7th Conference on Local Computer Networks 1982

7th CONFERENCE ON LOCAL COMPUTER NETWORKS

~~000536~~

7th

HILTON INN
MINNEAPOLIS
MINNESOTA
OCTOBER 11-13, 1982

sponsored by

COMPUTER CENTER
UNIVERSITY OF MINNESOTA

IEEE COMPUTER SOCIETY
TC-COMPUTER COMMUNICATIONS

TWIN CITIES IEEE SECTION

IEEE CATALOG NO. 82CH1800-2
LIBRARY OF CONGRESS NO. 82-82281
COMPUTER SOCIETY ORDER NO. 433

COMPUTER SOCIETY PRESS

Order from: IEEE Computer Society
Post Office Box 80452
Worldway Postal Center
Los Angeles, CA 90080

IEEE Service Center
445 Hoes Lane
Piscataway, NJ 08854

Published by IEEE Computer Society Press
1109 Spring Street
Silver Spring, MD 20910

IEEE Catalog No. 82CH1800-2
Library of Congress No. 82-82281
IEEE Computer Society Order No. 433

Order from: IEEE Computer Society
Post Office Box 80452
Worldway Postal Center
Los Angeles, CA 90080

IEEE Service Center
445 Hoes Lane
Piscataway, NJ 08854

The Institute of Electrical and Electronics Engineers, Inc.

PREFACE

This proceedings represents the conference record of the 7th Conference on Local Computer Networks. The Conference was held October 11-13, 1982 at the Hilton Inn in Minneapolis, Minnesota.

The conference general chairman was Dr. Abe Franck. The program chairman was Mr. Richard Tett. Sponsoring groups were the University of Minnesota Computer Center, the Twin Cities Chapter of the IEEE, and the IEEE Computer Society Technical Committee on Computer Communications.

We wish to extend our thanks to all those who contributed to making the conference a success. To the authors, whose knowledge and efforts create the body of the conference, we give special thanks and hope that they have gained from their participation. We also wish to thank the persons who presented tutorials: David C. Wood, Harvey A. Freeman, and William J. Wood.

We would like to especially acknowledge the two keynote speakers, Bob Metcalfe, President, 3COM Corporation, and Harry Saal, President, Nestar Systems Inc. for sharing their knowledge and experience with those who attended the conference.

Rick Tett
Abe Franck

TABLE OF CONTENTS

SESSION 1

Implementation Strategies

WORKSTATION CLUSTERING IN A PERSONAL COMPUTER LOCAL NETWORK

Barry A. Trent

Architecture Technology Corporation
Minneapolis, Minnesota

Abstract

This paper describes the design and implementation of a workstation Cluster Server in a local computer network. The Cluster Server provides up to six non-homogeneous microcomputers with virtual disk and printer facilities derived from resources available on the local network. The Cluster Server is a local network node which connects to each of the clustered workstations via a RS-232 link. The server acts as a communications processor for the workstations, translating their requests for virtual disk and printer access into remote requests to file servers and printer servers on the local network.

The workstation clustering scheme allows for indirect connection of non-homogeneous processors to local networks where direct connection is not possible due to interface incompatibilities. The particular implementation described here is based upon the Nestar Cluster/One, a homogeneous local network of Apple microcomputers, and allows non-homogeneous CP/M-based microcomputers such as the Osborne I, the Xerox 820, the IBM Personal Computer, and S-100 bus machines to have access to network disk and printer services.

Introduction

The past two years have seen the growth of tremendous interest in local computer network technology and the introduction of local computer network hardware by a variety of manufacturers. This trend is continuing and accelerating, with particular emphasis being placed on local networks based on personal computers. Many local network products currently available are designed for the homogeneous computing environment, however, in that they allow only for the connection of a single type of processor to the network, or for the connection of the processors of a single computer manufacturer [1,2]. The problem of connecting non-homogeneous processors to local networks is one which must be addressed before local computer networks can gain widespread acceptance, if for no other reason than the fact that a large installed base of non-homogeneous personal computers already exists in organizations throughout the world.

