DOCUMENT RESUME

ED 027 731 24 EM 006 864

A Feasibility Study of a Central Computer Facility for an Educational System. Final Report.

General Learning Corp., Washington, D.C.

Spons Agency-Office of Education (DHEW), Washington, D.C. Bureau of Research.

Bureau No-BR-7-9000

Pub Date Feb 68

Contract-OEC-1-7-079000-3525

Note-173p.

EDRS Price MF-$0.75 HC-$8.75

Descriptors-Budgeting, Centralization, Computers, Cost Effectiveness, Electronic Data Processing, *Electronic Equipment, Evaluation Methods, Expenditure Per Student, Facilities, *Feasibility Studies, Information Processing, Information Systems, Scheduling, School Administration, Simulation, Systems Development, *Time Sharing

The feasibility of using a centralized data processing facility to sever a large group of secondary schools and the capabilities of two alternative systems were investigated. The population to be served included 100,000 students in grades 9-14 attending 50 schools in an area having a hundred mile radius. Service requirements were defined to include administrative uses, instructional and practice use by students, and faculty analyses in support of academic studies. Procedure was then developed for translating parameters of school usage into load estimates a system would be required to meet. Following a review of the experience of twelve institutions currently using computer systems, two alternative systems--time-sharing with keyboard terminals and remote batch-processing with reader/printer terminals--were selected. Finally a computer program simulated user loading behavior and systems capabilities. On the basis of cost effectiveness, a system with remote reader/printer terminals was favored. The concluding recommendation supported inclusion of the planning and budgeting requirement for a centralized facility in the near term program of the U.S. Office of Education. (SS)

Final Report
OEC-1-7-079000-3525

A Feasibility Study of a Central Computer Facility for an Educational System

february, 1968

U.S. Department of Health, Education and Welfare
Office of Education
Bureau of Research

GENERAL LEARNING CORPORATION • EDUCATIONAL SERVICES DIVISION

FINAL REPORT

Contract No. OEC - 1 - 7 - 079000 - 3525

A FEASIBILITY STUDY OF A CENTRAL COMPUTER FACILITY FOR AN EDUCATIONAL SYSTEM

GENERAL LEARNING CORPORATION
The Educational Affiliate of Time Incorporated and General Electric Company
5454 Wisconsin Avenue, Washington, D. C. 20015
February, 1968

U. S. DEPARTMENT OF
HEALTH, EDUCATION, AND WELFARE

Office of Education
Bureau of Research

ACKNOWLEDGEMENT

The General Learning Corporation wishes to thank the staff and administration of the following institutions, who gave generously of their time, and provided data and thoughtful suggestions during our visits:

Fairfax County Board of Education
Iowa Educational Information Center
New England Education Data Systems
Oakland County Board of Education
Palo Alto School System
Philadelphia Public Schools
Pontiac High School
Portland Community College
Stanford University
U. S. Air Force Academy
U. S. Military Academy

We also wish to give special thanks to Dr. G. E. Anderson, of the University of Massachusetts, and this year's President of the Association for Educational Data Systems. Dr. Anderson contributed ideas in several parts of the study, reviewed the work of the GLC staff, and applied his experience and skill to portions of the report where the need was greatest.

TABLE OF CONTENTS

Acknowledgement ii

List of Tables, Figures and Graphs v

1.0 Introduction 1

2.0 Analysis of Requirements 6

2.1 System Application Requirements and Constraints . . . 6
2.2 Current Educational Computer Use 8
2.3 The Trend of Computer Use In Schools (Grades 9-16) . . 9

3.0 Recommendations 14

3.1 Time-Sharing vs. Remote Batch-Processing 14
3.2 System Recommendations 15
3.3 System Performance 17
3.4 System Cost 18
3.5 Economic Feasibility 18

4.0 Application & Implementation 21

4.1 Systems Design 21
4.2 Management Strategies 22
4.3 Implementation Strategies 22
4.4 User Training 23
4.5 General Workflow Considerations 25
4.6 School Operations Model 26

5.0 Study Procedure 35

5.1 Functional Analysis 35
5.2 Design Synthesis 36
5.3 Design Evaluation via Simulation 36
5.4 Documentation 36

6.0 Detailed Functional Requirements 37

6.1 Introduction 37
6.2 Characteristics of Member Schools 37
6.3 Characteristics of Student Use 42
6.4 Computer Service Requirements 45
6.5 Administrative Data Processing 53
6.6 Storage Requirements 67

7.0	Design Synthesis	71
7.1	Technical vs. Economic Feasibility	71
7.2	General Design Approach	71
7.3	Communication Lines	72
7.4	Terminals	72
7.5	Central Computer Facility	78
7.6	Central Facility Peripheral Equipment	86
7.7	Special Communications Unit	86
7.8	System Programming Requirements	87
7.9	Additional Design Considerations	90
7.10	Design Summary	91
8.0	System Cost and Cost Allocation	99
8.1	Equipment Cost	99
8.2	Other System Costs	99
8.3	Cost Allocation	103
9.0	Simulation of System Performance	105
9.1	Introduction	105
9.2	Hardware Configuration	106
9.3	Educational Time-Sharing System	111
9.4	Remote Batch-Processing System Simulation Model	122
APPENDIX 1	Student Program Running Time Estimates	131
APPENDIX 2	Administrative Data Processing Assumptions	137
APPENDIX 3	Systems and Application Standardization Across Many Regions	146
APPENDIX 4	Transmission System and Teletype Analysis	151

LIST OF TABLES, FIGURES AND GRAPHS

Table No.	Title	Page
2.1	9th Grade Science	10
2.2	Assigned Computer Problems per Student	11
2.3	Assigned Computer Problems per Student Grades 13-16	11
3.1	Per-Student Cost	15
6.1	School/Student Population Model	39
6.2	Enrollments in Selected Subjects as Fractions of Total Students in Grade	40
6.3	Enrollments in Selected Subjects as Fractions of Total Students in Grade	40
6.4	Degree of Computer Use	41
6.5	Special Parameters	41
6.6	Penetration Factor	42
6.7	Assigned Problems Per Course	43
6.8	Average Problems Assigned Per Student/Subject/Grade Unit	42
6.9	Average Problems Per Student/Grade Unit	44
6.10	Average Problems Per Student/Category	44
6.11	Average Problems Per Student/Grade	45
6.12	Estimated Usage Parameters	45
6.13	Time-Sharing Terminal Requirements	46
6.14	Distribution of Time-Sharing Terminals	47
6.15	Runs Per Student by Grade	49
6.16	Total Runs by Grade	49
6.17	Input/Output Requirements	50
6.18	Estimated Input/Output Rates	51
6.19	Reader/Printer Loads by School Size	52
6.20	Student Master File	54
6.21	Report Cards	54
6.22	Attendance	55
6.23	Scheduling	55
6.24	Testing	56
6.25	Personnel Master File	57
6.26	Inventory Master File	58
6.27	Accounts Payable	59
6.28	Budgetary Accounting File	60
6.29	Example of Daily Loading for Peak Day	62
6.30	Example of Daily Loading for Non-Peak Day	63
6.31	Student User Storage Requirements in Characteristics	69

Table No.	Title	Page
6.32	Major Administrative File Sizes	70
7.1	Distribution of Communication Equipment	76
7.2	Time-Sharing System Core Memory Requirements	81
7.3	Remote Batch-Processing System Core Memory Requirements	82
7.4	Time-Sharing System Secondary Storage Requirements	83
7.5	Remote Batch-Processing System Secondary Storage Requirements	84
7.6	Administrative File Storage Requirements	85
7.7	Time-Sharing System Terminals Communication Equipment	96
7.8	Remote Batch-Processing System Terminal and Communication Equipment	96
7.9	Time-Sharing Central Computer System Equipment	97
7.10	Remote Batch-Processing System Central Computer System Equipment	98
8.1	Estimated Cost for Terminals and Communication Subsystem-Time-Sharing System	100
8.2	Estimated Cost for Terminal Equipment and Communications Subsystem-Remote Batch-Processing System	100
8.3	Estimated Cost for Central Computer System-Time-Sharing System	101
8.4	Estimated Cost for Central Computer System-Remote Batch-Processing System	102
8.5	System Cost Comparison	103
9.1	User Program Statistics	111
9.2	Command Type Distribution	112
9.3	Command Input/Output Message Length (In Characters)	112
9.4	Communication Facility Utilization	113
9.5	Input Conditions for Simulation Runs	116
9.6	Summary of Normal Load Results	117
9.7	Summary of Peak Load Results	118
9.8	Summary of Maximum Load Results	119
9.9	Summary of Maximum Load Results	120
9.10	Summary of Maximum Load Results	121
9.11	Summary of EMPSim Output	125

Figure No.	Title	Page
2.1	Computer Usage Parameters	12
6.1	Peaking Conditions of Administrative Applications	66
7.1	Time-Sharing System	93
7.2	Remote Batch-Processing System	94
7.3	Central Computer System	95
A4.1	Even Distribution Data Requirements	115

Graph No.		
9.1	Normal Load	123
9.2	Peak Load Situation	123
9.3	Maximum Load	123
9.4	Job Type One Versus All	126
9.5	Job Type Two Versus All	126
9.6	Job Type Three Versus All	127
9.7	Job Type Four Versus All	127
9.8	Job Type Five Versus All	128
9.9	Job Type Six Versus All	128
9.10	Educational Multiprogramming System Thruput	129

1.0 Introduction

This study was performed by the General Learning Corporation acting as a contractor to the U. S. Office of Education. Its two main goals were to provide:

- an analysis of the functional requirements applicable to a central computer facility having remote terminals in 50 educational institutions with a total enrollment of 100,000 students, and

- preliminary specifications for such a facility.

Given the complexity of today's computer technology and the specific results desired, the goals stated above are not sufficiently detailed to define the study. Additional concepts and constraints have been provided by the Office of Education, and the results of this study can be fully understood only with reference to them. They will be listed in Section 2 of this report.

In the earlier days of automatic computing, the computer's ability to multiply man's calculating speed found satisfactory expression in a kind of speed ratio — the ratio of computer speed to human speed. As this ratio went higher, one computer could serve the needs of an increasing number of individual users. The principal thrust in the advance of computing technology has been toward achieving multiple use. Practically all computer programming concepts are derived in part from the notion that the computer is to be a common tool for many users. The same also holds true for many equipment design concepts.

Economically, of course, multiple users have always been desirable. Computers are and always have been expensive, even though cost-per-computation has decreased at an astonishing rate. But, despite major advances in convenient programming languages for users, the increasing degree of multiple use has brought with it one important disadvantage. Perhaps best described as a loss of accessibility, this disadvantage makes the computer less of a responsive tool and more of an impersonal, bureaucratic, mail-order service.

Recognizing this loss of accessibility, computer engineers and programmers began to seek a solution to the problem several years ago. Two major concepts, time-sharing and multiprogramming, have resulted from this search.

Time-sharing is the rapid time-division multiplexing of a central processor unit among the jobs of several users, each of whom is on-line at a typewriter-like console. The rapid switching of the processor unit among user programs is a particular form of multiprogramming.

Multiprogramming is a more general term, and means the operation of a computer system with several independent programs residing and operating within it.

The two terms are related very closely. In current usage, the difference is that in time-sharing the users are presumed to be at their consoles, making requests of the system, entering data, and waiting for responses. An adequate time-sharing system must switch among programs fast enough to give each user service which is acceptable in his on-line mode. In multiprogramming, there is no implication that the users are on-line. Switching among programs is done on whatever basis is appropriate to the environment; often, it is simply to get maximum utilization of the central processor unit.

Research and development on time-sharing and multiprogramming systems are continuing at many institutions, public and private. Working systems of both types exist. One can buy a subscription to time-shared computing service anywhere in the United States that is served by telephone. At the same time, neither time-sharing nor multiprogramming has reached the point where a set of standard designs or techniques has been established.

More recently it has been recognized that access to computing power may be of significant value in education. Teachers and administrators, faced with ever increasing complexities in required subject matter, vocational opportunities for students, and administrative procedures, view the computer as a potentially powerful tool in their work. This recognition is superimposed on the search for general solutions to the time-sharing and multiprogramming problems.

This study is, in effect, an attempt to identify those concepts which are common to the needs of education and to the current body of knowledge of time-shared and multiprogrammed systems. In computer business parlance, it is a feasibility study.

- What is a Feasibility Study?

  In the folklore of computing, the concept of a "feasibility" study arose in the context of a computer sales effort. Part of the job of selling the customer included convincing him that he could really use a computer for his work - that the computer proposed was, in fact, "feasible" for the organization. Early feasibility studies rarely went into the structure of the organization or into its problems in any depth. In its original (and somewhat narrow) concept, a feasibility study was literally that of seeing if a particular piece of machinery could do a job that needed doing.

  Today it is rather obvious that the computer is useful in education. The computer is already being applied in many educational areas, at least in a piecemeal fashion. The feasibility question is not just one of whether a job can be done; rather it is one of relative economics and merits, contrasting various ways of doing it (including improved non-machine methods). This implies that a great deal is known about the job to be done. In reality, the concept today is that of a "systems" study, examining the job to be done, the organizational setting of that job, the information required to do it, the roles of people, and various alternative ways of accomplishing the desired objectives. The more appropriate questions for today are how to build a more effective information (management) system in which the computer is but one tool. One of the last questions to ask is whether or not computer "X" is feasible. The fact that something <u>can</u> be done by computer does not always mean it <u>should</u> be done.

The following questions should receive careful consideration:

- <u>What is the Job to be Done?</u>

  In order to determine and evaluate ways of getting something done, it is first necessary to determine as specifically as possible what that something is. In theory, this should be in a "how things really ought to be" frame of mind instead of "this is the way things are". In practice, it is very hard to foresee carrying on many of our activities in any radically different way; at best, the usual definition of the job to be done is in terms of present practices and concepts. Occasionally other factors, such as the pres of having a computer, will provide a "pseudo-definition" of the job.

  The history of automating present practices shows that automating tends to change the nature of the job itself more often than not, and in many subtle ways that were not foreseen. By now, we have enough experience with the computer to know that this occurs, and to anticipate some of the effects of automating. To the extent possible, the specific definition of the job to be done should take these effects into account. While consensus about present practices has a place, so do educated guesses about the future.

- <u>What Information is Needed to do the Job?</u>

  "Information" in this context includes the facts, figures, and projections with which to make necessary decisions. In the case of mathematics instruction, libraries of procedures and data might be included.

  Here again, an attempt should be made to foresee what will be really useful instead of what is now in use. In some cases, this will mean reducing a large volume of raw data to meaningful summary statistics; in other cases, it could mean acquiring and displaying new information to aid the decision process.

- How Can This Information Be:
  - Obtained
  - Processed
  - Stored
  - Presented (displayed)?

This part of the study is concerned with looking at the various possibilities rather than the selection of any particular one. Technical assistance can be very valuable in determining the possible contributions of the computer and information technology to the job at hand.

- What is the Present "Information" System?
  - Actual information
  - Roles of people

There are several reasons for taking a close look at what is now being done. Careful analysis may show there is no real reason for change, or that the cost of potential benefits is unreasonably high. Whatever is done, present information of value should not be lost. Regardless of what is on the present organization chart, key people are probably performing tasks somewhat differently. It pays to ask who does what and why.

- What are the Identifiable Costs?
  - Now
  - During transition periods
  - When a proposed new system is operating as designed

The reasons for trying to get the best possible cost picture should be obvious. Yet, two kinds of mistakes appear frequently in feasibility studies: inadequate comparisons of present and future costs and inadequate indications of the costs of transition. The latter is particularly unfortunate for if resources are not available for an orderly transition, either the new "system" will not function as anticipated, or an expensive facility may not be well utilized.

- What are the Identifiable Transition Problems?

These will differ from situation to situation. In general, people need to learn how to use and interact with the new system. Pockets of resistance can endanger the functioning of the new system. Ignorance and lack of understanding on the part of the people involved hinder the transition.

- What are Projected Workloads?

  Such problems as rate of change-over, phase-in, and rate of growth have implications for how much machinery and staff are needed and when. Both volume and timing are important.

- What are Actual Intermediate and Long Range Goals, and the Strategies for Meeting Them?

  After the facts are in, alternative possible solutions may be considered and accepted, rejected, or modified. It may be necessary to rethink the problem in view of the constraints that have developed in the course of study. Normally machine selection, program specification, site preparation, and staff selection occur after this step.

These questions obviously interact in many different ways; the foregoing is not a checklist to be followed. Any actual design of an information system, whether a computer is to be used or not, will undoubtedly be a compromise among many factors uncovered in the study process. Technical advice may be very valuable in ascertaining that the actual strategy proposed is workable, and that a reasonable gain for the time and money to be invested will be attained.

Regardless of the scope and quality of service which a central computer facility may offer, each school will have to do some part of the system design work. This is especially true for the personnel roles and the implementation strategies. However, the central facility staff should serve to provide guidance in these areas, and to limit the range of alternatives which must be considered.

In summary, if the central facility does its job well, the quality of its service will be high enough so that member schools are given meaningful alternatives in deciding how to use it. In addition, it should widen the horizon for each member school by allowing it to use more information better and more quickly.

2.0 Analysis of Requirements

The first section of this report stated the questions which ought to be asked in a feasibility study. This section answers the first of these, namely, what is the job to be done?

The answer is in three parts. First, there are application requirements and constraints which were specified by the Office of Education, and which serve to define the scope of the study. Second, there is a brief review of the observations made during the study on current practices in computer use in education. Third, the requirements for educational computer use in the time period of the possible installation of a central computer facility are analyzed. These will be considered in order.

2.1 System Application Requirements and Constraints

In addition to the general statement of purpose given in Section 1, there are important application requirements and constraints serving to define the feasibility study. The recommendations given apply only in the context of these requirements and constraints specified by the Office of Education as follows:

- The system will serve fifty educational institutions having a total enrollment of 100, 000 students.
- The enrollment of 100, 000 students encompasses grades 9 - 16 with the emphasis on grades 9 - 14. Major universities offering post-graduate work are excluded.
- The fifty institutions served are within a region roughly 100 miles in diameter. No specific actual region is implied.
- The facility will provide service to the following type of users:
    - Students taking courses in programming using languages such as FORTRAN, COBOL, ALGOL, and the like.
    - Students and faculty performing calculations in support of various academic subjects.
    - School administrators and faculty preparing schedules and reports, maintaining records, and performing other data processing tasks.

The study procedure and goals are further defined by certain requirements specified by the Office of Education:

- Two system designs are to be analyzed and designed with different terminal subsystems:
  - Keyboard-type consoles operated on a time-sharing basis.
  - Moderately high-speed readers and printers operated on a remote batch-processing basis.
- The analysis of feasibility and the preliminary specifications are to be presented in functional or parametric form. That is, items of equipment are to be analyzed and specified in terms of their levels of performance, without reference to specific manufacturers' products. At the same time, the system specified must be capable of being installed in 1969 - 1970; levels of performance must not exceed those attainable in that immediate future period.
- For both system designs, the following specifications are to be provided as a minimum:
  - The size of the central computer facility,
    - Central processor speed
    - Size of main memory
    - Size, access time, and transfer rate of auxiliary memory
  - The type of communication network required,
  - Average turn-around times for various usages,
  - Approximate cost of the facilities reflected in the preliminary designs, and
  - A recommended or appropriate billing system which would be compatible with such a facility.
- The study is to include, as an element of procedure, a survey of a number of institutions now using computers so that data on present educational usage may be collected.

2.2 Current Educational Computer Use

An assessment of computer use in education must take into account various types of use and their relative importance to education.

Computers are widely used in administration and research in public education in the United States. There are many examples, some well out of the experimental stage, of computer use for problem-solving by students. There is broad interest in a number of computer-aided-learning projects.

Despite the obvious interest in these computer applications, however, it is clear that the actual penetration of computer service into operating educational institutions is very small. Even in administrative applications, where the penetration is greatest, relatively few institutions have done more than use simple processing routines to implement long-standing procedures. The use of computers for research work is, of course, very well established, but this type of use is not a factor in grades 9 - 14. The long-term possibilities of computer-aided-learning are extremely interesting, but current work is clearly experimental.

For this study, student involvement with computers depends upon problem-solving in support of various academic subjects, and upon courses directly related to the equipment such as programming courses. In these two areas, current penetration in public schools is very low. It is true that most large school districts, and many smaller ones, are at least considering means of providing access to a computer for student problem-solving, and interesting results are being reported from the few who have reached an operating level. Even in these cases, however, it is easy to overestimate actual student involvement. The reason for this is that computer access, if it exists at a school, tends to be used only to augment the standard curriculum, and therefore inevitably becomes an enrichment facility for the bright students.

Computer access as an enrichment device is certainly not undesirable in itself. Indeed, one mode of curriculum expansion involves evolution through the phase of individual enrichment to general offering. Even so, the impetus for this study comes from the anticipation of benefits for nearly all students in the population, not for just the fast learners.

During the course of this study, many opinions were received and discussed as to how computer-based problem-solving can be extended through the curriculum, particularly in grades 9 - 14. To analyze the costs and benefits of such an extension is a major study in itself. Matters of faculty orientation and training, new materials, and changes in physical facilities are involved.

To satisfy the requirements of this study, it is necessary to assume that these matters will be handled in a satisfactory way. As a result, requirements for computer access will be generated by nearly all students in selected courses, and computer problem-solving will be extended into such areas as social studies and business, as well as the sciences and mathematics.

It is necessary also to assume significant progress in the handling of computer programming as a skill to be learned. Among four-year colleges, there are many currently offering programming courses. In this environment, the skill tends to be viewed as a tool to aid concurrent and later studies. In grades 9 - 12, on the other hand, programming tends to be considered as a terminal skill. In the few 9 - 12 institutions where it is taught, programming is usually found in a vocationally-oriented curriculum.

In this study, it was assumed that programming courses would be offered at all levels in the 9 - 14 range. It was also generally assumed that almost all students would take a first course, in grade 9, 10, or 11, and that a significant number would pursue the subject further in grade 12, 13, or 14. No distinction was made between programming courses having a vocational emphasis and those having a supportive skill emphasis.

It must be observed that the very limited use of computers by students in the grade 9 - 16 population cannot be ascribed simply to the fact that computers are new. Attempts to involve both teachers and students have been made over a period of time beginning at least ten years ago. These early efforts were extremely informal and were made possible by the largesse of various computer manufacturers. They certainly were not part of an administration-supported curriculum expansion. Even so, it may seem strange that the seeds thus planted did not grow more rapidly than they did. This issue was not closely examined during this study. The limited observations that were made suggest that the slow penetration has resulted from the lack of a visible, understood educational need, rather than from economic considerations.

2.3 The Trend of Computer Use in Schools (Grades 9 - 16)

For this study, the decision has been made to assume an extension of computer use across as much of the curriculum as possible, and across the entire range of student abilities. In addition, it has been assumed that data processing for administrative functions will be extended about as far as it is now used in the most advanced (in EDP) school systems today. It is not intended to show that this extensive use of computers will actually occur in the next two or three years. The question is one of feasibility, and it seems best to consider full usage of computer facilities in developing an answer.

In addition to the survey of institutions which comprised the first part of the study, a number of published data sources were used to establish the expected computer involvement. These included:

(1) Subject Offerings and Enrollments in Public Secondary Schools, OE No. OE-24015-61, 1965.

(2) 1966 Digest of Educational Statistics, OE No. OE-10024-66.

(3) Computers in Higher Education, Report of the President's Science Advisory Committee, February 1967. Popularly known as the "Pierce Report".

Using data from these sources, together with trend estimates gathered during the survey, a set of tables was constructed giving degree of computer involvement as a function of subject and grade level for grades 9 - 12. In addition to data on enrollment, these tables contain a number of computer-related parameters such as number of problems assigned, estimated number of computer sessions per problem, and computer time per session. For grades 13 - 16, instead of grade level, the terminology of the Pierce Report was used to classify the student population. This terminology identifies degree of student involvement as "substantial", "limited", or "casual".

The tables will be treated in detail in Section 6. To illustrate the magnitude of the problem, however, Table 2.1 shows the computer-load estimating sequence for 9th grade Science.

% of 9th Grade Students Enrolled in 9th Grade Science	62%
Fraction of Science Enrollment Actively Using Computer	0.6
Assigned Programs per Course	2

Table 2.1 9th Grade Science

The first entry in this table is taken from Source (1). The second and third entries are estimates based on opinions taken during the survey. These data and estimates were assembled for five elements of the 9 - 12 curriculum:

- Programming
- Mathematics
- Science
- Business Education
- Industrial Arts

Limiting the analysis to these elements of curriculum is not to imply that computers will not be used in language arts or social studies. The intent is to identify the major uses and to provide enough margin in later estimates to allow for expansion.

By combining these estimates in an appropriate way, it is possible to state an average number of computer problems per student for each year. Table 2.2 gives these values.

Grade	9	10	11	12
Average Number of Assigned Problems	3.9	4.2	5.2	6.6

Table 2.2 Assigned Computer Problems per Student

It cannot be overemphasized that the numbers in this table are derived from a collection of estimates similar to those used in Table 2.1. In Table 2.2, these estimates become a summarized forecast of the average number of computer problems each student will be assigned during a year in the specified grade.

For students in grades 13 - 16, the corresponding forecast is given in Table 2.3.

User Category	Casual	Limited	Substantial
Average Number of Assigned Problems	2.3	6	30

Table 2.3 Assigned Computer Problems per Student, Grades 13 - 16.

The next step in the analysis of computer use is to estimate a sequence of usage parameters on a "per-problem" basis. The sequence of parameters is given in Figure 2.1.

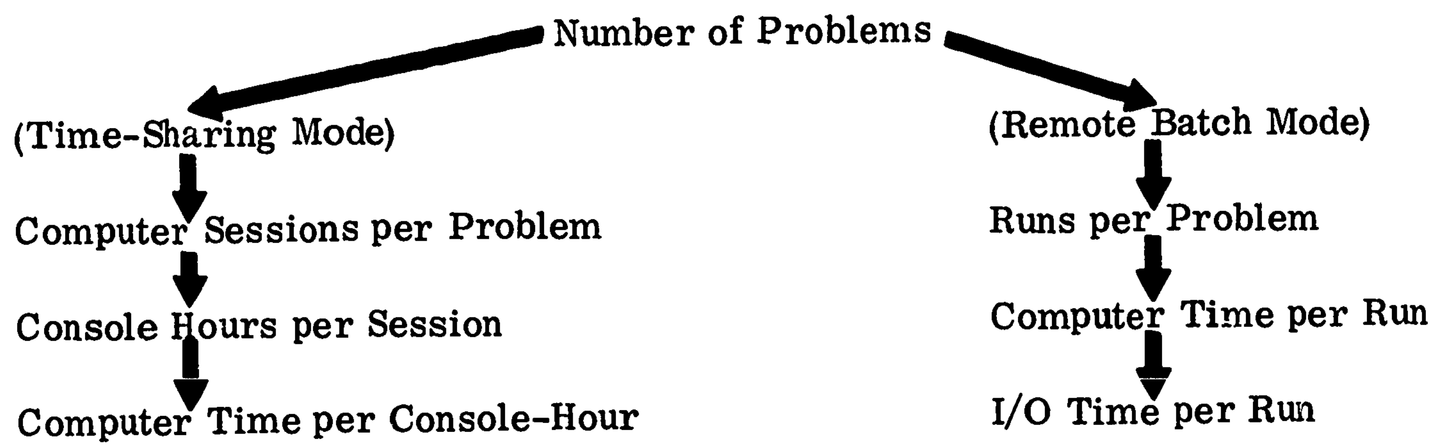


Figure 2.1 Computer Usage Parameters

Tables of these parameters, as estimated for this study, are given in Section 6. The end result of this estimating sequence is a measure of the amount of system computing and data-handling capacity needed to serve the given population.

It is necessary to accommodate a large computer load of administrative data processing in addition to what is required for student use. It was found, however, that this requirement did not have a significant effect on the total system capacity. The reason for this is simply the assumption that almost all administrative work could be handled at times other than the hours when students would be using the system. As will be seen in Section 3, Recommendations, some features must be added to the system to accommodate administrative data processing, but they add relatively little to system cost.

Full understanding of the derived measures of capacity can be obtained only by studying Section 6. The main conclusions, however, can be summarized as follows:

For the given population, a time-sharing system would have to include about 1000 terminals, distributed among the schools roughly in proportion to students enrolled. Each school would be connected to a central computer facility through leased telephone lines. The central computer facility would contain the computer itself plus a number of memory, input-output, and communications devices. To handle the load, the computer would have to be an extremely fast, powerful machine, in the class of the most powerful computers now installed (1967).

For a remote batch-processing system, the terminal requirements are, of course, very different. In this case, medium-speed card readers and printers would be installed in the schools. One or two reader-printer combinations would go into each school, depending on enrollment, connected to the central facility by telephone lines. As before, the central computer would need to be a very powerful machine.

There are a number of indications, including opinions taken in the survey, that the ideal system to satisfy user needs would contain both time-sharing and remote batch-processing capability. It seemed evident that such a system would necessarily be more expensive to install than a single-function system, and, as a result, it was not considered. The advantages of a mixed system are, however, quite compelling.

3.0 Recommendations

The previous section briefly stated the requirements to be placed on the proposed system. Later sections will present those requirements in much greater detail together with the designs which will be developed to satisfy them. Before the presentation of the detailed analysis, however, it is desirable to offer a summary of the significant results.

3.1 Time-Sharing vs. Remote Batch-Processing

The first conclusion of the study is that the two types of systems considered are significantly different, not only in equipment configuration and cost, but also in the nature of student use. Users of both types of systems are enthusiastic about their educational potential, but they recognize that each type has distinct advantages.

It is important to remember that the students' approach to using a computer is strongly influenced by the type of terminal equipment available to him.

With a keyboard-type terminal, a user expects to conduct a dialog with the system. The approach taken to problem-solving is adjusted to the dialog procedure. The language used may be more or less restrictive. The waiting time for responses may vary widely and even approach the limit of patience. Still, the user stays with the system as long as he thinks it is working on his problem. Access to the computer, in this case, is the same as access to a terminal.

With terminal equipment which reads cards and prints lines of output, the user does not expect dialog. He expects the type of service one might expect from a trained worker who can take a defined task back to his office and return with the completed work in a reasonable length of time. The user has no need to stay with the system with this type of terminal equipment. It works in his absence. The approach taken to problem-solving adjusts itself to this environment.

In this study, it is necessary to consider both types of access for three types of systems use:

- Teaching programming
- Student problem-solving
- Administrative data processing

It cannot be said, however, that identical work can be done on each of the systems. It has already been observed that this cannot occur because the terminal determines what work is done to a significant degree. Of course, any recommended system must handle all three uses in an acceptable manner.

In the case of student use, questions of teaching effectiveness become important. Also, existing administrative procedures must affect the use of the system for administrative data processing. Strictly speaking, this study does not include an analysis of teaching effectiveness or of possible redesign of administrative procedures. Instead, it approaches the design of the system with the understanding that ultimate system effectiveness will be achieved as the users adopt teaching, problem-solving, and administrative practices appropriate to the type of terminals available.

3.2 System Recommendations

The second conclusion drawn from this study is that it is technically feasible to provide the required computing services for the specified school/student population using central computer facilities. Furthermore, either of the approaches, that is, either time-shared remote keyboard terminal or remote reader/printer units, can be used.

Both approaches are expensive if their costs are looked at relative to average per-student secondary education costs. Of the two, the system using remote reader/printer units will be significantly less expensive. Specifically, the cost conclusions reached are shown in Table 3.1 below.

System Type	Annual average cost per student (approximate)
Time-shared keyboard terminals	$30
Remote reader/printer terminals	$22

Table 3.1 Per-Student Cost

It is clear that the nature of the student/computer interaction is quite different in the two types of system. However, it is also clear that the application requirements, as specified by the Office of Education, will be met by either system. Moreover, the administrative applications will be performed much more conveniently through reader/printer terminals. Therefore, the recommendation is that primary consideration be given to the implementation of a system using reader/printer units as the in-school terminals.

In this recommended system, the in-school equipment will consist of one or more identical terminal "subsystems". The terminal subsystem will contain a card reader, a line printer, control devices for each, and communications equipment to handle data transmission to and from the central computer facility. Data will be transmitted over leased telephone lines between each of the member schools and the central location. At the central location, a large-scale computer system will operate in a multiprogrammed mode, handling work requests from and returning output data to the member schools.

Work requests will be handled by the central computer facility according to an automatic priority system. Priorities will depend upon a number of factors, including the type of application (student use or administrative data processing), the expected running time, memory required, and the like. During school hours, student work will tend to receive high priority. Typically, output for a student problem will be received within five minutes after submission. Some types of student problems will be placed in an overnight service category; for example, problems which have unusually high data-storage requirements should be given low priority because they tend to clog the system for other users.

Administrative data processing will generally receive overnight service. There will be exceptions to this, as in the case of daily attendance reporting. Other exceptions may occur during particularly critical periods as, for example, class scheduling times.

