AN ABSTRACT OF THE THESIS OF

Yaqin Zhou for the degree of Master of Science in Electrical and Computer Engineering presented on March 15, 1989.

Title: CLASSLAN: An Experimental Personal Computer Network for Classroom Education

Abstract approved: Redacted for Privacy
James H. Herzog

Interest in personal computer networks has grown rapidly in recent years. As the personal computer becomes more and more popular, implementing computer networks on those small computers has become an interesting topic. Such networks are usually inexpensive and can be easily installed and maintained.

In this thesis work, CLASSLAN, such a network system, is designed and evaluated. The system consists of IBM PC's and microprocessor controlled network interface units. The network is connected via twisted pair wires.

The emphasis of this research is in the network's higher layer development. A basic set of educational utilities for CLASSLAN have been developed. They include:

- Lecture and conversation
- Preparing and administrating examinations
- Class monitoring
- Questioning during exams
- Timer functions
- Instant help menu
- Completely menu-driven with pop up windows

Most aspects related to a network's performance are covered, including user friendliness, layered structure, error control, and error recovery.

CLASSLAN: An Experimental Personal Computer Network for Classroom Education

by

Yaqin Zhou

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Completed March 15, 1989

Commencement June, 1989

APPROVED:

Redacted for Privacy

Associate professor in charge of major

Redacted for Privacy

Head of department of Electrical & Computer Engineering

Redacted for Privacy

Dean of Graduate School

Date thesis is presented March 15, 1989

ACKNOWLEDGEMENTS

I would like to extend my sincere thanks to Dr. James H. Herzog, my major professor, for his encouragement and guidance throughout this study.

I warmly thank all of my friends and colleagues whose assistance and encouragement has been invaluable.

TABLE OF CONTENTS

LIST OF FIGURES

CLASSLAN: AN EXPERIMENTAL PERSONAL COMPUTER NETWORK FOR CLASSROOM EDUCATION

CHAPTER 1

INTRODUCTION

Computer aided instruction systems or CAI systems have become more and more popular with the development of local area computer networks. Such systems are well suited in scientific education, demonstration, and tutoring. They are also found useful in elementary and high school education and the training of mentally retarded children.

Local area computer networks (LANs) have the following characteristics that make them well suited in CAI applications:

1) the classroom environment is easily formed by connecting the student stations and the instructor station to the network;

2) local area networks usually offer both addressed and broadcast communication, both are necessary for educational applications;

3) Many LANs allow simultaneous initiations of

message transmission from users. This is similar to the competition of students that usually occur in a classroom; and

4) Low cost networks (such as those implemented by personal computers and microprocessors) can be used because there is no stringent requirement on communication speed and capacity.

Several experimental LAN projects have been carried out at the Department of Electrical and Computer Engineering at Oregon State University by the research group of Dr. James H. Herzog. They covered the master-slave, token ring and CSMA/CD protocols. These are named as the COLAN series network I thru IV, with increasing capabilities and complexity. The emphasis of these works have been on data framing, routing and storage. Most low-level network tasks could be performed with the COLAN network.

The goal of this thesis is to explore these ideas with the development of CLASSLAN, an education-oriented experimental local area network. This research was performed with COLAN IV, a low cost serial communication network. The emphasis of this thesis is on the higher network layer software development. Hence, some

imperfections in the underlying structure were not corrected. The user interface was built on top of the communication utilities of COLAN IV. Useful educational functions were developed. The CLASSLAN works quite well and most of the design goals are realized.

The system is named "CLASSLAN" to denote its specialization to educational applications and offers a minimum set of functions needed in classroom education (the CLASSLAN users, however, need not be in a common classroom). The major user functions include (for the complete menu description see chapter 2):

Instructor Functions:

- Preparing class materials
- Setting up classes
- Administrating examinations
- Lectures and Conversations
- Filing students' records
- On-line help menu
- Friendly, completely menu-driven users interface that makes the physical network transparent to the user

Student Functions (driven by the instructor):

- Student identification

- Conversation with instructor during lectures
- Examinations
- Questioning during exams

The hardware consists of personal computers and microprocessors, which make the network distributed and inexpensive. The network interface unit has been implemented using an INTEL 8051 single board computer.

The network functions of CLASSLAN consist of four hierarchical layers: the application layer, the network layer, the data-link layer and the physical layer (see Figure 3.1).

CLASSLAN's communication medium is a half-duplex serial communication bus. In each user station there is a personal computer and a network interface board where two software programs are running interactively. A machine code program (COLAN IV software) resides in the network interface unit board. It controls the communication hardware and forms the physical layer, data-link layer and part of the network layer of the network. The rest of the network functions are the tasks of this thesis. The education-oriented local area network is realized by two

high level language programs that run simultaneously on the personal computers in user stations.

User friendliness is important in educational applications. The programs are designed in such a style that persons with little experience with computers are able to use CLASSLAN the first time. The CLASSLAN main menu is always shown on the right side of the screen to remind the user about the available options of function and operations. The menus are selected by moving the highlighted window using the Up/Down arrow keys. When a menu is entered (by pressing Enter), the selected function is run or the submenus (if any) pop out to display a second level menu. Therefore the user can hardly forget where he is. The on-line help is available by pressing the F1 key. The messages related to the current operation pop onto the screen and disappear with a depression of the ESC key. When there is an operation error, such as reading a non-existant file, the error message is displayed and the operation is cancelled.

The various aspects of the work are discussed in detail in the following chapters. Chapter 2 describes the menu functions of CLASSLAN that a user would see. With these user interface functions, the actual network is

made transparent to users. Chapter 3 describes the functional structure of CLASSLAN. Chapter 4 describes the software algorithms developed to deal with major communication problems. These techniques were used to tolerate the inherent transmission errors and small buffering capacity associated with the simple, low-cost network interface unit.

The primary interest of this work is in the implementation of an educational local area network. More general descriptions about the educational applications of computer networks can be found in the various literatures. For example, Hofmeister described equipment used by an educational network in detail [7]. Hick and Hunka discussed the design strategy, development and evaluation of computer aided instruction (CAI) [6]. Heins described interesting design strategies for screen display used in CAI applications [4].

CHAPTER 2

CLASSLAN APPLICATION FUNCTIONS

CLASSLAN consists of a number of network stations at different locations connected by a simple twisted pair wire. In each station there is a personal computer and a network interface board (Fig. 2.1). There is one station dedicated to being used as the instructor's station. All other stations are used as student stations.

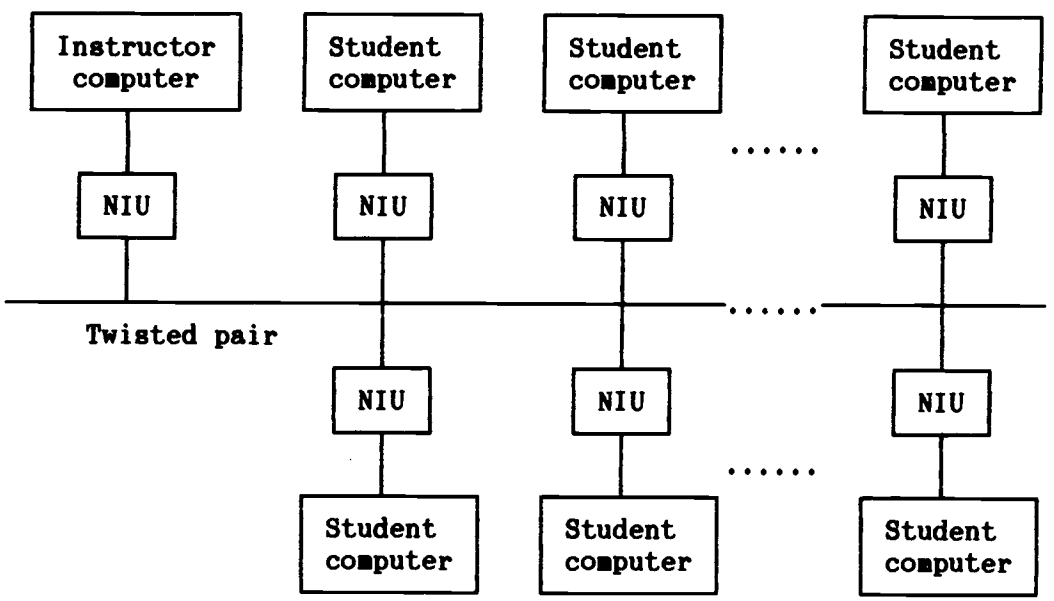


Fig. 2.1 CLASSLAN Layout

The software that runs in the personal computer of the instructor's station is CLPRO.EXE. It interacts with the PCs at student stations where another software

CLSTU.EXE is running. CLPRO acts as the master or driver. CLSTU's at student stations are slave programs. They jointly perform the network functions of CLASSLAN.

The CLASSLAN software supports a series of specialized functions that are used for educational purposes. They form the application layer, or the user interface of the network. Therefore, the main requirement of this layer is to realize educational functions and provide user friendliness.

To be "friendly" to users, the instructor program (CLPRO) is completely menu-driven with an ever-on-screen menu. Figure 2.2 illustrates the screen display after the user types "CLPRO" from DOS. The instructor needs to know the DOS file directory system. A word processor must be available in the system in order to edit CLPRO's document sheet.

The student program (CLSTU) is designed so that minimum number of commands (if any) are needed to use it. CLSTU is a "listener" to CLPRO. It executes appropriate functions automatically under the control of CLPRO. Normally, students need not to worry about what to do next, since they are always instructed by the messages on the screen.

Fig 2.2 CLPRO Main Menu

There are twenty menus in CLPRO. Some of them have their counterparts (listener or slave) in CLSTU. Figure 2.3 shows these menus. Note that some menus in CLPRO are the subentries of parent menus.