The approach which has been used until now to provide local network connections by many manufacturers is to offer a special interface card which plugs into an unused expansion slot of the personal computer and allows network communication to occur. This approach has the advantage that it allows network parameters and hardware connection strategies to be optimized for the capabilities and limitations of the processor in question, but has disadvantages associated with it as well. Using this scheme, a separate hardware interface card must be designed and produced for each processor to be connected to the network which is not based on a standard backplane bus interface. As personal computers with non-standard or proprietary buses proliferate, this sort of connection approach will entail an ever increasing hardware development and support effort. Further, many personal computers have been designed and are being designed today which have no expansion bus (e.g., the Osborne 1 and the Xerox 820). Obviously, new methods will have to be found to connect these processors onto local networks.

This paper describes a different approach to local network connection technology which is more amenable to the non-homogeneous networking environment. The approach to be described is similar to the kind of technology which has been used to allow connection to long-haul networks such as ARPANET [3], which has supported communication between non-homogeneous processors for many years. The basic idea is to configure a computer which is

CH1800-2/82/0000/0003$00.75

directly connected to the local network as a 'front-end' or 'communication processor' around which a variety of non-homogeneous user workstations may be clustered. The communication processor provides a variety of specific services derived from resources available on the local network to the clustered workstations, communicating with them over standard RS-232 serial lines. Specifically, this paper discusses the design of such a workstation clustering device which allows non-homogeneous personal computers running the CP/M operating system to be indirectly connected to the Nestar Cluster/One local network through an Apple II processor which acts as a communications front-end and provides access to virtual disk storage and virtual printing facilities. This approach does not require the development of any new computer hardware and requires only minimal software changes to the CP/M input/output drivers resident on the clustered workstations. The software which runs in the Cluster Server itself is implemented in Apple II Pascal and a small amount of 6502 assembly language.

Overview of the Nestar Cluster/One Local Network

The workstation Cluster Server was implemented on a Nestar Cluster/One Model A local computer network with file & print server facilities (see Figure 1) [1]. The Cluster/One is a network designed to interconnect up to 64 Apple II and Apple III personal computers. A special interface board is inserted into an unused slot on the Apple's bus to allow communication between Apples over a 16 conductor flat cable at a raw data rate of 240 KBPS. A modified CSMA/CD convention controls the access of an individual computer to the network. The network operates at distances up to 1000 feet.

Nestar File Server

The Nestar 'Network File Server' (NFS) provides complete disk storage facilities for the Apple computers connected to the local network, with up to 66 MBytes of storage online at a single Apple II NFS station. The key concept on which the NFS is based is that of the 'virtual disk'. The hard disks connected to the NFS can be partitioned into a set of virtual disks which are read and written transparently (i.e., just like real floppy disks) by Apple stations on the network. A special 'NFS Command' program is provided which runs in user stations and communicates with the NFS, allowing virtual disks to be 'MOUNTED' (equivalent to inserting a real floppy disk into a floppy disk drive), 'UNMOUNTED' (equivalent to removing a floppy disk from a floppy disk drive), 'CREATED', and 'DELETED'. Virtual disks are addressed using a pathnaming convention whereby the virtual disk's name describes a path through the tree-structured file system supported by the NFS. Virtual disks can be mounted in one of several access modes (read/write, read only, shared, exclusive, etc.) and, in the Apple Pascal environment, may be created with a size ranging from 512 bytes to 16 MBytes. In addition to the NFS Command program, a set of user-callable subroutines is provided which allow application programs to communicate with the NFS in order to manage virtual disks