While the system described above appears the most desirable both functionally and economically, the alternate approach, using time-shared keyboard terminals, certainly merits consideration. In terms of equipment, the greatest difference would be in the in-school configuration, but there are also significant differences at the central facility. In this case, each member school would have one or more groups of keyboard terminals, each group consisting of from five to twenty terminals. Each terminal group would be connected through a communication device to the central computer facility. The central facility would consist of a large-scale computer system with a configuration very similar to that required to service the remote reader/printer terminals. The major difference at the central facility would be that more high-speed memory (core memory and magnetic drums) would be required to handle the time-shared keyboard terminals. Also, there would be major differences in the computer programs required to monitor the operation of the two types of system.

Operationally, the keyboard terminal system will be quite different. In this case, student problem-solving and programming exercise will occur in an interactive or "conversational" mode. Student requests will consist primarily of commands to the central computer facility to "run", "save", "list", and so on, programs prepared at the keyboard. Typically, responses to these requests will occur within a few seconds, giving the student an immediate opportunity to select the next step toward solution of his problem. A series of these request-response interactions will constitute a problem-solving session. On the average, an individual student session will take about twenty minutes of terminal time, but will consume only a few seconds of actual processing time in the central computer.

The processing of administrative data will also be quite different in the keyboard terminal case. Where rapid turn-around is required, as for daily attendance reporting, the terminals will be used. For most administrative data processing, however, low-cost keyboard terminals do not offer adequate speeds for data input and output. To handle this work, a courier service would be required, providing regularly scheduled deliveries of input and output data at the central location and the member schools.

After considering the modes of operation of the two types of systems, it was concluded that the most desirable arrangement would be a mixed system, that is, a system with both types of terminals in the member schools. The varied educational needs would certainly be better satisfied with such a system. However, it should not be assumed that the technical feasibility of the two systems as proposed implies the feasibility of a mixed system. The indication is that the mixed system would be technically feasible, but that the technical problems in its design could be solved only by combining elements which would be substantially more expensive than either of the systems having only one terminal type.

Even with the added cost, however, the attempt should be made to include some time-sharing capability if a remote batch-processing system were to be implemented. This could be done on a relatively small scale without incurring the large additional cost referred to above.

3.3 System Performance

The later section on design synthesis gives the required performance parameters of each of the system elements for both systems. The general arrangement of the systems has already been described. Of greatest interest here is how each type of complete system will perform in the prescribed environment.

Overall system performance is predicted by the use of simulation programs which are described later in the report. The simulation work provides encouraging results.

3.3.1 Time-Sharing System Performance

Simulation of the time-sharing system was done for several sets of the most important system parameters. The result is the prediction that under normal load conditions, the average turn-around or response time will be five to six seconds for a request calling for actual computing, and less than two seconds for any other type of request.* Under peak load conditions, these response times may go up to about four times the normal values. These response times are averages, of course, and the variance can be expected to be quite large.

On the basis of experience with operating time-sharing systems, these response times, although certainly not negligible, are considered to be satisfactorily small.

3.3.2 Remote Batch-Processing System Performance

For this system, the simulation indicates that average turn-around time at the central computer will be less than one minute for all types of jobs considered. The limiting factor on actual turn-around time at the school location will be the wait time before reading input data and before printing output data. However, it appears that for the expected job sizes, very few jobs would have turn-around times greater than ten minutes, and most would be finished in five minutes.

3.4 System Cost

Cost estimates are derived from published price information on the various system elements. A detailed breakdown is given in Section 8. The cost summary from that section is presented on the following page.

3.5 Economic Feasibility

Both of the systems described are expensive. In both cases, costs are about 5% of average per-student annual school operating expenses. Few school districts will be persuaded to make such a commitment easily. But this does not mean that the systems are not economically feasible. Economic feasibility depends in part on absolute purchasing power, of course, but it also depends on the cost-to-benefit ratio. The immediate and potential benefits which these systems may produce are unknown.

It must be clear that the benefits are really unknown. The set of requirements, stated broadly by the Office of Education and defined more closely in this study, are a definition of a specific addition to a learning environment, and are not to be confused with learning itself.

---

* In currently operating time-sharing systems, less than 15% of requests call for actual computing.

		Time-Sharing System	Remote Batch-Processing System
Terminals & Communication Equipment	Total $ per month	$ 164,100	$ 118,800
	$ per student-year	$ 19.70	$ 14.30
Central Computer System	Total $ per month	$ 82,400	$ 62,400
	$ per student-year	$ 9.90	$ 7.50
Total System	Total $ per month	$ 246,500	$ 181,200
	Equiv. purchase *	$ 9.8 million	$ 7.3 million
	$ per student-year	$ 29.60	$ 21.80

Table 8.5 System Cost Comparison

* Based on 40 months lease equivalent

The desire to enrich the learning environment with computing capability comes from observing how similar enrichment has produced benefits in other activities. Little may be known about the learning process, but it must depend, at least in part, on a flow of information. Computers outdo people in causing information to flow accurately, reliably, and quickly. There is good reason to assert that the installation and use of one of the proposed systems would effect benefits of sufficient magnitude to justify the cost.

This situation has the usual elements of a risk venture, with the additional element of tremendous public interest whenever the public schools are involved. A full discussion to determine how best to distribute the risk is beyond the scope of this report. But it seems clear that the major cost of initial development of one of these systems must be borne by a public agency, such as the U. S. Office of Education. A public agency is in a postion, not only to take the risk, but also to evaluate and promulgate the results far beyond the scope of the selected local school-administrative unit.

The concluding recommendation is that the U. S. Office of Education should include in its near-term program detailed planning, scheduling, and budgeting for the implementation of an educational computer system similar to one of those described in this report.

4.0 Application and Implementation

This feasibility study has attempted to project workloads and look at possible hardware configurations that would handle these workloads. It has become obvious that many other things need to be done before a hardware configuration could be installed and made to work well in a school context.

4.1 Systems Design

Although the functioning of hypothetical systems has been assumed for purposes of this report, the actual systems design work for the schools involved still needs to be done. Even though much of this will involve the centralized design work outlined below, some will need to be done in each school administrative unit.

4.1.1 Administrative Systems

School administrative data processing systems tend to develop in a sequence corresponding to local priorities and capabilities. The result of this growth, when compared among different schools, is a collection of systems which contain many of the same processing steps, but which differ markedly in their external features. It will be extremely desirable to effect some standardization among schools in this respect. This will be difficult, but it will become even more difficult if delayed. Along with standardization in processing systems, an effort should be made to design standard operating procedures for school personnel to follow. The future effectiveness of the central facility will be enhanced directly in proportion to the quality and degree of acceptance of these efforts.

4.1.2 Software

Once an administrative system has been designed, the appropriate computer programs will have to be specified, written, tested, and documented. Actual data file structures, forms design, and error detection and correction procedures will have to be designed. A great deal can be learned from existing educational data processing activities, thereby reducing the amount of original design work required.

In addition, the software for student programming must receive attention. It is reasonable to expect that the manufacturer of the computer selected will supply appropriate compilers and assemblers; it is not as reasonable to assume that the monitor/control system will provide for appropriate teacher monitoring of student work. It is suspected that the attributes of a good teacher feedback monitor system will need study and implementation before much student usage occurs.

4.2 Management Strategies

Although data processing has been managed successfully in many different ways in the educational enterprise, it is apparent that many if not most successes are the result of accidental evolution rather than careful design and planning.

4.2.1 Political Milieu

It is evident from both education and industry that administrative data processing as well as organizational change itself requires interaction with top management. A central computer facility, regardless of the excellence of its hardware, software, and personnel, is not likely to succeed if simply superimposed on existing school organizations.

The factors leading to probable success in school settings need to be clearly identified and steps planned to see that political conditions are as nearly optimum as possible before a central computer facility is installed.

Preinstallation study and plans will need to be made for the control of a central facility. It is not easy to answer the question of what kind of administrative setting (e.g., control board) should be established.

4.2.2 Operational Management

The success of an actual central computer facility will depend in part on proper staffing and day-to-day operation, both at the facility itself and at the "other end of the line" in the schools. There is an increasing amount of literature from business and industry on effective computer and data processing center management. Study should be given to the development of similar guidelines for the educational setting, including management standards, staffing patterns, operator responsibilities, and clerical responsibilities.

4.3 Implementation Strategies

It was assumed that the workload requirements, although dynamic, were based on an implemented and working system. Unless the central computer facility replaces a number of well-functioning local data processing activities, it is not likely that the workload anticipated would develop immediately. Nor is it likely that the schools could immediately make use of a far greater computer capability than they have had previously. Similarly, it is reasonable to assume that not all administrative and student processing capabilities will be ready for use at once — some will take more time than others to implement.

It is important to avoid wasting money and premature expectations in the schools to be served. It is also important that the implementation of the central computer facility proceed with realistic speed. Obviously suitable training programs for the school staff members will need to be devised and executed. In this process, attention will have to be given to the psychological mechanisms of change, many of which are not yet clear.

4.4 User Training

While the details and mechanisms of user training may require further analysis, several guidelines are now apparent.

It is recommended that plans be made for the development and testing of suitable training courses at various levels well in advance of any actual installation. In addition to the immediate operational training, much of the training that is long-range must have an early start.

Since some of the skills needed are those traditionally taught in professional courses in colleges and universities, it will be highly desirable to work closely with the institutions of higher education offering such training to school people within the area.

4.4.1 Mathematics Teachers

Every mathematics teacher in participating schools should receive instruction in the compiler/assembler/monitor system that will be available. Even though many of these teachers will not initially be involved in the active use of the computer or have students who are, it would appear to be better strategy to include all teachers rather than a select few.

Appropriate text materials and student problems will need to be found or created prior to the inclusion of computer programming as an integrated part of mathematics courses. It is anticipated that the revolution in mathematics curriculum will be greatly accelerated by the availability of significant amounts of computer time for use by all mathematics students. Mathematics teachers will need time to participate in and understand these curricular changes.

4.4.2 Vocational and Business (Commercial) Teachers

Some of these teachers will need to learn administrative (as opposed to scientific) programming so they may give appropriate instruction to their students. All teachers of courses involving office procedures, machine operation, and/or data manipulation will need to learn how to use the computer as a tool, perhaps with the emphasis on data management rather than programming.

As with the mathematics teachers, it may be expected that widespread use of the computer will have curriculum implications of major proportions, many of which cannot now be predicted in any detail.

4.4.3 Teachers of Other Subjects

Many teachers of science, economics, psychology, and other courses in which the occasional manipulation of data could be valuable will need to have an appreciation of what can be done and how. It is probably too much to hope that all such teachers will become comfortable about programming; however, they should have the opportunity to learn programming if they wish. Data management skills will be more important, and new emphasis on the gathering and interpretation of data probably will have to develop.

The curricular implications of computer availability in these areas is probably not as drastic as it is in the aforementioned areas, but some time and attention should be available. Research will be needed to determine the value of different kinds and amounts of exposure to data.

4.4.4 School Administrators

School administrators at all levels will need training both in the mechanics of using the new facility for their purposes and in the uses of information. The latter could be one of the most beneficial results of a well-designed school information system.

Continuing instruction will be needed in such areas as simulation and modeling, projections, operations research as applied to education, research techniques using the data at hand, and the overall systems approach. It is anticipated that the advent of readily available computer facilities will spur a growth of interest in these areas well beyond anything practiced in education today.

Training for school administrators, including guidance counselors and teachers insofar as they are consumers and producers of information, will of necessity be a long-term effort. The change between present practices and new practices is too great to be bridged by a short-term training program, although such a program will be necessary for the almost immediate implementation of standard applications.

4.4.5 Operating Staff

The operating staffs of both the central computer facility and the participating schools will need training in the simple operation of the facility.

Ideally, all operating procedures should be documented so that training in this case will be primarily familiarization with equipment and procedures.

4.5 General Workflow Considerations

The in-school equipment is not only an element of the total computer system, it is an element of the personnel-procedures-equipment system within the school. Its physical installation must be designed for the personnel-procedures environment in which it will operate. Some aspects of this environment appear to be common among schools, and lead to the following recommendations.

- All terminals should be used under adult cognizance.
  - In the work center (student lab), the center operator controls signing up to use terminals, and sees that they are properly used.
  - In classrooms and laboratories, the teacher concerned oversees terminal usage.
- Some terminals will be used intermittently in classrooms and regular laboratory settings rather than in a special work center. Such terminals might be portable enough to be locked up when not in use. Further protection may be afforded by switchboard connections in the control center that actually assign terminals to computer lines.
- Rooms with teletypes, printer, and punched card machinery should have adequate soundproofing.
- Any student-operated teletype should be capable of being monitored by a teacher on any other teletype by means of switching in the control center.
- The center operator should be able to monitor every teletype in use, in turn, briefly and unobtrusively.
- A physical division should exist between regular student areas and the control center. A counter or suitable windows would do. Some controllable doorway to permit adult (teacher, control center operator) passage when needed is also envisioned. In part, this is for protection of administrative data.

- A school will have more terminal equipment than it can operate at one time. A school will establish its own priorities for using the available line and computer time.

- Mark sense or punched cards will be used when possible, even for student work, to reduce terminal requirements without significantly slowing student turn-around time.

- A school administrator schedules the data and job through his local coordinator (center operator or director), who in turn arranges any necessary clerical work, computer time, messenger service, etc. In turn, the local data coordinator keeps the administrator informed of the system's capabilities (and problems).

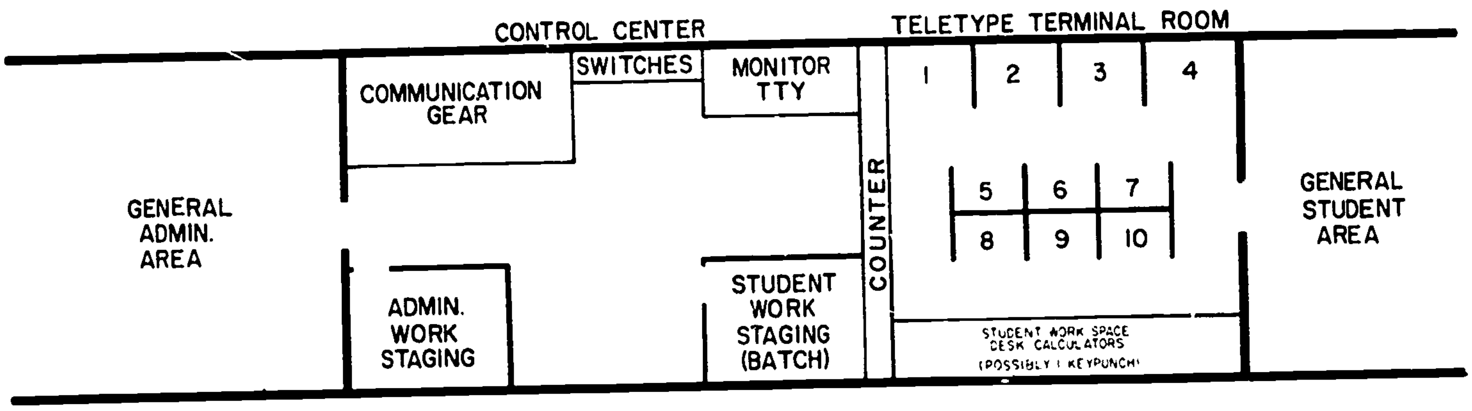
4.6 School Operations Models

As a further step in anticipating the implementation of a centralized computer facility, several possible operations models have been outlined. Some of these models go beyond the system capabilities recommended. They are included to show the range of operational possibilities which may appear at a later time.

Four examples of operations models are presented in the remainder of Section 4. For each case, a diagram showing the placement of equipment and working areas is followed by an outline which describes how people and activities relate to the model under discussion. The first example begins on the following page.

4.6.1 School has teletype and messenger capabilities.

All administrative and some student work by messenger.



Additional terminals or outlets for portable terminals:

Teachers' Stations:
- 6 math. classrooms
- 4 science classrooms
- 4 business education classrooms

Student Stations:
- 2 science laboratories (desk calculator mode)
- 6 misc. stations - e.g., instructional resources center

Other student use at school's center.

Center Operator:

Assigns teletypes to students (and to lines)
Monitors terminal use
Reports defects and malfunctions to source of correction
Receives and returns student batch-processing work

Organizes administrative work for batch-processing
Distributes returned administrative material
Keeps track of supplies needed - both administration and students
Controls administrative data files
Routes communications
Acts as contact person for both administrative and student problems
(May have other administrative duties)

Teacher:

Notifies center operator of anticipated student use
May monitor actual student use if desired (1 student at a time)
Receives reports of student progress
Requests special terminal arrangements
Submits administrative work requests and data via center operator (e.g., scoring teacher-made test)

Student:

Hands in runs for batch-processing and picks up results
Hands in material to be available at a terminal at a specified time
Signs up for terminal time
Requests assignment to a specific terminal when ready to run
Uses terminal in science laboratory as integral part of lab
Gets needed supplies from center operator

Parameters of Messenger Service:

Overnight turn-around on all runs
May go either to the main center or to a sub-center with reader-printer capabilities
May provide service several times daily depending on volume, distance and cost of waiting time
May deliver prepared data forms - test answer cards or sheets

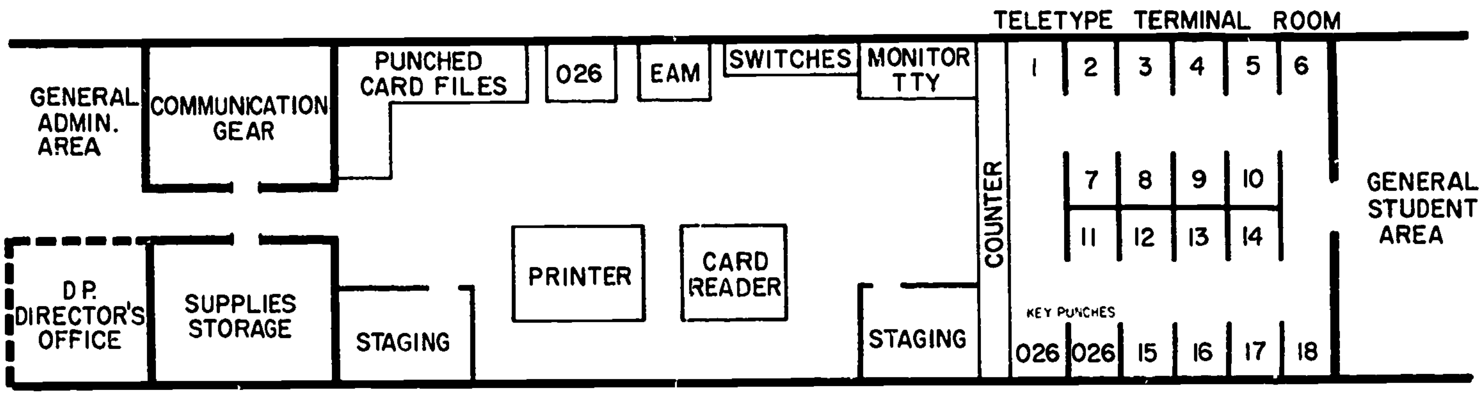
Types of batch-transactions:

- Self-contained normal batch-type programs and data
- File up-date data transactions (also possible terminal)
- Large files of administrative data (e.g., report cards)
- Requests for reports from central data files (also possible terminal)
- Programs and data to be available at a terminal when specified (including teacher monitor programs for student assignments)

Terminal transactions and communications:

- Desk calculator use for quick equation evaluation
- Self-contained student programs or interaction with teacher monitor
- Interaction with previously stored materials
- Limited administrative interrogations

4.6.2 School has reader-printer-teletype capabilities. (Messenger service irregularly)



Additional terminals or outlets for portable terminals:

6 Math classrooms
2 Science classrooms
2 Business education classrooms
2 Science laboratories
10 Other misc. outlets

Data Processing Director:

- Supervises rest of staff
- Determines school priorities for use of computer resources
- Schedules work flow
- Works cooperatively on systems development
- Handles local analysis problems
- Assists school personnel in use and understanding of data
- Contact person for administrative and student problems
- Directly controls administrative work
- May do some maintenance programming

Center Operator (2 shift operation):

Assigns teletypes to students (and to lines)
Monitors terminal uses
Reports defects and malfunctions to director
Receives and returns student batch-processing work
Organizes file maintenance and administrative work
Routes communications and administrative output
Operates card reader for batch-processing and for student runs to save teletype time
Operates printer to receive printed output under control of monitor teletype
Manages I/O for schools with more primitive facilities

Teacher — Same as 4.6.1

Student — Same as 4.6.1, except:

Student may be able to hand in program and have it available at a terminal almost immediately, and
May request printed output on the printer and get it quickly

Messenger Service:

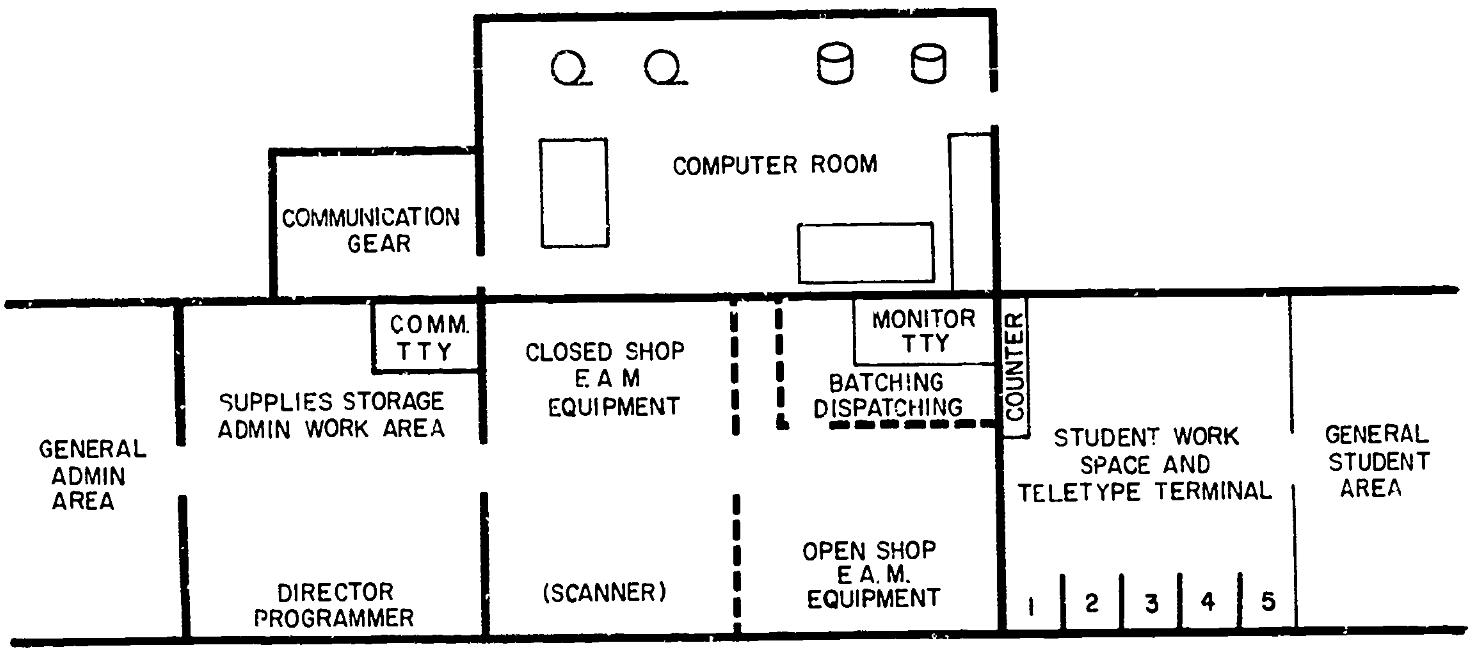
Overloads
Supplies delivery, including prepared data forms if necessary
Punched card output delivery

Batch-transactions — Same as 4.6.1, except:

Student card input and printed output of any volume may be included if the work "mix" permits

Terminal transaction — Same as 4.6.1

4.6.3 School has small-scale computer capability and teletypes. (Messenger service irregularly)



Additional terminals or outlets for portable terminals:

12 Classrooms
3 Science Laboratories
10 Misc. Outlets

Data Processing Director — Same as 4.6.2, except more staff.

Maintenance Programmer:

- Maintains local computer operating systems
- Maintains administrative and teaching programs used locally
- Participates in overall systems development effort

Dispatcher — Functions of Center Operator (4.6.2), plus:

- Deciding, within guidelines, what will be done locally vs. centrally depending on requirements of job and demands on equipment

Operators (Probably 2 at a time) (2 shift Operation):

Perform work under direction of Dispatcher-Supervisor

Teachers — Same as 4.6.1, with:

Possibility of direct computer use (particularly for voc. ed.)

Student — Same as 4.6.1, with:

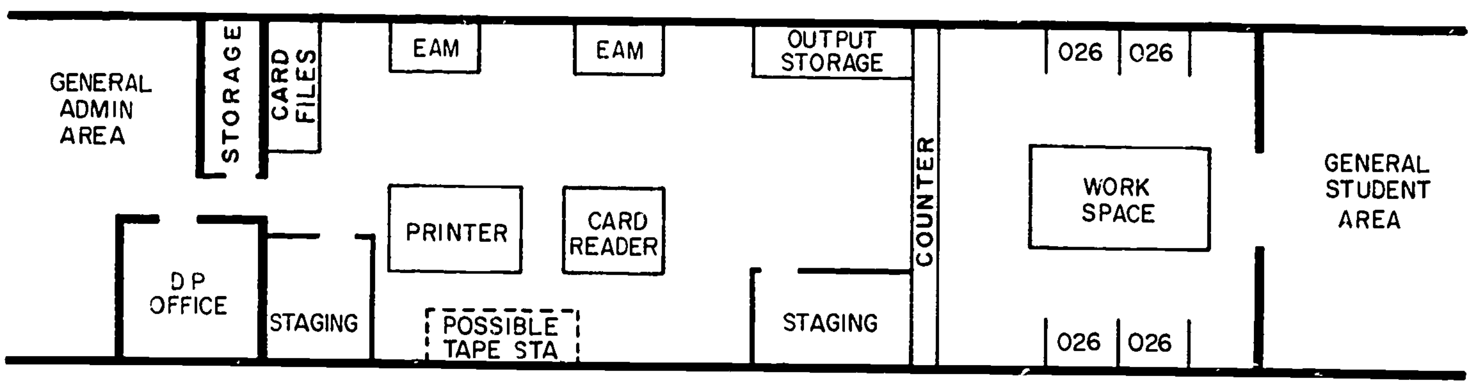
Opportunity to operate computer directly or observe computer operation when appropriate to type of work student is doing

Messenger Service:

Rarely used except for communication overload or equipment failure

NOTE: Each school may have considerable variation in its use of teletypes. For example, a school with a well-defined instructional resources center may want teletypes there. A school may want one or more teletypes in a department office.
It may be a good idea to provide many more outlets than needed to allow for contingencies and changed philosophies.

4.6.4 School has reader-printer capabilities; no teletypes. (Messenger service irregularly)



Data Processing Director — Same as 4.6.2

Center Operators (1 or 2 shift operation):

- Receives and returns student batch-processing work
- Organizes file maintenance and administrative work
- Routes communications and administrative output
- Operates card reader and printer
- Manages I/O for schools with less advanced facilities

Teacher:

- Notifies director of anticipated student work
- Receives reports of student progress
- Submits administrative work requests and data via center operator

Student:

- Hands in runs for batch-processing and picks up results

Messenger service — Same as 4.6.2

All transactions batch mode

5.0 Study Procedure

This study was comprised of four main tasks:

- Functional Analysis
- Design Synthesis
- Design Evaluation via Simulation
- Documentation

5.1 Functional Analysis

This first task was divided into two main parts. The first was a survey of twelve institutions currently using computers in one of the modes of interest. The survey included visits to the institutions, discussion of computer applications with school personnel, review of equipment, forms, and procedures, and documentation of the observations made.

The second part, conducted after the survey was complete, was the development of a procedure for translating the parameters of student computer usage into meaningful parameters of loading on the computer systems to be designed.

The survey included the following twelve institutions:

Fairfax County Board of Education
Iowa Educational Information Center
New England Education Data Systems
Oakland County Board of Education
Palo Alto School System
Philadelphia Public Schools
Pontiac High School
Portland Community College
Stanford University
U. S. Air Force Academy
U. S. Military Academy

At each institution, an attempt was made to gather information on computer usage which could be directly related to the design process to come later. In most cases, it was not possible to do this. Although a great deal of useful general information was obtained, there was very little data available on the actual statistics of usage. This is not a reflection on the institutions visited; in fact, in those cases where computer usage has become a regular part of the academic program, as at the military academies, there are effective procedures for gathering information on usage, and this was very helpful to the study team.

But in most of the institutions, particularly the 9 - 12 institutions, the student involvement with computing is so new that measurements are not yet effective.

The second task of the functional analysis, developing a load-estimating procedure, was made more important by the paucity of reliable usage data. It was felt that, if the data to be used was not fully substantiated, the procedure used to manipulate it should be very clear.

5.2 Design Synthesis

In this study, synthesizing system designs was primarily a matter of identifying the performance parameters for individual system elements. The general organization of system elements could be fairly well established directly from the study requirements and knowledge of existing time-sharing and remote batch-processing systems. The logic of the design procedure actually followed closely the order of presentation in Section 7 of this report. At many stages of this procedure, it was necessary to review the choices being made relative to earlier choices, so that there could be continuing assurance that minimum total system cost would be achieved.

5.3 Design Evaluation via Simulation

Initially, when the goals of this study were considered, it was clear that computer simulation of the systems designs would be most desirable. Fully one-third of the effort in the study went into the concepts, design, programming, computing, and interpretation of system simulation. This simulation was a complex computation requiring the use of a large-scale computer. It involved the expression of various load parameters in the form of probability distributions in an attempt to match the random pattern in which the actual system will be loaded. The simulator analyzed the effect of each event occurring in the system, and gave indications as to which system elements were most heavily used, as well as many other performance statistics.

5.4 Documentation

This final phase of the study has been approached with the idea that the report contents should be meaningful to as wide an audience as possible. There are important decisions to be made relative to the implementation of the systems proposed. They are educational and economic decisions as much as technical ones. The questions of technical feasibility and equipment specifications are important, but not as important as the questions relating to the penetration of computer service into the structure of educational resources. It is hoped that this report will be of most use to those who are prepared to consider these most important questions.

6.0 Detailed Functional Requirements

6.1 Introduction

A summary of what the proposed system must do was presented in Section 2. The detailed analysis in support of the summary is presented in this section.

The task of functional analysis, as originally conceived, consisted of visiting a number of educational institutions, observing current computer work, and analyzing the data collected. This survey was made. In the judgment of the study team, the computer work observed at the various schools was generally well-planned and well-executed. There was, however, no instance in which computer use had penetrated the curriculum to the degree specified for this study. Also, while teachers and administrators were eager to discuss plans for expanding computer use, these discussions did not result in a clear picture of what could be achieved across the curriculum in 1969 or 1970. Since such a picture is essential to the estimation of computer loading, it was constructed, following the survey, from several data sources, and using rather broad assumptions regarding individual computer use. The pattern of expected computer use which is described in this section is consistent with opinions gathered during the survey, but it goes well beyond any of the actual situations observed.

The factors which will affect the loads on the computer system can be classified as follows:

- Factors which characterize the schools
- Factors describing individual student use
- Factors describing administrative data-processing use

These three sets of factors will be considered in order.

6.2 Characteristics of Member Schools

It is necessary to specify a distribution of school sizes so that the specified population of 100,000 students will be properly distributed among the fifty schools. Using the numbers given, the average school size will be 2000. Since the average U. S. high school enrollment is less than 400, over-all U. S. statistics will be of little use in making the distribution.

To construct an appropriate distribution, the following assumptions were used:

- Since emphasis is to be placed on grades 9-14, only 2 four-year colleges will be included.

- The number of students in four-year colleges will be much smaller than national statistics would give, since it is assumed that four-year colleges, not included in the distributions, exist in the region to serve most of the college-bound 12-grade graduates.

- The number of students in two-year colleges will be higher than national statistics now would indicate.

- For 9-12 institutions, the minimum size will be 500 students, the maximum size 4,000 students.

Following these assumptions, and liberally rounding off enrollments to convenient numbers, the distribution shown in Table 6.1 was derived.

School Size (1000's)	No. of Schools	Students in Grade 9	10	11	12	Total in Category (1000's)
4	6	1,150	1,100	950	800	24
2	24	600	550	450	400	48
1	7	300	275	225	200	7
0.5	2	150	140	110	100	1
Total	39	-	-	-	-	
Total in Grade		23,700	22,005	18,295	16,000	80

School Size (1000's)	No. of Schools	Students in Grade 13	14	15	16	Total in Category (1000's)
4	1	2,200	1,800	-0-	-0-	4
2	4	1,150	850	-0-	-0-	8
1	4	575	425	-0-	-0-	4
2	2	750	550	400	300	4
Total	11	-	-	-	-	
Total in Grade		10,600	8,000	800	600	20
					Grand Total	100

Table 6.1 School/Student Population Model

Another important characteristic of the schools is the distribution of enrollment among subject offerings. Since computer use is to be related to subject offering, data on computer use per student must rely on this basic enrollment distribution. For grades 9-12, the best source of the information is a report, Subject Offerings and Enrollments in Public Secondary Schools, HEW, 1965, OE-24015-61.

