUIPNET, Paper for Techinical Group on Computer, I.E.C.E., Dec. 1973, (in Japanese).
3) S.M. Ornstein et al: The Terminal IMP for the ARPA computer network, AFIPS, SJCC, 1972, p. 243.
4) T. Sakai et al: Inhouse Computer Network for Information Processing and some Applications, preprint for the seminar on Computer Asisted Chemical Research Design, July, 1973.
5) S.D. Crocker et al: Function-oriented protocols for the ARPA Computer Network, AFIPS, SJCC, 1972, p. 271.
6) T. Sakai et al: Computer Network, J. Soc. of Instrument and Control Engrs, Japan, Vol.12, No.11, 1973, p. 863, (in Japanese).
7) A. Mckenzie et al: Host/Host protocol for the ARPA Network, NIC 8246, Jan. 1972.
8) T. Sakai et al: Inhouse computer network KUIPNET and HOST software, Proceeding of the 14th Annual Convention, Information Processing Soc. of Japan, 1973, (in Japanese).