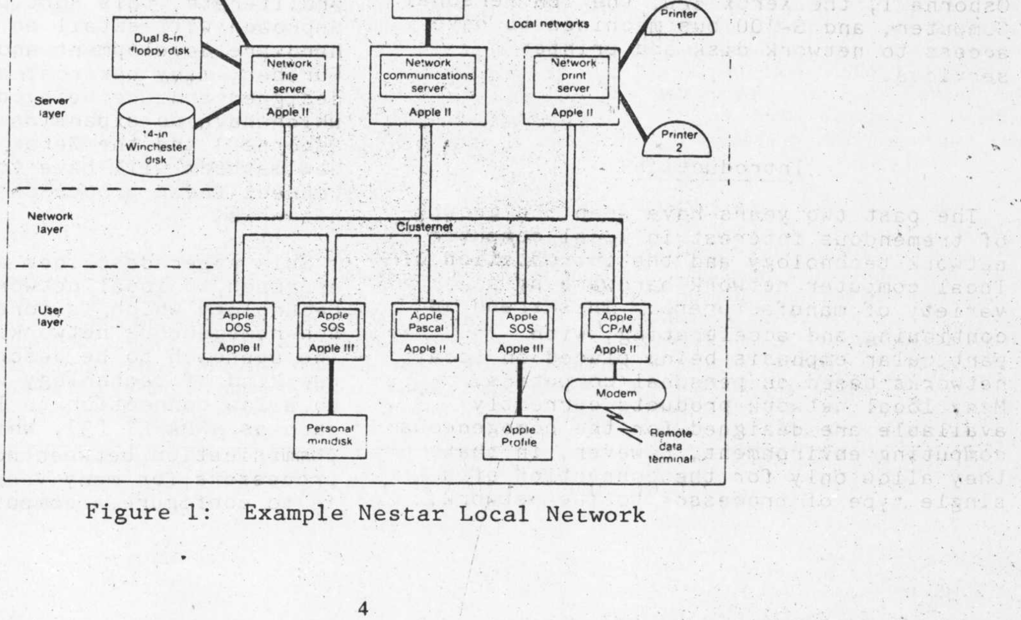

Figure 1: Example Nestar Local Network

under program control. A user station on the network may have up to 10 Pascal virtual disks mounted at one time.

Nestar Printer Server

The Nestar 'Network Printer Server' (NPS) provides shared printing services for the Apples connected to the local network. Requests for files to be printed are entered into a queue maintained on the file server, and the printer server periodically checks for new entries in the queue, printing out files as requested. The NPS can support up to 6 attached printers, and allows print requests to be tagged with requests for special printing forms and printing priorities.

Overview of CP/M Input-Output Drivers

The device-dependent I/O drivers of the CP/M operating system are grouped together in a section of the OS generally referred to as the BIOS (Basic Input Output System). BIOS routines are accessed through a jump vector in memory, with parameters being passed in predefined CPU registers [4]. The following BIOS calls are pertinent to the operation of the Cluster Server :

Disk I/O Driver Calls

SELDSK - select a specific disk drive for subsequent disk operations.

HOME - move the head on the selected disk drive to track zero.

SETTRK - set the track address for subsequent operations on the currently selected disk.

SETSEC - set the sector address for subsequent operations on the currently selected disk.

SETDMA - set the memory address of the disk I/O buffer. The buffer is 128 bytes long.

READ - read the 128 byte disk sector currently being addressed into the disk I/O buffer.

WRITE - write the contents of the disk I/O buffer into the currently addressed 128 byte disk sector.

Printer Driver I/O Calls

LIST - send one character to the 'LST:' device (typically the printer).

LISTST - return the status (ready/busy) of the 'LST:' device.

Usage of the Cluster Server

The Cluster Server provides each of the clustered CP/M workstations with two simulated 102 Kbyte floppy disk drives and a virtual printer capable of buffering arbitrarily large print files for eventual spooling to a printer server on the local network. In describing the usage of the Cluster Server, an example of its operation with an Osborne 1 computer will be used. Operation with other CP/M machines would be virtually identical.

The Osborne 1 is a Z-80 based personal computer with two built-in 5 1/4-inch floppy disk drives, an asynchronous RS-232 serial interface, an IEEE-488 bus interface, and 60 Kbytes of user-programmable RAM [5]. The two built-in disk drives are addressed by the user under CP/M as drives 'A:' and 'B:'. The Osborne is booted from a real floppy disk in drive 'A:' which contains the CP/M operating system and a modified version of the standard Osborne CP/M BIOS (modifications to the CP/M BIOS are discussed below). The two disk drives built into the Osborne function just as they would in a 'stand-alone' configuration. The two virtual floppy disk drives provided by the Cluster Server are addressed by the user as drives 'C:' and 'D:'. The virtual printer is addressed as the CP/M 'LST:' device, just as a real printer attached to a CP/M machine would normally be addressed. Control over virtual disk mounting and virtual printer operations is facilitated by the use of a standard CP/M program called NETCMD. The NETCMD program may be loaded into memory either from one of the built-in disk drives or from one of the virtual disk drives. The NETCMD program provides the following commands to the user :

1) MOUNT a virtual floppy disk for reading and writing in virtual drive 'C:' or 'D:'.

2) INITialize the virtual printer. The maximum allowable size of the print file to be created is specified at initialization time.

3) CLOSE the virtual printer, spooling the buffered print file to the Nestar printer server.

4) PURGE the virtual printer, destroying the buffered print file.