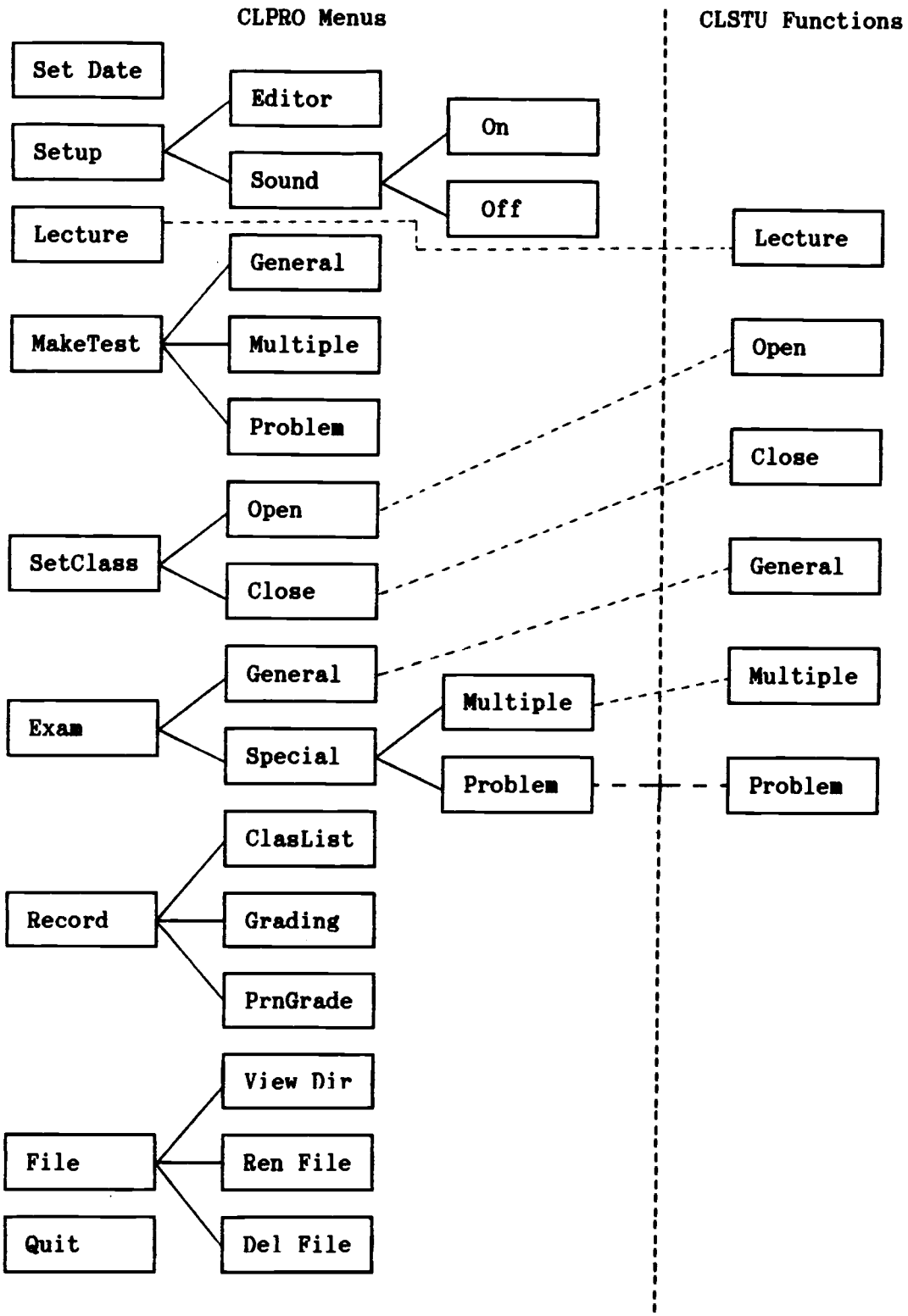


Fig. 2.3 CLASS Menus from Users Point of View

2.1 CLPRO Menus

CLASSLAN's application functions are organized in a tree-type menu directory as follows:

1) Set Date
2) Setup/Editor
3) Setup/Sound/On
4) Setup/Sound/Off
5) Lecture
6) MakeTest/General
7) MakeTest/Multiple
8) MakeTest/Problem
9) SetClass/Open
10) SetClass/Close
11) Exam/General
12) Exam/Special/Multiple
13) Exam/Special/Problem
14) Record/ClassList
15) Record/Grading
16) Record/PrnGrade
17) File/View Dir
18) File/Ren File
19) File/Del File
20) Quit

This section describes the twenty menus of CLPRO. Each subsection will explain the function of a particular menu.

SET DATE

When the shaded block is moved to window "SET DATE" with arrow keys and followed by a carriage return, this window is highlighted and entered. The help line at the bottom shows the basic options of operations. The program will automatically determine the type of the PC video adapter and set the appropriate foreground and background color or shade.

Menu SET DATE allows you to set the date and time for the system. Since date and time will appear in all exam papers, it is advisable to set the correct date and time after the program is started.

SETUP/EDITOR

In order for the user to be able to create test paper files and to edit grade sheets, CLASSLAN requires a word processor in the computer. In MAKE-TEST menu, for instance, the word processor is invoked automatically by CLASSLAN. To do this, CLASSLAN needs to know the name of

the word processor. Menu SETUP/EDITOR allows the user to specify the word processor to be used by CLASSLAN by entering the word processor's program name (the same name as used under DOS to run it). The selection is stored in the configuration file "clconfig.cl". Different computers can use different word processors.

SETUP/SOUND/ON

The return key is pressed to turn on the CLPRO sound effect. As a result, a short clicking sound will be generated to acknowledge commands or for other purposes. The option is remembered (written to the configuration file) so that the selection will remain in subsequent program runs. Figure 2.4 shows an illustration of this operation.

SETUP/SOUND/OFF

Hitting the return key will turn off the CLPRO sound switch. As a result, no sound will be produced when running the program. The action is remembered in following program runs.

LECTURE

LECTURE is designed to simulate a class room

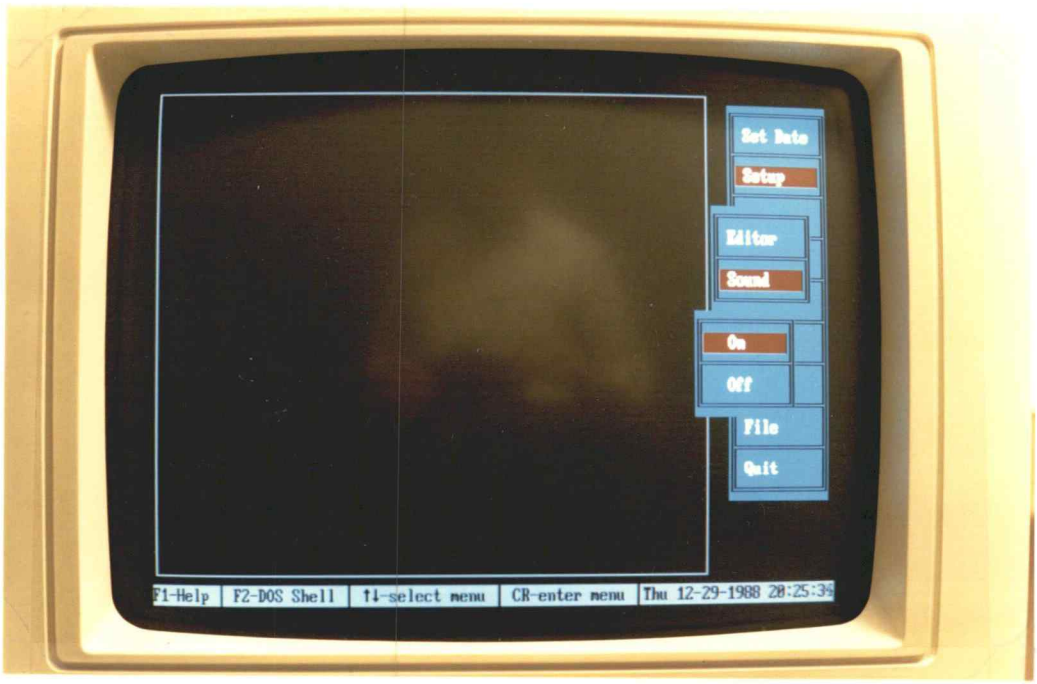
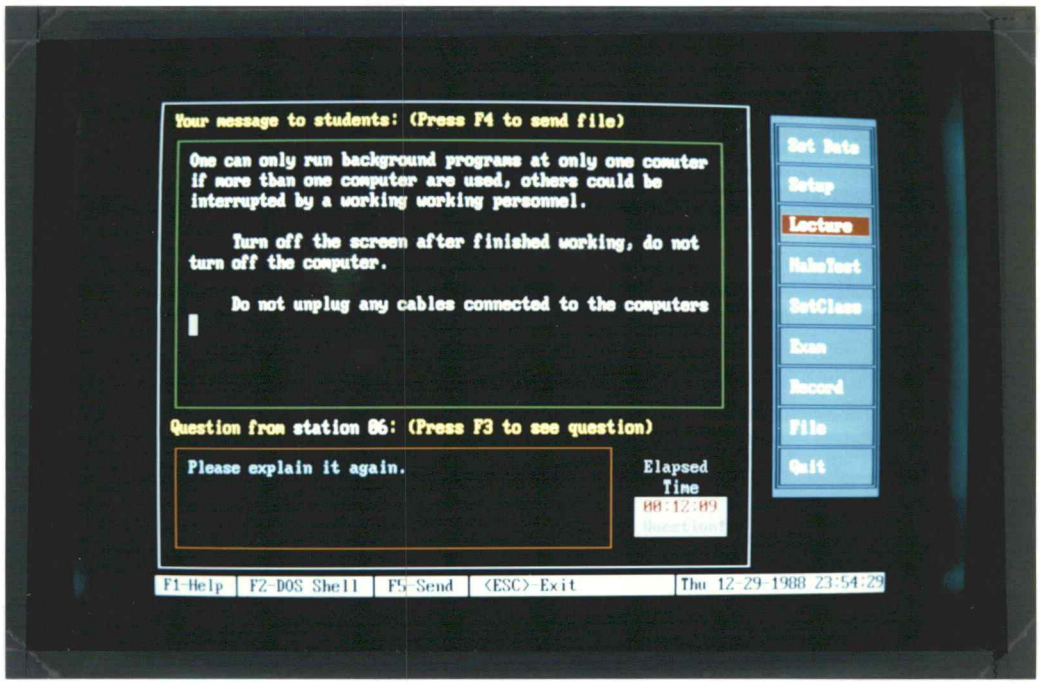