Table 6.2 shows enrollment in selected subjects as fractions of total students in grade. The subjects shown in this table are the ones used as the basis for computer use in this study.

Subject	9	10	11	12
Mathematics	1.08	.45	.40	.49
Science	.64	.80	.37	.38
Bus. Ed.	.45	.47	.49	.50
Ind. Arts	.23	.24	.25	.25
Programming	.35	.30	.25	.20

Table 6.2 Enrollments in Selected Subjects as Fractions of Total Students in Grade

In this table, fractions greater than unity occur because of scheduling anomalies. Subjects were selected on the basis of high relative enrollment and ease of computer involvement. The data for the subject "Programming" is not taken from the reference cited in the text, but is an additional subject assumed for the purpose of this study.

As a matter of interest, Table 6.3 is included to show the fractional enrollment in other high-enrollment high school subjects.

Subject	9	10	11	12
English	.98	.98	1.20	.98
Social Studies	.30	.69	1.20	1.07
Foreign Lang.	.26	.28	.29	.28
Health & Phys. Ed.	.84	.76	.60	.55
Music	.26	.28	.29	.29

Table 6.3 Enrollments in Selected Subjects as Fractions of Total Students in Grade

Enrollment data by subject offering are not available for grades 13 - 16. For this grade range, it was decided to use the February 1967 Report of the President's Science Advisory Committee, entitled Computers in Higher Education, popularly known as the Pierce Report, after the committee's chairman. That document contains an analysis of enrollment by major area of study and by degree of computer usage.

Several 13 - 16 institutions were visited during the survey portion of this study. Observations made at those institutions were generally consistent with the forecasts in the Pierce Report. In fact, the use of the forecasts was recommended during two of the visits.

Table 6.4 shows the Pierce Report data on percentage of enrollment at various levels of computer use.

	Casual	Limited	Substantial
Fraction of enrollment 13 - 16	.25	.40	.35

Table 6.4 Degree of Computer Use

The data in Tables 6.1, 6.2 and 6.4 will be used to characterize the member schools in this study. They are believed to represent a reasonable forecast for the period 1969-1970. In addition, the following specific parameters will be used:

	9 - 12	13 - 16
Hours per School-day	8	14
School-days per year	180	200
Use Time-factor	2 /3	4/5

Table 6. 5 Special Parameters

The use time-factor is the assumed proportion of the school-day in which student computer use can be expected. It is inserted to account for various types of lost time which are certain to occur.

The hours per school-day are admittedly greater than in current practice; it is believed that the increasing diversity of school facilities will induce a trend toward keeping the facilities open longer.

For grades 9 - 12, an additional characteristic of the schools will be estimated. Since the time period being considered is 1969-1970, it would be unreasonable to assume that all students in a given subject area will use the computer system. Penetration of computer use cannot occur so fast. To account for this, the "penetration factor" is introduced, equal to the fraction of students in a given subject area who will actually use the computer.

Estimated values of the "penetration factor" are given in Table 6.6.

Subject	9	10	11	12
Mathematics	0.7	0.8	0.9	0.9
Science	0.6	0.7	0.8	0.9
Business Education	0.5	0.5	0.5	0.5
Industrial Arts	0.4	0.3	0.2	0.1
Programming	1.0	1.0	1.0	1.0

Table 6.6 Penetration Factor

It is not necessary to estimate a "penetration factor" for grades 13 - 16, since the Pierce Report data on estimated computer use includes the effect of limited penetration.

6.3 Characteristics of Student Use

Student use of the computer system, for problem-solving and programming, will be treated in terms of numbers of problems, numbers of "sessions" (at a terminal), and numbers of runs. Also, it will be necessary to estimate various times, for example, average time (minutes) per session.

6.3.1 Grades 9 - 12

For grades 9 - 12, if the number of computer probler assigned per subject can be estimated, the estimates of course enrollment and penetration factor can be used to give average computer problems assigned per student/subject/grade unit. Both programming and problem-solving applications were observed and discussed during the survey, but the range of activity and opinion is very wide. The required estimates were made after considerable discussion of the range of possibilities. Table 6.7 gives the result.

Subject	9	10	11	12
Math, Science, Business Education, Industrial Arts	2	2	3	4
Programming	3	6	9	12

Table 6.7 Assigned Problems Per Course

Obviously, this data has little meaning without a measure of problem difficulty. One rough measure is simply comparative; the problems are assumed to be of the same type, and of somewhat greater difficulty, as typical problems now being solved by high-school students in the few instances where computers are now being used.

Another measure of problem difficulty will appear when estimates are made of sessions and runs per problem.

The data and estimates of Tables 6.2, 6.6, and 6.7 can now be combined to give Table 6.8.

Subject	9	10	11	12
Mathematics	1.51	.72	1.08	1.76
Science	.77	1.12	.89	1.37
Business Education	.45	.47	.74	1.00
Industrial Arts	.18	.14	.15	.10
Programming	1.05	1.8	2.25	2.40

Table 6.8 Average Problems Assigned Per Student/Subject/Grade Unit

For further use in the analysis, Table 6.8 can be condensed. There is no need to carry separate subjects beyond this point. By using the enrollment data once more, the condensation of Table 6.9 can be derived.

9	10	11	12
3.96	4.25	5.11	6.63

Table 6.9 Average Problems Per Student/Grade Unit

It is important to note that the entries in Table 6.9 are now overall averages. To illustrate this, it is easy to calculate the problem load for a specific student. A 12th grader, heavily involved in using the computer for mathematics and science, and taking a second course in programming, would have to solve twenty problems during the school year, or about three times the per-student average.

6.3.2 Grades 13 - 16

A different and simpler analysis is required for this grade range. The Pierce Report includes estimates of problems per student/category, as shown below.

Casual	Limited	Substantial
2.3	6.0	30.0

Table 6.10 Average Problems Per Student/Category*

A minor difficulty arises here, however. The Pierce Report took a view of higher education across grades 13 - 16 and beyond. The distribution of student population used in this study (Table 6.1) is deliberately arranged to exclude a high proportion of 15th and 16th graders. The data of Table 6.4, which gives the Fractions of 13 - 16 enrollment in the casual, limited, and substantial use categories, can not be applied with their original level of confidence to the specific 13 - 16 enrollment distribution used here. However, it is believed that the errors introduced by so using it will not be large; also, such errors will tend to be conservative, that is, they will tend to yield computer use estimates on the high side.

*The terms "Casual", "Limited", and "Substantial" are taken from the Pierce Report.

6.3.3 Combined Data for 9 - 16

Using the Pierce Report data, as discussed previously, yields a table of average problems per student/grade unit for grades 13 - 16, in which all table entries are equal. Combining this with the data for grades 9 - 12 (Table 6.9)gives the complete range shown in Table 6.11.

9	10	11	12	13	14	15	16
3.96	4.25	5.11	6.63	13.5	13.5	13.5	13.5

Table 6.11 Average Problems Per Student/Grade Unit

This completes the first part of the analysis of student use characteristics. A reasonable per-student computer problem load has been developed. The next task is to determine the computer service requirement which will be generated by student solution of these problems. This task must be done twice, once for a time-sharing system (keyboard terminals in the schools) and again for a batch-processing system (reader-printer units in the schools).

As a matter of interest, the weighted average of the data in Table 6.11 can be computed. The result is 6.6 problems assigned (for the school year) for the average of all students in the population of 100,000.

6.4 Computer Service Requirements

6.4.1 Terminal Capability: Time-Sharing System

The number of sessions at a time-sharing terminal estimated for grades 9 - 12 and the three categories of students in grades 13 - 16, respectively, are:

	Grades						
	9	10	11	12	C	L	S
Sessions/Problem	4	3.5	3	2.5	4	3	2
Terminal hours/Session	.55	.6	.65	.7	.45	.5	.55

Table 6.12 Estimated Usage Parameters

The figures for C, L, S (grades 13 - 16) are derived from the Pierce Report. The decrease in time required to de-bug and satisfactorily run a program in a time-sharing system can be explained by assuming that the heavier users have more experience, are therefore better programmers, hence require less time. In the Pierce Report, the C, L, and S users are assumed to require 2, 1.5, and 1 hours per problem, respectively. The equivalent values in this model are the products of the row elements, namely 1.8, 1.5, and 1.1 which are very close to the "Pierce times" for hours per problem. It is then reasoned that 9th graders, just beginning to program, will be less sophisticated than the casual users and require somewhat more time at the console for their programs, and that they would improve with increasing experience through the remainder of their high school days. Thus, the values for 9 - 12 are arrived at, yielding console-hours per problem of 2.2, 2.1, 1.9, and 1.7.

For the time-sharing case, enough information has now been assembled to permit an estimate of the terminal requirements. Data from Tables 6.1, 6.11, and 6.12 can be combined to produce Table 6.13.

	Grade								
	9	10	11	12	13	14	15	16	Total
Students (100's)	23.7	22.0	18.3	16.0	10.6	8.0	0.8	0.6	100
Terminal-hours per student per year	8.71	8.93	9.97	11.60	19.7	19.7	19.7	19.7	
Terminals	215	205	190	193	93	70	7	5	978
Student Terminal Ratio	110	107	96	83	114	114	114	120	102

6.13 Time-Sharing Terminal Requirements

Calculations of numbers of terminals are based on assumed hours/day and use factors given earlier.

Table 6.13 is interesting in two ways. First, it shows a very large number of terminals required, nearly 1,000, to serve the given population. There is no doubt that this number will be questioned because it is so large. The analysis preceding it shows which parameters could be changed to reduce the total. However, if the system scope and purpose originally stated by the Office of Education is to be maintained, it would not seem likely that the total terminal requirements can be reduced appreciably except by reducing the population to be served.

The second interesting feature of Table 6.13 is that the student-terminal ratio is about constant over the range of grades. This is the result of compensating trends in various contributing factors, and can be regarded as somewhat coincidental. At the same time, it provides a handy rule of thumb for evaluating variations in the assumed grade mix in the population model.

If the student-terminal ratio is applied to the original population model, the result is Table 6.14 which shows the required numbers of terminals distributed among the schools.

School Size (1000's)	Grade Range	No. of Schools	Total Students in Categories	Terminals Per School	Total Terminals
4	9 - 12	6	24	40	240
2	9 - 12	24	48	20	480
1	9 - 12	7	7	10	70
.5	9 - 12	2	1	5	10
4	13 - 14	1	4	35	35
2	13 - 14	4	8	18	72
1	13 - 14	4	4	9	36
2	13 - 16	2	4	18	36
				Total	979

Table 6.14 Distribution of Time-Sharing Terminals

6.4.2 Terminal Capability: Remote Batch System

For student use in a remote batch-processing mode, the processing parameters which must be estimated are number of problems (per student, as before), runs per problem, and terminal requirements per run.

The numbers of assigned problems will be taken from Table 6.11, the same data that was used for the time-sharing analysis.

There are several sources of data on number of runs per problem.* Analysis of this data indicates that an average of between four and six could probably be safely applied over the whole student population. However, there are several important factors relating to this average which should be discussed briefly.

It is difficult to find a correlation between the problem difficulty and the number of runs required for successful solution. One reason for this is that students working at the harder problems usually appear to be more diligent in checking a program before trying to run it. Another reason is that faulty logic, which is the source of program failure, seems to occur almost as frequently in the programming of easy problems as in programming difficult ones. These effects appear across the grade 9-16 without a clear correlation.

There is a correlation, however, between runs per problem and grade level, and this seems to follow from a more important correlation; namely, the relationship between runs per problem and the source language used for programming. Specifically, runs per problem average about three when a simple programming language like BASIC is used, and from five to seven when a more complex (and richer) language like FORTRAN is used. Specifically, in the following tables, runs per problem are estimated at three for the simple language case, and six for the advanced language case. These estimates should not be used to compare FORTRAN unfavorably with BASIC, of course; there are many compensations for this seeming disadvantage.

Students in the earlier grades of the range will tend to use the simpler programming language, while students in later grades will tend to use a more advanced language. Thus, the average runs per problem will be higher for a grade 15-16 student than for a student in the early grades.

Table 6.15 summarizes the results of these effects. In the table, estimates, based on judgment, are used for the proportionate use of the two languages as a function of grade level.

---

* Appendix 1, page 131.

	Grade							
	9	10	11	12	13	14	15	16
Proportion of problems in simple language	.9	.8	.6	.5	.4	.4	.3	.3
Proportion of problems in advanced language	.1	.2	.4	.5	.6	.6	.7	.7
Problems/student (from Table 6.11)	3.96	4.25	5.11	6.63	13.5	13.5	13.5	13.5
Runs per student in simple language	10.7	10.2	9.2	9.9	16.2	16.2	12.2	12.2
Runs per student in advanced language	2.4	5.1	12.3	19.9	48.6	48.6	56.7	56.7
Total runs per student	13.1	15.3	21.5	29.8	64.8	64.8	68.9	68.9

Table 6.15 Runs Per Student by Grade

Combining the data from Table 6.15 with the original student distribution, the total number of runs can be calculated.

	9	10	11	12	13	14	15	16	Total
Total Runs in Simple Lanugage (1000's)	253	225	168	158	172	130	10	7	1123 (40%)
Total Runs in Advanced Language (1000's)	57	112	225	318	515	389	45	34	1696 (60%)
Total Runs (1000's)	310	337	393	477	687	518	55	41	2819 (100%)

Table 6.16 Total Runs by Grade

Table 6.16, showing total runs required, gives one measure of the student-imposed load on the remote batch system. It is necessary, however, to carry the analysis further to establish the actual terminal requirements.

It would be possible to estimate an overall average terminal requirement per run. Since the difference between a simple programming language and an advanced one has already been introduced, however, and since it is a significant difference as regards to terminal requirements, the dual analysis will be continued through this step.

The data used earlier to estimate runs per problem* will be used again to estimate input-output requirements per run and per problem. Analysis of this data provides an estimate of about 2,800 characters for the average length of a program written in an advanced programming language. Although there would doubtless be a tendency for students in higher grades to write longer programs (in a given language), there is insufficient data to show this trend. The 2,800-character average will be used across the grade range.

For programs written in a simple language, current experience indicates that the average program will be about half as long as the advanced language average.

For purposes of analysis, the I/O requirements for student programs will be estimated as shown in Table 6.17.

To form the estimates in the table, an additional important assumption is made. This is that the normal run input will be equal to the program length in source language form, and that the normal run output will be equal to the program length in object language form, that is, in the language format produced by compilation during the run. Obviously, there will be many runs for which this assumption is not valid. But, experience with batch-processing systems indicates that the assumption gives reasonable results for I/O loading when most of the processing load is from solution of relatively short problems.

	Advanced Language	Simple Language
Avg. Program Length (Source Language)	2,800 ch.	1,400 ch.
Object Language Multiplier	5	8
Object Language Program Length	14,000 ch.	11,200 ch.
Runs per Problem	6	3
Total Input per Problem	16,800 ch.	4,200 ch.
Total Output per Problem	84,000 ch.	33,600 ch.

Table 6.17 Input/Output Requirements

* See Appendix 1, page 131.

It is now possible, of course, to calculate a grand total input-output requirement for the student population. To avoid very large numbers which would have little meaning, this calculation will be put in the form of input-output characters per hundred students per minute. In this form, the data can be extended easily to apply to any of the schools in the original distribution.

Table 6.18 shows input-output rates for student batch-processing load, in characters per minute per hundred students, and cards per minute and lines per minute calculated per hundred students with 30 average characters per card and 50 average characters per line of print.

	9	10	11	12	13	14	15	16
Input Rate	37.5	49.5	81.9	120.9	118.0	118.0	131.0	131.0
Output Rate	266.0	322.5	477.0	675.0	642.0	642.0	692.0	692.0
Cards/Min	1.25	1.65	2.72	4.02	3.93	3.93	4.36	4.36
Lines/Min	5.32	6.45	9.55	13.50	12.84	12.84	13.84	13.84

Table 6.18 Estimated Input/Output Rates

From Table 6.18, the next step is to calculate reading and printing loads for specific school sizes. The original distribution in Table 6.1 is used to give the results shown in Table 6.19.

Table 6.19 must be interpreted with care. As the sequence of calculations makes clear, the reading and printing loads are averages, and would apply for both readers and printers operating concurrently and continuously over the school day, except for lost time covered by the 2/3 and 4/5 schedule factors introduced earlier.

School Size (1000's)	Grade Range	No. of Schools	Total Students in Category (1000's)	Cards/Min. per school	Lines/Min. per school
4	9 - 12	6	24	90.5	331
2	9 - 12	24	48	45.0	218
1	9 - 12	7	7	22.5	109
.5	9 - 12	2	1	11.2	41.0
4	13 - 14	1	4	157	514
2	13 - 14	4	8	78.6	257
1	13 - 14	4	4	39.3	128
2	13 - 16	2	4	81.6	264

Table 6.19 Reader/Printer Loads by School Size

With this in mind, the assignment of reading and printing capacity to the schools would have to be made quite liberally.

This completes the analysis of terminal requirements for student use of a remote batch system.

6.5 Administrative Data Processing

6.5.1 Introduction

The usefulness of the computer as an administrative tool is currently being demonstrated throughout the country. However, because of differences in the needs and inclinations of individual administrators, the administrative activities which have been mechanized and the degree of mechanization vary from school district to school district.

The activities described in this section do not exhaust the possibilities. They do, however, comprise that set of activities from which most administrators have made selections for mechanization.

Almost all administrative applications of computers require the editing, storage, retrieval, manipulation, and display of vast quantities of data. In any particular application all of these operations may not be necessary. But all applications involve the ordering (sequencing) of records in files. In the educational context a student record may consist of his name, address, sex, date of birth, parents' names, and other pertinent information. An orderly arrangement of a group of such records comprises a file. A distinction is often made between a master file, which contains relatively permanent or unchanging data, and a working file which contains data which change frequently or which need to be stored for relatively short periods of time.

6.5.2 Administrative Data Files

The administrative functions covered in this section are best described in terms of the various files of data involved and the activities centering on the storage, retrieval, up-dating, and use of file data. Tables 6.20 through 6.28 give the major file usage parameters for the more important data-processing activities.

For each activity: a) the data file(s) associated with it is (are) described; b) the series of file transactions comprising the activity are spelled out; and c) the procedures associated with each transaction are described. The required user input and computer output of data are described in terms of the numbers of records, punched cards, and lines of print which comprise the system loads. (The inputs do not include the control cards required to cause the execution of a transaction program.)

A master file is created which contains pertinent student data. Each student record is approximately 4000 characters in length during the student's senior year. The size of the file is a function of the number of students enrolled in a school.

Transaction	Input	Output	Description
File Update	1 set of cards/student update, or 1 record/student update	(See Description)	Card(s) or tape are read. Student record is updated. Log data is printed. (Number of lines depends upon the type of record update.)
File Print	1 record/student	40 lines/student	Dump of master file.
Report Generation	1 record/student	3 lines/student	Retrieve data (one pass thru master file). Write working file. Process working file. Print.

Table 6.20 Student Master File

REPORT CARDS

A working file containing grade reporting data is created from the Student Master File. The size of the working file is a function of the number of students enrolled in a school. The record length is approximately 900 characters. The file is used for the following transactions:

Transaction	Input	Output	Description
Marking Document Generation	1 record/student	10 cards/student	A set of mark sense cards, one card per course, with pre-punched identifying data is generated for each student.
Mark Input & Generation of Verification List	10 cards/student	10 lines/student	Mark sense cards are read and edited. Grades recorded in file. List of marks with editing comments for each section is prepared for teacher verification.
Corrections	1 card/correction	1 line/correction	Correction cards are read. Errors in working file are corrected. Log entry is printed.
Print Report Cards	3 records/student	20 lines/student	Using student master file and report card and attendance working files as input, print report cards. (Master file is used for address info if report cards are mailed.)
File Maintenance	1 card/change	1 line/change	Correction and/or change cards are read and file updated. Log entry is printed.
Master File Update & Gummed Label for Permanent File	1 record/student	10 lines/student	At end of school year, data in working file is summarized and recorded in Student Master File. Gummed labels for inserting summary data in "hard copy" permanent files are generated.

The following lists are prepared from the report card working file:

Failure/Incomplete List	1 record/student	1 line/failure or incomplete
Honor Roll	1 record/student	1 line/honor roll
Rank in Class	1 record/student	1 line/student
Mark Distribution	1 record/student	20 lines/teacher, or 1 report/request

Table 6.21 Report Cards

A working file for containing attendance data is created from the Student Master File. The size of the working file is a function of the number of students enrolled in a school. The record length is approximately 100 characters. The file is used for the following transactions:

Transaction	Input	Output	Description
Generation of Reporting Document	1 record/student	1 card/student	A mark sense card with pre-punched identifying data is generated for each student.
Control & Daily Bulletin	1 card/absentee	4 lines/absentee	Cards of students who are not in normal attendance are read. Bulletins are printed. (4 different orders: for use by teachers, guidance counselors, central office and special reports.) Attendance working file is logged.

Data on working file is summarized and used to update student master file and to print the following reports:

Register Sheet	1 record/student	1 line/student
Summary/Student	1 record/student	1 line/student
District Statistics	1 record/student or summary records	1 line/student or 1 report/request
Gummed Labels for Permanent File	1 record/student	2 lines/student

Table 6.22 Attendance

The size of the working file created for this activity is a function of the number of students enrolled in a school. The record length used to record student request is approximately 80 characters.

Transaction	Input	Output	Description
Creation of Working File	1 card/student	1 record/student	Cards submitted by students indic .c-ing course requests are read and file created.
Edit & Verification	1 record/student	10 lines/student	File is edited for following errors: prerequisite, loading, etc. File is printed with edit comments for verification by students and guidance counselors.
Course Lists	1 record/student	1 line/requestor	File is searched to determine requestors of designated courses, e.g., advanced orchestra. Lists are printed.
File Maintenance	1 card/change	1 line/change	Correction and/or change cards are read and file updated. Edit performed. Log entry is printed, with edit comments.
Tally	1 card/course	1 line/course	Working file is expanded to include course catalog data by reading input cards. Expanded tally of course requests is determined. Tally is printed.
Cross Tally	(working file)	30 lines/course	Cross tally of requests is determined. Results are printed.
Creation of Master Schedule Sub-file File Maintenance	(I-O dependent upon type of master scheduling)		Read cards creating sub-file describing master schedule. Correction cards are read as required.

Table 6.23 Scheduling (cont. on next page)

Transaction	Input	Output	Description
Assignment Simulation	(working file)	15 lines/conflict	Simulate student assignments. Print conflicts.
Student Assignment	(working file)	10 lines/student	Generate tape of student assignments. Tape is printed.

Student assignment tape may be sorted to generate the following printed lists. Each list requires an unique sort.

Class List	1 record/student	1 line/student/course
Home Room List	1 record/student	1 line/student
Study Hall List	1 record/student	1 line/student/study hall

Student assignment tape is used to update student master file. Master schedule sub-file is used to update personnel master file with teacher assignment and load data.

Table 6.23 Scheduling (cont. from last page)

Transaction	Input	Output	Description
Document Preparation	1 record/student	2 cards/student	Student Master File is used to create sets of mark sense cards for use as test answer forms. Cards pre-punched with identifying data.
Scoring	2 cards/student	1 record/student	Answer cards are read. Working file created. Size of record will depend on amount of data to be retained.
Report Generation	1 record/student	1 line/student	Test results are edited. Test results with edit comments are printed. (Results of additional processing included when requested.)
Gummed Labels for Permanent File	1 record/student	2 lines/student	Gummed labels for inserting standard test results in "hard copy" permanent files are generated.
Make-up List	2 records/student	1 line/absentee	Using as input, student master file and test working file, generate list of student names not taking test. Print list.
Item Analysis	2 cards/student or 1 record/student	200 lines/test	For teacher made test, an item analysis may be requested. Analysis performed. Results printed.

Table 6.24 Testing

A master file is created which contains pertinent employee data. Each employee rocord is approximately 1200 characters. The file size is a function of the number of employees per school.

Transaction	Input	Output	Description
File Update	1 set of cards/ updated record or 1 record/employee	(See Description)	Card(s) or tape are read. Employee record updated. Log data is printed. (Number of lines depends upon the type of record update.)
File Print	1 record/employee	15 lines/employee	Dump of master file.
Report Generation	1 record/employee	3 lines/employee	Retrieve data (one pass thru master file). Write working file. Process working file. Print.
Substitute Teacher List	1 record/employee	5 lines/substitute teacher	Master file is processed to determine substitute teachers status data. List is printed.

A sub-file containing personnel leave data is created for the Personnel Master File. Each record is approximately 100 characters in length. The size of the sub-file is a function of the number of employees per school. Sub-file is maintained as changes are made to the Personnel Master File.

Transaction	Input	Output	Description
File Update	1 card/entry	2 lines/entry	Leave (sick, personal, vacation, etc.) data is input by card. Working file record is updated. Log data is printed.
Payroll Summary	1 record/employee	1 record/employee	Data is summarized for a pay period. Summary record is input for payroll program.
Report Generation	1 record/employee	3 lines/employee	Summary data may be used to print reports.

A sub-file containing payroll data is created for the Personnel Master File. Each record is approximately 300 characters in length. The sub-file size is a function of the number of employees per school. Sub-file is maintained as changes are made to the Personnel Master File.

Transaction	Input	Output	Description
Document Generation	1 record/non-exempt employee	1 card/non-exempt employee	Generate mark sense time-cards with pre-punched identifying data. Used for reporting non-exempt employees payroll data.
Payroll Input	1 card/non-exempt employee + 1 card/exception to regular exempt personnel status + 1 card/employee assigned to special project	1 record/employee + 1 line/input card	Read non-exempt time cards. Read cards reporting exceptions to exempt employees pay records. Read card reporting "special project" time data. Print verification lists.
Payroll Computation	2 records/employee (payroll + leave records)	1 line/employee	Compute wages and deductions. Record current wages and deductions and update year to date data. Edit, using control parameters provided by payroll clerk. Print edit comments.

Table 6.25 Personnel Master File (cont. on next page)

Transaction	Input	Output	Description
Check Generation	1 record/employee	1 check/employee or 1 bank deposit form entry/ employee and 1 line/employee	Print checks and/or earning statements. (Print data necessary for direct bank deposits, when requested by employees.) Print list of check numbers with check data for payroll clerk.
File Update	1 card/entry	2 lines/entry	Adjustments to payroll records are read. Record updated. Log data is printed.
Report Generation	1 record/employee	3 lines/employee	Reports (W-2, FICA, etc.) are generated as requested.
Budget Summary	1 record/employee	1 record/costing center	Write working file for updating budgetary accounting file.

Table 6. 25 Personnel Master File (cont. from last page)

A master file is created which contains pertinent inventory data. The record length is approximately 300 characters. The size of the file is a function of the number of items carried in the inventory. File is initially set up for the control of consumable items. Following working files are created: On-order file, Loading file, Order file, Holding file and Costing file. A sub-file containing pertinent historical data will be maintained.

Transaction	Input	Output	Description
Receiving - Update	1 card/order + "On-Order working file" + adjustment cards	(See Description)	Cards are read indicating receipt of order. (Item adjustment cards are included. Receipt of partial order indicated with control cards.) On-order working file is updated. Inventory file is posted. "Reserved item" indicators are set. Holding file maintenance performed. Log data is printed. Update Encumbrance Suspense File and generate voucher.
File Print	1 record/item	1 line/item	Dump of inventory file.
Adjustment-Update	1 card/adjustment	1 line/adjustment	Read adjustment card. Update record. (Deletion & insertion of items included.) Log data is printed.
Issuing-Update	1 card/item requisition	(See Description)	Read requisition card. Edit for clerical and control purposes. Determine if item is available and take appropriate action: (1) Enter shipping data in loading file and update inventory file; record costing data in working file for budgetary accounting; "reserved item" indicators are updated when applicable; history sub-file is updated; print action performed data; or,

Table 6. 26 Inventory Master File (cont. on next page)

Transaction	Input	Output	Description
			(2) enter request data in order file and print form notifying requestor of items which will be ordered; enter request data in holding file; or, (3) print form notifying requestor of items currently on-order; enter request data in holding file.
Loading & Delivery Lists	1 record/item	2 lines/item	Generate loading and delivery lists from "loading" working file.
Holding File Maintenance	1 record/entry	(See Description)	Maintenance performed during Receiving-Update. Holding file is processed to determine if request can be filled. If item available, action performed as indicated in (1) of Issuing-Update description.
Inventory	1 record/item	1 record/item to be ordered + 1 line/item to be ordered	File is checked to determine if inventory is below established minimum and item is not on order. When applicable, enter request data in "order" working file. List of "short" items is printed.
Print Order File	(working file)	1 line/item/ requestor	Generate detailed item list (by school or department) for administrative review.
Adjustment Order File	1 card/adjustment	2 lines/adjustment	Read adjustment cards. Generate report to requestor affected by adjustment. Print log data.
Bid Request	(Order working file) + (Vendor Data file)	1 set of forms/ request	Bid request are generated when applicable.

(Purchase Order generation is described in following activity-Accounts Payable)

Table 6.26 Inventory Master File (cont. from last page)

A master file is created for accounts payable data. Working files for the following functions are created: Encumbrance Suspense File, Voucher Suspense File and a check request file. A Vendor Data File is created.

Transaction	Input	Output	Description
Purchase Order Generation	(Order Working file) + (Vendor Data file)	1 form/request + 1 record/request	Generate Purchase Order. Encumbrance is entered in account records (with distribution in designated cost areas.) Entry made in Encumbrance Suspense File. On-Order working file (as described for use with the Inventory activity) is updated. Items deleted from order file. Log data is printed.
Encumbrance Suspense	(Encumbrance Suspense File)	1 line/update	Set indicators to tag encumbrances for which items or services have been recieved. Print log data.

Table 6.27 Accounts Payable (cont. on next page)

Transaction	Input	Output	Description
Voucher Generation		1 form/request + 1 record/request	Voucher is prepared. Entry made in Voucher Suspense File. Accounts Payable File is updated.
Voucher Reports	(Voucher Suspense File)	1 form/request	Generate voucher register and reports for board upon request.
Pay Voucher	(Voucher Suspense File)	2 records/voucher + 1 line/action	Enter record in working file used for writing checks. Update Voucher Suspense File, Encumbrance Suspense File, and Accounts Payable File. Update Budgetary Accounting File. Encumbrances are reversed. Print log data.
Check Generation	1 record/check	1 check/request	Print checks. Print list of check numbers with check data for payroll clerk.
File Adjustments	(Accounts Payable File & working files) + adjustment cards	1 line/adjustment	Adjust records as requested. Print log data.

Table 6.27 Accounts Payable (cont. from last page)

BUDGETARY ACCOUNTING FILE

Budget accounts (for cost distribution and expenditure control) are created. These records are updated as indicated in the description of the activities pertaining to inventory, payroll and accounts payable. Accounts payable updates are input with cards directly to the budgetary accounting file.

Transaction	Input	Output	Description
Update Accounts	(working files) or cards	1 line/entry	Accounts are updated (includes encumbrance procedures and adjustments). Log data is printed.
Report Generation	(Master file with associated working files)	1 report/request	Retrieve data. Write working file. Process working file. Print, both standard and special reports.

Table 6.28 Budgetary Accounting File

Tables 6.29 and 6.30 present two examples of the daily loading of the central computer system. The first presents a hypothetical loading for a day during late January. The second shows a hypothetical loading for a non-peak day.

Requests for system service are identified using the Activity and Transaction designations described in previous tables. (College registration, not previously described, is similar to the scheduling activity.) A request for the execution of a transaction refers only to the procedures described in the tabular description of the transaction. For example, Print Report Cards refers only to the actual final processing and printing and not to the acquisition of the data to be printed.

Number of Requests indicates the total number of requests from all of the schools using the system.

Input/Request and Output/Request indicate the number of cards to be read or punched and the number of lines to be printed for each request. The amount of data which will be retrieved from a master or working file is not indicated. The chief purpose of these two columns is to specify the requirements to be met by the card reader, card punch, and line printer. The input device is assumed to be a card reader capable of reading both punch and mark sense cards.

The sizes of the files processed are shown in the last column.

The values for input/output and file size are based on a high school with an average enrollment of approximately 2,000 students. The values for the number of requests and the set of requested transactions are hypothetical.