Command: MOUNT /main/joe/data,c

Result: The virtual disk named '/main/joe/data' is now readable and writable on virtual drive 'C:'.

Command: INIT,SIZE=10000

Result: The virtual printer is initialized. The file(s) to be printed will contain less than 10000 characters.

Command: CLOSE,NAME=myfile,FORMS=standard

Result: The contents of the virtual printer are spooled to the printer server. The printout will be identified with the name 'myfile', and will be printed on 'standard' forms.

Command: PURGE

Result: The contents of the virtual printer are destroyed.

Figure 2 : Examples of NETCMD commands

Examples of valid commands are given in Figure 2.

Operation of the virtual disks and virtual printer is transparent to the user once virtual disks have been mounted and/or the virtual printer has been initialized using the NETCMD program. Programs which read to or write from disk through standard CP/M operating system calls function identically using real or virtual disks, or both. Similarly, programs which send data to the 'LST:' device for printing will have the data buffered in the virtual printer for eventual spooling to a printer server. Since all modifications to CP/M occur at the BIOS level, no existing programs need to be re-compiled or re-linked in order to work with the Cluster Server.

Operation of the Cluster Server

The Cluster Server functions as a communications processor for the workstations which are connected to it via RS-232 serial interface cards plugged into the Apple II's bus, as shown in Figure 3. The workstations send communication instructions over the serial links to the server, which processes the instructions one at a time. Communication instructions consist of a one byte operation code, an optional track/sector address and an optional 128 byte block of data. The presence or absence of the optional data fields is specific to the individual communication instructions. For disk read and control instructions, the Cluster Server responds to instructions with a 128 byte block of data. The basic structure of a communication instruction is shown in Figure 4.

The server continually polls the input buffers of the serial interface cards waiting for the reception of a one byte op code. Once an op code is received, the server sends back an 'ACK' character and the rest of the data associated with the instruction and its processing are transferred over the serial link. After one of the workstations with a communication instruction to be processed sends the op code byte of the instruction, it waits to receive the 'ACK' (or 'NAK', for invalid op codes or other errors) back from the server before sending or receiving the remaining instruction data. Since the serial interface hardware is always capable of receiving and buffering one byte of data, this scheme insures that no instruction data is lost thru transmission before the server is actually ready to process the instruction. Workstations may have to wait a short period for their instructions to be processed if the server is already executing a communication instruction from another workstation when the op code byte of the instruction is sent.