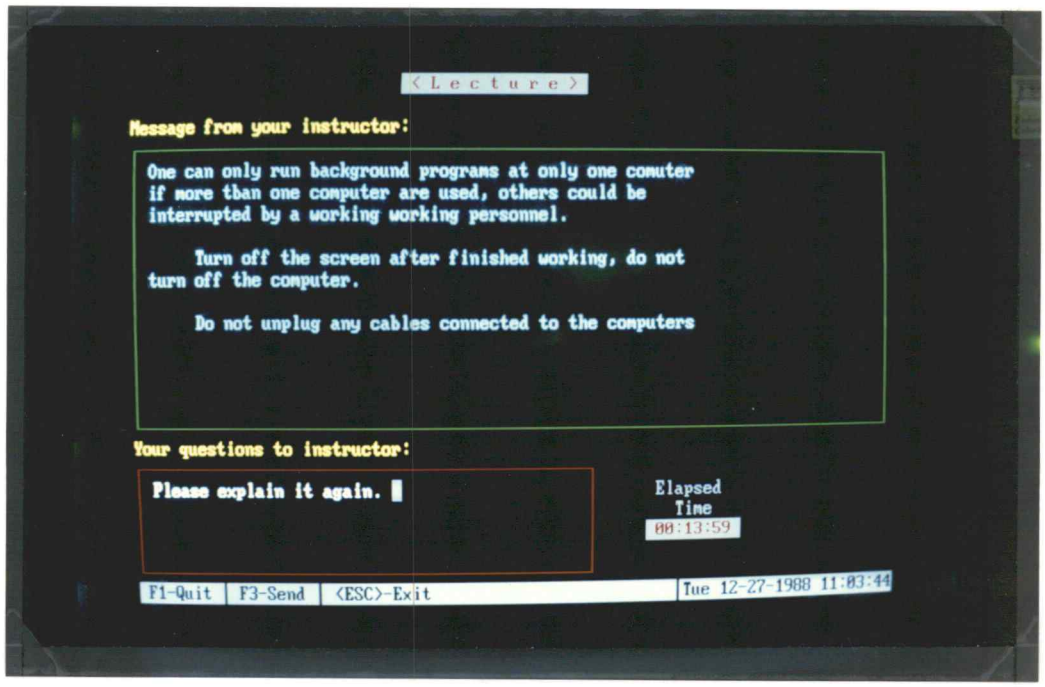
Fig. 2.4 Turning on the Sound Switch

environment where the instructor and students communicate with each other (Fig. 2.5). The instructor can send messages to students in two ways. He can send an pre-existing text file (not longer than 12K at a time, to meet COLAN IV's capacity). Alternatively, he can type the text in LECTURE's editor window (the larger rectanguler box inside the frame) and send the whole window of text. A full editor window has about 1000 characters.

The instructor can also choose the destination when sending messages. The destinations are identified by



(a)



(b)

Fig. 2.5 Lecture
(a) CLPRO Sends Message to Students;
(b) CLSTUD Receives Message from Instructor.

their station numbers. If the destination is not specified, all student stations will receive the message.

The smaller rectangular window is used to display student's questions. LECTURE is always monitoring messages from student stations. When a message from students arrives, a flashing "question!" will appear in the lower right corner of the frame. The instructor may then interrupt his lecture and answer the question.

The way LECTURE handles student's questions is similar to a real classroom lecture. When a student sends a question to the instructor, the same message is also sent to every other student on the CLASSLAN.

MAKE-TEST/GENERAL

This menu is used to write the test paper of a "general" exam. The word processor is invoked for the user to write the test paper. The test paper already has a written header prepared by the CLASSLAN program. The user then enters the text and saves the file when done. The file is saved on the disk and is named <test name>-P.GEN under the directory "/<course title>". The letter P denotes that the file is a Problem set for an exam. As

will be seen shortly, the file name of a student's Answer sheet contains an "A" and that of a Grading sheet contains a "G". For example, the user may see the screen shown in Figure 2.6 when he enters this menu to make an exam paper called "Final". It is assumed that an "ASCII" word processor such as PCWRITE is used.

Fig. 2.6 Writing a General Test Paper

MAKE-TEST/MULTIPLE

As the menu name indicates, this menu is used to make specially formatted multiple-choice test files. When making such test papers, the instructor just types in the required items prompted by the computer. All the information will be sorted and packed into a test file called <test name>-P.MUL. The test file is saved under directory "/<course title>". Figure 2.7 illustrates this operation.

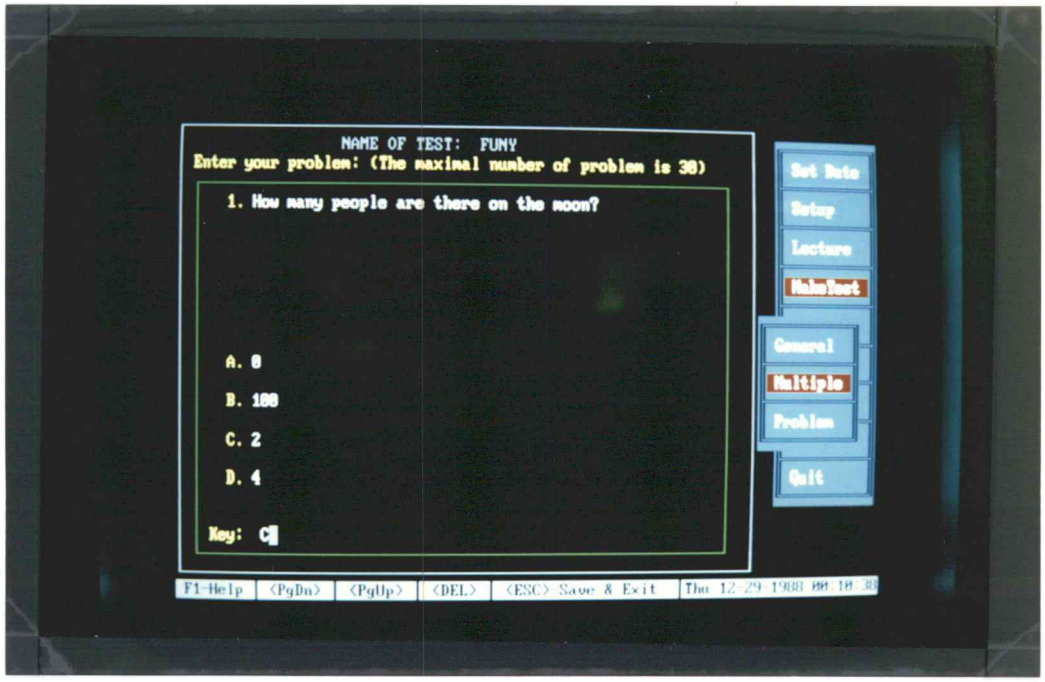


Fig. 2.7 Writing a Multiple Choice Test

MAKE-TEST/PROBLEM

As the menu name indicates, this menu is used to make a specially formatted short answer test file. When making such test papers, the instructor just types in the required items prompted by the computer. All the information will be sorted and packed into a test file called <test name>-P.PRO. The test file is saved under directory "/<course title>".

SET-CLASS/OPEN

The purpose of this menu is to perform a roll call of a class. It asks students to report their names and ID's. These are associated with the check-in time, the station numbers and are saved in the computer for future reference.

The OPEN menu can be used at the beginning of a class session. As the students come to the class, they will be asked by CLSTU to report to the instructor. The instructor can monitor the student's check-in and close the check-in at any time.

If a class session is expected to contain exam(s), then OPEN must be run first to collect the information

about the students. The information include names, ID's, check-in times and station numbers, and will be used in the automated grading later on.

It is not mandatory to OPEN the class if there will be no exams given in the session. Figure 2.8 illustrates the OPEN operation.

SET-CLASS/CLOSE

Running CLOSE at the end of a class will clearly inform the students that the session is over, but it is

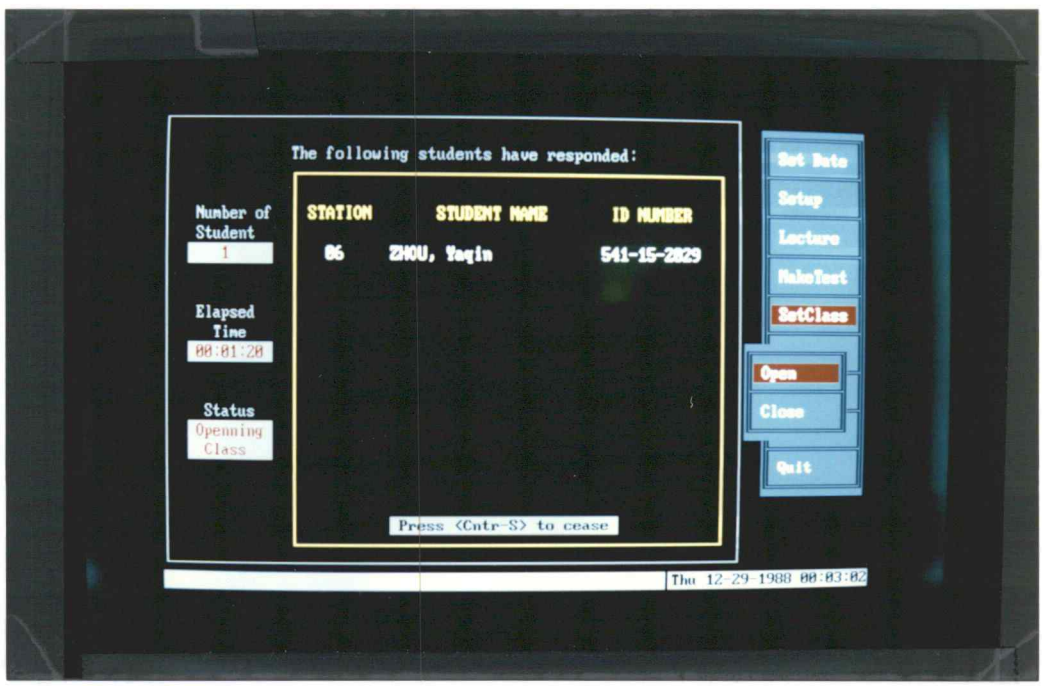


Fig. 2.8 Opening Class

not mandatory. If you open the class before an exam, it is suggested that you close the class after the exam so that the next instrutor can open class again.