Example of Daily Loading
for
Late January Peak Day

Activity	Transaction	Number of Request	Input/ Request	Output/Request	Processed File Size/Request
Attendance	Daily Bulletin	40	200 cards	800 lines (4 reports)	40,000 words
	Generation of Reporting Document	25	*	2000 cards	1.3 million words
Testing	Scoring	100	200 cards	1 record/student	4,500 words
	Report Generation	100	*	120 lines	4,500 words
Report Cards	Print Report Cards	5	*	40,000 lines	300,000 words
Student Master File	File Update	5	*	8,000 lines	1.3 million words
	Report Generation	10	*	6,000 lines	1.3 million words
College Registration		1	5000 cards	50,000 lines	67,000 words
Scheduling	Student Assignment	1	*	20,000 lines	27,000 words
Personnel Master File	File Update	40	*	400 lines	20,000 words
Payroll	Check Generation	20	*	1,000 lines	5,000 words
Accounts Payable	Voucher Generation	20	*	50 lines	45,000 words
	Purchase Order Generation	20	*	50 lines	60,000 words

* Input Data Contained in File Storage. Run request input with set of control cards.

TABLE 6.29

Example of Daily Loading
for
Non-Peak Day

Activity	Transaction	Number of Request	Input/ Request	Output/Request	Processed File Size/Request
Attendance	Daily Bulletin	40	200 cards	800 lines (4 reports)	40,000 words
Testing	Scoring	50	200 cards	1 record/student	4,500 words
	Report Generation	50	*	120 lines	4,500 words
Student Master File	File Update	5	*	8000 lines	1.3 million words
	Report Generation	2	*	6000 lines	1.3 million words
Personnel Master File	File Update	1	*	400 lines	20,000 words
	Leave Record File Update	40	5 cards	10 lines	2,000 words
Accounts Payable	Voucher Generation	20	*	50 lines	45,000 words
	Purchase Order Generation	20	*	50 lines	60,000 words
Inventory Master File	Issuing-Update	30	1 set of cards/ requisition (set 20 cards)	30 lines	90,000 words

* Input Data Contained in File Storage. Run request input with set of control cards.

TABLE 6.30

6.5.3 Peak Activity Periods

Figure 6.1 roughly shows the peaking conditions of administrative applications during the year. It is not drawn to scale so that the heights of peaks for two different activities cannot be compared. The purpose of the chart is to indicate the major activity periods during the year. Four periods of high activity can be identified. They are associated with the opening weeks of the school year, the weeks encompassing the close of the fall semester and the start of the spring semester, the closing weeks of the school year, and the weeks of late August when much of the scheduling activity is in progress.

The creation of the master files for a school district is essentially a one-time operation. The amount of input for a transaction varies as the previous descriptions indicated, but in the majority of cases the data to be processed is already contained in either a master or working file. Also, the computer processing time required for accomplishing a transaction is dependent upon the hardware configuration. For example, the size of the memory and its allocation will determine the block size to be used in processing the administrative files.

Each of the activities described in this section generates a vast amount of printed data. The output requirements placed on the system by the administrative applications are critical in determining what configuration can meet the requirements of a system which is oriented toward student use with administrative applications in a secondary role.

From the standpoint of printing requirements, Figure 6.1 and Tables 6.29 and 6.30, which further point out the differences between peak and non-peak conditions, may be summarized as follows:

(1) During peak conditions, a central printer with an effective speed of 1,000 lines per minute may be utilized between 8 and 16 hours per day. This assumes that all printing requests are executed at the central facility. Using the central printing facility also assumes the existence of an effective messenger service between the central office and the local school user.

(2) During peak conditions, a local printer with an effective speed of 100 lines per minute may be utilized between 12 and 24 hours per day. This assumes all of the printing is local, i.e., there is no use of the central printer.

(3) For a non-peak day, the central printer facility may be utilized between 2 and 4 hours per day.

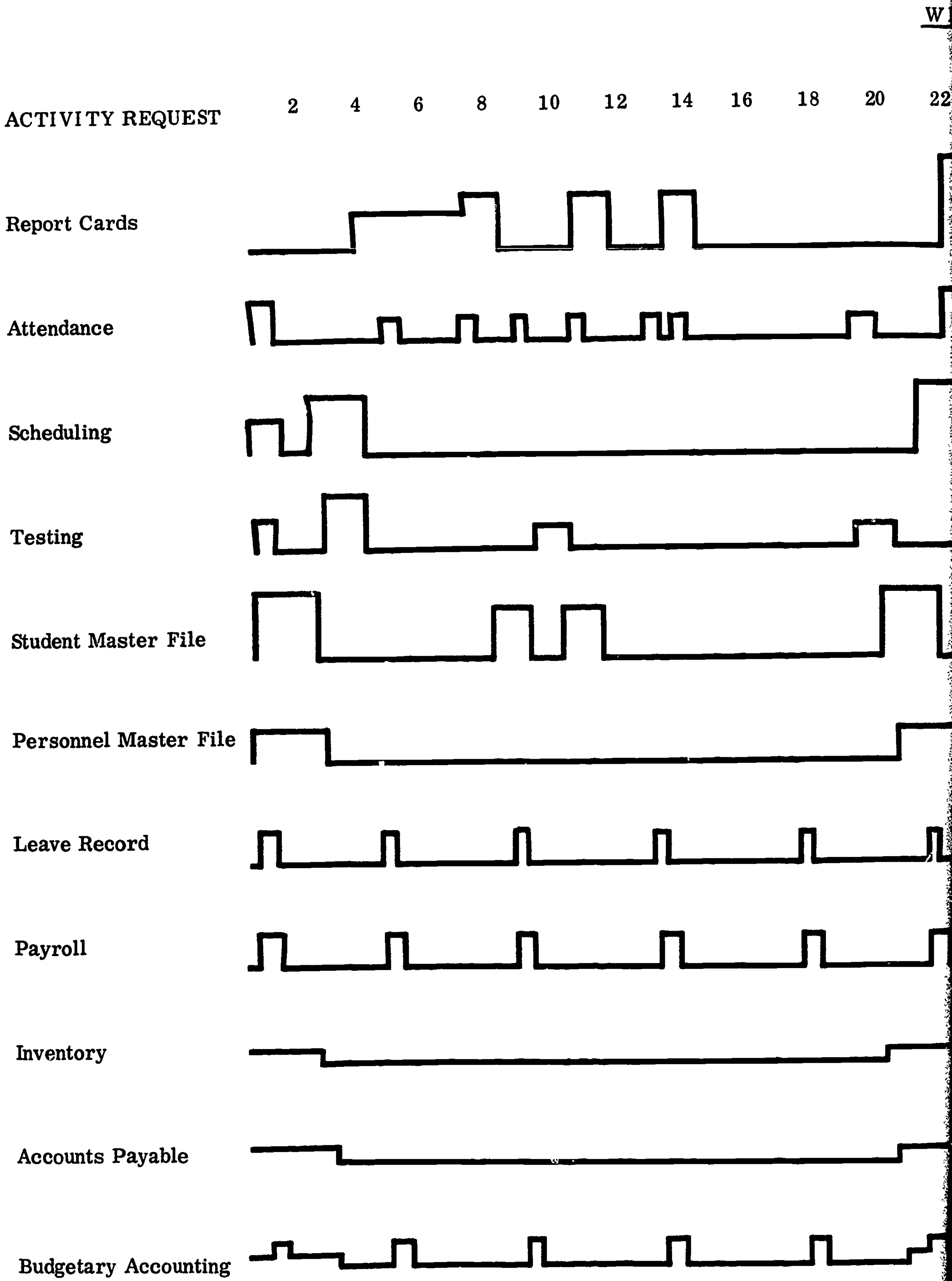
(4) For a non-peak day, a local printer may be utilized between 3 and 6 hours per day.

Assuming a system which provides for high volume printing to be accomplished at the central facility and low volume printing to be executed at the local school, the following additional summaries may be hypothesized:

(5) During peak conditions, the central printer may be utilized between 5 and 9 hours per day while a local printer may be used between 3 and 6 hours per day.

(6) For a non-peak day, the central printer facility may be utilized for less than 1 hour and a local printer between 3 and 6 hours.

The significant difference between the printing requirements of peak and non-peak conditions suggests the possibility of acquiring additional printing capacity during the four peak periods of the year. Assuming its availability, printing time may be purchased from other government or industrial computer facilities as needed.


W
ACTIVITY REQUEST
2 4 6 8 10 12 14 16 18 20 22
Report Cards
Attendance
Scheduling
Testing
Student Master File
Personnel Master File
Leave Record
Payroll
Inventory
Accounts Payable
Budgetary Accounting

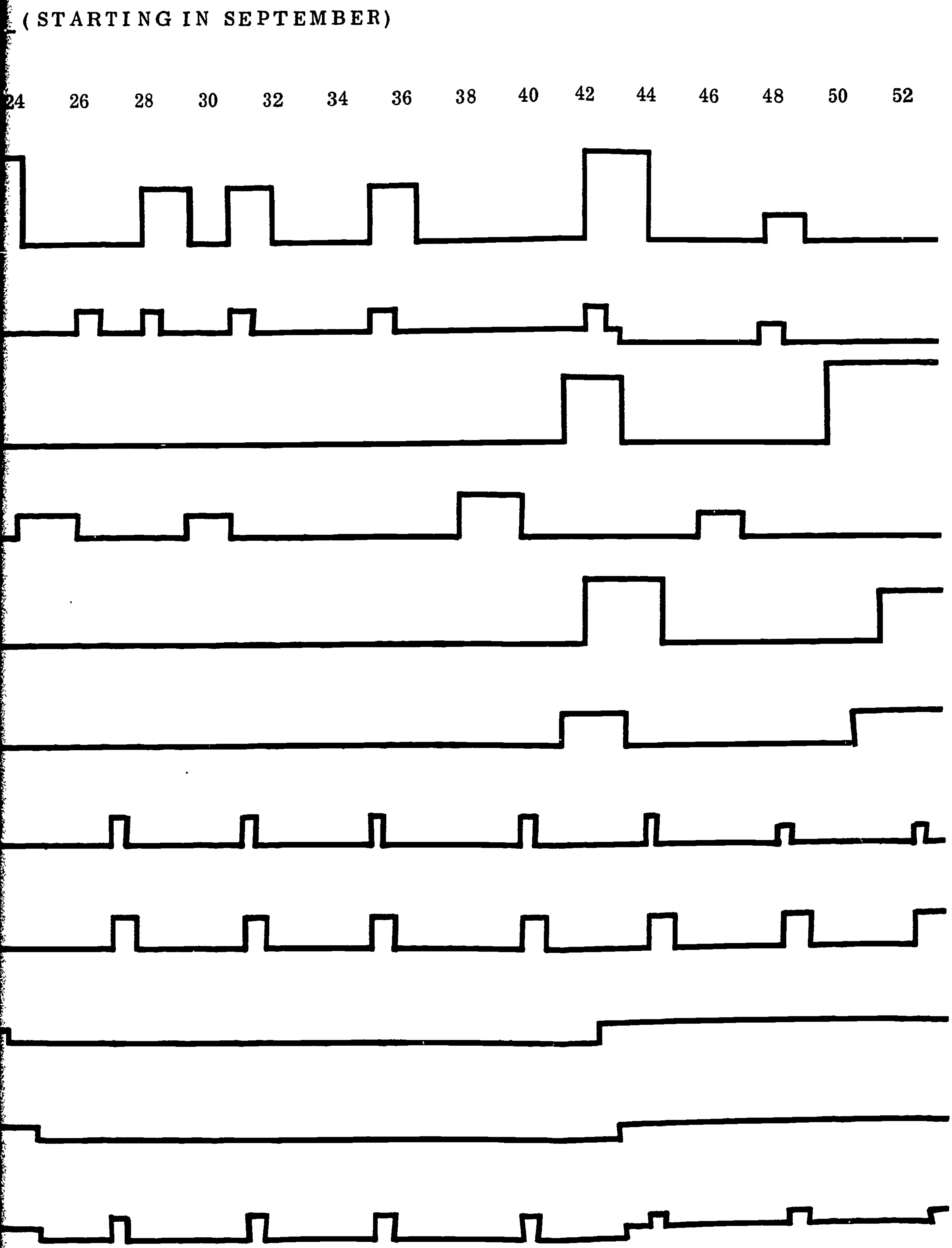


Figure 6.1 Peaking conditions of administrative applications.

6.6 Storage Requirements

6.6.1 Student User Requirements

Each student user of the system will require four types of information storage within the system:

(1) Working storage for his program and data during compilation and execution of his program.

(2) Long-term storage for his programs and data between sessions at the terminal.

(3) Working storage for compilers and utility routines needed during the execution of his program, plus storage for buffers and other internal system functions.

(4) Permanent storage for the library of utility programs and subroutines which are expected to be used with individual programs.

In this list, the first two types are distinctly individual; the system must provide this storage for each system user. The first type must be provided only for the number of users actively working at terminals at one given time. The second type must be provided for all users all the time.

The third and fourth types of storage can be provided on a common basis; that is, compilers, utility routines, and the subroutine library need not be stored separately for each user. Even in the case of a compiler being executed by an active user (storage type 3), the use of re-entrant compilers will make duplicate storage unnecessary.

Total storage requirements can be estimated by considering each of the four classes of storage.

6.6.1.1 User Program/Data Storage

Individual user storage requirements will vary over a wide range. Experience with one group of about one hundred students, grades 11-12, showed a range of from 200 to 500 six-bit characters of storage required for individual programs, including storage reserved for data. The average of these programs was about 1,600 characters. A student population covering grades 9-14 would probably produce a somewhat greater average program length. It is reasonable to require the system to provide 3,000 characters of program and data storage for each student user, with the understanding that this is an expected average, and that the system must accommodate longer programs up to a maximum length of about 10,000 characters. Programs exceeding this maximum length could be handled by the computer facility, but not during time-shared operation.

For the time-sharing system, the amount of long-term student program/data storage required (type 2) is simply the average per user, estimated at 3,000 characters, times the number of users. The number of users is actually a fraction of the total student population determined. It seems safe to assume that about 40% of the student population would be users during a semester or half-semester time period. This would make the type 2 storage requirement 120,000,000 characters.

In the remote batch system, student programs and data will be maintained in card files at the school locations. No long-term storage of this class is required centrally.

The amount of working storage required during compilation and execution (type 1) is a more difficult quantity to estimate. It depends heavily on the overall design approach taken for the central computer and its associated operating system programs. The best current guide in this area is the experience to date in installed time-sharing and remote batch systems. This experience is reflected in the tables and discussion of this storage requirement in Section 7. Only the result is given here.

For remote batch operation, the type 1 storage requirement would be less than for time-sharing operation. The amount of reduction will be significant, even though the need to store fewer programs concurrently is partly offset by the tendency for programs to be longer when written for this mode. This requirement is also analyzed in Section 7.

6.6.1.2 Storage for Compilers and Utility Routines

Considerable experience has been gained in estimating storage requirements for the compilers and utility routines which are in current use. However, the programs which will perform these functions in time-shared systems will be more complex because of the requirement that these programs be re-entrant. From a functional point of view, the programs in this category which must be stored are as follows:

Language Processors:
- FORTRAN
- BASIC

Catalogs:
- Math. routines
- File-manipulation routines
- Application programs
- User program abstracts

Operating System Support Programs:
- Peripheral equipment control
- File-to-peripheral
- File-to-file
- Scheduling algorithm programs
- Special diagnostics
- Maintenance log
- Billing log & processor

The actual amount of storage required for these programs is a matter of software design. A reasonable overall estimate, to cover the list, would be 20,000,000 characters.

In addition to the program functions listed, additional storage must be provided in the remote batch-processing system. Basically, this additional requirement is for buffer space, and it is, in a sense, a substitute for long-term storage of user programs and data. It will add about another 20,000,000 characters to the type 4 storage requirement.

The type 3 requirement for working storage for active programs must be estimated as before on the basis of current practice.

Bearing in mind that the time-sharing system will require multiple active compilers (at least 2, possibly 3), as well as extensive buffering and control areas, it seems wise to estimate this requirement at about 800,000 characters. Even this amount might be considered too low on the basis of experience with current systems. However, it is reasonable to expect some improvements in system programming techniques, before the 1969-1970 anticipated operational period, which would tend toward economy in this type of storage.

For remote batch operation, the same considerations apply as those used for user programs and data. The number of routines to be stored concurrently will be smaller than for the time-sharing system. The reduction will tend to be offset by the larger routines which will accompany the more complex user programs which are expected in the remote batch system, but the net effect will be to reduce the requirement by about one-half.

Table 6.31 summarizes the estimated storage requirements.

	Time-Sharing System	Remote Batch System
Program/data Working Storage	200,000	140,000
Program/data Long-Term Storage	120,000,000	-0-
Utility Working Storage, Buffers, and Control	800,000	400,000
Utility Library	20,000,000	40,000,000

Table 6.31 Student User Storage Requirements in Characters

6.6.2 Administrative Data Storage Requirements

The tables in Section 6.5 contain estimates of the size of individual files in the administrative system. Combining these estimates for the entire population gives total file sizes required. In addition to the files, storage will be needed for application programs to do the various administrative jobs. This storage requirement is small, however, relative to the files themselves; it can be considered to be included in the utility library discussed in the previous section.

Table 6.32 lists total major file storage requirements.

File Name	Size in Characters
Student Master File	400,000,000
Report Cards	90,000,000
Attendance	10,000,000
Personnel Master	8,000,000
Inventory Master	600,000
Accounts Payable Vendor Data Budgetary Accounting	500,000

Table 6.32 Major Administrative File Sizes

7.0 Design Synthesis

This section presents the sequence of design considerations and conclusions which have as their result the system recommendations given earlier. This is a user-oriented study, and the design process is user-oriented. Specifically, this orientation means that first consideration is given to the known characteristics of system users, and their need to communicate with the central facility. Thus the first system element considered is the communication subsystem between individual schools and the central location. The second element is the large number of terminals which will be placed in the schools.

After terminal and communication facilities have been treated, the central computer and its associated devices will be specified. Finally, some statements will be made about the programming system which will be necessary to allow the entire assemblage of equipments to work as an effective system.

Cost considerations will appear in the design sequence. The final listing of cost estimates will appear in the next section.

7.1 Technical vs. Economic Feasibility

The requirements summarized in the previous section exceed the capabilities of any known, currently installed, system. However, several organizations are planning or developing systems which are designed to meet or exceed these requirements in the 1969 - 70 time period. It is believed that the system designs developed in this report are technically feasible.

The design effort takes cost into account in that minimum subsystem configurations are selected which will perform at the required performance levels. However, no pre-selected cost target is used. Thus, while the economic parameters of supplying the required services have been estimated, it is not to be inferred that the specified system is clearly feasible economically.

7.2 General Design Approach

It is a design requirement that the equipment parameters specified be within the capabilities of equipment manufacturers' standard units. This requirement has been met. However, the state-of-the-art in multiprogramming systems and in time-sharing is advancing rapidly. It seems reasonable to anticipate this advancement, and to assume that the system under design could derive significant performance benefits from it.

Another element of the design approach is to take advantage of the nature of the educational environment which may simplify the functional requirement. As compared with multiprogramming systems in other environments such as research laboratories or large business operations, an educational environment will usually permit clusters of identical terminals to be located together. Also, the storage allocated to individual students can be smaller than that required by users of more general systems.

Finally, the design parameters are kept in functional terms as much as possible, so as to encourage creativity in later implementation proposals.

7.3 Communication Lines

The approach given above has a very specific meaning in the design of communication facilities for a time-sharing or multiprocessing system. With a few exceptions, all current and planned time-sharing systems use the facilities of a common carrier for point-to-point communications. There simply is no economic alternative.

This constraint solves a major design problem, in that the selection of communication line parameters is narrowed down to the relatively few standard line types available. The load estimates given earlier indicate that the basic communication medium should be a dedicated (leased) line, of the conditioned (4kc) type. Dial access from terminals would not be satisfactory because of heavy terminal loading and the loss of time in dialing. A single conditioned line will handle the multiplexed transmission to and from a terminal group, which is nominally 20 terminals of the keyboard type or one reader-printer combination.

Some schools will have combined transmission rates sufficiently high to justify the use of Telpak circuits if considered on an individual basis. However, the design of the communications interface equipment at the central location will be simplified if all incoming lines are of the same type. For this reason, and also because the transmission distances are relatively short, Telpak is not recommended.

Since no specific region is identified, it is not possible to determine the desirability of WATS (Wide Area Telephone Service). However, the specified area, with an approximate diameter of 100 miles, is so small that the WATS tariffs probably could not be applied. This particular service arrangement will have to be analyzed if a region is selected for implementation which includes portions of more than one state.

7.4 Terminals

The basic transmission medium having been selected, the next step is to determine whether suitable terminal facilities can be interfaced to that medium. The time-sharing and remote batch-processing systems must be treated separately since the primary difference between them is in the terminal equipment each will use.

7.4.1 Time-Sharing Terminals

The design choice here is between an automatic typewriter terminal and a CRT display with keyboard. The choice is a difficult one because the number of different types of display devices available is increasing, along with the types of control and communication interface arrangements offered. The automatic typewriter is recommended, primarily on the basis of cost and because it provides printed output.

In the time-sharing system being specified, the number of terminals required is very large, approaching 1000. The cost of terminals will be one of the largest items in the total system cost, even if the least expensive terminal devices are used. Every consideration must be given to keeping the cost per terminal down. The lowest terminal cost can be achieved with a unit such as the Teletype Model KSR 35 or ASR 35, or the IBM 1050. Moreover, as the time-sharing market grows, it is highly probable that lower-priced devices with characteristics similar to these units will become available, possibly including CRT consoles, the prices of which are steadily dropping.

If the prices for CRT terminals are reduced to be competitive with typewriter-type terminals, the CRT's should definitely be considered. However, in the educational environment, hard copy output is mandatory for both administrative and problem-solving applications. There are several ways to augment a CRT-terminal system to provide hard copy:

- A low-speed line printer can be added to the terminal. This would satisfy the need best, but adds substantially to the cost per terminal.

- A printer can be provided for each group of terminals at a given location. Each user decides when a copy of displayed information is needed and requests it. The task is assigned to the local printer (through software); the user proceeds with other activity and the hard copy is available approximately five minutes later to be picked up or delivered to the user, depending on procedures. This method is used in some business systems. It would probably satisfy administrative requirements in the school environment and it has the feature that the local printer could be used for batch-processing output at the school location during non-problem-solving periods. It is clearly not the optimal arrangement for the problem-solving user, but it is probably the best compromise considering the advantages of CRT display and the hard copy requirement.

- A third alternative is to provide line printing at the central location and to deliver hard copy to users via courier or mail. This approach has been made to work in some specialized management systems, but it has obvious disadvantages which eliminate it from consideration here.

It may seem strange that fast advancing technology has not yet provided a reasonably priced, integrated CRT/Keyboard/Printer. It is certainly possible that such units may be available in the 1969 - 72 period; if so, their use in an educational system should be evaluated. In the absence of such devices, however, the most attractive CRT terminal system would include a local printer to serve a group of terminals.

Despite the desirable visual properties of the CRT terminal, its greater cost and lack of hard copy output demands that the recommendations be given to the simple typewriter terminal.

For use by students and faculty, for teaching programming, and for support of academic work, the adequacy of typewriter terminal I/O rates has already been demonstrated by currently operating time-sharing systems. The output rate, of about 10 characters per second, sometimes causes delays during highly interactive use. Improved designs will certainly provide higher output rates. However, the 10 cps rate is acceptable, and to increase it significantly will lead to problems in multiplexing these terminals.

For administrative data processing, the picture is entirely different. Typewriter terminals cannot handle this load.

One example will serve to illustrate the problem. For report card printing, calculations were made to determine what typewriter terminal facilities would be required. For the average-size school in the region under study (2000 students), four typewriter terminals would be tied up for an entire twenty-four hour period. This is clearly unfeasible. Furthermore, report card printing is but one of several peak load requirements during the school year.

If a typewriter-terminal time-sharing system is to be implemented, other means will have to be provided for transmission of administrative data. This is not an obstacle; several groups of schools are already sharing computer facilities for data-processing using couriers and regular mail service. The region under study is sufficiently compact geographically to make courier service fast and economical. It would cost about $ 30,000 per year. Compared with other costs in the total system, this is relatively small.

7.4.2 Multiplexing for Time-Sharing Terminals

Since terminals in this system will be grouped at school locations, the use of multiplexing is an obvious way to reduce the cost of data transmission to and from the central computer. To minimize cost, the selection of specific multiplexers should be based on the actual distribution of terminals among the various locations (Table 6.14). As shown in Section 6, the number and distribution of terminals is very sensitive to the assumptions made regarding individual student use. If a level of student use different from that described in Section 6 is used in subsequent plans for implementation, then the multiplexing scheme recommended will have to be reexamined.

A reasonable design goal is to have as many schools as possible served by a single voice-grade leased line, but to avoid inefficient use of expensive multiplexers at schools having only a few terminals.

For the loads and student distribution assumed, more than half of the schools will have twenty terminals or more and the minimum number of terminals at a school will be five. Thus, a multiplexer which will accommodate twenty (20) terminals will meet the design goal.

The Western Union DALCODE (Data Line Concentrator/Deconcentrator) is such a device. It can receive data on a real-time basis from up to thirty (30) teletypewriters, concentrate it, and transmit it over a 4kc line via data sets to a computer. It consists of one or two concentrator/deconcentrators mounted in a single cabinet. The second unit, when used, provides redundance. The concentrator combines the teletyped data into a single high-speed output channel for transmission over a voice-grade channel. Only one DALCODE system is required when coupled to a computer interface. More detailed information on the DALCODE equipment is provided in Appendix 5, "Transmission System".

The identification of a specific equipment for multiplexing is not intended as an exclusive recommendation. The requirements are not severe; other equipments are available in the same performance range. Availability of the equipment from the common carrier is an added advantage, however.

Table 7.1 shows the distribution of time-sharing terminals, multiplexers, and leased lines among the schools.

School Size	Grade Range	No. of Schools	Terminals per School	Lines per School	Total Lines
4	9 - 12	6	40	2	12
2	9 - 12	24	20	1	24
1	9 - 12	7	10	1	7
.5	9 - 12	2	5	1	2
4	13 - 14	1	35	2	2
2	13 - 14	4	18	1	4
1	13 - 14	4	9	1	4
2	13 - 16	2	18	1	2
Total lines and multiplexers					57

Table 7.1 Distribution of Communication Equipment

7.4.3 Remote Batch-Processing Terminals

7.4.3.1 The Remote Reader

The reading device selected will be very much faster as an input data source than keyboard input. This device will be used for reading in relatively large amounts of information, such as user programs or associated files of data. These programs and data will be generated (in most cases) at local sites.

A choice must be made between reading punched cards, punched paper tape, and typewriter documents. All three reading approaches have advantages. Optical reading equipment accepts the source data in a form most convenient to system users. Individual documents can be interpreted visually and some hand manipulation is feasible locally. Source data preparation uses conventional typewriters, although special fonts may be required. Relatively high cost of the reading equipment is the main disadvantage.

Punched paper tape is the least convenient medium from the user's standpoint. It is difficult to read visually and difficult to manipulate locally. Its only advantage is the relatively low cost of equipment to prepare it manually, read it, and punch it as system output. Off-line equipment used to prepare it is relatively inexpensive and the visual reading disadvantage is partly overcome by the fact that the preparation device may produce a typed copy concurrently with the tape.

Punched cards are the most common media for source data. Cards are easy to handle and, through the use of an interpreter, selected portions of a card can be printed for visual reading. Card handling equipment ranks between a tape reader and a document reader in cost.

Keeping in mind the various uses to be made of this system, card reading equipment is recommended.

For maximum flexibility, the card reader and controller should have the following characteristics:

Speed: 100 cards per minute minimum

Reading:

- Full 80-column character field
- Binary field
- Coded marks on card front
- Coded marks on card back (not mandatory)

7.4.3.2 The Remote Printer

Analysis of the output loads at individual schools (Table 6.19) indicates that only two of the remote locations will require a line printer with a speed greater than 300 lines per minute. In some cases, higher speeds would permit more efficient handling of administrative data-processing; higher effective speed may also be obtained from multiple units. The line printers used should print at least 120 characters per line, with a character set of 60 characters or more.

7.4.3.3 Keypunch Units

Since cards are proposed as the basic input medium for batch-processing, it is necessary to provide keypunch equipment. It is difficult to analyze potential student use of keypunches; it would be advisable to install a reasonable number and to be prepared to modify that number when the actual load is observed. As an overall average, one keypunch unit can probably serve about 400 students in the specified environment.

The distribution of remote batch-processing units will consist of one 100 card-per-minute reader and one 300 line-per-minute printer, with controllers and communications interface at each member school, except for the seven 4000-student schools (six 9 - 12 and one 13 - 14). Each of these large schools will have two sets of this same equipment. Also, each school will have a number of keypunch units determined by the school size. In effect, this means that the same distribution of communication lines shown in Table 7.1, for the time-sharing system, will apply to the remote batch systems as well.

7.5 Central Computer Facility

This section describes two designs, one which will handle the time-sharing terminal load, and one which will handle the remote batch-processing load. In a number of respects, these designs are quite similar. The main differences are in the amount and type of memory which is required, and the required central processor speed.

In general, the relationship between desired system performance and the various design parameters is extremely complex for a time-sharing system or a remote batch system. The design parameters established in this section will be tested by simulating the operation of the system; the simulation is described in Section 9.

7.5.1 Time-Sharing System: Processing Requirement

It is quite difficult to estimate the computing power required to handle a given large number of terminals. Experience with time-sharing systems indicates that the traditional measures of computer speed, such as cycle time or average time per instruction, are of questionable value for this purpose.

The matter of computing power may be approached in a general way by considering the total number of terminal interactions the system will be required to handle. For example, it is observed that, on existing time-sharing systems, the mean time between a user's receiving a response and his next request is about thirty seconds. This period, called dead time, varies over a wide range, from practically zero to many minutes. If the 30-second mean is used, however, it can readily be calculated that the system proposed here will have to process a request about every 30 milliseconds. At first consideration, this may not appear extremely difficult. Modern large-scale computers can process thousands of instructions in 30 milliseconds. However, these same computers cannot, generally, retrieve information from large files in this short time, because of the delay which occurs in the storage devices used to store the large files. Also, if a request involves the compilation of a program in FORTRAN, for example, thirty milliseconds may be short indeed. Modern compile speeds reach 200 statements per second and higher, but even at this speed, thirty milliseconds would allow compilation of only six statements.

Clearly, it is extremely important to know the kinds of requests being made from time-sharing terminals. Also, it is important to use a system design approach which permits the central computer system to conduct several operations concurrently without mutual interference. Finally, it is important to use, in the system, a processor , storage devices, and other devices whose performance is as high as current technology permits.

Because of the type of load being put on the system, it is very probable that compilation rate will be a limiting performance factor. As a design requirement, it seems reasonable to specify a very high compilation rate. Simulation of the specified system will show whether such a rate is actually required.

The 200 statement per second compilation rate mentioned earlier is attainable with current technology. By 1969 or 1970, significant improvement can be expected. Also, this rate assumes a FORTRAN language richer than the student users should require. Based on these considerations, a design goal rate of 500 FORTRAN statements per second will be set.

Compilation rate is, of course, a software-dependent system parameter. It is vital to the performance of this system, however, and is a much more meaningful expression of processor speed than cycle time or average instruction execution time.

7.5.2 Remote Batch-Processing System: Processing Requirement

This system presents a less difficult problem than the time-sharing system, as far as computer speed estimates are concerned. In this case, Table 6.15 and Table 6.16 provide some direct information on numbers of runs and program length.

The greatest load on the system will occur during the 8-hour period when all school installations are operating. The load during this period can be calculated from Table 6.15:

$$\text{9 - 12 Load} = \frac{(310 + 337 + 393 + 477)\ (1000)}{(180)\ (8)\ (2/3)\ (60)}$$

$$= \text{26.3 runs per minute}$$

$$13 - 16 \text{ Load} = \frac{(687 + 518 + 55 + 41)(1000)}{(200)(14)(4/5)(60)}$$

$$= 9.7 \text{ runs per minute}$$

$$\text{Combined Load} = 36.0 \text{ runs per minute}$$

To get an estimate of required computer speed, it will be assumed that all runs have the average input amount, which is 70 cards*, that all runs require compilation of the source deck, and that one-third of the runs require full execution of the program. Data from Appendix 1 gives the estimate that execution time is about 75% of compilation time for student programs.

The required effective compilation rate can now be calculated:

$$\text{Compilation Rate} = (36)(70)\left[(2/3)+(1/3)(1.75)\right]$$

$$= 3150 \text{ cards (statements) per minute}$$

$$= 52.5 \text{ statements per second}$$

This result is interesting, first because it is so different from the estimate given for the time-sharing case, and second because it is well within the capabilities of existing computer systems. It should not be inferred from this that the remote batch system simply handles the student load much more easily. Compilation rate was the determining factor in both estimates; if the required computer speed comes out lower for the remote batch system, the correct inference is that this system is required to do less compiling. In other words, it is expected that the average student user will request fewer compilations using the remote batch-processing equipment as compared with the time-sharing terminals.