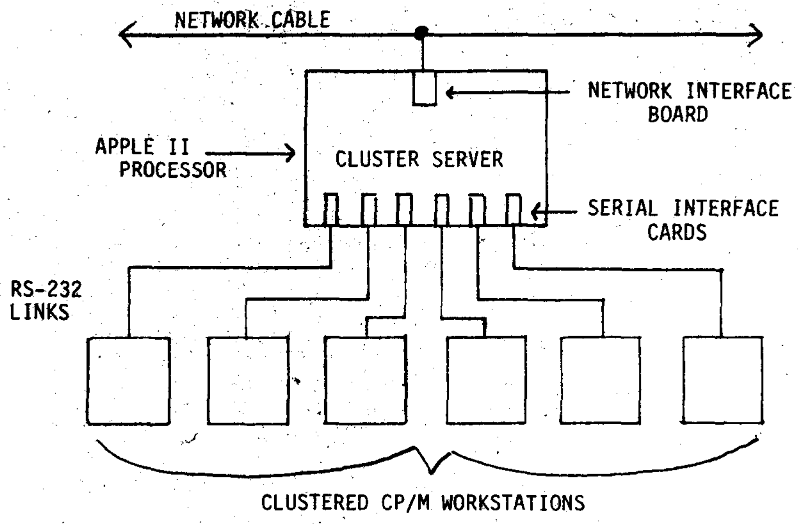

Figure 3: The Cluster Server

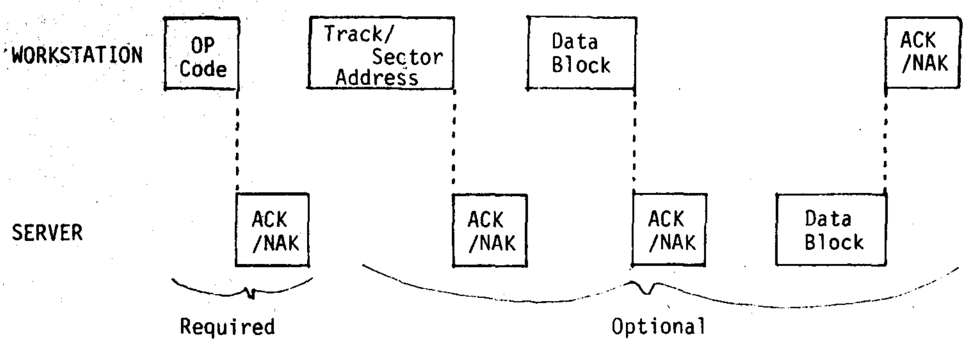

Figure 4: Basic Communication Command Structure

Virtual Disk Read/Write Operations

The CP/M BIOS routines SELDSK, HOME, SETTRK, SETSEC, SETDMA, READ and WRITE are modified to facilitate virtual disk read/write operations. If SELDSK is called from the operating system and one of the real disk drives is selected for subsequent operations, calls to HOME, SETTRK, SETSEC, SETDMA, READ and WRITE are simply passed on to the standard BIOS routines which handle real floppy disk I/O. If SELDSK is called and one of the virtual disks is selected for subsequent operations, calls to HOME, SETTRK, SETSEC and SETDMA cause the disk track, sector, and buffer addresses to be saved within the modified BIOS routines. When READ and WRITE are called with virtual disks selected, a communication instruction is sent to the Cluster Server in order to read or write the addressed disk sector. The structure of VREAD and VWRITE communication instructions is shown in Figure 5.

Virtual Printing Operations

The CP/M BIOS routines LIST and LISTST are modified to facilitate virtual printing operations. Each time LIST is called the character to be printed is stored in a 128 byte buffer within the BIOS. When the buffer is full the data to be printed is sent to the cluster server in a VPRINT communication instruction. The structure of the VPRINT communication instruction is shown in Figure 5. The LISTST BIOS entry is modified to always indicate that the 'LST:' device is ready to receive data, since no real printer is actually attached to the workstation.

Virtual Device Control Operations

The four virtual device control commands are sent to the Cluster Server for processing using a command channel constructed using the modified BIOS routines SELDSK, SETDMA, and WRITE. The channel is set up via a call to SELDSK which selects a 'disk' whose address is 255. This is an invalid disk address for normal CP/M operations and serves to signal the BIOS that a virtual device control command is about to be sent to the Cluster Server. SETDMA is then called to set the address of the 128 byte buffer in memory which contains the command entered by the user. Finally, WRITE is called and a VCONTROL communication instruction is sent to the server which contains the text of the command typed in by the user. The server processes the control instruction and returns a message to be echoed at the workstation's console. This message is copied back into the 128 byte buffer. The structure of a VCONTROL communication instruction is shown in Figure 5.

Communication Instruction Processing

The Cluster Server is capable of processing four basic instructions, VREAD, VWRITE, VPRINT, and VCONTROL. These instructions perform virtual disk reads, virtual disk writes, virtual printing. and virtual device control, respectively. A Device Control Table which describes the virtual disks and virtual printer status of each clustered station is maintained in the Cluster Server. Since the Cluster Server is designed to support at least 6 clustered workstations, each with two virtual disk drives and a virtual printer, the 10 virtual drives provided to each Apple II Pascal user station by the NFS is not necessarily sufficient to allow all virtual disks in use at any one time to be mounted on the Cluster Server simultaneously. To overcome this problem, virtual disks are MOUNTed and UNMOUNTed from the Cluster Server on a LRU (Least Recently Used) basis.

The manner in which each of the communication instructions is processed is described briefly below :

VCONTROL - the text of the user's command is processed and any changes to currently mounted disks or printer status is reflected in an update to the Device Control Table. If the virtual printer is to be initialized, a NFS virtual disk is created in order to buffer data written to the virtual printer. When the virtual printer is closed, a print request is entered into the NPS queue. A textual message is returned to the workstation which issued the command indicating whether the control operation succeeded or whether some error occured. This message is echoed on the workstation's console by the NETCMD program.

VPRINT - the 128 byte block of data to be printed is written out to the NFS virtual disk being used to buffer the print file currently being created. This virtual disk is marked as the most recently used, or the 'youngest' disk currently mounted on the Cluster Server.