EXAM/GENERAL

The program sends examination papers (files) to students and then monitors the class. During the test, students may ask questions and the instructor may give responses. Students coming late will also report to the instructor. It is up to the instructor whether or not to give them the test. At the end of an exam, CLPRO will receive the student's answers sent by CLSTU. The answers will be saved in a file named <test name>-A.GEN under the proper class directory.

EXAM/SPECIAL/MULTIPLE

This program sends examination papers (files) to students and then monitors the class. During a test, students may ask questions and the instructor may give answers. Students coming late will also report to the instructor. It is up to the instructor whether or not to gives them the test. After the exam, each student's answers will be collected and saved.

The answer keys are sent to the students together with the test paper so that students will know their scores immediately after they have turned in their test papers.

EXAM/SPECIAL/PROBLEM

The program sends examination papers (files) to students and then monitors the class. During a test, students may ask questions and the instructor may give responses. Students coming late will also report to the instructor. It is up to the instructor whether or not to give them the test. After the exam, each student's answers will be collected and saved.

RECORD/CLASSLIST

This menu is used to enter the class list which contains the names and i.d. numbers of the students. A sample format is given on the screen for the user to follow when entering information. The finished list is saved in a file <course title>.LST under directory "\<course title>".

RECORD/GRADING

Each student's answer sheets are graded using this

menu. At the end of grading, the computer will make a grade sheet. In the grade sheet all students in the class list are listed. Their grades are placed next to their names. The class average is also calculated and shown. The grade list is saved in a file and can be printed right away. Figure 2.9 shows a grade sheet generated by computer.

```
GRADE LIST
COURSE TITLE: EE570
TEST NAME: FINAL
-------------------------------------------------
NAME                      ID NUMBER       SCORE
-------------------------------------------------
ZHOU, YAQIN               541-15-5765     90
KIM, LAO                  123-23-1234     77
DAWN, DON                 345-11-2029
JOHNSON, MIKE             145-33-3039     95
SHEN, BEN                 145-44-4049
HUA, YU                   544-11-1339
MCDONALD, BOB *           563-77-4767     100
JACK, DAI *               644-75-5874     90

*: not in the class list
Total number of student taking test = 5
Average Score = 90.4
```

Fig. 2.9 Grade Sheet

RECORD/PRINT-GRADE

This menu prints the grade list which has been prepared by GRADING. Of course, you should check if the printer is ready before start printing.

FILE/VIEW DIR

As the menu name indicates, this menu is used to view directories of the PC's files (Fig. 2.10).

Fig. 2.10 View Directory

FILE/REN FILE

As the menu name indicates, this menu is used to RENname the file.

FILE/DEL FILE

As the menu name indicates, this menu is used to DELete the file.

QUIT

Hitting the return key will terminate the CLPRO and return the user to DOS.

2.2 CLSTU Menus

This section describes the six menus of CLSTU. Menus in the student program CLSTU differ from menus in the CLPRO in that they are not operator-driven. In other words, the student program CLSTU is controlled by the instructor program CLPRO via the network. CLPRO directs CLSTU to enter and exit any of its 6 menus. CLSTU stays in the "idle" mode until it receives a "go somewhere" command from CLPRO. Figure 2.11 shows the CLSTU in idle mode.

LECTURE

The program enters menu LECTURE when the instructor program sends a "go Lecture" command. The function is designed to simulate a class room environment where

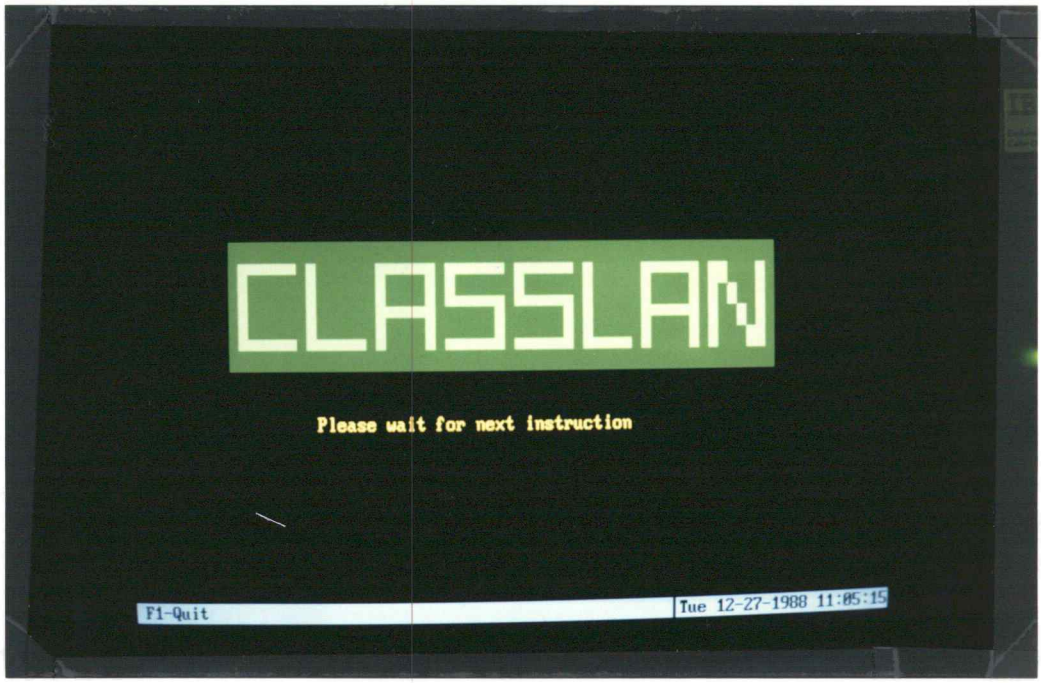


Fig. 2.11 Idle Mode of CLSTU

instructor and students communicate by conversation. Students read the instructor's messages from the screen. They can ask questions at any time during lecture. Figure 2.5 illustrats the classroom conversation during a CLASSLAN lecture.

OPEN CLASS

When CLSTU receives an "open class" command, it enters the OPEN CLASS menu. In this menu, students will be asked to report their names and ID numbers. CLSTU then

sends this information and the check-in time to the instructor's computer where CLPRO is running.

CLOSE CLASS

When CLSTU receives a "close class" command, students will receive a "goodbye" message from the instructor. The program will then return to the monitor loop (idle mode). The screen will show a big "CLASS CLOSED" message.

GENERAL

When the instructor administrates a "general" test, the CLPRO program on the instructor side sends a "general test" command to each student station. The student side then enters menu GENERAL. Students receive the message and exam "paper" from the instructor. The word processor is then automatically invoked by CLSTU to allow the students to answer the test paper. They can press the F2 key to ask questions and the F3 key to finish and turn in the test. Each student's test paper will be sent to the instructor when they press F3.

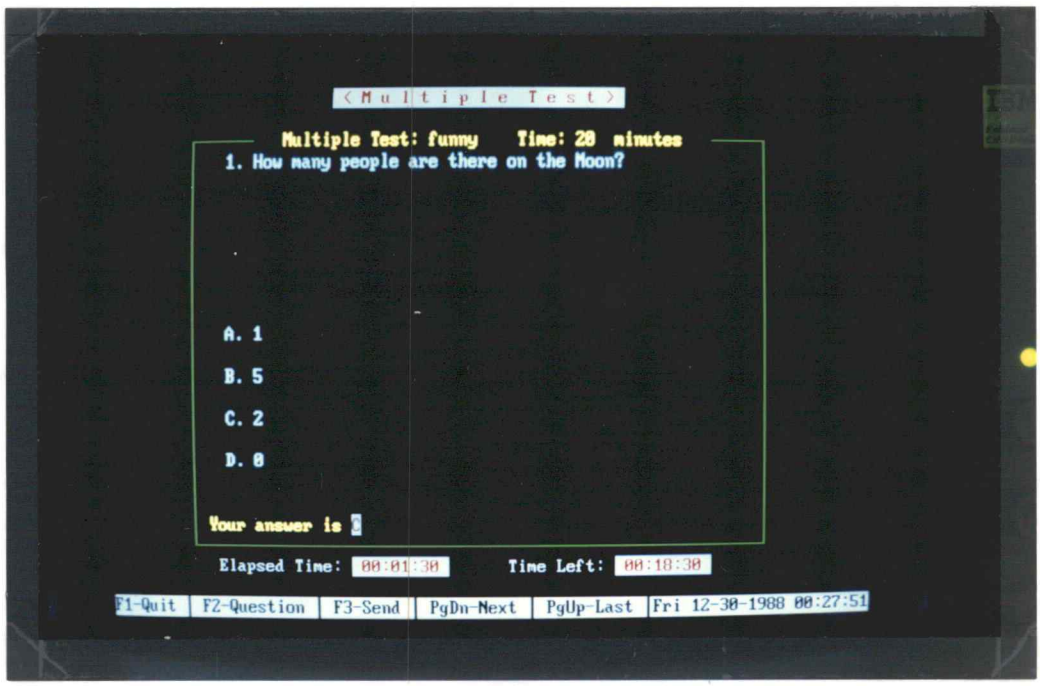
MULTIPLE

When the instructor administrates a "multiple"