To take some account of peak periods, a design goal of 100 statements per second will be set for the remote batch central computer system.

---

* 2100 characters at 30 characters per card.

7.5.3 Time-Sharing System: Core Memory Requirement

The high-speed memory requirement cannot be derived from the functional analysis. Many of the functions which high-speed memory provides in a time-sharing system develop out of various internal requirements, such as buffering, storage of system commands and tables, storage of the control program itself, and so on. These requirements have been rather well developed in various developmental time-sharing projects. Table 7.2 summarizes the requirements for the proposed system.

Table 7.2 shows, in addition to the design requirement, a memory allocation which could be used for a less extensive system. These lower estimates are based on a system which would serve about 400 terminals (referred to as a minimum system), rather than the 1,000 terminals specified in Section 6.

Function	Design Goal Storage Required characters	Minimum System Storage Required characters
Control Program	300,000	200,000
Time-Sharing Commands	60,000	60,000
Terminal Buffer Area (256 char. per term)	240,000	100,000
User Ident. Table	100,000	40,000
Resident Compilers	100,000	100,000
User Prog. Work Area	200,000	100,000
Totals	1,000,000	600,000

Table 7.2 Time-Sharing System Core Memory Requirements

7.5.4 Remote Batch-Processing System: Core Memory Requirement

The same comments apply for this memory requirement that were made for the time-sharing system. Table 7.3 summarizes the requirements, based on estimates derived from current experience with similar systems.

Function	Design Goal Storage Required characters
Control Program	200,000
Terminal Buffer Area (512 char. per terminal)	30,000
User Ident. Table	60,000
Resident Compilers	100,000
User Prog. Work Area	140,000
Total	530,000

Table 7.3 Remote Batch-Processing System Core Memory Requirements

7.5.5 Time-Sharing System: Secondary Storage Requirement

Based on the number of users and terminals specified, Table 7.4 shows the secondary storage requirements for the time-sharing system.

Data Base Name	Avg. Record Size (thousands of characters)	Max. No. of Records	Storage Requirement (millions of characters)	Storage Medium
System Library	2.0	-	2.0	drum
Swap File	2.0	1000	2.0	drum
Current File	2.0	1000	2.0	drum
Users Catalog File	2.0	7500	15.0	disk
Users Saved Prog. File	2.0	60,000	120.0	disk
Total Drum Storage Required			6.0	
Total Disk Storage Required			135.0	

Table 7.4 Time-Sharing System Secondary Storage Requirements

7.5.6 Remote Batch-Processing System: Secondary Storage Requirements

Based on the number of users and the number of reader-printer units specified, Table 7.5 shows the secondary storage requirements for the remote batch-processing system.

Data Base Name	Avg. Record Size (thousands of characters)	Max. No. of Records	Max. Storage Rqmt. (Disk) (millions of characters)	Avg. No. of Records	Avg. No. Storage Rqmt. (Drum) (millions of characters)
System Library	2.0	-	-	-	1.2
Current File (System In)	2.0	10,000	20	200	0.4
Swap File (System Out)	2.0	10,000	20	200	0.4
Total Drum Requirement					2.0
Total Disk Requirement			40		

Table 7.5 Remote Batch-Processing System Secondary Storage Requirements

7.5.7 Secondary Storage for Administrative Files

In addition to the storage requirements stated in the preceding sections, both the time-sharing and the remote batch systems will require secondary storage at the central location for administrative files. These requirements were discussed briefly in Section 6. Table 7.6 gives a more complete statement of this requirement.

Data Base Name	Record Size	No. of Records	File Size (millions of characters)	Storage Medium
Student Master File	4,000	100,000	400	disk
Report Card File	900	100,000	90	disk
Attendance	100	100,000	10	disk
Scheduling	80	100,000	8	disk
Personnel Master	1,200	6,700	8	disk
Inventory Master	300	2,000	0.6	disk
Accts. Payable Vendor Data Budgetary Acctg.	- - -	- - -	0.5	disk
Total Disk Storage Requirement			517	

Table 7.6 Administrative File Storage Requirement

7.5.8 Additional Secondary Memory

The secondary memory should include magnetic tape. Tape is the most economical means of storing and using very large files of administrative data, and is efficient for direct interaction with the processing unit for applications in which sequential processing is used.

The number of tape units to be included can be relatively small, since so much storage and file manipulation capability is provided by the disks.

The magnetic tape subsystem should have the following characteristics:

- 4 tape units, with 2 x 8 controller, so that additional tapes may be added later. The controller must allow for direct data transfer between magnetic tape units and the following other system elements:

    - o the disk memory
    - o the controllers for the central peripheral devices

- Transfer rate: 50,000 - 100,000 characters per second

- Density: 200, 556, and 800 bits per inch, 1600 bits per inch desirable.

7.6 Central Facility Peripheral Equipment

In the time-sharing system large volumes of input and output data (primarily administrative) will be handled at the central facility, and may be transported to and from member schools by courier or mail. High speed card readers, printers, and card punches will be located at the central facility. A suitable complement for this subsystem is:

Card Readers:	2 units, 800 - 1200 cards per minute
Printers:	3 units, 800 - 1200 lines per minute
Card Punches:	1 unit, 200 - 300 cards per minute

In the remote batch-processing system, almost all administrative data will be handled at the in-school reader-printer stations. Some central reading-printing-punching equipment will be needed, however, both as back-up for individual schools and for the use of the system operating staff. For this, one each of the units described above should suffice.

7.7 Special Communications Unit (SCU)

Of major concern is the equipment at the central facility which will receive and transmit data over the large number of leased lines. The demands on this equipment are very high, in terms of existing time-sharing or remote batch-processing systems. In fact, current system designs usually use a separate processor for this function. In terms of today's approach, this system element could well be termed a special communications processor. However, as time-sharing systems become more common, equipments may be developed specifically for this type of service.

Since this equipment must be considered a special item, the most effective way to specify it is to list the functions which it must perform:

- Buffering. The SCU will receive and transmit single characters over the leased lines. This data must be buffered in the SCU for the composition of messages which may be up to 60 characters in length for the time-sharing system, or up to 150 characters in length for the remote batch system.

- Synchronization. Characters coming in from the lines will be identified from timing relative to a synchronizing bit. During buffering of these characters to message length, an address must be generated to accompany the transmission of the message to the core memory (or other central system element). Data coming from the central system must be identified from address code, and synchronized, character by character, for transmission over the leased lines.

- Error Detection. The SCU must recognize invalid codes and parity errors and send appropriate error messages to the terminals.

- Line Servicing. The SCU must perform the above functions for up to 60 leased lines, each having a maximum transmission rate of 2400 bits per second.

7.8 System Programming Requirements

7.8.1 Operating System Structure

The operating system program, sometimes referred to as the executive program, has the responsibility for continuously monitoring and controlling the operation of the system. The degree of control that actually takes place in the computer will vary depending on system requirements, but "housekeeping", accounting, and control are absolutely necessary.

- Supervisor

The supervisor determines the sequence in which various user programs are executed. Each user is allotted a fair share of available processor time and sufficient space in memory. The supervisor contains the scheduler which assigns priorities, forms, queues, optimizes the use of the system, and controls interactions to allow jobs to proceed on time with minimum delays. The scheduler also records the amount of system resources expended by each user.

- Command System

The command system examines the input from the user terminal looking for calls expressed in user-oriented language. Messages from users are translated and converted to calls for procedures which are routines that cause work to be done on the user's input. These procedures could cause messages to be transmitted to the user, a program to be called into memory from storage, the proper compiler or translator to be made available, or data to be processed.

- File Control

File control should free the user from concern over the physical location in the system of his stored data. Only a reference to a name should be necessary to make his programs or data available when required. It also protects data from accidental destruction, and use by non-authorized persons. In addition, file control keeps statistics on file use and moves the more frequently used files to faster access storage, and the less frequently used ones to slower access storage. The establishment of common files would also be possible, as well as the sharing of files by authorized users. A directory of existing programs and files will be maintained by the system indicating file name, physical address, time and date the file was created or last modified, when last referred to, number of times referred to, and status as to its availability to others or only to one user.

- Input/Output Control

The I/O control performs all the necessary communication and controls among the remote devices, peripherals, and the central processor. Necessary code conversions, terminal queuing, buffering, and error recovery are handled here.

7.8.2 Important Characteristics

Various possible approaches are available to meet the requirements of the proposed systems. A more detailed description of specific user needs would be necessary before describing the one that would best fulfill requirements. Various methods of multiprogramming, multiprocessing, program-swapping, and memory-sharing are in operation or in development stage. It is possible to do time-sharing and/or multiprogramming to one degree or another on almost any of the newer product lines available from computer manufacturers. A brief discussion of some of the many factors that need evaluation follows.

- Swapping and Queuing

In order to achieve the maximum degree of efficiency of which today's computing systems are capable, it is necessary that more than one program be available for execution and that computing and I/O operations overlap. As programs and data are submitted by users, a queue indicating the status of each must be built. A priority scheme may also be established, which gives precedence to important work requiring access to the control processor over nonessential work. In many systems, programs are kept in the state of execution only for relatively short time periods; they get "swapped" between auxiliary storage and the processor memory. This allows each user to get his fair share of computation time. When a program is "road blocked" by lack of input, it is swapped out until sufficient input is received to continue. The processing of swaps does, however, require some computer time and increases system "overhead".

- Memory Allocations

Attempts are being made to decrease the number of swaps required by putting two or more programs in memory at once and alternating time slices between them. Swaps are generally used only if there is no input or if the output buffers fill up. In order to use memory better, such techniques as dynamic relocation, fragmentation, paging, and list processing have been developed.

- Scheduling

The scheduler should handle the allocation of processor time, forming of queues, and the establishment of priorities. The use of the system is optimized by mixing programs of various types to fully utilize the capacity of the system.

- Error Control

The detection or correction of certain types of errors can be under computer control. It is a matter of personal opinion and cost as to how much control should be built into the hardware, how much into the software. In addition to checking for many system malfunctions, users' programs can be diagnosed in order to "flag" possible mistakes. The "flags" or error messages displayed to the user add to the system load; the more extensive the error checking, the more complex the programming for software. There is usually a trade-off between cost and practicability.

- File and Storage Requirements

The key to rapid turnaround time will be the amount and efficiency of use of the data storage capacity available both in the processor and secondary media. The amount of processor storage is a vital factor affecting multiprogramming efficiencies and the frequency of swaps required. Secondary storage affects access time to private files, common files, and stored data. Currently, the most common secondary storage medium used is the magnetic disc.

However, some of the larger systems use combinations of high-speed drums; intermediate-speed, larger capacity disks; and slower access, large capacity magnetic card files. Programs and data are moved from one medium to another depending on demands for and frequency of use.

- Memory Protection and Security

It is imperative that when operating in a time-sharing or multi-programming environment that a user's program not be allowed to "clobber" any other information in the system. Most hardware is currently equipped with means which prevent programs from leaping out-of-bounds. Various file lockout and file-protect devices are also available on some peripherals. Software-implemented safety features include checking user numbers and allowing access only to files for which that user is cleared. In critical applications, system back-up facilities are often required to maximize storage protection and minimize downtime.

7.9 Additional Design Considerations

7.9.1 Mixed Keyboard-Reader-Printer Subsystem

During the discussion of CRT - Keyboard terminals, the possibility arose of using a local printer to provide hard copy for a group of terminals. If this were to be done, the further addition of, say, a paper tape reader/punch would give a mixed terminal subsystem. Such a subsystem would have, at least from the I/O equipment standpoint, interfaces for both time-sharing via individual terminals and batch processing.

While such an approach leads to some overlapping in functions, it also has advantages which should not be overlooked. The mixed system offers students both access methods, and there is a clear benefit in learning the mechanics of both modes of use. While there is no adequate substitute for the individual terminal for teaching programming in a high-level language, so there is no substitute for medium speed (at least) readers and printers.

7.9.2 Local Processing Considerations

In any remote batch-processing system, the desirability of local processing capability arises. Although not analyzed during the study, it is appropriate to list some trends or conditions under which this type of system can become the most effective approach. Local processing capability should be considered if one or more of the following conditions arise:

- Increased administrative data-processing using individual school files,
- Increased geographic dispersal of user schools, with consequent increases in communications costs, or
- Increased system use for support of business-subject courses.

7.9.3 Error Rate Considerations

Some thought was given to the design of a model for calculating detected and undetected error rates. This is of somewhat questionable value for several reasons. First, the bit error rate is under the control of the common carrier. Second, a complex burst error analysis would be required, which uses the Mertz* burst error model, for example. Third, the choice of modems and their internal design becomes of importance. Finally, the consequences of an occasional error in this system are not disastrous. From an economic standpoint, it is difficult to justify forward error correction in the system design. Therefore, it is proposed that the approximate message error rates (detected and undetected) be calculated manually after the preliminary design is complete. Since the error rate is primarily under the control of the common carrier, an unsatisfactory message error rate will dictate a change in message code structure.

7.9.4 Synchronization

It can be assumed that clocking will be obtained from the computer so that subscriber terminal transmission will be synchronized from the receiver clock. Consideration must also be given to the problem of synchronization loss after a temporary line failure. Ultra-high stability clocks at subscriber terminals can probably not be justified. The solution likely lies in a combination of coding plus line monitoring and re-synchronization initiated from the computer station.

7.10 Design Summary

Design information on the complete time-sharing and remote batch-processing systems has been developed in the preceding sections. This section summarizes the information. Figures 7.1, 7.2, and 7.3 are schematic diagrams of the time-sharing system, the remote batch-processing system, and the central computer system, respectively. The tables following these diagrams give the various design parameters as developed earlier.

---

* Mertz, Pierre. "Model for Error Burst Structure in Data Transmission." Proceedings of NEC, Vol. 16, 1960.

Table 7.1, shown earlier, gives the distribution of communication lines and multiplexers. The distribution of time-sharing terminals and remote reader-printer units was given in Section 6. These will not be repeated in this section.

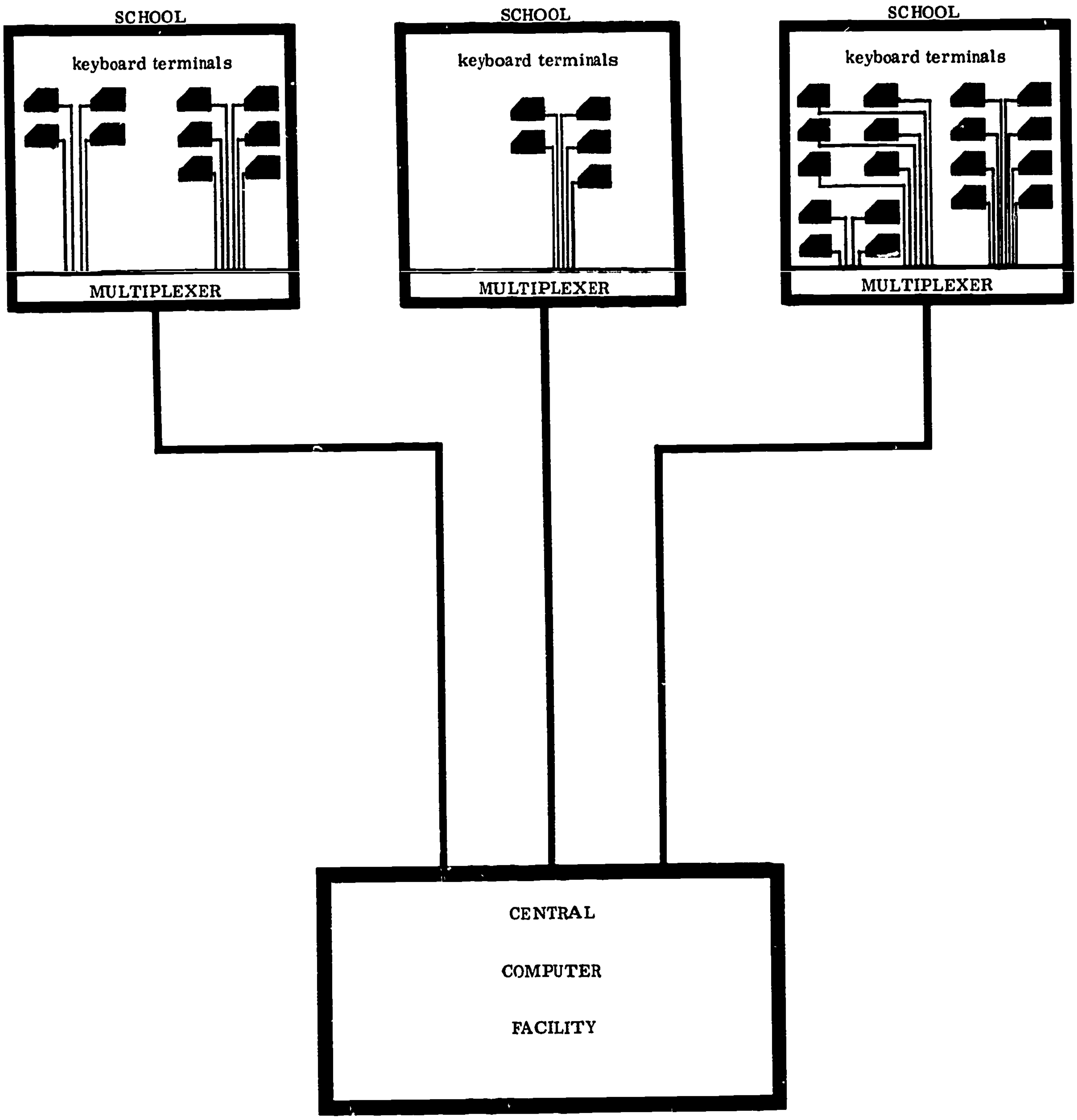


Figure 7.1 Time-Sharing System

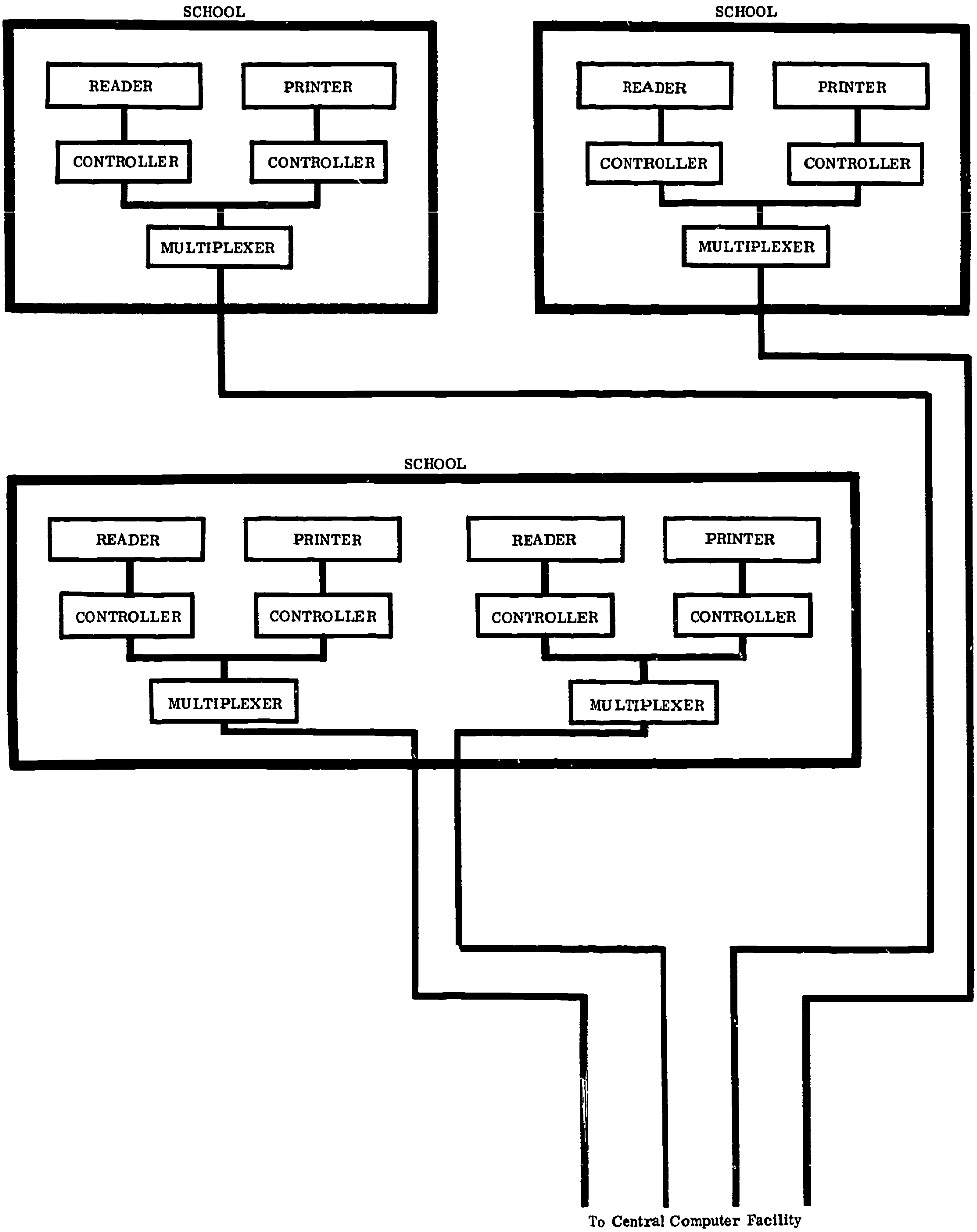


Figure 7.2 Remote Batch-Processing System

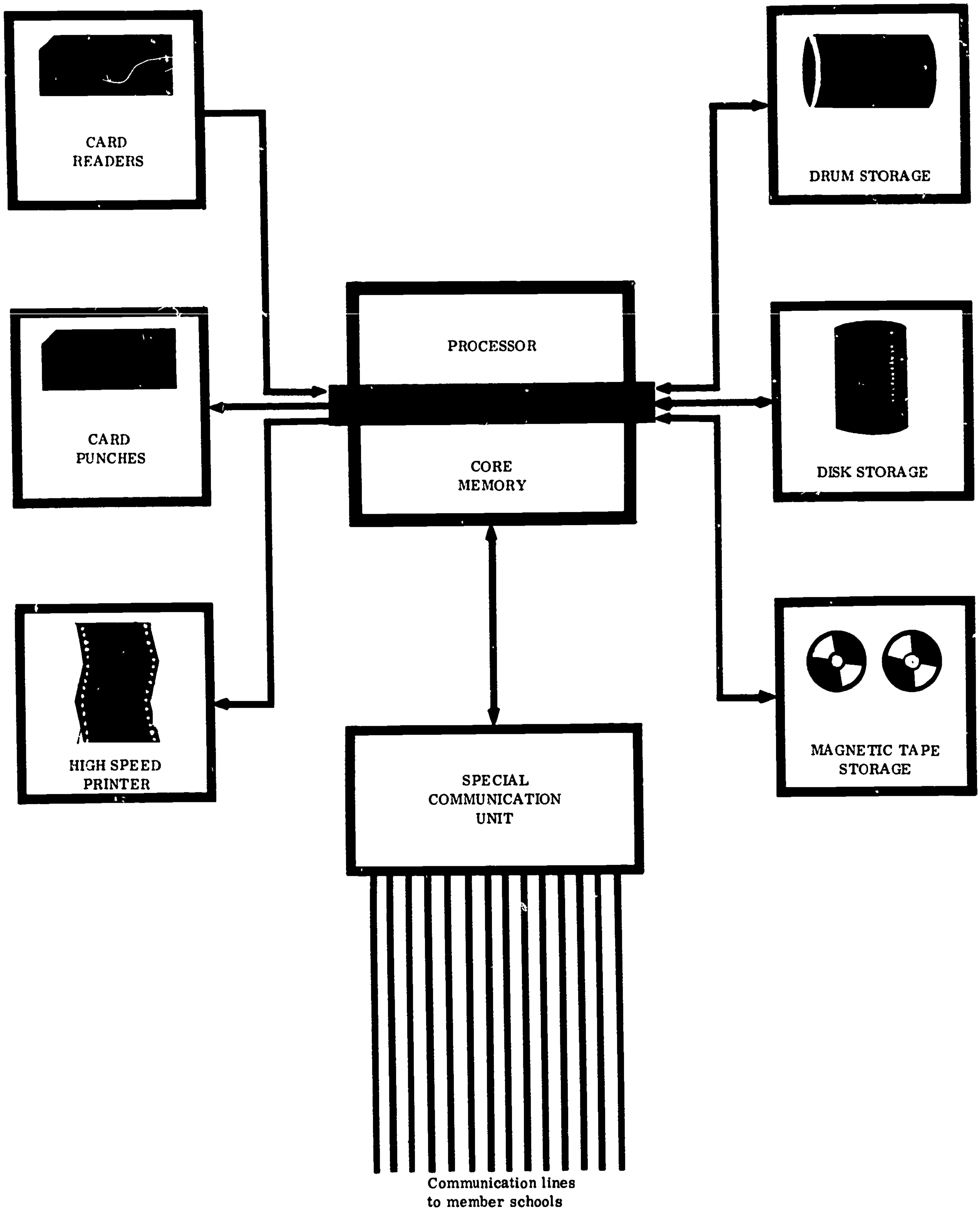


Figure 7.3 Central Computer System

System Element	Quantity	Model or Performance	Comments
Time-Sharing Terminal	978	Typewriter terminal Teletype ASR 35 or equivalent	Distributed as shown in Table 6.14
Multiplexors	57	Will multiplex 20 (minimum) 10 char/sec keyboard terminals to one 2400-bit/sec. (4kc) communication line	Distributed as shown in Table 7.1
Communication Lines	57	4kc conditioned voice-grade lines, average length 30 miles	Distributed as shown in Table 7.1

Table 7.7 Time-Sharing System Terminals Communication Equipment

System Element	Quantity	Model or Performance	Comments
Key-Punch Units	250	Standard Keypunch, IBM — or equivalent	Distributed approximately one per 400 students
Card Reader & Controller	57	100 cards/min	Distributed as described in Para. 7.4.3
Line Printer & Controller	57	300 lines/min	"
Communications Interface Unit	57	Will interface one read controller and one print controller to one 2400 bit/sec (4kc) communication line	"
Communication Lines	57	4kc conditioned voice-grade lines, average length 30 miles	"

Table 7.8 Remote Batch-Processing System Terminal & Communication Equipment

System Element	Quantity	Model or Performance	Comments
Central Processor	1	Large-scale computer, required speed given by 500-statement per second FORTRAN compilation rate	See Para. 7.5.1
Core Memory	1 million characters	Access time not specified, must match processor capability	See Para. 7.5.3
Secondary Memory			
Disk Units	3 - 4	Capacity: 150 - 200 million characters per disk unit Access Time: 150 - 250 milliseconds average Transfer Rate: 100,000 - 200,000 char. per second	Must provide total capacity of approximately 600 million characters
Drums	1	Capacity: 6 million characters Access Time: 15 - 25 milliseconds average Transfer Rate: 500,000 char. per second.	
Mag. Tape Units	4	Transfer Rate: 50,000 - 100,000 characters per sec. Density: 200,556,800 b.p.i.	
Central Peripheral Equipment			
Card Readers	2	800 - 1200 cards per min.	
Line Printers	3	800 - 1200 lines per min.	
Card Punches	1	200 - 300 cards per min.	
Special Communications Unit	1		See Para. 7.7

Table 7.9 Time-Sharing System Central Computer System Equipment

System Element	Quantity	Model or Performance	Comments
Central Processor	1	Large-scale computer, required speed given by 100-statement per second FORTRAN compilation rate	See Para. 7.5.2
Core Memory	530,000 characters	Access time not specified, must match processor capability	See Para. 7.5.4
Secondary Memory			
Disk Units	3 - 4	Capacity: 150 - 200 million characters per disk unit Access Time: 150 - 250 milliseconds average Transfer Rate: 100,000 - 200,000 char. per second	Must provide total capacity of approximately 600 million characters
Drums	1	Capacity: 2 million characters Access Time: 15 - 25 milliseconds avg. Transfer Rate: at least 500,000 char. per second	
Mag. Tape Units	4	Transfer Rate: 50,000 - 100,000 char. per second Density: 200, 556, 800 b.p.i.	
Central Peripheral Equipment			
Card Readers	1	800 - 1200 cards per min.	
Line Printers	1	800 - 1200 lines per min.	
Card Punches	1	200 - 300 cards per min.	
Special Communications Unit	1		See Para. 7.7

Table 7.10 Remote Batch-Processing System Central Computer System Equipment

8.0 System Cost and Cost Allocation

8.1 Equipment Cost

Tables 7.7 through 7.10 provide the basis for estimating the costs of the equipment systems proposed. In estimating these costs, currently published manufacturers' prices are used. Since, in general, the system elements called for do not correspond to a specific manufacturer's model, the prices of several units, having performance in the desired range, were compared to derive a cost estimate.

The cost estimates for the different system elements vary in their reliability. Some are almost certainly accurate within a 10% range because of the specific characteristics required. The secondary memory units and standard peripheral equipments are in this category. The estimates for central processor units, core memories, and communication interface units cannot be regarded with the same confidence. The designs of the various manufacturers differ substantially in the way they achieve the performance desired, and prices will differ accordingly.

The most critical cost element, for both time-sharing and remote batch-processing systems, is the in-school terminal equipment. The quantities required are very large, and the unit price of these units has strong leverage on total system cost. It would be reasonable to expect quantity price adjustments on these items, but this possibility has not been included in the estimates. Tables 8.1 through 8.4 show cost breakdowns for the two systems, and Table 8.5 shows a comparison of the system costs. In the tables, cost of secondary storage units and peripheral equipments include the costs (pro-rated where necessary) of the controllers required.

8.2 Other System Costs

In addition to the equipment costs, the installation and use of either of the proposed systems will generate other costs. These additional costs will arise from the following necessary items:

- Facility preparation and operation
- Training of personnel
- Operating and support personnel
- Maintenance of owned equipment
- Support programming
- Forms and supplies
- Courier and vehicle operation
- Administration

System Element	Unit Cost ($/mo.)	System Cost ($1000's/mo.)
Time-sharing Terminals	125	122.2
In-school lines plus multiplexers	465	26.5
Leased lines plus termination equipment	270	15.4
	Total for System	164.1
	Total per student-year	$ 19.70

Table 8.1 Estimated Cost for Terminals and Communication Subsystem - Time-Sharing System

System Element	Unit Cost ($/mo.)	System Cost ($1000's/mo.)
Keypunch Equipment	60	15.0
Card Reader & Controller	200	11.4
Line Printer & Controller	800	45.6
Communications Interface Unit	550	31.4
Leased lines plus termination equipment	270	15.4
	Total for System	118.8
	Total per student-year	$ 14.30

Table 8.2 Estimated Cost for Terminal Equipment and Communications Subsystem - Remote Batch-Processing System

System Element	Unit Cost ($/mo.)	System Cost ($1000's/mo.)
Central Processor	23,000	23.0
Core Memory	24,000	24.0
Disk Units	5,500	16.5
Drums	3,000	3.0
Mag. Tape Units	1,000	4.0
Card Reader	650	1.3
Line Printer	1,600	4.8
Card Punch	800	.8
Special Comm. Unit	5,000	5.0
	Total for System	82.4
	Total per student-year	$ 9.90

Table 8.3 Estimated Cost for Central Computer System - Time-Sharing System

System Element	Unit Cost ($/mo.)	System Cost ($1000's/mo.)
Central Processor	18,000	18.0
Core Memory	14,000	14.0
Disk Units	5,500	16.5
Drums	1,800	1.8
Mag. Tape Units	1,000	4.0
Card Reader	650	.7
Line Printer	1,600	1.6
Card Punch	800	.8
Special Comm. Unit	5,000	5.0
	Total for System	62.4
	Total per student-year	$ 7.50

Table 8.4 Estimated Cost for Central Computer System - Remote Batch Processing System

		Time-Sharing System	Remote Batch-Processing System
Terminals and Communication Equipment	Total $ per month	$ 164,100	$ 118,800
	$ per student-year	$ 19.70	$ 14.30
Central Computer System	Total $ per month	$ 82,400	$ 62,400
	$ per student-year	$ 9.90	$ 7.50
Total System	Total $ per month	$ 246,500	$ 181,200
	Equiv. purchase *	$ 9.8 million	$ 7.3 million
	$ per student-year	$ 29.60	$ 21.80

Table 8.5 System Cost Comparison

* Based on 40 months lease equivalent.

No attempt will be made here to estimate these additional costs. Each one is potentially important, however, and the list should be covered thoroughly in an implementation plan for either of the systems.

8.3 Cost Allocation

The allocation of cost among the users of the proposed systems should be made according to a specific objective which should be the same as the overall system objective. That statement appears unnecessary, but it is not. Unless the principle is followed carefully, it is entirely possible that utilization of the system will be undesirably reduced as a result of ill-considered attempts to reduce allocated costs.