VWRITE - the track/sector address of the disk sector to be written is converted to a

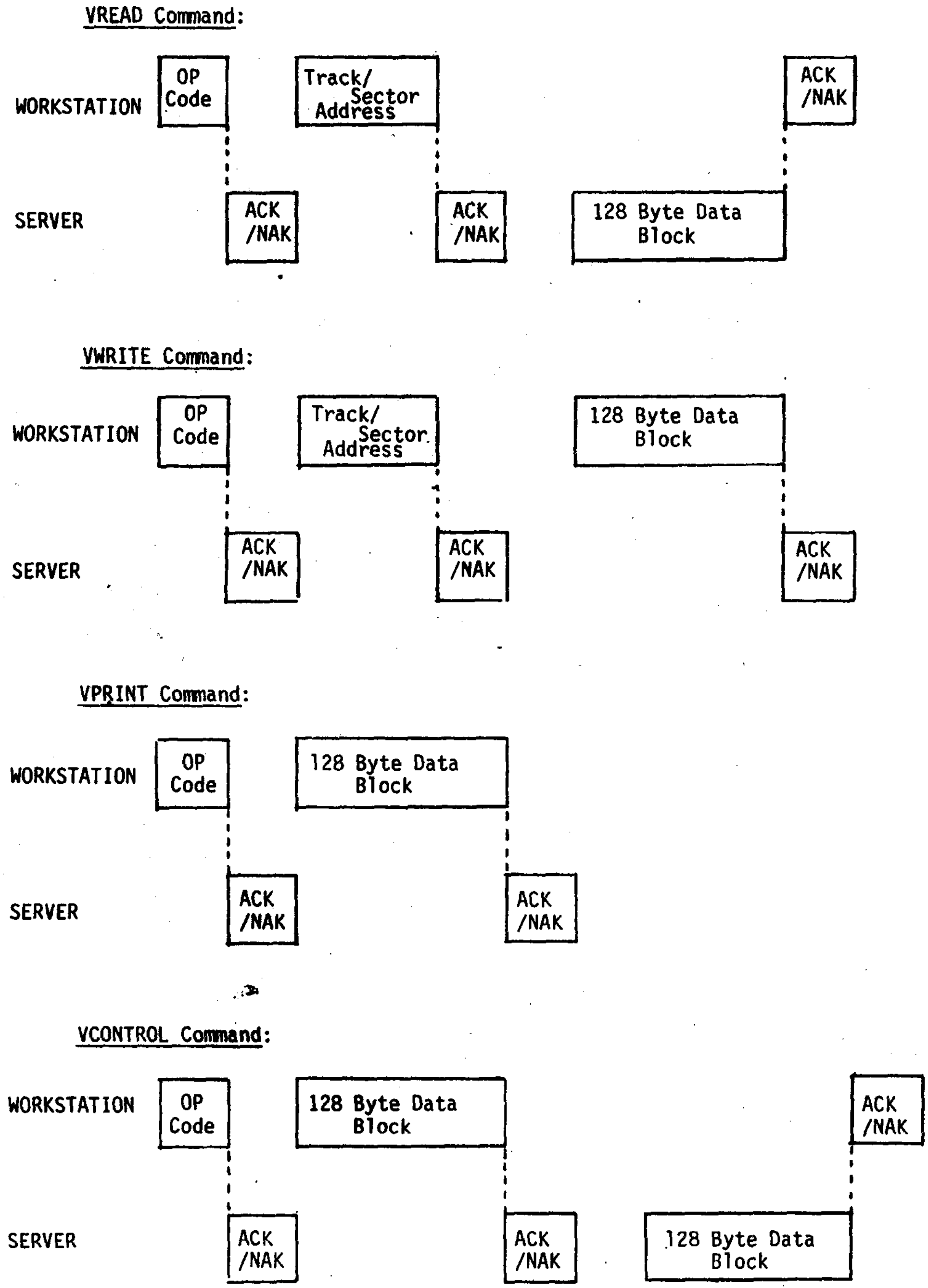

Figure 5: Communication Commands

single integer. This integer is used to access one file block of the Pascal virtual disk which is being used to simulate the CP/M virtual disk currently mounted on the user's workstation. The appropriate file block is written to the disk. The disk is marked as 'youngest' currently mounted on the Cluster Server.

VREAD - the track/sector address of the disk sector to be written is converted to a single integer. This integer is used to access one file block of the Pascal virtual disk which is being used to simulate the CP/M virtual disk currently mounted on the user's workstation. The appropriate file block is read from the disk, and the data sector addressed is sent back to the user's workstation. The disk is marked as the 'youngest' currently mounted on the Cluster Server.