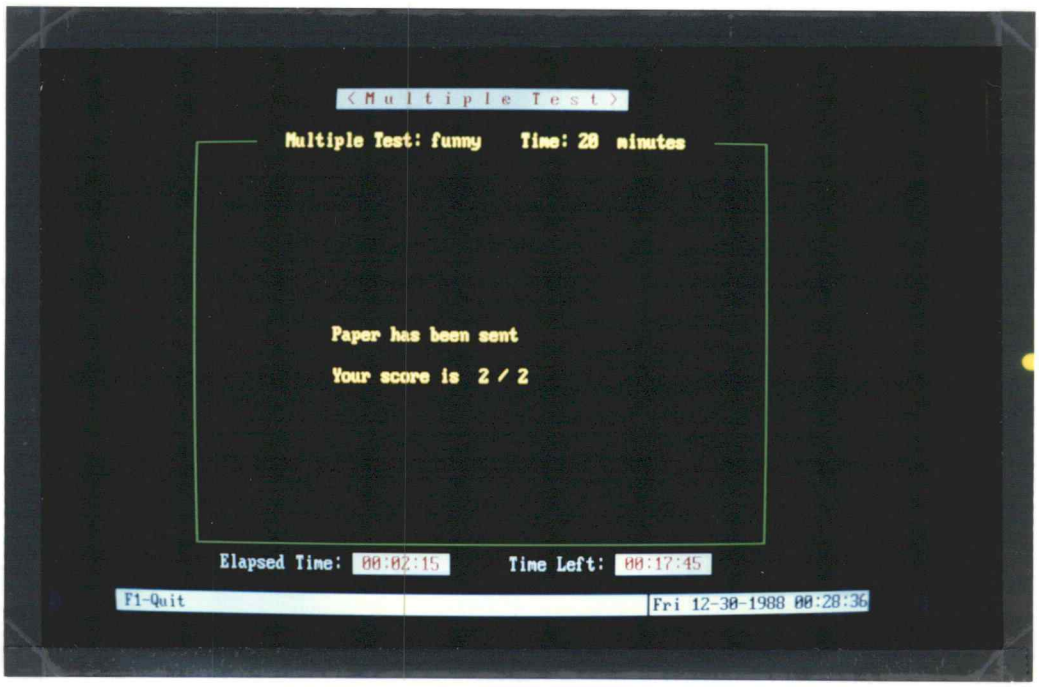
test, the CLPRO program on the instructor side sends a "multiple choice test" command to each student station. The student side then enters menu MULTIPLE. Students will see the test paper on screen. The PgDn and PgUp keys are used to select a problem. As in GENERAL test, they can press the F2 key to ask questions about the test and press the F3 key to finish the test. Student's test papers will be sent to the instructor. Each student will know his/her score immediately, since the answer key has been sent to student's computer together with the test problems. Figure 2.12 shows how a student takes the multiple choice test.

PROBLEM

Students use the PgDn and PgUp keys to select problems and enter their answers. As in MULTIPLE test, They also can press the F2 key to ask questions about the test and press the F3 key to finish the test. Student's test papers will be sent to the instructor immediately when F3 is pressed.



(a)



(b)

Fig. 2.12 Multiple Choice Test
(a) Answering Test
(b) Turning in Test

CHAPTER 3

FUNCTIONAL STRUCTURE OF CLASSLAN

The network functions are categorized into several classes. Functions interacting with users are considered as "top layer" functions. Functions that control the actual electrical signal transfers are thought as "bottom layer" functions. There are other classes of function between the top and bottom layers, which coordinate the network activities. CLPRO and CLSTU differ at the top layer.

3.1 CLASSLAN Network Layers

The functional structure of CLASSLAN is a reduced form of a standard network recommended by ISO/OSI (International Standard Organization / Open System Interconnection). The network functions are divided into several "layers", each with clearly defined input and output interface conditions. The layers of CLASSLAN are shown by Figure 3.1.

The idea of "layers" frees a network designer from having to know details of all the network activities. Instead, he/she could first define the interfacing conditions between layers and then concentrate on one

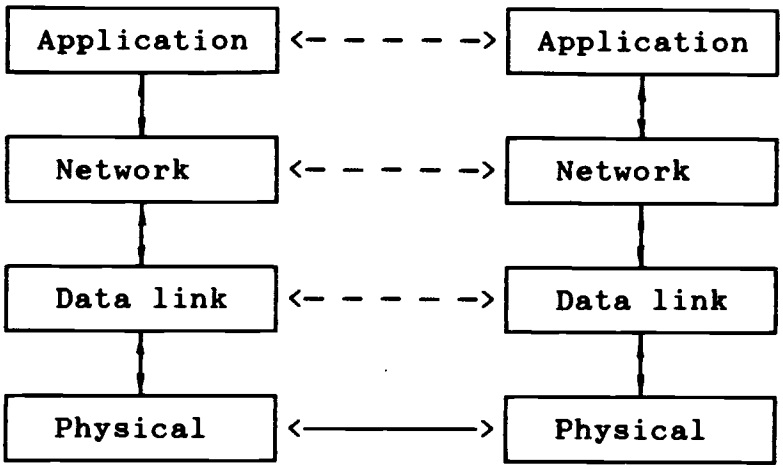


Fig. 3.1 Function Layers of CLASSLAN

layer at a time, much like "modular" design. Another distinct advantage of the layers is that once the layer function designs are completed, they can be used by different applications. The modifications are done only in the top layer (application layer). The four layers of the CLASSLAN network functions are realized by the software and hardware in two devices: a personal computer (PC) and a microprocessor network interface unit (NIU). The division of work is shown by Figure 3.2.

The PC and the NIU communicate through an asynchronous serial communication link at 9600 Baud. Network functions in the network interface unit are programmed in assembly language of the Intel 8051 microprocessor. The NIUs were developed by the research

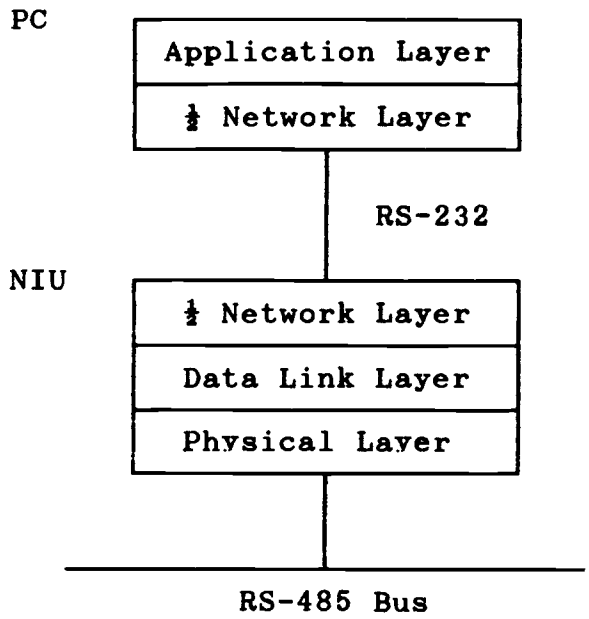


Fig. 3.2 Division of Work in CLASSLAN

group of Dr. James H. Herzog at the Department of Electrical and Computer Engineering at Oregon State University. The NIUs are connected by a common bus. The signal is transmitted in a differential voltage mode with the data form of RS-485 to achieve better noise rejection at longer distances. The CSMA (Carrier Sense Multiple Access) protocol is used in the data transmission routines. In such algorithm, all NIUs listen to the shared serial bus, transmit data when the bus is free and back off if there is collision (simultaneous transmission). The basis of the NIU software is the TaskMaster

operating system [5] with which the network application COLAN I thru IV was developed. The details of these previous works can be found in [2], [9], [14], [15].

Network functions in the personal computer are programmed in Turbo Basic, a high level language for the IBM PC. It communicates with both the NIU and the user, recovers from communication errors and realizes the special functions of an educational network.

3.2 Software Structure

The software of CLASSLAN consists of three independent programs: the 8051 assembly language program for the network interface unit board, the high-level programs CLPRO for the instructor's personal computer, and the high level programs CLSTU for the student's personal computer. The latter two are the emphasis of this thesis; their program structures are described below.

3.2.1 Program Structure of CLPRO

The program running in the instructor's PC is named CLPRO. The program is written in Borland's Turbo Basic for IBM PC's and compiled to a machine code version CLPRO.EXE. There are 9 major modules in the program as

show in Figure 3.3. The relationship between these modules is shown in Figure 3.4.

Module name	Type	Position
Menu Driver	Monitor loop	Application Layer
CLASSLAN Menus	Subroutines	
Arrow Key Interrupt Handler	ISR *	
Help Key Interrupt Handler	ISR *	
Timer Interrupt Handler	ISR *	
Communication Routines	Subroutines	Network Layer
COM1 Interrupt Handler	ISR *	
Error Interrupt Handler	ISR *	Shared
Utility Routines	Subroutines	

* ISR = Interrupt Service Routine

Fig. 3.3 CLPRO Program Modules

Module MENU DRIVER is an infinite loop which monitors the position of the highlighted menu window entry. When a carriage return is entered it branches to the appropriate CLASSLAN menu to start a menu task. When the task is completed, control is passed back to the monitor loop.