If it is assumed that at least a limited system objective is high utilization by direct requests from member schools, the cost allocation method can help induce this. In this case, which certainly seems like a reasonable one, the member schools should be directly assessed for both fixed and variable cost, in proportion to the best estimates of anticipated use which can be obtained. The assessment process should occur frequently enough so that estimation of future use tends to become an accurate procedure.

As a check on this approach, it would be reasonable to apply an additional cost allocation for use above an estimated minimum. If this is done, however, the added use charges should reflect as accurately as possible the actual incremental cost of such use.

Also, the operating staff at the central facility should conduct careful analysis of types of usage to determine the most equitable procedure for allocating costs from usage estimates.

A number of interesting cost questions come up. For example, operating personnel located at a member school would be on that school's staff. It should probably be left to each member school to budget for these personnel, even though the item is a part of total system operating cost. On the other hand, it would probably not be desirable for each school to pay for its own communication costs since these will vary markedly with distance from the central facility, and the location of the central facility is presumably selected on the basis of economy and convenience for the total school membership.

9.0 Simulation of System Performance

9.1 Introduction

It has already been observed that there is great difficulty in making a direct prediction of the performance of a proposed time-sharing or remote batch-processing system. An indirect approach, but a very useful one, is to simulate the actual operation of the system, using estimated load parameters. This type of simulation is itself a lengthy computing process, and usually involves a major task of computer programming and a substantial amount of computer time.

Computer simulations of the two proposed systems were performed as a part of this study. This section describes the approach to simulation which was used, and gives the major results. A more complete description of the simulator program and its use is given in Appendix 5.*

The purpose of this simulation study was to examine the dynamic and stochastic behavior of the various loading conditions on the Central Computer Facility. In order to study the trade-offs between the various potential time-sharing and multiprogramming configurations and software strategies, three separate simulation models were developed by the Special Information Products Department of General Electric Company under a subcontract from GLC. Two of the three simulation models are used to study the time-sharing system. The third is used to study the remote batch-processing system.

Although the three simulation models emphasize different aspects of the system operations, they deal with the same set of user profiles. All simulation input conditions are identical for the first two time-sharing simulation models. However, the multiprogramming system cannot handle some of the applications that are available under the time-sharing system. The number of terminals attached to the system is considerably lower for this case than for the time-sharing system. All three simulation models are written in RAND SIMSCRIPT and the IBM 7094 is used for the simulation runs.

9.1.1 Time-Sharing Simulation

The Educational Time-Sharing Environmental System Simulation Model (ETESim) was developed to study an environment in which the Central Computer Facility is used to service the various terminals located at different schools. This simulation model can determine the steady state load on the

* Appendix 5 is contained in a separate document available upon request from the U. S. Office of Education.

Central Computer Facility under various conditions of terminal use. Basically, this model simulates the communications aspects of the system.

The Educational Time-Sharing System Simulation Model (ETSSim) is principally a queuing model. This simulation model uses a fixed command and file structure to process 11 classes of user terminal commands. The software strategies are designed to minimize the blocking effects. The hardware configuration is designed to achieve the desirable level of performance.

9.1.2 Remote Batch-Processing Simulation

The Educational Multiprogramming System Simulation Model (EMPSim) was used to simulate operation of a proposed remote batch-processing system. It is designed to use the same basic file structure as used in the Educational Time-Sharing System. The principal features of the model are time slicing, dynamic core allocation, and concurrent input/output and processor operations. This simulation model studies the effect of six different types of remote job entry applications.

9.2 Hardware Configuration

An educational data processing system will require extensive inter-connection of information handling equipment. There are three basic types of equipment.

- Data Communications — to provide channels and devices for getting data to and from the system

- Data Storage — to store data required in the system

- Data Processing — to perform arithmetic and logic operations with data

The same basic hardware configuration is used in all three simulation models. The number of terminals attached to a system may vary as a function of the application and load conditions. This basic configuration is presented schematically in Figure 9.1 on the following page.

9.2.1 Data Communications Facility

The individual loads imposed by the student consoles may vary. The model of this environment consists of a description of what each user does during an elementary operation at his console. Simply stated, the interaction consists

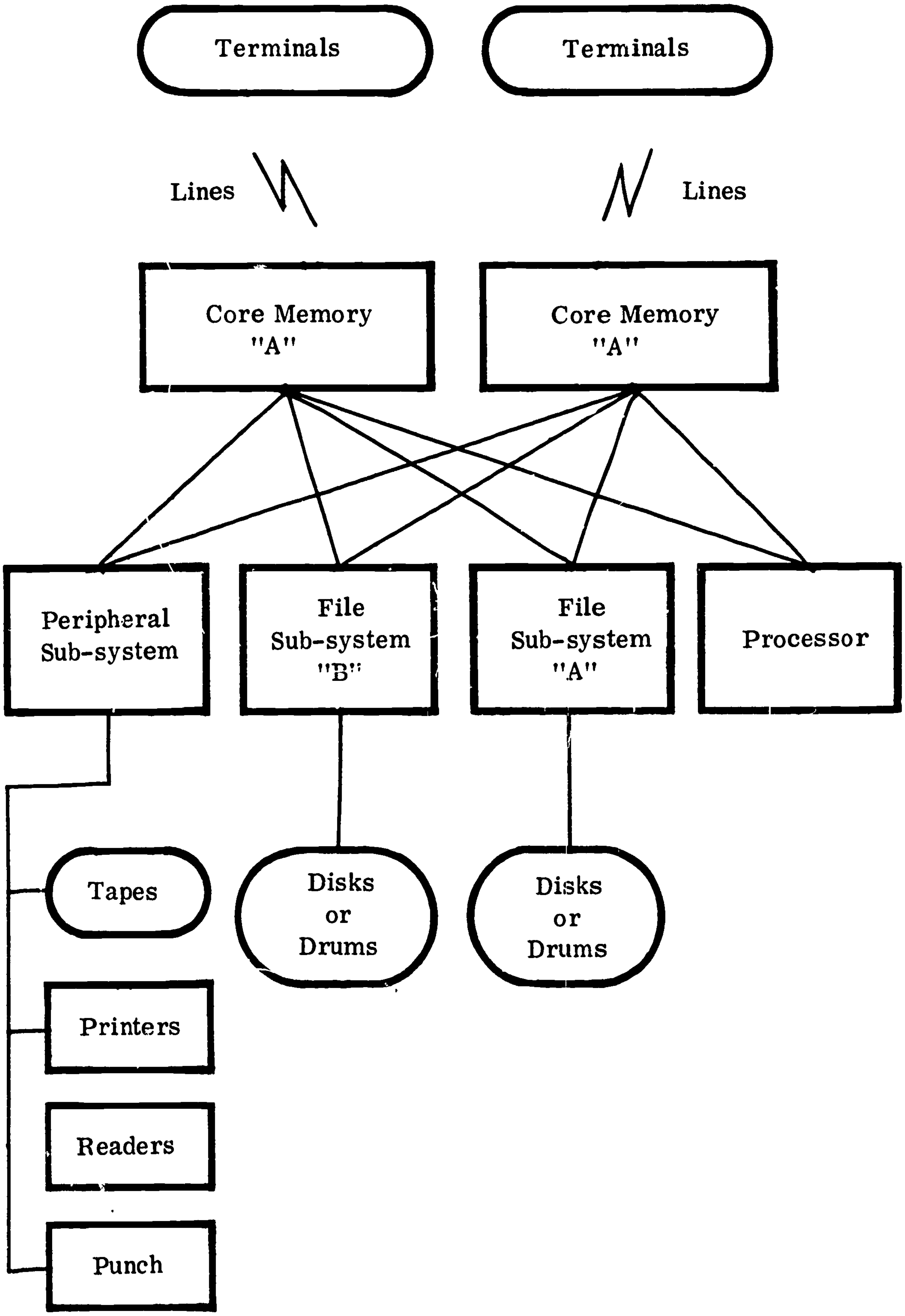


Figure 9.1. Simulated Hardware Configuration

of the user requesting and receiving service from the system. The usual form of an interaction involves the user thinking, typing input at his remote console, waiting for the response from the system, and finally receiving the computer output.

Student Consoles

From the system's point of view, a remote console may be thought of as being in either one of two states — "active" or "idle". If a remote console is "active", it indicates that the user in interacting with the computer. A "transaction" that represents the request of a user is somewhere in a computer. If a remote console is "idle", it does not interact with its computer during a time interval called "dead time".

In order to construct a simulation model to describe this system environment, a system designer must define the following parameters according to his application:

- Device Characteristics — expected data transfer rate, display technique (visual or hardcopy), page size (visual display only), line size, remote buffer size, etc.

- Use of the Device — data input media (paper tape, console), data output media (printer, console, paper tape), data entry technique, data entry rate, dead time between entries, terminal activity statistics, etc.

- User's Requirements — message size, message type distribution, etc.

Some of these parameters define the variables in a simulation model. Others define the facilities available in the simulation model. From the results of initial simulation runs, some variables may be changed to study system design alternatives and to perform trade-off analyses.

Communications Network

Communications networks link the remote consoles to a concentrator, pre-processor, or the host computer in a system. A communications network consists of transmission channels, interfaces, and concentrators. For a far-flung time-sharing system, the cost of communication is high. Economic factors usually determine the selection of appropriate telegraph/telephone/wide-band service.

To describe the specific layout of a communication system, a system designer must define the following parameters:

- Communication Line — number of communication lines and number of remote terminals attached to each line.

- Buffer — buffer size for each terminal or line, location of the buffers (local or remote to the computer).

- Concentrator — location of each concentrator, number of input lines and output lines for each concentrator.

- Service — common user telegraph service (TELEX or TWX) or leased telegraph channel (TELPAK).

9.2.2 Data Storage Facilities

An educational data system uses three types of data storage facilities: core, drum, and disk.

Core

The core memory is divided into three parts:

- memory required for data buffer areas,

- memory required for resident control and compiler programs, and

- the remainder of the total memory which is available for user programs. This is called the slave mode work area.

The slave mode work area is partitioned into pages. The simulation model allows each user to define the memory and page size. If a request calls for three pages of core, the slave mode work area program will allocate it unless there is not enough work area storage available. In that case, the request will be deferred and filed into a request queue until it is serviced.

Drum

The drum file subsystem consists of one pair of data channels with cross channel switch capability and one or more drum storage units. This file subsystem is designed to allow two simultaneous accesses to any part or parts of the drum storage units. In the simulation model, the drum storage units are assumed to be large enough to store the Control Program Library, Current File, and Swap File.

Control Program Library size varies according to the complexity of the software system. Current File should be large enough to serve maximum number of active user programs and data in the system. For example, if there are 200 terminals attached to the system, the absolute maximum storage unit requirement should be no less than 200n when n is the maximum program size allowed in the system at any time. However, the actual storage usage may be substantially less than that requirement.

Swap File size also varies. The simplest approach is to partition the file into even slots, one for each terminal attached to the system. This, of course, will require more data storage. The alternative is to allocate Swap File storage on a dynamic basis. For example, a system can have a Swap File of 320K words and be divided into one thousand 320-word links. If a request calls for 400 words, two 320-word links will be allocated to satisfy such a request.

Disk

The disk file subsystem consists of one pair of channels with cross channel switch capability and one or more disk storage units. This file subsystem is designed to allow simultaneous accesses to any part or parts of the disk storage units. In this simulation model, the disk storage units are assumed to be large enough to store the Catalog File and the Program File.

All members of the Catalog File are identified by their user numbers. Each user can have many entries. Several links may be used to store those entries. The address of the first link is identified by the user number. Other links are chained to the first link through a set of address pointers.

All members of the Program File are identified by their program names. The address of those programs can be located in the Catalog File. In general, Program File is organized as a linked file. File maintenance and housekeeping routines are required to reorganize the file so that the links belonging to the same program will be placed adjacent to each other. One seek per program rather than one seek per link will be required to locate the entire program.

9.2.3 Data Processing Facility

The data processing facility may have one or more central processing units. The performance of those central processing units are usually represented by their instruction time and processor speed. In the more advanced system, their performance is often described by the amount of compute times required to perform each of the software functions (such as allocate core, free core, buffer full, buffer empty, schedule, and input/output supervisor, etc.). In order to evaluate the equivalent performance between various computers, Gibson Mix can be used to calculate their relative processor speed and effective performance.

9.3 Educational Time-Sharing System

An Educational Time-Sharing System should be evaluated on the basis of the number of active users it can handle concurrently, the turnaround time given to those users for a given load mix, and the cost to those users at that level of performance. The overall cost of the system can be further broken down to the degree of utilization achieved for each of the major components such as the core memory, input/output channel and processor, etc. In order to obtain steady state statistics, we have used two simulation models (ETESim and ETSSim) jointly to perform this study.

9.3.1 ETESim and ETSSim Input

Table 9.1 shows the statistics used to define work load and compute time requirements for the Educational Time-Sharing System. Table 9.2 and Table 9.3, on the following page, show the statistics used to define the command mix for the same system.

SUBJECT	DISTRIBUTION				
	Minimum	Medium	Mean	Maximum	Unit
Compile and Execution Time	0	10.0	153.5	2000	Milliseconds
Program Size	500	1900	2100	6000	Character
Keyboard Input Per Request	10	33	41	200	Character
Program Listing Per Request	270	680	1250	2700	Character
Dead Time	0	11	35.2	454.2	Seconds

Table 9.1 User Program Statistics

Command Type	Individual Probability in Percent
TRIVIAL	10
COMPILE	13
ENTER	12
NEW	1
END	23
SAVE	8
LIST	9
RESEQ	10
RELEASE	2
PERMIT/REVOKE	1
OLD	11

Table 9.2 Command Type Distribution

Command Type	Input Size	Output Size
TRIVIAL	20	15
COMPILE	10	20% of the source program length
ENTER	300	See Table 3.1.1
NEW	2700	10
END	10	10
SAVE	10	10
LIST	10	See Table 3.1.1
RESEQ	20	10
RELEASE	10	10
PERMIT/REVOKE	20	10
OLD	20	10

Table 9.3 Command Input/Output Message Length (In Characters)

9.3.2 ETESim Results

The Educational Time-Sharing Environmental System Simulator (ETESim) produces the following statistical information concerning the utilization of various communications facilities:

Number of Lines	299 Lines	397 Lines
Simulated Time	7200 seconds	7200 seconds
Number of Commands Simulated	24,592	32,343
Maximum Number of Lines:		
Input	154	198
Output	89	116
Total	243	314
Maximum Number of Requests in Processor	31	43
Average Requests in:		
Processor	8.6	11.4
System	180.7	239.7
Percent of Lines with a Job in the Processor (average)	2.86	2.86
Percent of Lines Busy (average)	60.42	60.45
Maximum Number of Lines in the Processor	10.4%	10.8%

Table 9.4 Communication Facility Utilization

9.3.3 ETSSim Results

From the ETESim simulation results, the ETSSim runs are set up to study the effects of the 299 and 397 line systems under three different load conditions — normal, peak, or maximum. Normal load represents the steady state average work load that a central computer facility has to handle during a typical day. Peak load represents the maximum work load that a central computer facility has to handle when the work load is built up gradually during a 2 hour interval. However, if the work load is built very fast (such as assuming all terminals are occupied by the students concurrently), then the maximum load the control computer facility has to handle is called "maximum". The ETSSim output contains the following type of statistics:

- Turnaround Time — the length of time between the arrival of the last character of an input message at the buffer area and the arrival of the first character of the output message at the student console. Turnaround time is also called response time.

- System Time — the length of time between the arrival of the last character of an input message at the buffer area and the arrival of the last character of the output message at the student console. System time is important simply because it represents the time a user must wait at his console.

- Work Area Utilization — the percent of slave mode work area being occupied by user programs, data base, or transient programs weighted by time.

- Channel Utilizations — the percent of the channel time allocated to service input/output requests such as software overhead, seek time, rotational delay, and data transfer time. For the drum storage units, there is no seek time.

- System Thruput — the number of transactions serviced per unit of time, e.g., 50 per minute.

- File Thruput — the number of events serviced by the channels attached to a data base per unit of time, e.g., 20 events per minute. An event represents one input/output request from the time a request is filed into a waiting queue to the time this request is serviced.

- Maximum Queue Length — the maximum number of requests in the same queue at any time.

- Average Waiting Time — the average length of time the request stays in a queue.

- Processor Utilization — the percent of time a processor is used to perform control program services, application programs processing, compile and execution, or background program processing respectively. The sum of the processor utilizations should equal 100%.

- R/E Ratio — the ratio between response time (R) and the execution time (E) for all compute interactions. This number is very meaningful for evaluating the maximum number of lines a processor bound system can handle.

The input conditions for ten simulation runs are listed on Table 9.5 on the following page. The results of these runs are summarized on Tables 9.6 to 9.10, beginning on page 117 .

Run Condition	Run Number	# of Lines Attached to the System	# of Active Users on Processor	Core Memory Size (in K Characters)
Normal	1	128	4	25
Normal	2	256	8	25
Peak	3	299	31	25
Normal	4	299	9	25
Maximum	5	299	52	25
Maximum	6	299	52	45
Normal	7	397	12	25
Peak	8	397	43	25
Maximum	9	397	78	45
Maximum	10	397	78	109

Table 9.5 Input Conditions for Simulation Runs

Table 9.6

SUMMARY OF NORMAL LOAD RESULTS

NUMBER OF USERS	DYNAMIC CORE SIZE	CHANNEL WAITING TIME (MILLI-SECONDS)				CHANNEL QUEUE LENGTH				LOGICAL FILE THRUPUT (PER MINUTE)				SYSTEM THRUPUT	FACILITY UTILIZATION				WAITING TIME (MILLI-SECONDS)						RESPONSE TIME (SECONDS)				REMARKS
		Current	Swap	Catalog	Program	C	S	C	P	C	S	C	P	(Request/Minute)	Processor	$Chan_A$	$Chan_B$	Work area	PQ1	PQ2	PQ3	PQ4	PQ5	PQ6	Compute	Data Base	Trivial	All	
128	25K	17.95	5.24	21.34	0	2	2	2	1	849.9	468.9	232.3	98.5	923.30	79.89%	26.30%	43.48%	11.45%	0	10.48	10.59	3.16	0	0	0.44	0.19	0.09	0.21	
256	25K	48.0	14.75	75.4	0	6	4	5	1	1043.6	577.4	295.0	125.6	1137.9	100%	22.50%	53.69%	12.74%	0	68.59	422.71	4.57	0	0	1.63	0.34	0.18	0.49	
.99	25K	58.18	18.42	94.51	0	6	4	7	1	1079.4	580.0	286.6	116.2	1154.1	99.58%	33.36%	52.59%	25.74%	0	144.09	486.02	5.35	0	0	2.02	0.35	0.20	0.56	
397	25K	45.38	12.30	58.71	0	6	4	6	1	1062.8	591.1	300.0	122.4	1157.2	100%	31.11%	54.85%	33.79%	0	151.32	1186.19	4.85	0	0	3.19	0.44	0.27	0.78	

Table 9.7

SUMMARY OF PEAK LOAD RESULTS

NUMBER OF USERS	DYNAMIC CORE SIZE	CHANNEL WAITING TIME (MILLI-SECONDS)				CHANNEL QUEUE LENGTH				LOGICAL FILE THRUPUT (PER MINUTE)				SYSTEM THRUPUT	FACILITY UTILIZATION				WAITING TIME (MILLI-SECONDS)						RESPONSE TIME (SECONDS)				REMARKS
		Current	Swap	Catalog	Program	C	S	C	P	C	S	C	P	(Request/Minute)	Processor	$Chan_A$	$Chan_B$	Work area	PQ1	PQ2	PQ3	PQ4	PQ5	PQ6	Compute	Data Base	Trivial	All	
299 Lines	45 K	34.67	19.15	123.6	0	6	4	8	1	1051.8	568.0	293.0	118.0	1130.5	100%	32.34%	53.02%	93.78%	0	148.24	4823.9	6.47	0	0.10	11.70	0.89	0.71	2.23	
397 Lines	25 K	3725.4	1124.83	451.4	128.08	8	4	7	1	1097.1	601.4	303.0	129.3	1189.5	100%	33.95%	54.25%	99.28%	2.41	150.05	5794.3	470.83	988.1	539.05	12.00	1.42	0.98	2.87	
84 Lines	25 K	58.8	18.42	94.51	0	6	4	7	1	1079.4	586.0	286.6	116.2	1154.1	99.58%	33.36%	52.59%	25.74%	0	144	487	5.4	0	0	2.02	0.35	0.20	0.56	

Table 9.8

SUMMARY OF MAXIMUM LOAD RESULTS

NUMBER OF USERS	DYNAMIC CORE SIZE	CHANNEL WAITING TIME (MILLI-SECONDS)				CHANNEL QUEUE LENGTH				LOGICAL FILE THRUPUT (PER MINUTE)				SYSTEM THRUPUT	FACILITY UTILIZATION				WAITING TIME (MILLI-SECONDS)						RESPONSE TIME (SECONDS)				REMARKS
		Current	Swap	Catalog	Program	C	S	C	P	C	S	C	P	(Request/Minute)	Processor	$Chan_A$	$Chan_B$	Work area	PQ1	PQ2	PQ3	PQ4	PQ5	PQ6	Compute	Data Base	Trivial	All	
299	25K	44.51	13.30	58.29	0	5	3	6	1	1019.0	566.9	286.6	120.8	1110.3	100%	31.51%	51.99%	87.31%	0	138.4	9333.2	5.2	0	0	21.82	1.34	1.19	3.90	
299	45K	9341.1	2993.4	1559.2	439.3	12	5	10	1	1024.3	561.0	288.1	119.2	1112.6	99.96%	31.79%	52.[illegible]2%	99.28%	2.06	187.4	5416.8	1154.6	2778.7	711.8	13.77	2.63	1.19	3.90	

Table 9.9

SUMMARY OF MAXIMUM LOAD RESULTS

NUMBER OF USERS	DYNAMIC CORE SIZE	CHANNEL WAITING TIME (MILLI-SECONDS)				CHANNEL QUEUE LENGTH				LOGICAL FILE THRUPUT (PER MINUTE)				SYSTEM THRUPUT	FACILITY UTILIZATION				WAITING TIME (MILLI-SECONDS)						RESPONSE TIME (SECONDS)				REMARKS
		Current	Swap	Catalog	Program	C	S	C		C	S	C	P	(Request/ Minute)	Processor	$Chan_A$	$Chan_B$	Work area	PQ1	PQ2	PQ3	PQ4	PQ5	PQ6	Compute	Data Base	Trivial	All	
299	45K	9341.1	2993.4	1559.2	439.3	12	5	10	1	1024.3	561.0	285.1	119.2	1112.	99.96%	31.79%	52.02%	99.28%	2.06	187.4	5416.8	1154.8	2778.7	711.8	13.77	2.63	1.19	3.90	
397	45K	10203.0	3274.6	1546.2	438.0	11	4	11	1	1064.5	580.2	296.8	125.8	1150.	99.95%	53.78%	32.81%	99.6%	4.06	187.5	11264.3	1115.2	2748.5	707.3	24.54	2.77	1.79	6.10	

Table 9.10

SUMMARY OF MAXIMUM LOAD RESULTS

NUMBER OF USERS	DYNAMIC CORE SIZE	CHANNEL WAITING TIME (MILLI-SECONDS)				CHANNEL QUEUE LENGTH				LOGICAL FILE THRUPUT (PER MINUTE)				SYSTEM THRUPUT	FACILITY UTILIZATION				WAITING TIME (MILLI-SECONDS)						RESPONSE TIME (SECONDS)				REMARKS
		Current	Swap	Catalog	Program	C	S	C	P	C	S	C	P	(Request/Minute)	Processor	$Chan_A$	$Chan_B$	Work area	PQ1	PQ2	PQ3	PQ4	PQ5	PQ6	Compute	Data Base	Trivial	All	
397	45	10203.0	3274.6	1546.2	438.0	11	4	11	1	1064.5	580.2	296.8	125.8	1150.	99.95%	53.78%	32.81%	99.6%	4.06	187.5	11264.3	1115.2	2748.5	707.3	24.54	2.77	1.79	6.10	
397	109	59.23	19.51	130.27	0	8	5	8	1	1025.6	567.0	277.7	122.2	1118.	100.0%	31.76%	49.61%	56.630	0	185.3	13689.1	6.8	0	0	31.89	1.94	1.79	5.77	

Beginning on the following page, Graphs 9.1 to 9.3 show the trade-offs between response time, performance, and the number of lines attached to the system under three different load conditions — normal, peak, and maximum. In each graph, there are four curves representing each of the command types — trivial, compute, data base (all others), and all commands.

Trivial Interaction Curve

This curve is a near straight line because the trivial interactions do not have to wait in any queue. In the case of processor, a trivial interaction is assigned to higher priority than either data base or compute interaction and is usually processed directly by an interrupt handler.

Compute Interaction Curve

This curve shows that the response time is increased sharply as the number of lines attached to the system is increased. The user terminal command mix, compute size, and dynamic storage requirement have the most affect on this curve.

Data Base Interaction Curve

This curve is influenced by the size of compute quantum, dynamic core storage size, and channel performance. Those interactions usually take very little compute time, but may take as much core or channel time as most compute interactions.

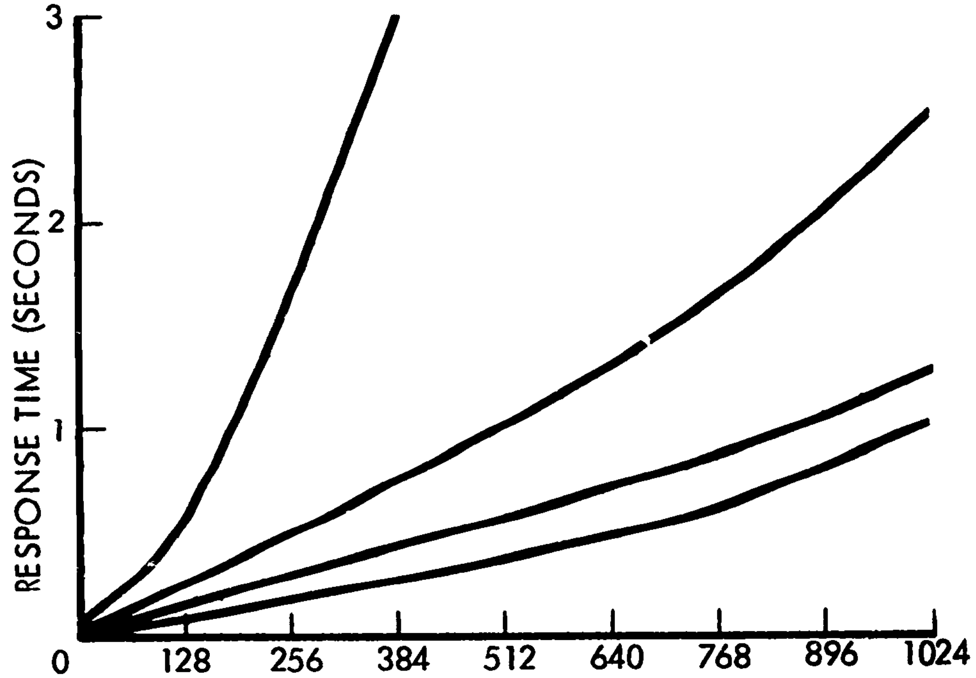
All Commands Curve

This curve shows the weighted average response time for all commands as a function of the number of lines attached to the system.

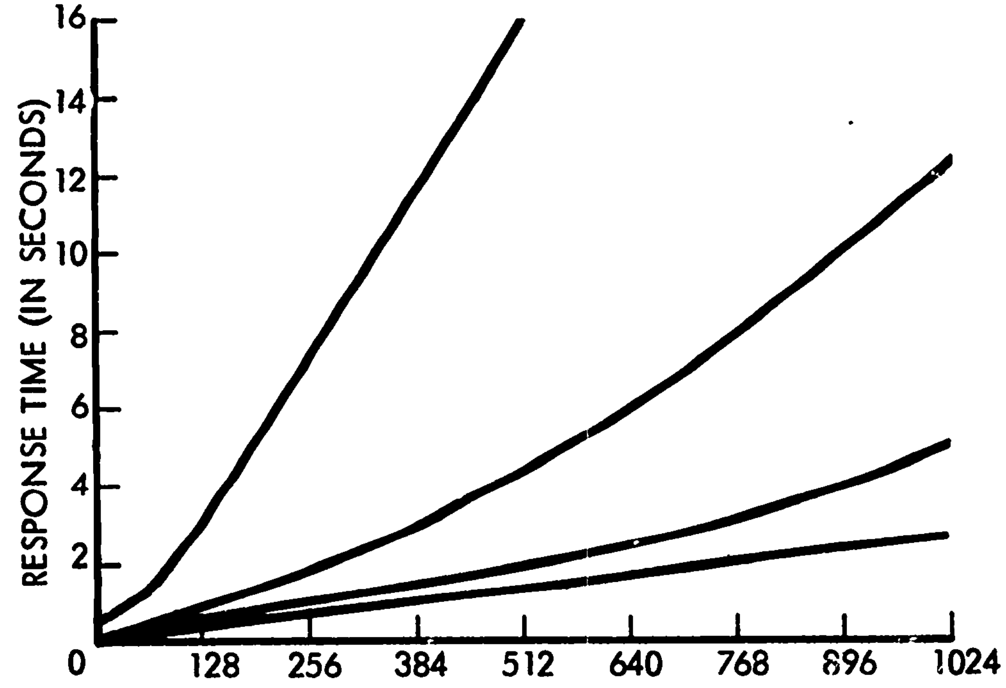
9.4 Remote Batch-Processing System Simulation Model (EMPSim)

9.4.1 EMPSim Input

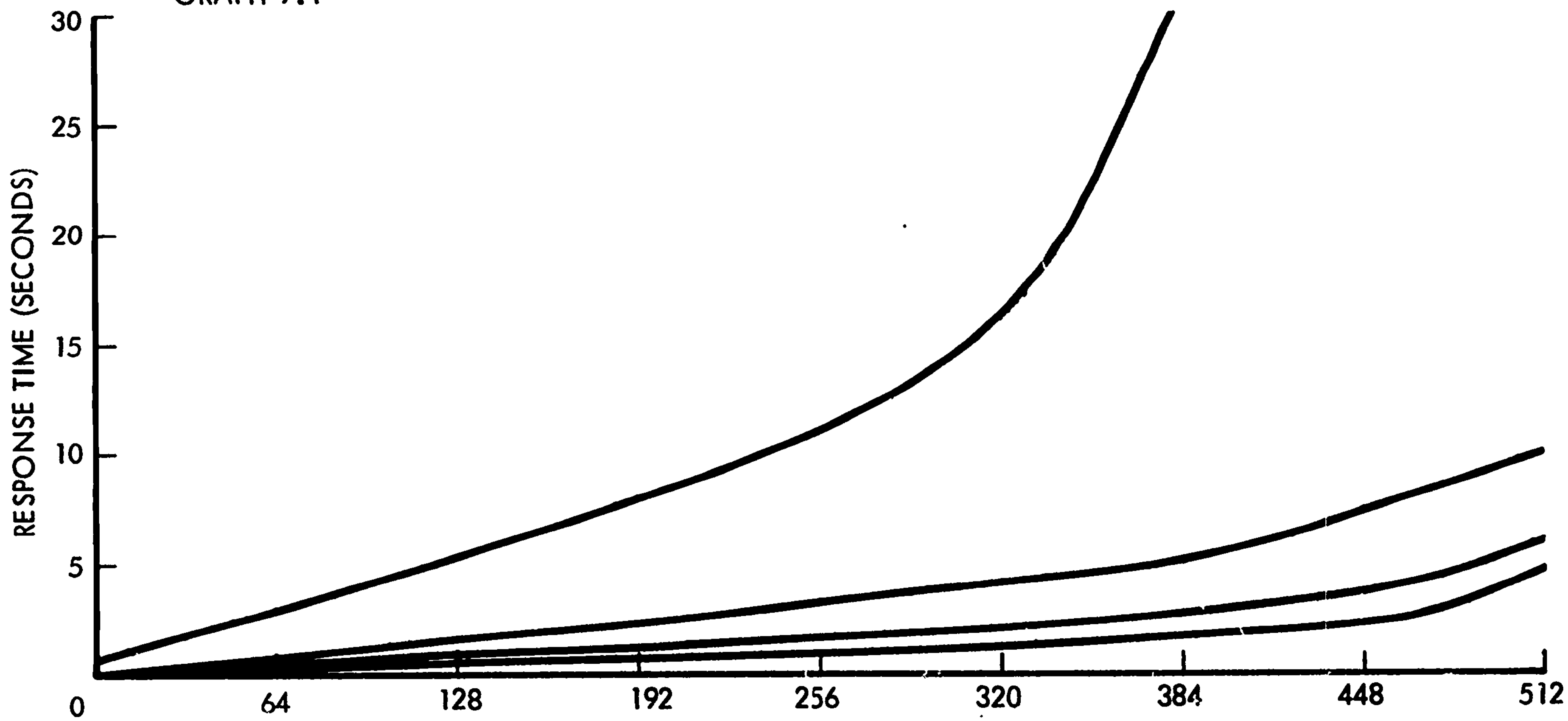
The system consists of 52 or 80 sets of remote terminals. A card reader serves as the input device and a line printer serves as the output device. One pair of channels is attached to the fast random access file system (drum) and one pair is attached to the slow random access file system (disk). The maximum core storage size is 22 slots with each slot capable of storing the largest program size.