It is important to note that each time a VREAD, VWRITE, or VPRINT command is executed by the Cluster Server the virtual disk to be accessed may not actually be mounted on the Cluster Server. In this case the 'oldest' disk currently mounted on the Cluster Server is replaced by the disk to be accessed.

Suggestions for Further Work

Work on the Cluster Server to date has included the implementation of the server itself in Apple II Pascal and 6502 assembly language, modification of the Osborne 1 CP/M BIOS to allow operations on virtual devices, and the implementation of a compatible CP/M BIOS for a S-100 bus machine based on the 8088 processor in 8088 assembly language. Modified BIOS routines will next be implemented for the IBM Personal Computer running CP/M-86 and the Xerox 820 running CP/M-80.

To insure the integrity of the 128 byte blocks of data which are sent back and forth between the users and the Cluster Server, each 128 byte block of data should be followed by a 16 bit checksum code, with erroneous blocks being re-transmitted. This has not been done to date, but the fact that each 128 byte block transfer is acknowledged by the receiving station with an 'ACK/NAK' character should make the addition of this feature trivial.

The Cluster Server has been tested with two clustered workstations operating simultaneously. It is felt that the Cluster Server is capable of servicing at least 6 and possibly up to 12 clustered workstations. The testing of this hypothesis is contingent on the acquisition of at least four more personal computers and the implementation of the necessary BIOS modifications.

The performance of the Cluster Server, both in terms of performance as perceived by the user at a clustered workstation and loading of the local network should be quantified. This work is proceeding. The S-100 bus machine connected to the Cluster Server at present will be used to perform a random mix of communication instructions, thus presenting an artificial load to the Cluster Server in order that the server's performance can be analyzed.

At present, CP/M virtual disks which are used by the clustered workstations are simulated using Apple Pascal virtual disks on the NFS. Nestar now has a new version of the NFS program in beta-test which supports CP/M virtual disks on the network. These disks are to be used by Apple stations which have an auxiliary Z-80 processor board installed and can run the CP/M operating system. It would be advantageous if the Cluster Server could be modified to use virtual CP/M disks rather than simulating them with Pascal disks. This would allow data files to be shared between clustered workstations and appropriately equipped Apples connected directly to the network.

Finally, the virtual disk operations currently supported by the Cluster Server would be significantly improved if the limitation that virtual disks must have a fixed size of 102 Kbytes were removed. CP/M-80 itself is capable of supporting disks of up to 8 Mbytes [4], and the next version of the Cluster Server will allow variable size virtual disks to be accessed by users.

Conclusions

The design and implementation of the Cluster Server as a communications front-end processor suggests new architectural variations which should be explored in local networks of personal computers. In the short term, it appears that new personal computers will continue to be developed and introduced faster than local network interface boards which allow them to connect onto a variety of local networks. The Cluster Server approach will allow a large variety of personal computers to connect indirectly onto local networks with a minimum of implementation effort and delay.

In the long term, it seems that there will always be low-cost personal computers built which do not have the capability to connect directly onto local networks, and the Cluster Server is well-suited to allow such computers to share expensive resources such as hard disks and high-quality printers in the local network environment. Finally, not all applications of personal computers on local networks require the full communications capability of a very high-speed local network, and, by carefully choosing the number of stations to be clustered around a communications processor and the type and quality of services to be provided to the workstations, it is hoped that local networks of personal computers can be used to economically and efficiently solve the problems of users for many years to come.

References

[1] "Special Report: Nestar Systems Incorporated Cluster/One Model A," The LOCALNetter Newsletter (ed. K.J. Thurber), September, 1981.

[2] "Special Report: Datapoint ARCNET," The LOCALNetter Newsletter (ed. K.J. Thurber), December, 1981.

[3] Heart, F.E., Kahn, R.E., Ornstein, S.M., Crowther, W.R., and Walden, D.C., "The interface message processor for the ARPA computer network," Proc. AFIPS 1974 SJCC, Vol. 36, AFIPS Press, Montvale, NJ, pp. 551-567.

[4] Cortesi, D.E., Inside CP/M, Holt, Rinehart and Winston, New York, 1982.

[5] Hogan, T., Iannamico, M., Osborne 1 User's Reference Guide, Osborne Computer Corporation, Hayward, CA, 1981.