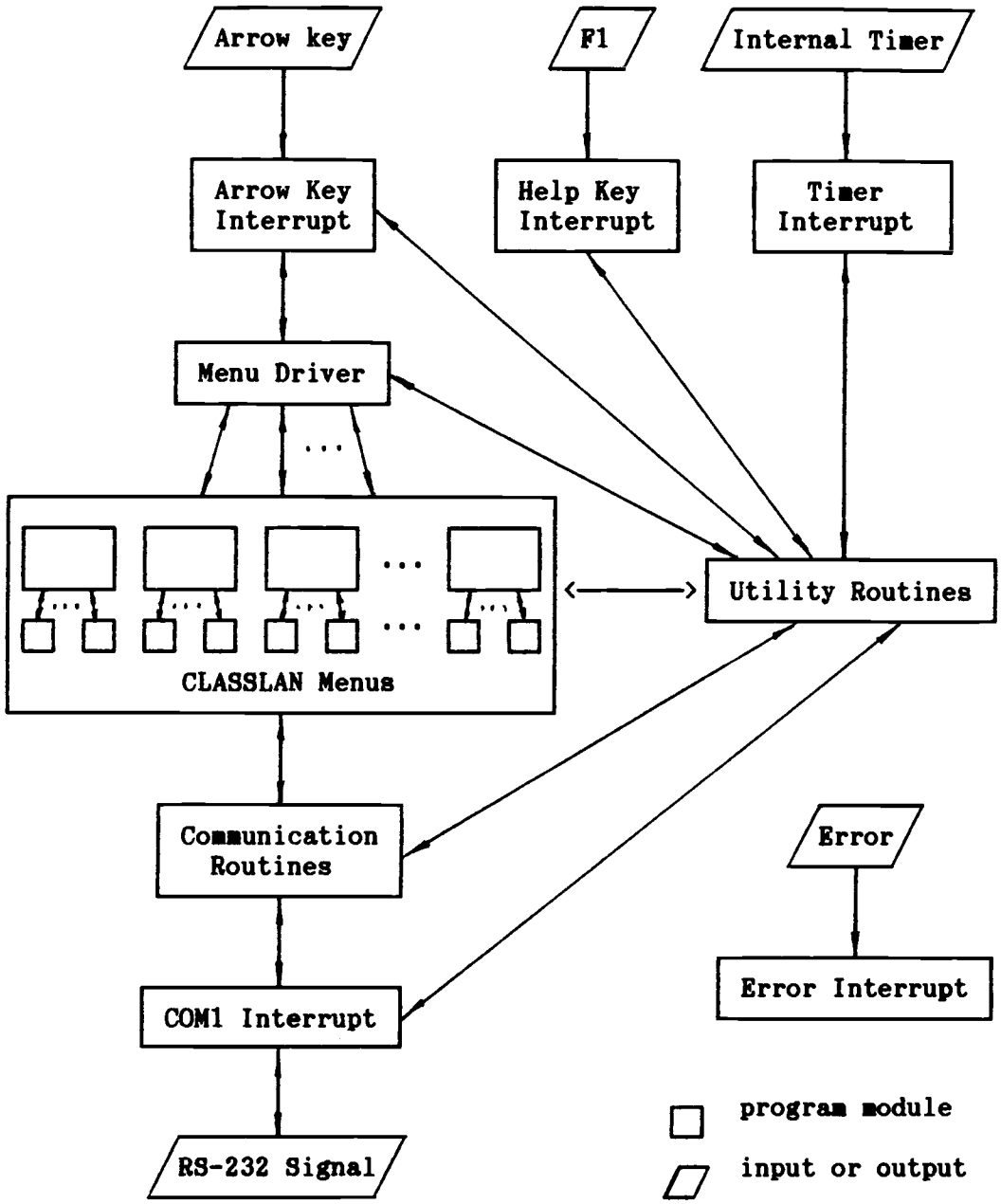


Fig. 3.4 Program Structure of CLPRO

Module CLASSLAN MENUS is a set of subroutines corresponding to the twenty CLASSLAN application menus. These are the functions of the network viewed by a user. Details are covered in Chapter 2 where the functions are defined and their usage described. The names of these menus are listed in the beginning of Chapter 2.

Module ARROW KEY INTERRUPT HANDLER is an interrupt service routine that is initiated when an up or down arrow key is depressed during the Menu Driver looping. It sets the current menu window entry and highlights that entry. The interrupt is disabled when any menu is entered and enabled when a menu finishes running (so that the menu driver loops again).

Module HELP KEY INTERRUPT HANDLER is initiated by depressing the F1 key. It detects the current program status (for example which menu is currently running) and then reads an appropriate record for the file CLHELP.CLH to display the help message on the screen. The previous screen display is saved in an array. When the <ESC> key is pressed, the previous screen is restored and the program returns to the interrupted process. This function is shown by Figure 3.5.

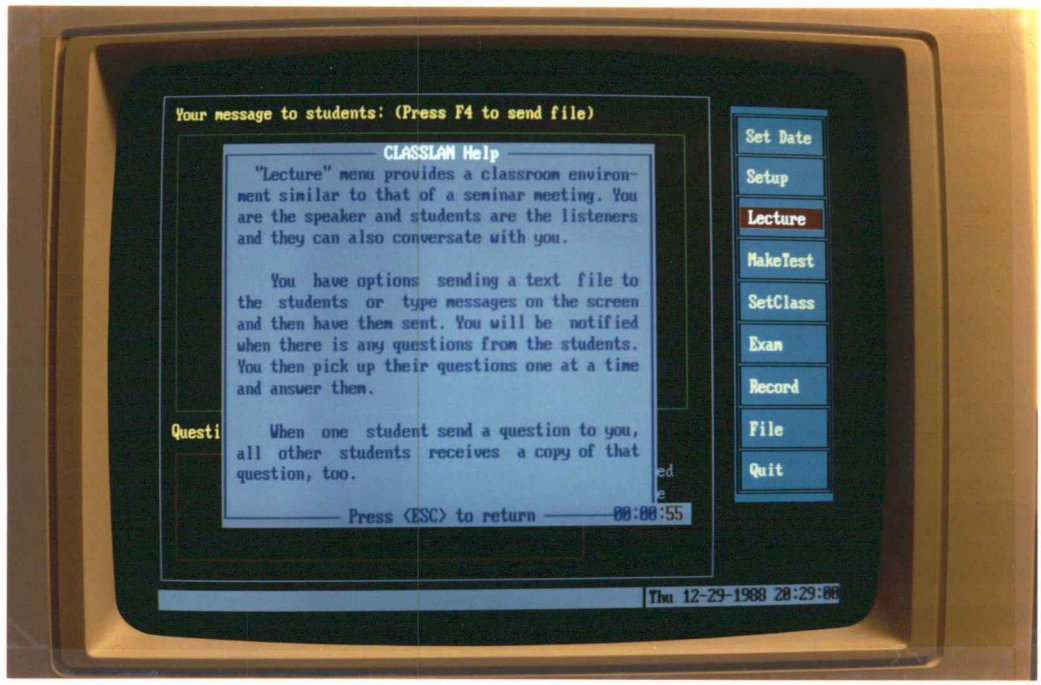


Fig. 3.5 Help Function

Module TIMER INTERRUPT HANDLER is executed at one second intervals. It updates the time variables and display.

Module COMMUNICATION ROUTINES is a set of subroutines for data transmission preperation. In the sending routine, headers and trailers are added to the messages passed from the calling process. The message frame is then sent to the serial port to which the NIU board is connected. The acknowledgement for the NIU is

analysed and proper actions are taken to either end transmission or re-transmit because of transmission errors. In the receiving routine, the message in a (incoming) communication buffer is loaded to a buffer array, which is then passed to the calling process. If the trailer is missing, the program will wait until it comes or a time out occurs. A flag is sent to indicate the error to the calling process.

Module COM1 INTERRUPT HANDLER is called automatically when a byte is received by the serial port COM1. The routine reads the byte and places it into the buffer array. The array pointer is incremented.

Module ERROR INTERRUPT HANDLER is called when an error occurs, such as a non-existant file, disk not ready, printer out of paper, etc. It contains a look-up table indexed by TURBO BASIC's error codes. When error occurs, the nature of error will be displayed and a procedure to resume from the error will be shown. Figure 3.6 shows such an example.

Module UTILITY ROUTINE contains many small subroutines for string processing, pattern display, sound generation, etc.

Fig. 3.6 Error Recovery

3.2.2 Program Structure of CLSTU

The program running in the students' PC is named CLSTU. The program is written in IBM PC's Turbo Basic and compiled to a machine code version CLSTU.EXE. There are 7 major modules in the program (Fig. 3.7). The relationship between these modules is shown in Figure 3.8.

Module TASK DRIVER is an infinite loop that monitors the command packet which comes from the instructor. When receiving a command, it branches to the appropriate task.

Module name	Type	Position
Task Driver	Monitor Loop	Application Layer
CLSTU Tasks	Subroutines	
Question Handler	ISR *	
Timer Interrupt Handler	ISR *	
Communication Routines	Subroutines	Network Layer
COM1 Interrupt Handler	ISR *	
Utility Routines	Subroutines	Shared

* ISR = Interrupt Service Routine

Fig. 3.7 CLSTU Program Modules

When the task is done, control is passed back to the monitor loop.

Module CLSTU TASKS consists of six subroutines corresponding to the six menus in the CLPRO. Each of the routines interacts with their counterparts in CLPRO. They are given the same names as in CLPRO:

1) Lecture
2) Open
3) Close
4) General
5) Multiple
6) Problem

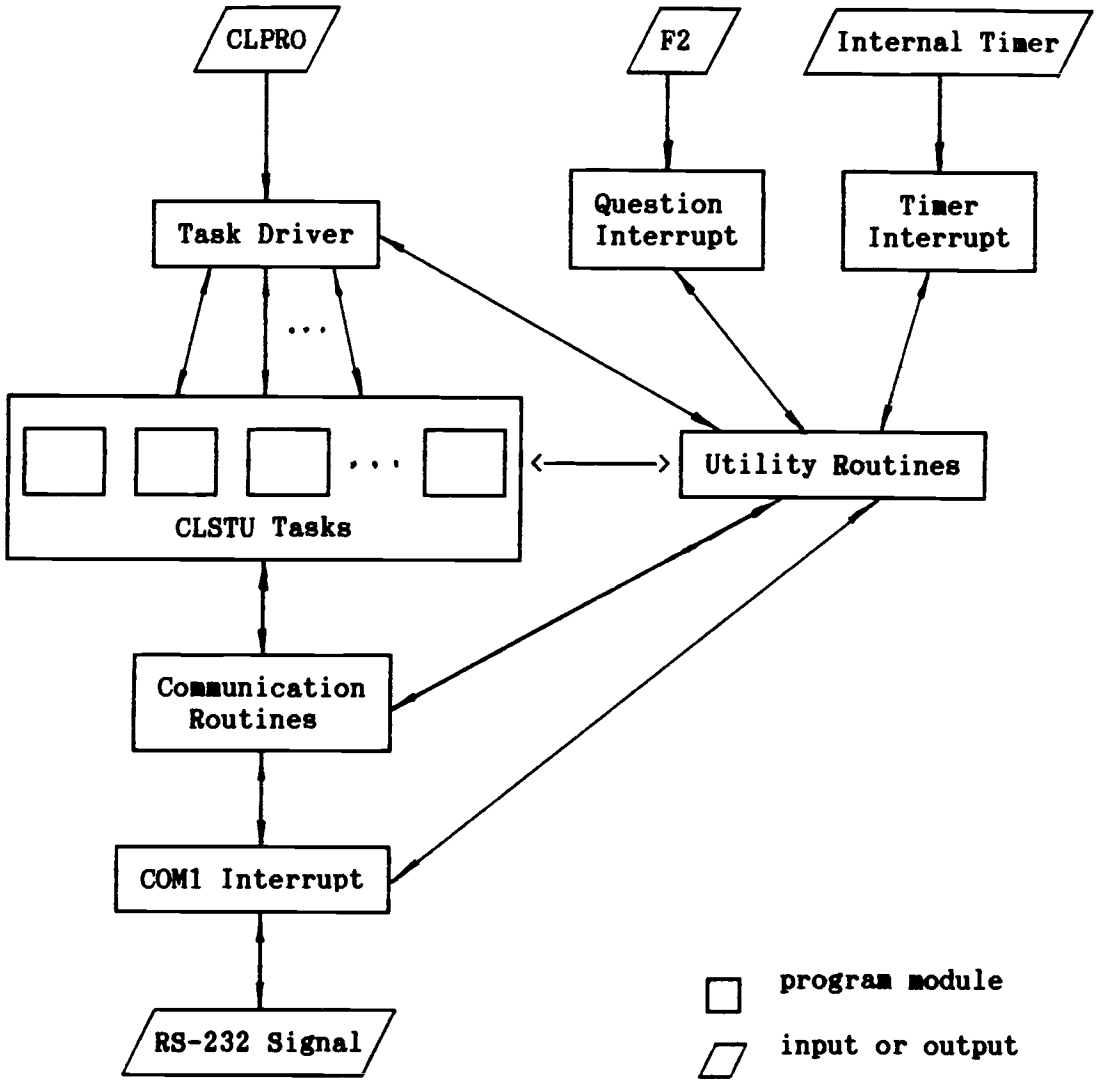


Fig. 3.8 Program Structure of CLSTU

Module QUESTION HANDLER is called when a student presses F2 during a class session. It allows the student to send several lines of message (a question) to the instructor during lectures and exams.

Module TIMER INTERRUPT HANDLER is executed once per second. It updates the time variables and display.

Module COMMUNICATION ROUTINES is a set of subroutines for data transmission preperation. In the sending routine, headers and trailers are added to the messages passed from the calling process. The message frame is then sent to the serial port to which the NIU board is connected. The acknowledgements for the NIU are analysed and proper actions are taken to either end transmission or re-transmit due to transmission errors. In the receiving routine, the message in a (incoming) communication buffer is loaded to a buffer array which is then passed to the calling process. If the trailer is missing, the program will wait until it comes or there is time-out. A flag is sent to indicate the error to the calling process.

Module COM1 INTERRUPT HANDLER is called automatically when a byte is received by the serial port

COM1. The routine reads the byte and places it into the buffer array. The array pointer is incremented.

Module UTILITY ROUTINES contains many small subroutines for string processing, pattern display, etc.

CHAPTER 4

IMPROVING COMMUNICATION RELIABILITY

One of the major technical problems needed to be solved while developing CLASSLAN was to improve the reliability of the communication network. Three typical phenomena have been observed: 1) Duplicate packets (2 or 3 similar messages) from the NIU; 2) A burst of random characters in a message; 3) Missing acknowledgements. The first and second are the most common transmission errors of the NIU. They occur roughly once during every twenty packet transmissions. The third one is quite rare.

Three error control mechanisms are developed accordingly, to solve or minimize the problem caused by the above three types of transmission errors. These are in CLASSLAN's communication subroutines and the algorithms are presented as follows.

4.1 Duplicate Message Packets

It has been observed that the NIU some times sends duplicate message packets (up to 3) to the PC. It has also been observed that the last packet is the least likely to contain errors. The algorithm to reject duplicate packets is illustrated by the chart in Figure

4.1. The key is to number the outgoing packets on the transmitting side, and to keep track of the received packet numbers. If a newly received packet has the same packet number as a previous packet, the previous one will be replaced by the new packet.

4.2 Bursts of Random Characters

One method to handle this problem is to check the length of a received packet. This is done to the CLASSLAN's Fixed-Length Packets only (see appendix for Packet Definition). Fixed-length packets are usually command packets and message packets. If the length is wrong, then the program waits for next packet to arrive, since it is very likely to see duplicate packets when there is error in the first packet. This process is repeated until a valid packet is received or three invalid packets have been received. In the latter case the incoming command packet is ignored. This is shown by Figure 4.2.

No length checking is done to CLASSLAN's long data packets such as packets used for file transmission.

4.3 Missing Acknowledgements

When the PC sends a packet to the NIU, the NIU is required to acknowledge the message by sending back to PC a short character string. The PC then begins to send the message to the destination computer(s). When the acknowledgement is missing, the NIU is probably overwhelmed by messages. Noise could also destroy the acknowledgement. CLASSLAN uses a time-out method to recover from this error, as illustrated by Figure 4.3.

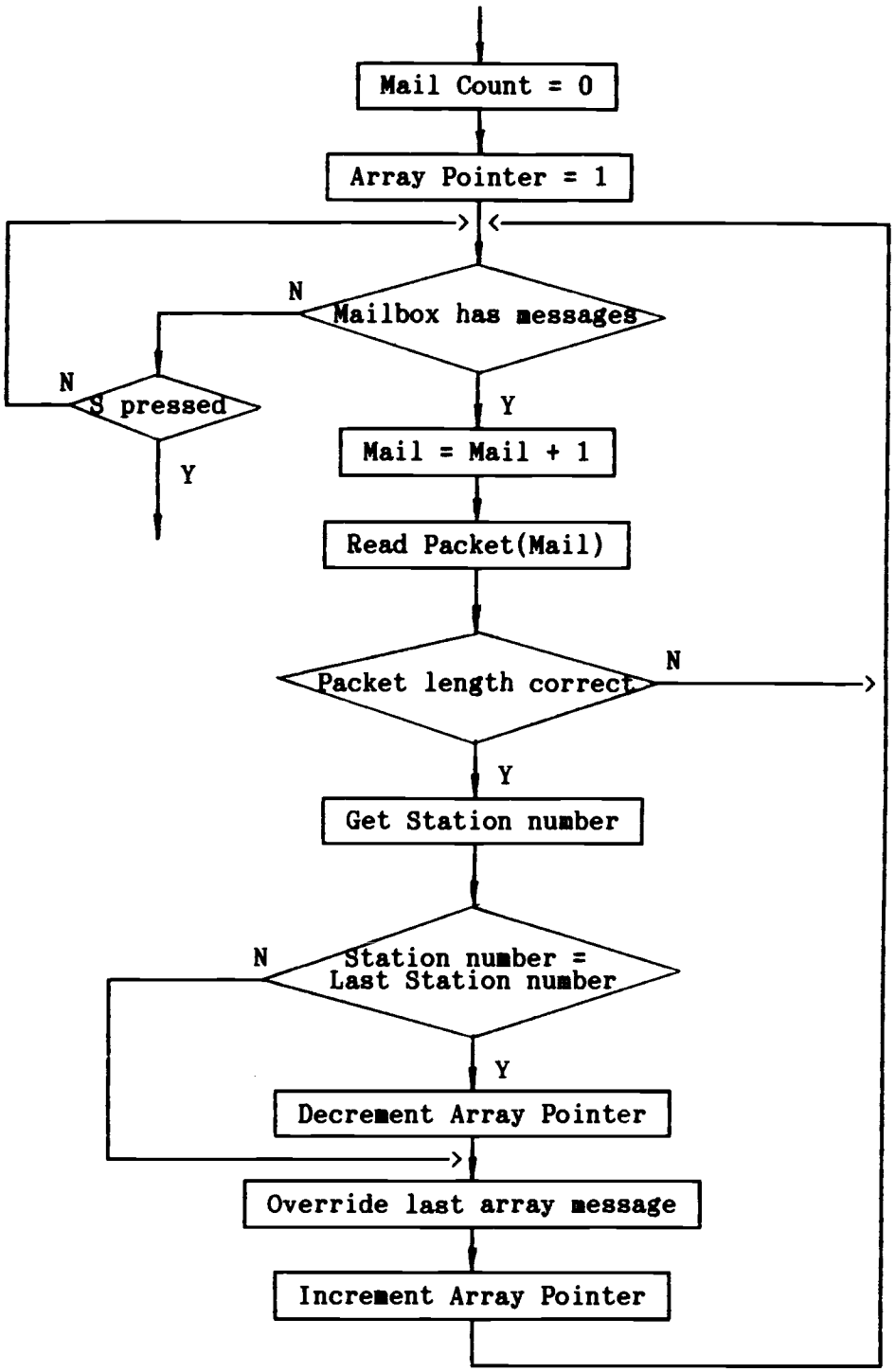


Fig. 4.1 Algorithm for Duplicated Message Rejection
(Used in Menu Open Class)