NORMAL LOAD
GRAPH 9.1



PEAK LOAD SITUATION
GRAPH 9.2



MAXIMUM LOAD
GRAPH 9.3

The six different types of jobs are:

(1) Input new program, execute it, but do not save it (60%)
(2) Input new program, save it, and execute it (10%)
(3) Input new program, save it, but do not execute it (5%)
(4) Execute the saved program (15%)
(5) Execute the saved program using other data bases (such as referencing other records) (5%)
(6) Write data bases (information retrieval or inquiry processing operations) (5%)

The record length distribution is:

Number of Words	Cumulative Probability
0	0.0
60	0.0
480	0.7
600	0.9
3,000	0.98
12,000	1.00

Program length distribution is:

Number of Characters	Cumulative Probability
83	0.0
167	0.1
500	0.4
1,500	0.8
2,500	0.98
6,000	1.00

9.4.2 EMPSim Output

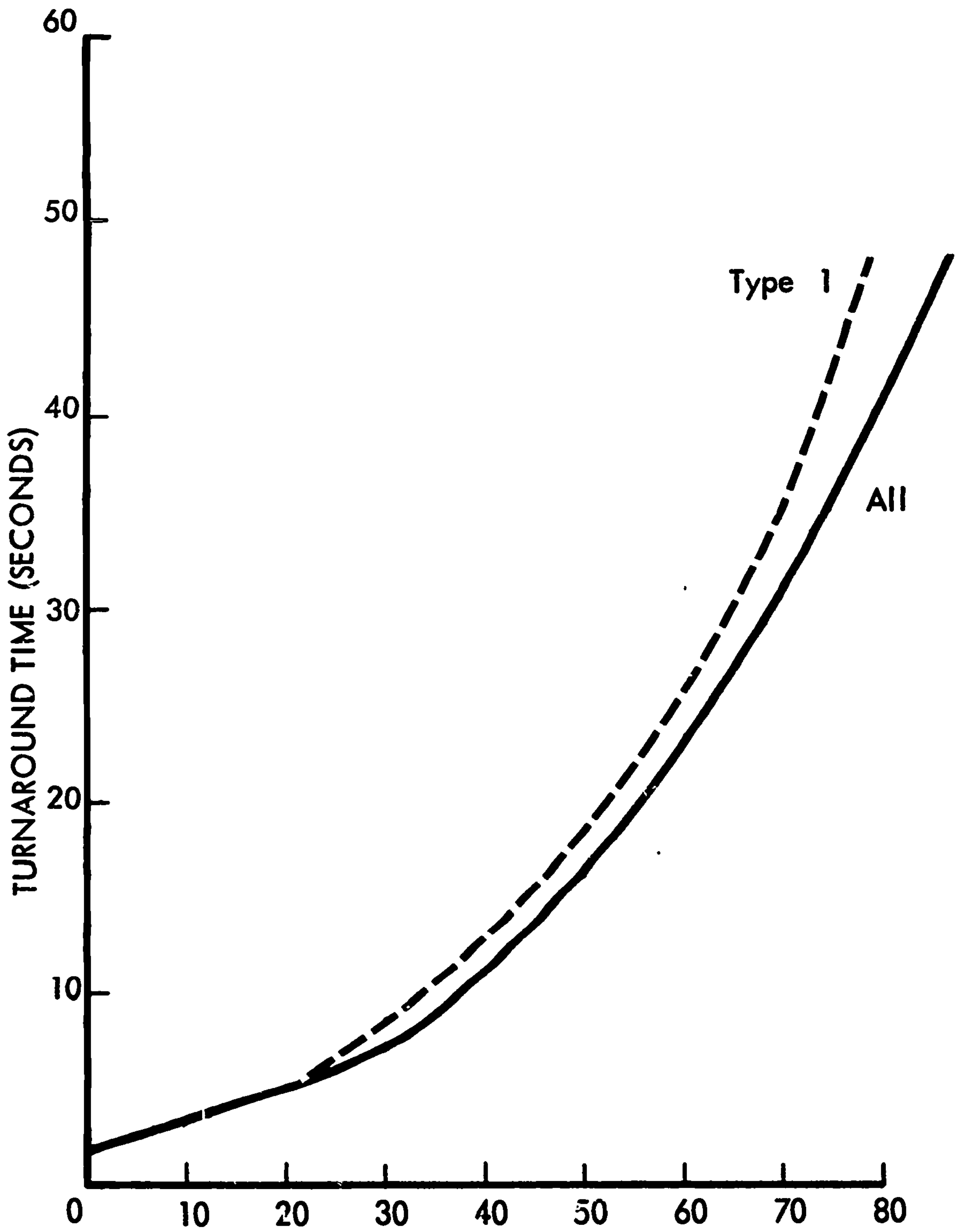
There are two simulation runs made to study the effects of different numbers of terminals attached to the systems. Run one simulates the load which the Central Computer Facility must handle during the peak hours. Run two simulates the normal load situation. Table 9.11, on the next page, summarizes the simulation output.

Seven graphs follow Table 9.11. Graphs 9.4 to 9.9 show the relative performance level of the system during the first twelve minutes of the operations. Graph 9.10 shows that the thruput capability is a function of the number of lines attached to the system.

Table 9.11

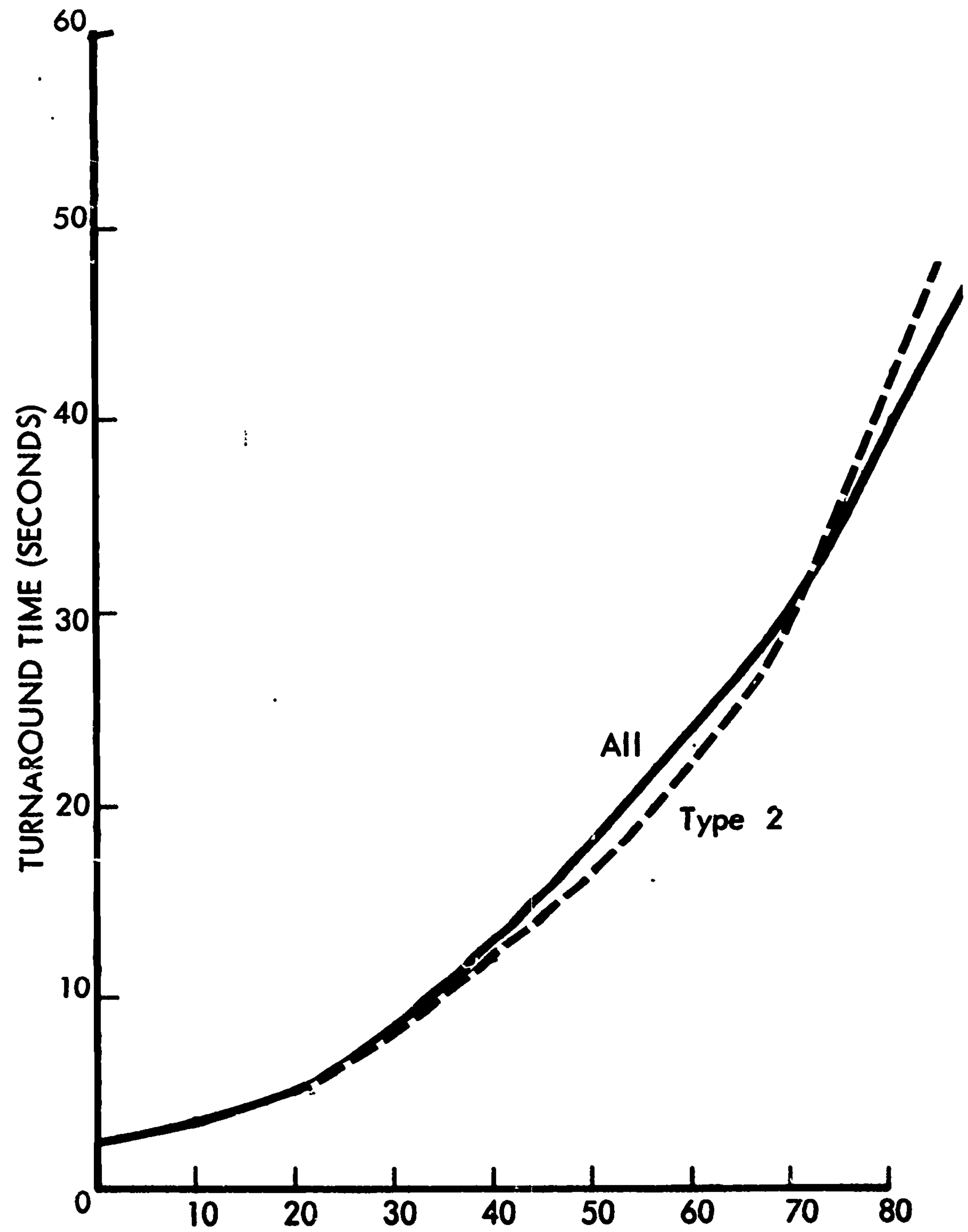
SUMMARY OF EMPSim OUTPUT

RUN NO.	NO. OF USERS	INPUT RATE (JOBS/ MINUTE)	CHANNEL WAITING TIME (MILLI-SECONDS)				CHANNEL QUEUE LENGTH				LOGICAL FILE THRUPUT/MIN.				SYSTEM THRUPUT REQUESTS/ MINUTE	FACILITY UTILIZATION (PERCENT)				WAITING TIME (MILLI-SECONDS)						JOB TYPE TURNAROUND TIME (SECS)							EFFECTIVE AVERAGE INTER-ARRIVAL RT
			Current	Swap	Catalog	Program	C	S	C	P	C	S	C	P		Processor	Chan A	Chan B	Workarea	PQ1	PQ2	PQ3	PQ4	PQ5	PQ6	1	2	3	4	5	6	All	(Per Min.)
1	80	80	13273.9	1407.8	245.3	0	11	3	8	1	73.8	53.4	19.2	8.9	30	77%	14.2	9.01	36.8	0	2165	14840	665	193 SEC	13.5 SEC	52.2	43.8	42.7	27.0	31.9	28.7	43.79	81.9
2	52	52	3698.3	753.6	1133.6	0	11	2	9	1	60.2	43.1	15.5	7.1	25	56.	11.5	7.3	21	0	1424	8796	294	79.4 SEC	5.7 SEC	32.3	28.0	29.6	20.7	22.8	21.5	28.49	67.4



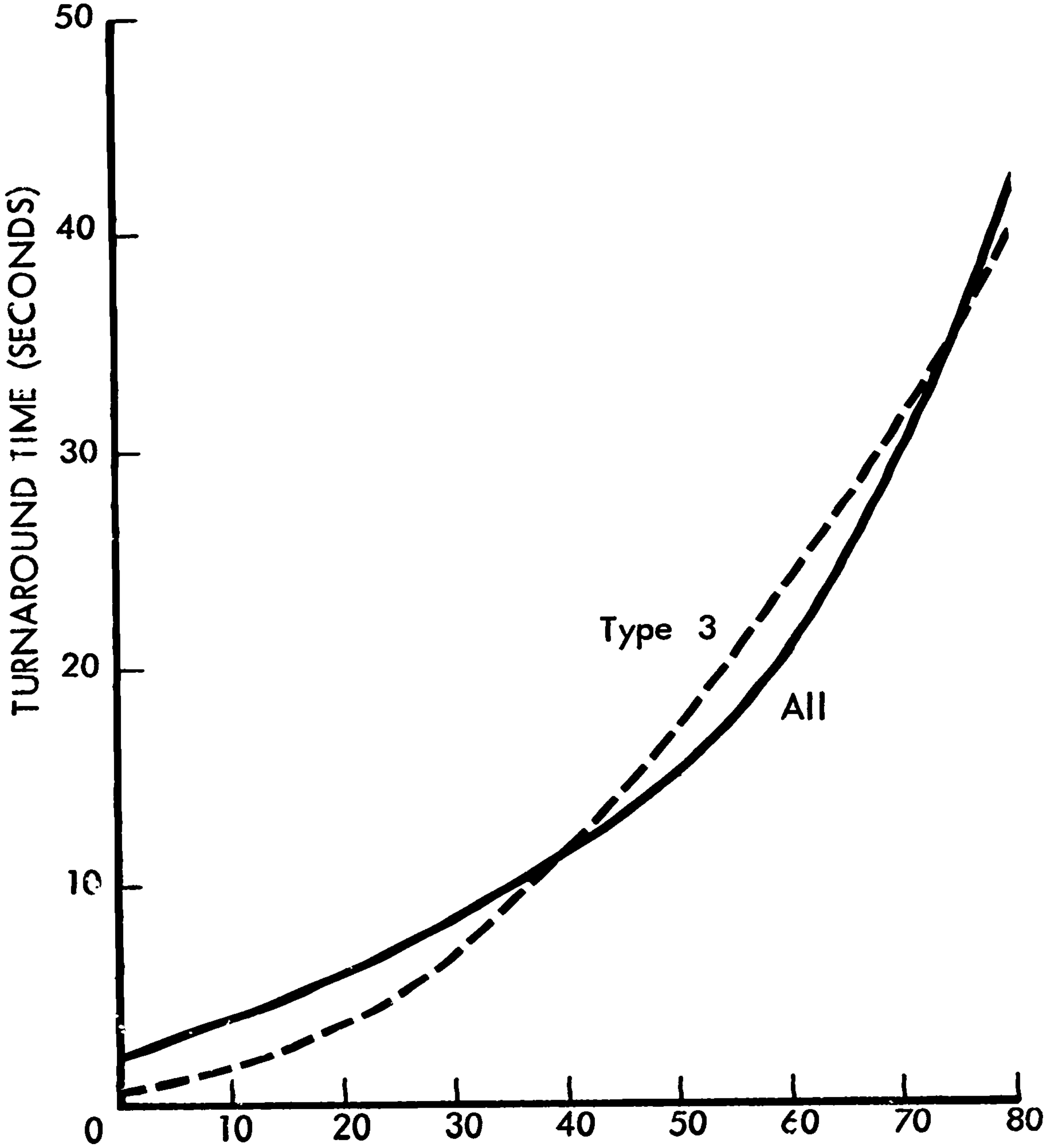
EFFECTIVE INTER-ARRIVAL RATE
(PER MINUTE)
JOB TYPE ONE VERSUS ALL

GRAPH 9.4



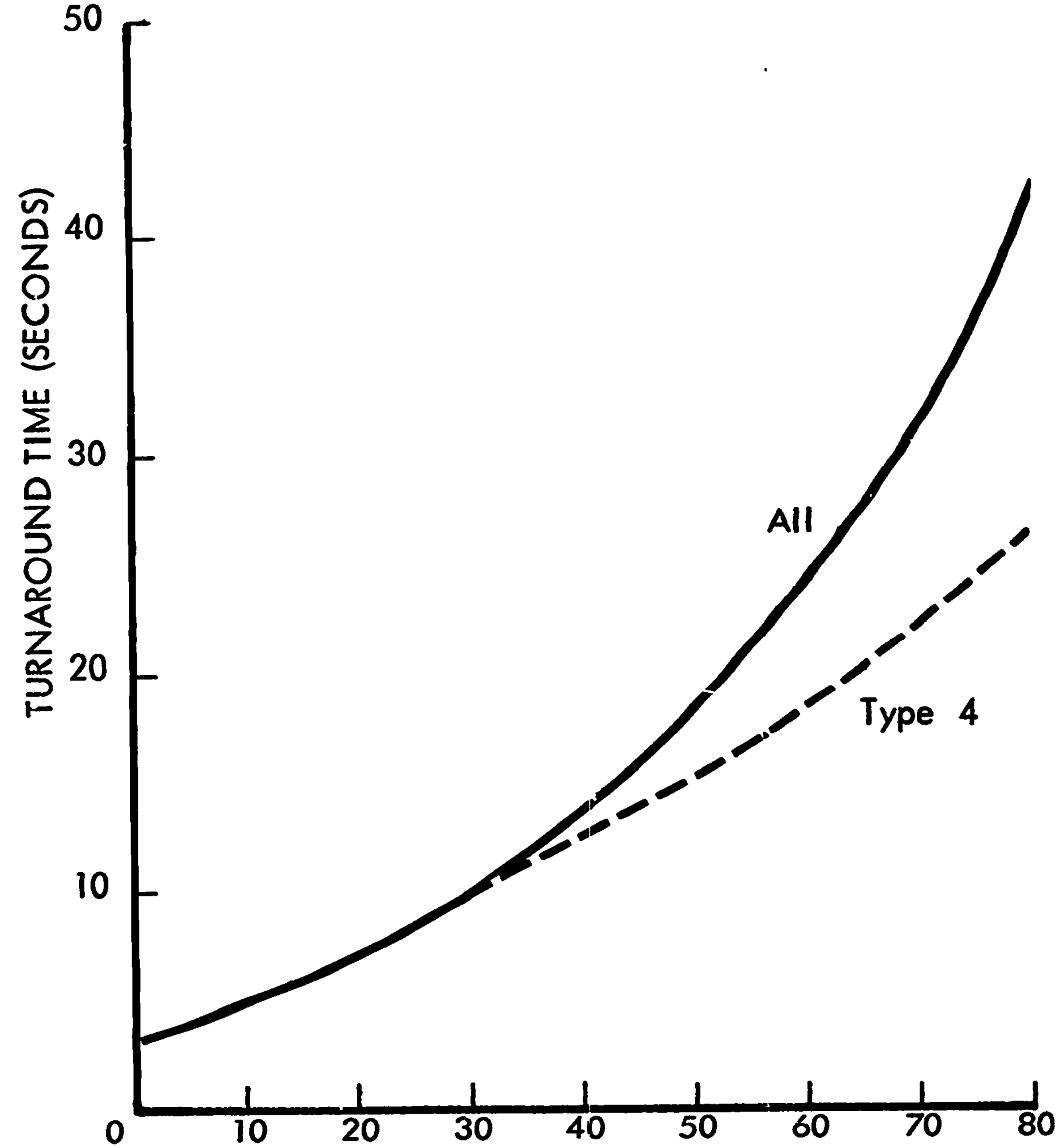
EFFECTIVE INTER-ARRIVAL RATE
(PER MINUTE)
JOB TYPE TWO VERSUS ALL

GRAPH 9.5



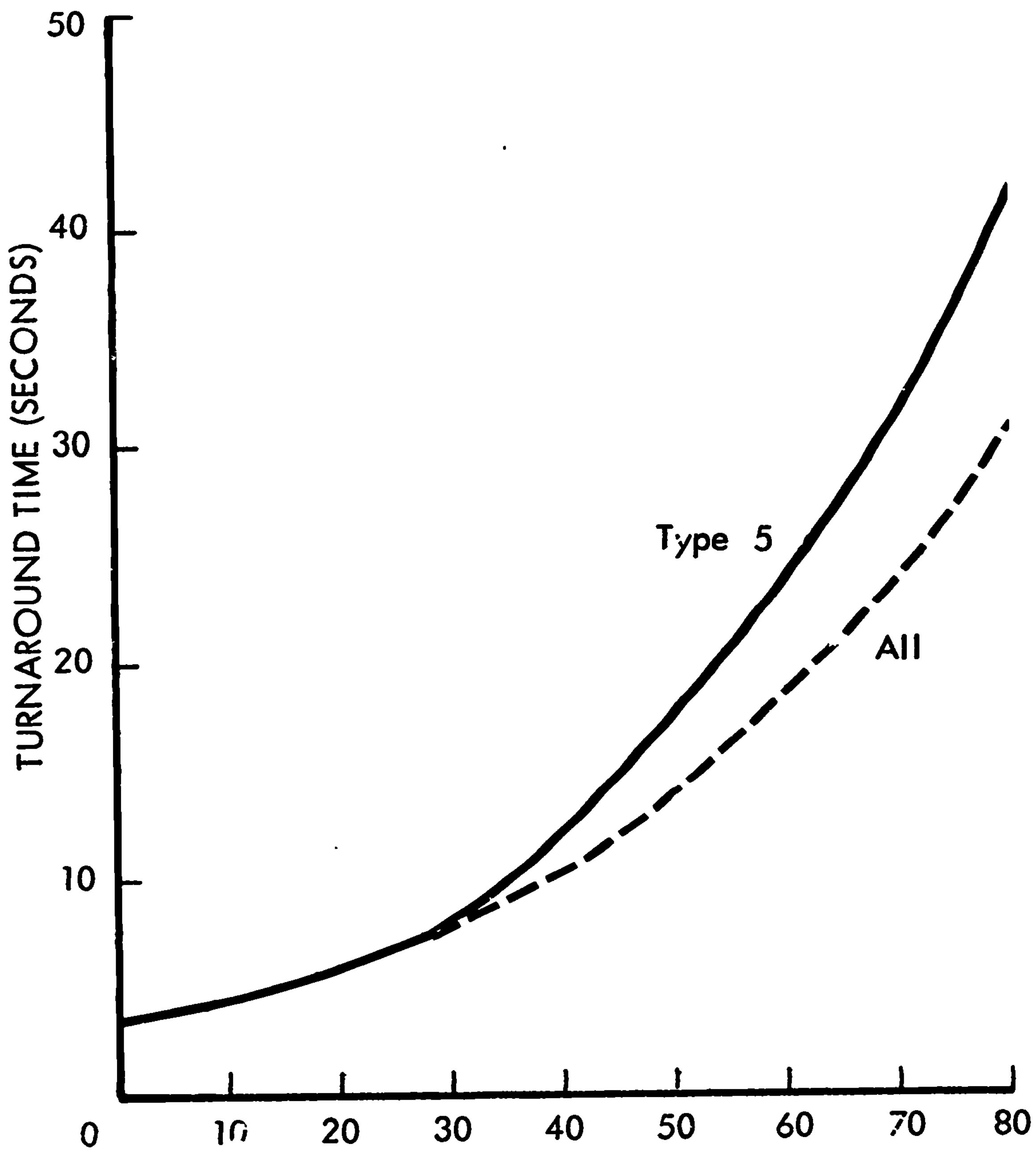
EFFECTIVE INTER-ARRIVAL RATE
(PER MINUTE)
JOB TYPE THREE VERSUS ALL

GRAPH 9.6



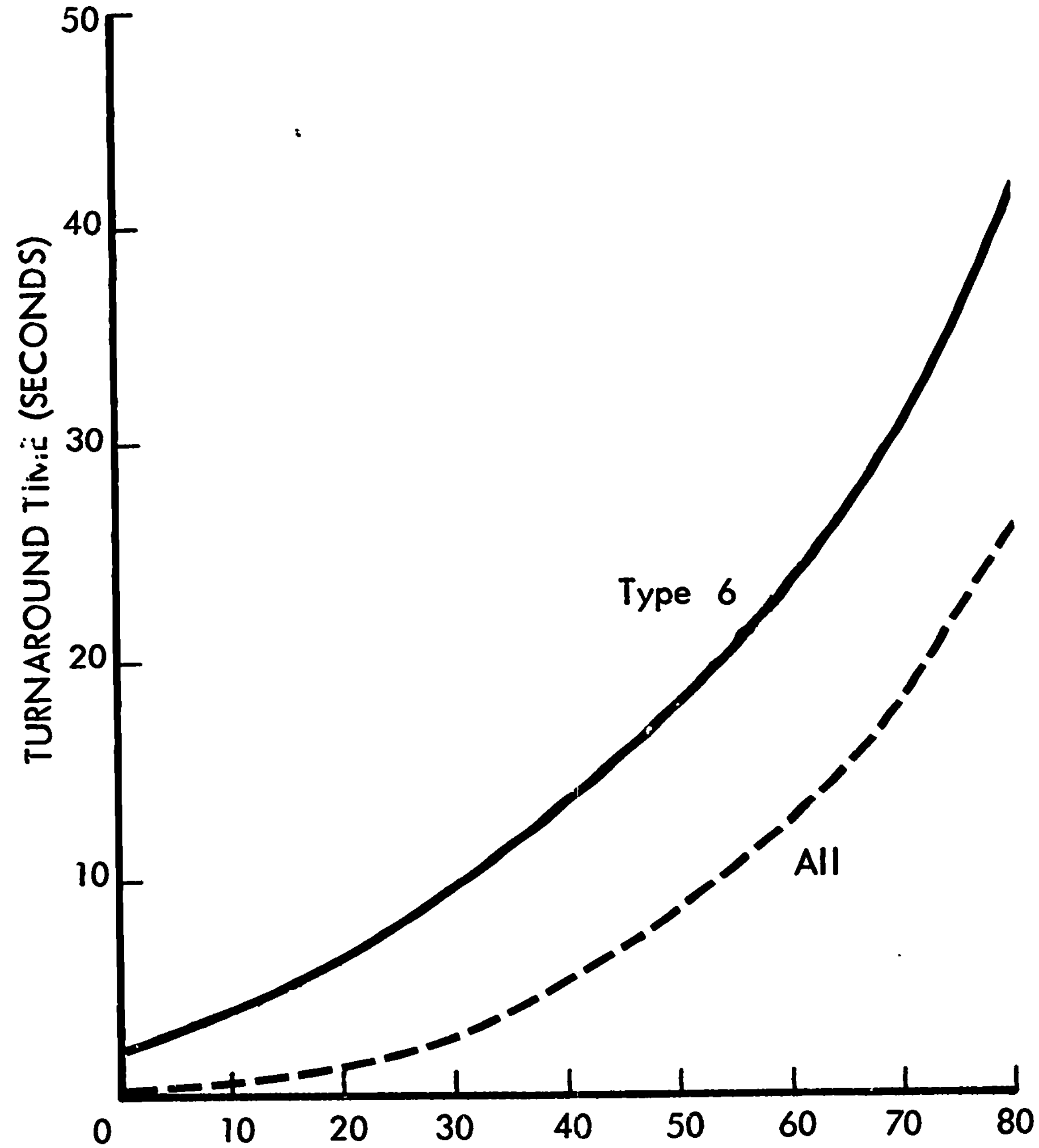
EFFECTIVE INTER-ARRIVAL RATE
(PER MINUTE)
JOB TYPE FOUR VERSUS ALL

GRAPH 9.7



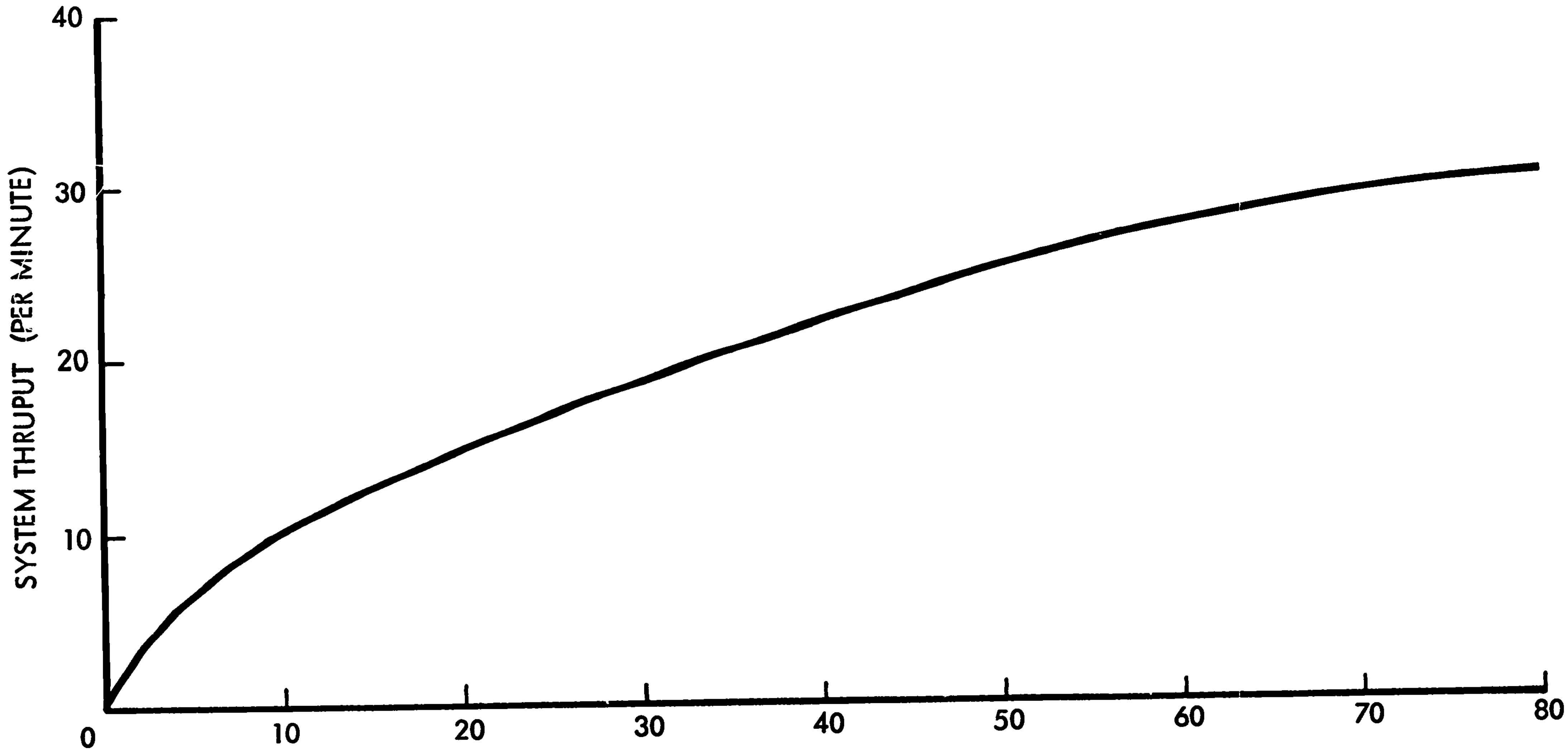
EFFECTIVE INTER-ARRIVAL RATE
(PER MINUTE)
JOB TYPE FIVE VERSUS ALL

GRAPH 9.8



EFFECTIVE INTER-ARRIVAL RATE
(PER MINUTE)
JOB TYPE SIX VERSUS ALL

GRAPH 9.9



EDUCATIONAL MULTI-PROGRAMMING SYSTEM THRUPUT

GRAPH 9.10

9.4.3 Analysis

Large core size can always improve the system thruput capability. However, if the core size is large enough to store eight jobs concurrently, then additional core space will not be able to improve the system performance significantly. The system thruput increases steadily as the number of terminals is increased until the number of terminals approaches 55.

Larger processor slice will reduce scheduler overhead and hence improve the system thruput capability. However, a time slice, such as two seconds, will not affect the system overhead significantly, but it can improve the job turnaround.

A high performance file can cut down core storage requirement. Since file access time (including waiting time) is calculated as part of the time a job will stay in the system, it can also affect the job turnaround time significantly.

In the simulation output, 100% utilization of the work area has not been achieved, even though there is a long waiting list of work area requests. EMPSim usually does not have paging facility; it does not allow a job to be scattered over discontiguous areas. In fact, most systems partition their core spaces into segments. It is very likely that a 4K segment of core may be allocated to store a program of 0.5K. The effective core utilization, therefore, is usually very poor.

EMPSim is designed to handle a larger work load than that of ETSSim. Because its turnaround time requirement can be as long as five minutes, the effect is to level off the work load during the peak hours. For example, if the processor is overloaded, it can take hours to work off its queue. The queue length will probably increase as long as the thruput is less than the total traffic generated from the various terminals.

APPENDIX I

Student Program Running Time Estimates

GENERAL LEARNING CORPORATION
Washington Office
September 1, 1967

OE/CCF TECHNICAL MEMORANDUM

TO: OE/CCF Study Team

FROM: P. Galidas

SUBJECT: Student Program Running Time Estimates

Estimates of the expected running times of computer programs written by students to solve problems were obtained from USAFA data. The amount of data made available was very large so that confidence in the statistics derived from the data is high. However, the total sample itself is small when viewed in the broader context of the present study. That is, the data represents the performance of a selected group of 14th, 15th and 16th grade students who are highly motivated. Whether the performance of the hypothetical population of 100,000 in grades 9 through 16 will be similar is seriously open to question. Although the USAFA computer was a two-processor B5500, and the language in which most programs were written was ALGOL, it is believed that the "translation" of running times on the B5500 to times on machines of the type under consideration in the study is much less open to question. To a good first approximation, the running times of similar programs are directly proportional to the cycle times of the machines on which they are run, all other things being equal. What _is_ open to question is the "similarity" of the programs.

The most important defense of the use of the USAFA data, in spite of serious reservations, is that the Academy was the only source of hard data among the institutions surveyed. Without these data, the estimates of running times would have been dubbed "guesstimates".