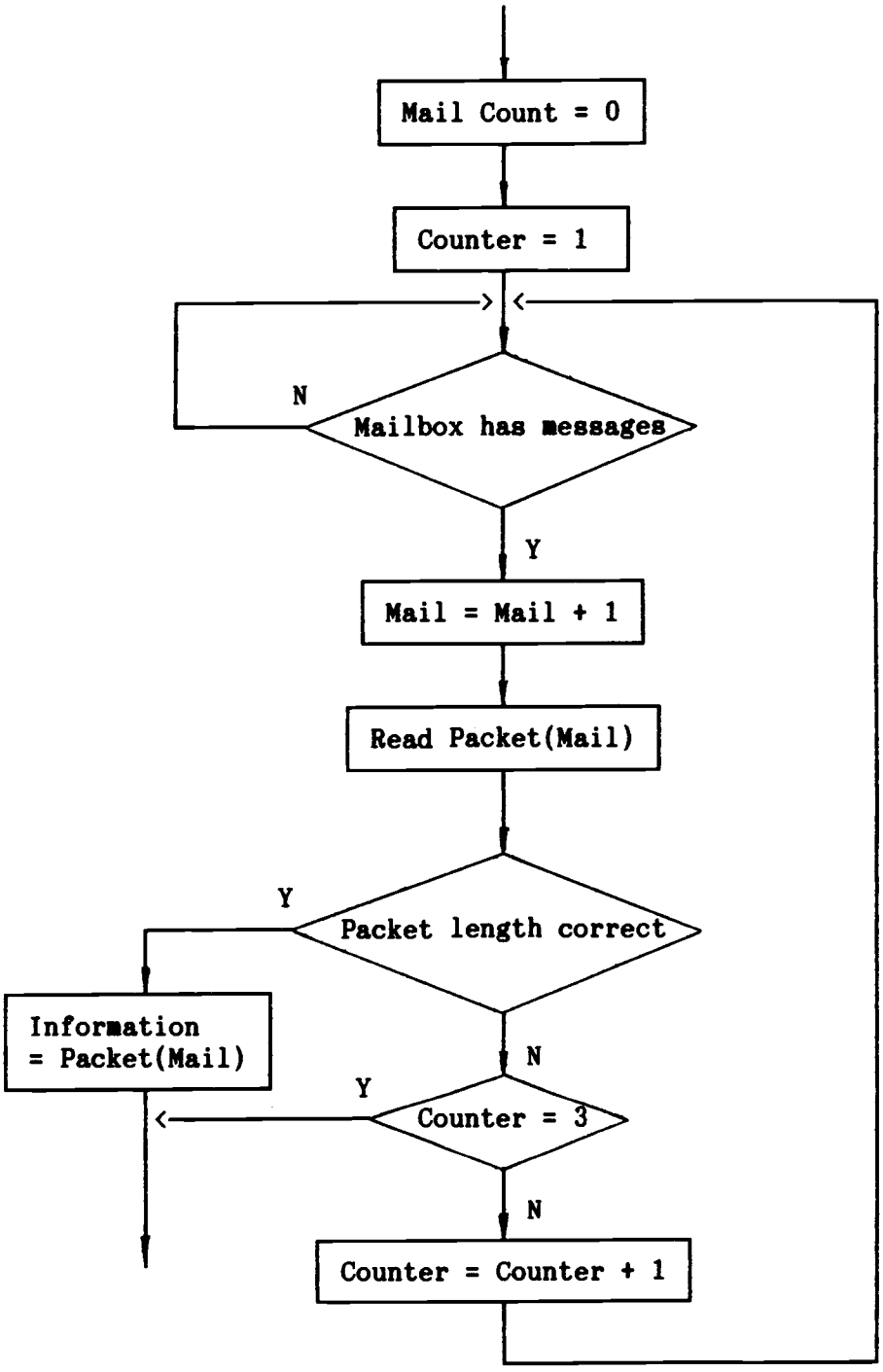


Fig. 4.2 Algorithm for Error Checking

(Used in CLSTU's Task Driver)

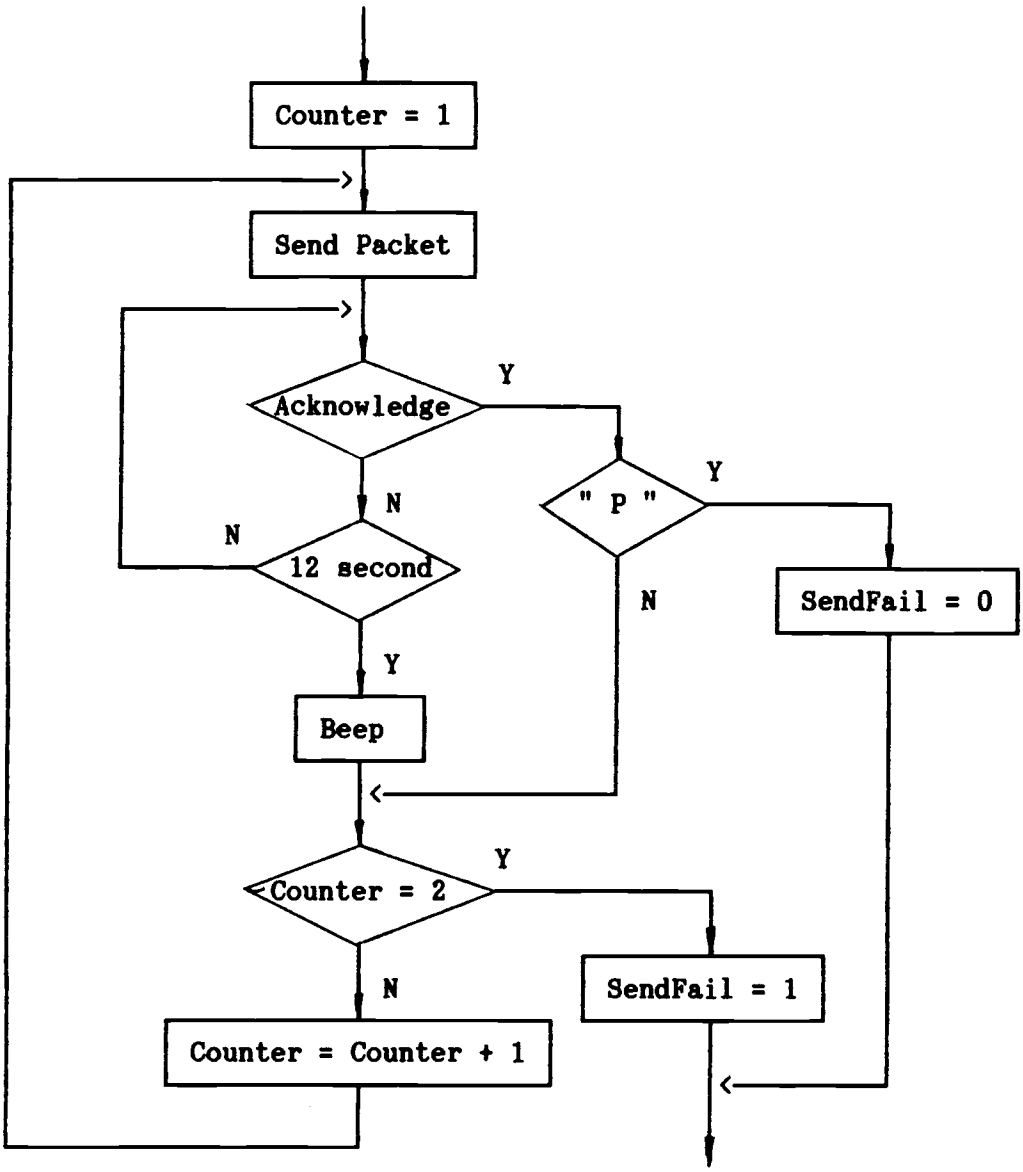


Fig. 4.3 Algorithm Used to Deal With

Missing Acknowledgement from NIU

CHAPTER 5

CONCLUSION

A simple yet complete educational computer network has been developed using microprocessors and personal computers. It realizes the following three functions:

1) It accomplishes user friendliness by having a graphic menu driven user interface with an online help menu. It also makes the physical network transparent to users.

2) It provides a basic set of functions required for a class room education environment.

3) It overcomes most of the communication errors associated with the currently used network interface unit.

In observing the nature of several typical communication errors (see Chapter 4) it was found that over 90% of transmission errors occurs between the NIUs, where communication is via a serial CSMA bus. A much lower error rate is expected if the NIU can employ a better error control scheme.

CLASSLAN is only a simple experimental system. It

can be improved in the following aspects:

1) The host program can be written in a more effecient programming language rather than BASIC to speed up program running and to reduce the code size (BASIC was used for its simplicity and real time debugging capability; the final version is a compiled BASIC program).

2) More work can be done to improve the communication error rate of the NIU. This will improve the overall performance of CLASSLAN greatly.

3) Currently the NIU has a "Purge Mail" command that purges all the mail messages. It would be nice to have an individual mail purge command. This will simplify the host program.

4) A reduced version of CLASSLAN can be developed that would allow only for simple tests. The student station could use only a dumb terminal and an application firmware system (could be included in the NIU). This would reduce the cost of CLASSLAN tremendously for the real classroom implementation.

BIBLIOGRAPHY

1. **Clark et al**, "An Introduction to Local Area Networks", *Computer Networks*, Dec. 1979

2. **Eum, D.**, "COLAN III, a Control-Oriented LAN Using CSMA/CD Protocol", Master Thesis, Oregon State University, 1987

3. **Goldenberg, E. P. and Carter, C. J.**, *Computers, Education and Special Needs*, Addison-Wesley 1984

4. **Heines, J. M.**, *Screen Design Strategies for Computer Aided Instruction*, Digital 1984

5. **Herzog, J. H.**, "TaskMaster User's Manual", Oregon State University, 1986

6. **Hicks, B. L. and Hunka, S.**, *The Teacher and the Computer*, Saunders 1972

7. **Hofmeister, A.**, *Microcomputer Applications in the Classroom*, CBS College Publishing 1984

8. **Joseph, B. M. and Marion, R. M.**, *Computers in the Classroom*, Spartan Books 1970

9. **Kao, S.**, "Design of COLAN II, a Control Oriented Local Area Network", Master Thesis, Oregon State University, 1987

10. **Landa, R. K.**, *Creating Courseware"*, Harper & Row 1984

11. **Lewis, R. and Tagg, E. D.**, *"Involving Micros in Education"*, Elsevier Science 1982

12. **Roberts, N.**, *Integrating Computers into the Elementary and Middle Schools"*, Printice Hall 1988

13. **Tanenbaum**, A. S., *Computer Networks*, Prentice-Hall, NJ., 1981

14. **Yang**, T., "COLAN IV, a Local Area Network for Communication and Control", Master Thesis, Oregon State University, 1987

15. **Zheng**, Y., "A Simple Local Area Network, COLAN (Control Oriented Local Area Network)", Master Thesis, Oregon State University, 1986

APPENDIX

APPENDIX

CLASSLAN CONTROL AND MESSAGE PACKETS FORMAT

CLASSLAN's message and command packets have the following format:

Field 1: Destination NIU number

Field 2: Source NIU number

Field 3: CLSTU task number

Field 4: Source network station number

Field 5: message string

Field 1, 2 are the old Fields of NIU. Field 3 thru Field 5 are the new fields used by CLASSLAN. CLASSLAN also adds two new delimiters "|" and "|" to seperate the message.