Table A 1.1 is a summary of the means and sample sizes of running times per run and runs per problem for seven USAFA courses; three second-year courses, one third-year course, and three fourth-year courses. The sample size of one fourth-year course (Code SC 9) is much too small to be statistically significant.

A computer program was written in GE-Dartmouth BASIC to perform the statistical analysis. The raw data is appended as Attachment 2 to this memorandum. Attachment 1 describes the raw data and the major assumption made in interpreting them. The data consisted of the total

CPU, I/O and overhead times for one or more runs of a program for a given problem. The individual times per run are not available. For each problem, total times were divided by number of runs to obtain the time per run. For example, if there were 7 runs for a problem, and the total CPU time was 70 seconds, it was treated as 7 runs of 10 seconds of CPU time per run.

Referring to Table A 1.1, it is observed in the column headed "Runs/Prob" that the average number of runs per problem for the three second-year courses (S7, S0, and S4) was between about 5 and about 7. Disregarding the extremely small course, SC9, the other three courses had Runs/Prob of between about 10 and about 12 or about double those for the earlier course. It can also be observed that the CPU, overhead, and I/O times for the second-year courses were generally lower than the corresponding statistics for the third and fourth-year problems. This was reflected in the assumed times per run used in the model described in the previous memo, except for overhead time.

It should be noted that the overhead times given in the raw data are not time intervals which were actually observed, but are prorated overhead "charges" made by the operating system. The means in Table A 1.1 seem to indicate that the overhead charge is approximately the sum of CPU time plus I/O time. This type of overhead scheme was not assumed in estimating number of consoles and CPU's required. Instead, a "flat" one second of overhead was assumed for the MP system. One of the reasons is that a central processor about ten times as fast as the B5500 was assumed, but the assumptions about the I/O devices were that they would not be substantially faster. A second assumption is that the compiler(s) will be efficient enough to make a flat overhead of one second/ run essentially correct.

P. Galidas

P. Galidas

Table A1.1 USAFA Data Means

USAFA Code (Course No.)	No. of Observations			Runs/Prob.			CPU & O'H secs/run			CPU sec/run			O'H secs/run			I/O secs/run		
	Total Probs.	Sample Probs.	Sample Runs	C	E	C+E	C	E	C+E	C	E	C+E	C	E	C+E	C	E	C+E
S7 CS 230	100	70	396	3.6	2.1	5.7	36.9	47.4	40.7	12.8	12.4	12.6	24.2	34.9	28.1	12.6	18.7	14.8
S0 CS 240	713	192	1,029	3.2	2.2	5.4	24.6	42.3	31.7	8.3	12.6	10.0	16.4	29.5	21.7	10.1	16.6	12.8
S4 CS 254	961	150	1,033	4.2	2.7	6.8	21.4	42.7	29.8	7.2	13.9	9.9	14.2	28.6	19.9	9.6	15.6	12.0
SC3 CS 381	106	103	1,188	5.1	6.4	11.5	48.7	11.3	84.3	13.0	29.3	22	38.7	83.6	62.3	16.9	27.0	22.5
S1 AST 451	77	71	886	6.1	6.3	12.4	31.5	57.5	44.7	13.0	17.2	15.2	18.5	40.3	29.5	11.7	19.0	15.4
S3 AST 453	448	98	987	4.9	5.2	10.1	37.9	66.2	52.4	15.0	22.2	18.7	22.9	44.0	33.7	13.1	22.9	18.1
SC9 CS 499	14	11	54	3.1	1.8	4.9	88.6	104	94.3	24.4	20.7	23.1	64.2	83.1	71.2	22.3	24.7	23.2

C = Compile only data
E = Execute data (includes compile time)
C+E = Compile and/or execute data
O'H = Overhead

ATTACHMENT 1

The Raw Data - Its Interpretation and Reduction

It was desired to obtain the means and frequency distributions of certain computer running times for problems assigned to 2nd, 3rd, and 4th-year students at the USAFA. Refer to Attachment 2 - Raw USAFA Data. The first column is a list of the cadet charge numbers for use of the computer. The first two digits ("S4") are a code number for the course, the final digit is the code number for the particular problem (one of several) for which the cadet wrote a computer program.

The next column (partly obliterated) is a list of the student's names. The "mail box" number is the number of the "pigeon-hole" to which the student's output is delivered for pick up at his convenience.

Referring to the first line, the four columns headed COMPILATIONS mean the following: the student made 11 compilation runs, which took 83 seconds of CPU time, 111 seconds of I/O time, and the prorated overhead charge for compilations was 149 seconds. The next four columns say that 4 (of the 11) compilations went through execution, and that the execution times were 27, 29, and 293 seconds of CPU, I/O, and O'H.

The statistics of interest were number of runs per problem and running times per run. Runs/prob were directly available. Times/run were not. The best estimates of these were averages. Thus, 83, 111, and 149 were divided by 11 to yield average CPU, I/O, and O'H secs/run for the compilations of 7.55, 10.1 and 13.6, and for the executions 27, 29, and 293 were divided by 4 to yield 6.8, 7.3, and 73.3. Since 4 of the compilations were those preceding the executions, the number of compilations was reduced by 4 to yield 7 runs which were compilations only (no execution), and the original average execution times were increased by the compilation times to yield 14.4 (=7.55 + 6.8), 17.4 (=10.1 + 7.3), and 86.9 (=13.6 + 73.3) which represent the average running time per run including both compilation and execution.

Attachment 2 Raw USAFA Data

Course Code	Problem Code #	Student Name	Mail Box #	COMPILATIONS No. of Runs	COMPILATIONS CPU Secs	COMPILATIONS I/O Secs	COMPILATIONS Prorated Overhead Secs	EXECUTIONS No. of Runs	EXECUTIONS CPU Secs	EXECUTIONS I/O Secs	EXECUTIONS Prorated Overhead Secs
S4M1	AB3	F	24	11	83	111	149	4	27	29	293
S4M1	AB5	F	249	16	107	155	214	5	11	32	35
S4M1	AC0	H	250	1	9	13	8	1	61	5	1
S4M1	AC1	H	250	9	39	69	76	5	26	23	41
S4M1	AC2	H	250	5	37	52	85	4	5	24	26
S4M1	AC3	H	250	19	112	193	310	10	24	55	95
S4M1	AD0	J	250	1	9	12	6	1	62	5	10
S4M1	AD1	J	250	6	57	79	119	3	22	14	29
S4M1	AD2	J	250	3	19	23	57	0	0	0	0
S4M1	AD3	J	250	1	6	9	5	0	0	0	0
S4M1	AD5	J	250	16	98	138	174	7	10	39	58
S4M1	AE0	K	251	1	7	11	5	1	57	5	20
S4M1	AE1	K	251	12	56	107	144	7	23	56	84
S4M1	AE2	K	251	10	96	88	144	4	13	24	90
S4M1	AE3	K	251	1	5	5	8	1	4	2	22
S4M1	AE5	K	251	9	56	79	112	5	19	27	38
S4M1	AF0	L	251	2	14	24	68	1	57	5	29
S4M1	AF1	L	251	8	40	77	91	4	25	17	17
S4M1	AF2	L	251	16	130	150	196	7	14	43	78
S4M1	AF3	L	251	3	127	20	91	2	2	6	14
S4M1	AF5	L	251	11	63	105	135	4	11	28	21
S4M1	AG0	P	252	1	10	17	23	1	57	4	6
S4M1	AG1	P	252	17	83	140	173	7	35	30	146
S4M1	AG2	P	252	10	60	81	115	3	4	11	31
S4M1	AG3	P	252	8	42	52	104	3	9	15	93
S4M1	AG5	P	252	16	128	161	205	7	41	93	84
S4M1	AH0	H	252	3	23	40	34	1	56	5	14
S4M1	AH1	H	252	11	59	104	147	4	19	25	68
S4M1	AH2	H	252	8	54	82	107	4	8	21	41

APPENDIX 2

Administrative Data Processing Assumptions

Administrative Assumptions

1. The Central Computer Facility will serve more than one school administrative unit. Each administrative unit will control its own data.

   Implications:

   a. "Standard" applications must permit considerable flexibility.
   b. Schools may do some "unique" things with their data.
      1. Own forms when real reason exists.
      2. Special computer programs written for a school. (For real reason and at the school's expense)
   c. Intensive effort to maintain generality in administrative development work.
   d. Intensive effort to train school people to manage their data remotely.

2. The Central Computer Facility will probably not be an integral part of any one of the school administrative units served.

   Implications:

   a. Careful balancing of control power so that all school administrative units feel they are being treated fairly.
   b. Careful cost-accounting and billing, so that no crucial part of the operation need be hidden in the administrative costs of a "host" school administration.

3. Most schools served will be relatively traditional in outlook and services required, at least at the beginning of the operation of a Central Computer Facility.

   Implications:

   a. Application, volume, and frequency forecasts may be made from what is now being done in advanced school data centers.
   b. Batch processing of administrative data will be satisfactory.
   c. Existing applications development can be capitalized on to shorten initial development time.
   d. New and improved ways to do traditional things will be acceptable.

4. A few schools will be actively experimenting with organizational and curricular patterns, or with the uses of information itself, and therefore require significantly more service from a Central Computer Facility.

   Implications:

   a. Much heavier demand for some standard services, e.g., scheduling, test scoring.
   b. Some programming capability to meet unanticipated, non-standard requests.

5. Administrative data systems will be continually evolving.

Implications:

a. Demands for machine time for administrative services will keep on increasing, even if student population remained static.
   1. Initial demand will be less than can now be foreseen due to time required to phase in new administrative systems.
   2. Demand in a very few years will exceed the machine capacity to do what is now foreseen.
b. Systems development, programming, machine time, and training resources will be needed, not only to implement what is now being done elsewhere but to facilitate new developments.

6. Once data files are in electronic storage, some capability will be needed to interrogate them during the daytime, even though most administrative processing will be done at night, e.g., input data editing (to allow rapid error detection and correction), queries and individual students, employees, accounts, etc.

Implications:

a. Basic data files will be essentially "on line" at all times.
b. Data input and editing will need to be done "background" simultaneously with student program processing.

7. A school complex (private schools included), with 100,000 students in high school and college (grades 9-12, 13-16) will have about

70,000 students 9-12
30,000 students 13-16
192,000 students K-8

292,000 total administrative load

Calculations based on 1966 Digest of Educational Statistics and supplementary information obtained by P. Galidas.

8. Batch-type overloads may be transferred to outside machinery for extra C.P.U. time, printer capacity, etc.

Implications:

a. Installed machinery need not be adequate to meet all anticipated peak loads.
b. Installed machinery and data file structures must be reasonably compatible with other machinery readily available.

9. Administrative work will be of two different types:

   a. Tightly controlled jobs, of predictable arrival time, accuracy. These will be primarily accounting type jobs.
   b. Less tightly controlled jobs, whose arrival time is less likely to be predictable, and with more errors in the input data. Most student records work will be of this nature.

   Implications:

   a. Administrative workload demand at any point in time will not be completely predictable, regardless of previous scheduling.
   b. Sophisticated error detection and correction techniques will be needed.
   c. Some jobs will need to be "done over" when data is corrected (e.g., financial accounts don't balance).
   d. Some jobs, such as scheduling, will be planned deliberately to be interative, where the number of iterations cannot be determined precisely in advance.
   e. Arrival times of administrative jobs are best described by a probability distribution.

10. To reduce the amount of electronic storage needed for administrative data, a "bucket" system is proposed. Each student, for example, would have a varying number of relatively short logical records, each identified as to type and date, instead of one much longer logical record with space for adding information as it becomes available. Thus, an elementary student record will have relatively few characters of information compared to a high school or college student record, and no electronic storage space is used until it is needed.

    A further advantage of the "bucket" system is the case of adding special information for some students. There need be no requirement or electronic storage r rved for all students having the same information.

11. It will be the role of the Central Facility to help each school district learn how to use the services available. Each school administrative unit will be responsible for determining how to use the information and processing capability available to it.

Outline of Administrative Applications

The following outline is a list of activities which may be included in a computerized administrative information system. The outline contains applications implemented in currently operating systems. It is expanded to also include additional applications which may be implemented in future systems.

The outline of the administrative information system is divided into the following sub-systems:

- Student Accounting
- Budget Accounting
- Employee Accounting
- Equipment Accounting
- Auxiliary Services Accounting

Each sub-system is described as a series of activities with each activity further divided into a series of transactions.

This outline was used in selecting the administrative activities described in Section 6 of the main report.

<u>Student Accounting System</u>

Demographic Records
Academic Performance
Schedule
Extracurricular Records
Health Records
Guidance
Financial Accounting Records
Attendance Records
Cumulative Record

- Demographic Records

    Census including all children (to the state)
    Age - grade enrollment report for only school children (to the state)
    Enrollment report
    Enrollment projection
    Family list

- Academic performance

    Classroom performance
        Report card
        Anecdotal record
    Standardized testing
    Ranking

- Schedule

    Course load report
    State annual report

Master Scheduling
- Feasibility simulation
- Schedule report
- Conflict report
- Class list
- Homeroom list
- Study hall list
- Grade level list

Individualized student schedule

- Extracurricular Records

  Extracurricular activity report
  Part-time job report

- Health Records

  Emergency record
  Physical handicap report
  - Permanent handicap
  - Temporary handicap

  Basic health record

- Guidance

  Performance profile
  Personality profile
  Data investigative reports
  - Specific student reports
  - Relationship reports (research)

  Transcript
  Placement
  Automated counselling

- Financial Accounting Records

  Activity fund reports
  Fee reports

- Attendance Records

  "Twice-daily" attendance report (Student/Homeroom)
  "X" period attendance report by reason for absence
  Special purpose attendance reports
  State annual (or more often) attendance report
  Attendance register

- Cumulative Record

    Demographic records - cumulative and updated
    Academic performance - cumulative
    Extracurricular - cumulative and updated
    Guidance - cumulative
    Attendance - cumulative

<u>Budget Accounting System</u>

- Budget development

    Program statement
        Program goals
        Program objectives
        Program elements
    Program cost
        Current expenditures
        Projected expenditures
    Anticipated revenues
        Local
        State
        Federal
    Budget analysis
    Budget document

- Budget implementation

    Expenditures
        Plan vs. actual
        Payroll
        Accounts payable
        Purchasing
    Revenues
        Plan vs. actual
        Accounts receivable
    Program evaluation
        Program cost analysis
        Program opinion surveys
        Program performance surveys

<u>Employee Accounting System</u>

- Employee personnel file
    Confidential reports (evaluations and assessments)
    Demographic reports

Experience resumes
Professional credentials
In-service activity reports
Career advancement reports
Community participation reports
Miscellaneous activity reports
Staff analysis reports
Employee health record and reports
Employee leave records

- Employee schedule and load reports
    - Load reports
    - Assignment reports
    - Scheduling reports

- Employee supervision reports
    - Evaluations
    - Staff development
    - Substitute teacher file
    - Employee negotiations
    - Contracted services

Equipment and Facilities Accounting System

- Building and grounds reports

    - Expansion and alteration reports
        - Projections
        - Proposals and specifications
        - Current projects
    - Maintenance reports
        - Projections
        - Regular preventive maintenance
            - Maintenance scheduling
            - Status reports
            - Contracted service
        - Replacement requirements
    - Inventory reports
        - Inventory control reports
        - Real estate report (to state)
        - Depreciation reports
        - "By whom - when used" reports

- Moveable equipment and supplies reports

    - Equipment additions and replacements

        Projections
        Proposals and specifications
        "On-order" reports
    Equipment maintenance
        Projections
        Regular preventive maintenance
            Maintenance scheduling
            Status reports
            Contracted service
        Replacement requirements
    Inventory reports
        Inventory control reports
        "Stock room" reports
        Depreciation reports
        "By whom - where used" reports

Auxiliary Services Accounting System

- Transportation scheduling
- Insurance program identification
- Cafeteria accounting system

APPENDIX 3

Systems and Application Standardization Across Many Regions

SYSTEMS AND APPLICATION STANDARDIZATION ACROSS MANY REGIONS

Repeated and widespread evidence from industry indicates that half or more of the costs of data processing installation are not for hardware, but for software in one form or another. Systems analysis of the situations to which automated techniques are to be applied, e.g., programming, documentation, training of people to interact effectively with new systems, and operation of new systems, all take time and money.

Traditionally, industry does not share its systems and programming work to any great extent, even though many similar situations exist in which the work of one company might be almost directly useable. At an abstract level, there may be some sharing of techniques of using a given piece of hardware, the principles of systems development, and occasionally actual algorithms of a scientific nature. Occasionally a service bureau will develop relatively standardized applications which are used by many companies, none of whom control the actual computer programs.

It might be expected that education would have no real economic reason not to share systems and programming work; in fact, there appears to be every economic reason to do as much sharing as possible. It would be expected further that there would be relatively little logical difference among the many ways school data are handled, even though outward appearances may be quite different. However, current evidence is that each administrative unit in education tends to develop its own data processing procedures and programs from scratch, with relatively little attention to what might already be available. On occasion, a "canned" package will be accepted from a machine vendor, or a utility routine (such as card to tape) from another user. But the basic pattern is one of programming, in some instances, without even doing a good analysis first.

There are several possible reasons for this state of affairs in education. An obvious one is that there is no central authority in public education which can say exactly how something is to be done, although occasional smaller geographic regions may have a county or state agency that is trying to function this way. Any centralized systems and program development effort must rely on voluntary acceptance today. This is a problem in all aspects of education, not just data processing. Any attempt to use data processing to force a degree of control in education, or one that is perceived as such, may meet with determined covert, if not overt, resistance because of the traditions of local autonomy that exist.

There are some genuine local differences that must be considered, at least as long as local autonomy, differences in state laws, and vigorous programs of experimentation and development in education exist. If the legal requirements for attendance accounting, for example, are radically different in two states, the data processing requirements, and hence the systems and programs developed, may have to be different. If a school wishes to have a Stanford type of flexible schedule, the computer programming and data file requirements can be quite different from those of a school wishing a traditional schedule.

Another problem is that of machine configuration differences. Any centralized systems development effort that includes computer programming would tend to dictate what computer equipment schools could rent or buy, much to the dismay of makers of other equipment. A possible solution to this problem is to establish program requirements that must be met by any vendor wishing to sell data processing and computing equipment to schools. The vendor must be restricted from using machine features that lead to incompatibility, although he may use nonstandard machine features that improve standard operations. *

The argument can be made for programming exclusively in a machine-independent higher level language, such as COBOL, FORTRAN, or PL/I. The problem here is that true machine-independence rarely exists. A file structure required for one computer may be very inefficient on a different computer. Higher level languages do not yet give efficient object programs on many computers; simple pressure of demand for machine time for many jobs would soon force many managers of school installations to increase their computer capacity or to reprogram so that far less machine time is required. Higher level languages also make it harder in many cases to manipulate data containing errors. **

These arguments would indicate that any attempt at centralized systems and program development for large segments of education will probably have to be a pluralistic effort, keeping certain data files comparable in definition, and preserving some common logic for processing where this can be ascertained. It should still be possible to reduce the systems and development effort in education by a large factor, perhaps exceeding 50%, if a way can be found to centralize this effort without destroying the ability to do important things differently that educators seem to value. Any such effort must also realize the dynamic effect of educational changes, and be prepared to facilitate rather than retard them.

---

* For example, the compressed tape feature of the IBM 7070 can improve hardware performance, but results in data files on tape that cannot be read properly on other machines.

** For example, a simple job such as tallying the number of requests for each offered course on the IBM 1401 computer requires approximately twice as much core storage and ten times as much operating time if programmed in Fortran instead of Autocoder. Further, it is inconvenient, if not impossible, to provide for on-the-spot detection and correction of mispunched cards in Fortran. More sophisticated higher level languages will remedy some of these problems.

Two more aspects of the problem still need to be considered. One is that a local systems analysis in many cases does serve a good purpose regardless of centralized systems and programming efforts. Smoothing the transition from manual or more primitive automated systems to a higher level system requires ferreting out who uses what information for what purposes and seeing that these people are correctly integrated into the new system. Such an analysis should bring to light the local power structure, and establish a reasonable compromise with or accomodation to it. In some cases, it may be desirable for the top school operating official (superintendent) to review the situation in his district carefully.

An observable characteristic of many of the data processing managers in schools is great pride in what they have developed. Perhaps this is related to some of the psychological insights resulting from the use of the "discovery" method of teaching: enthusiasm for what was discovered (through personal involvement), and a tendency to hang on to an erroneous conclusion quite defensively. A correlate of the pride of development is the attitude that "if I didn't write it, it isn't any good." Such attitudes are furthered by more or less typical school situations in which the people responsible are glad that something works at all, and its working is such an improvement over previous manual methods, that the new system is not critically analyzed.

The predictable consequence of this is reinvention, duplication, and a tendency to meet immediate problems in the local school district rather than to undertake the development of a more thorough information system. The only apparent solution to such problems would seem to be to help top school administrators become informed about computers and information systems to the point where they can make better decisions, or at least understand the consequences of decisions proposed by others on the school staff. Both the promise of potential benefit to education and the high cost of using computers would indicate that such an effort be undertaken.

To summarize, the best prospects for achieving the benefits of data processing at lower cost through centralized systems and program development work would appear to require:

a. Pluralistic systems development, allowing for a high degree of flexibility where this may be valuable, yet providing for comparable data, and readily available computer programs and school procedures for most situations.
b. Requiring vendors wishing to do business with schools to guarantee implementation of computer programs within the above framework in a reasonably efficient way on their machines.
c. Bring school administrators to a higher level of understanding about computers and data processing, so that decisions affecting both a

centralized development effort and the impact of data processing in the local situation could be made appropriately.

d. Developing guidelines for schools to do those parts of their own systems analysis work necessary for the effective functioning of administrative data processing regardless of where the systems, computer programs, and procedures were developed.

e. Guaranteeing that centralized systems and program development would not be used as a tool of political control, and would serve the schools as well as various government and other agencies.

APPENDIX 4

Transmission System and Teletype Analysis

I. Transmission System

A. Common Carrier

Interstate communications traffic is regulated by the Federal Communications Commission. Intrastate traffic is regulated by a state utility board and in some large cities by a city agency. The major carriers are Bell and Western Union. Interspersed with the larger systems are several smaller independent companies. The services available vary with the company equipment; thus, it is important to check with the local company to determine exact services available. For example, not all communities served by Bell are on the Touch-Tone System. Also, available equipment and services can affect rates.

B. Voice Band Facilities

These facilities use channels with bandwidths of about 3,000 to 4,000 cycles per second. The public telephone network uses channels with a bandwidth of about 3,000 cycles per second. Leased lines usually have bandwidths of about 4,000 cycles per second. Public switched service provides the user with access to a communications network by dial and/or push button. Leased service provides the user with exclusive use of the communication line. Narrow-band facilities provide data communications at up to 300 bits per second.

C. Transmission Channels

The rate of transmission in bits per second is related to the bandwidth of the circuit. As a rule of thumb, the maximum bit rate is usually half the bandwidth, which is expressed in cycles per second. The number of bits per cycle is determined by the efficiency of the digital subset and quality of the line. For most dialed channels in public voice networks, 1,200 bits per second for a 3 KC voice channel is available. The voice channel can be subdivided into several 150 cycle teletype channels, thus making it possible to transmit several teletype signals simultaneously over one voice line.

The equipment used in teletype remote terminals may be made up of a keyboard-printer and a remote control unit, supplemented when needed by a perforated tape reader and perforated tape punch. The keyboard-printer serves for transmitting and receiving intelligence, while the remote control unit accomplishes the functions inherent in the setting-up and breaking-down of connections.

The TTY terminals may be located up to ten miles apart. Telegraph transmission for distances greater than 10 miles depends on a long distance telegraph circuit in which digital subsets are used for connection with the line-terminating circuit.

In transmission, synchronization between the transmitter and receiver must be maintained to indicate the beginning and the end of a character; the start and stop bits are used for this purpose.

The speed of transmission is determined primarily by the remote terminal and available bandwidth to handle the transmitted character. Most teletype circuits have sufficient bandwidth to handle transmissions at 75 bits per second; the limitation is in the electromechanical line-switching circuits. However, through use of private lines, it is possible to transmit 110 bits per second (equal to 10 characters per second).

The eight level code specified here allows for control of errors that may arise in transmission.

D. Preliminary Specifications

1. System Specification
   a. Eight level code to consist of:

Start bits	1	
Data bits	8	
Stop bits	2	(In some instances, only one bit may be used)
	11	Bits character

   b. System to consist of TTY, Group MUX, Central MUX, and CPU
   c. TTY's to be remote up to 200 miles distance from CPU
   d. Voice band transmission lines to be used
   e. Dedicated transmission lines to be used between Group MUX and Central MUX

2. TTY Specification
   a. Each TTY to have its own channel to the Group MUX
   b. TTY to send and receive
   c. TTY to have manual input
   d. TTY input to be maximum of 10 characters per second (110 bits per second)
   e. TTY to be located at distances up to 10 miles from Group MUX
   f. TTY inputs 66 characters per line of type
   g. TTYs connected to Group MUX to total up to 32
   h. TTY input/output is direct from Group MUX

3. Group MUX Specification
   a. Group MUX to have connector for 32 TTY channels, full duplex, I/O

b. Group MUX to connect to Central MUX on dedicated lines, full duplex, 2,400 bits per second (bps)
c. Group MUX to receive 8 level code, 11 bits/character
d. Group MUX scanner designed such that there is no lost data
e. Group MUX may contain, bit, character, or work buffer on TTY I/O side
f. Group MUX may contain bit, character, or word buffer on input side from Central MUX
g. Group MUX to be capable of identifying which TTY is to receive or is sending data
h. Group MUX to be within fifty feet of data set

4. Central MUX Specification
   a. Central MUX to have connectors for 50 Group MUX, channels, I/O, full duplex
   b. Central MUX to output data to CPU and receive input data from CPU
   c. Central MUX to be capable of identifying which Group MUX and which TTY is to receive or is sending data
   d. Central MUX to be within 50 feet of data set
   e. Central MUX to be within 50 feet of CPU

5. Operation Specification
   a. TTY inputs to be manual or punched tape
   b. Inputs to be random
   c. TTY identity to be known at CPU for I/O of data
   d. TTY operator identity to be part of the message structure (software)
   e. System design to be such that TTY operator does not encounter "wait time"

6. Figure A-4.1 shows a schematic diagram of the specified system.

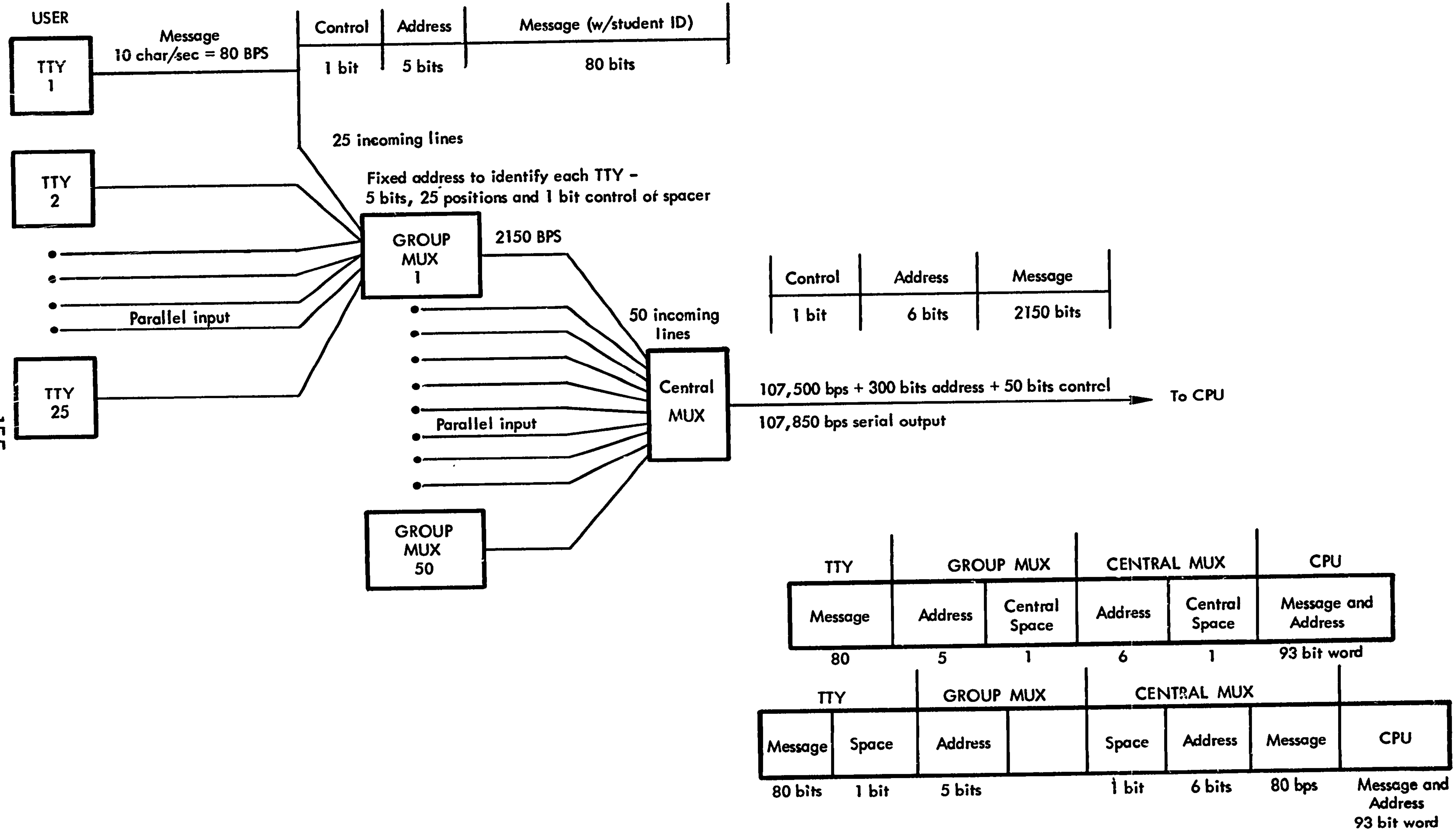


Figure A-4.1 Even Distribution Data Requirements

E. Supplementary Note

GENERAL LEARNING CORPORATION
Washington, D. C. Office

TO: D. G. Lewis

SUBJECT: Data Line Concentrator/Deconcentrator (DALCODE)

DATE: August 22, 1967

Discussion

DALCODE, a Western Union Design, is capable of receiving data on a real-time basis from up to 30 teletypewriters, concentrates the data and transmits over a 2400 baud line via data sets to a computer.

DALCODE consists of one or two concentrators/deconcentrators mounted in a single cabinet. The second unit, when used, provides redundancy. The concentrator combines the teletyped data into a single high-speed output channel which is transmitted over a voice grade channel.

The number of TTYs which can be accommodated by DALCODE for different data rates is shown in the table below.

Type of Low Speed Line		Number of Low Speed Lines DALCODE Can Accommodate	
Baud Rate	Bits/Character	Without Parity	With Parity
45.45	7.42	30	30
56.9	7.42	30	30
74.2	7.42	30	30
110	11	21	--*
150	11	14	--*
180**	11	11	--*

* Parity cannot be used with characters greater than 10 bits

** Higher baud rates are possible

Specifications

Output to Low-Speed Lines:	±6 V into 600 ohms
Output to Data Set:	±6 V into approximately 3.4 K ohms
Input Distortion:	42 % maximum
Power Requirements:	117 V, 60 Hz
Heat Dissipation:	2500 BTU/hr
Forced Air Cooling:	100 cubic ft/min (nominal)
Operating Temperature:	0^{o}C to 60^{o}C
Size:	72 3/4" H x 26" W x 29 1/2" D
Weight:	700 lb approximately

Example of DALCODE use plus cost

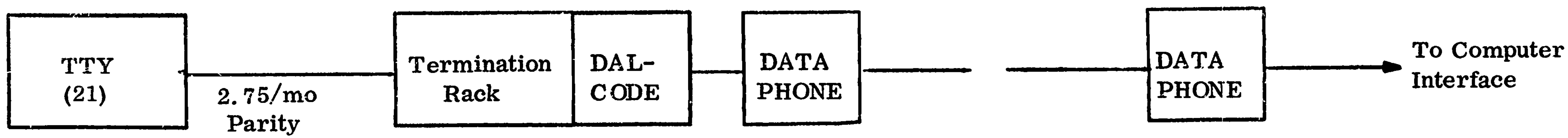
Next page.

S. Molello
Educ Svc Div

SM/smb

EXAMPLE

TRANSMIT OVER LEASED LINE VOICE GRADE



		Monthly Rental *	
21	Type 35 Teletypewriter KSR $ 35/mo **	$ 1,785	
21	Lines between TTY & Termination Rack ($2.75 ea/mo)	$ 58	(Nearest Dollar)
1	WU Termination Rack	$ 111	
1	DALCODE	$ 295 ***	
2	DATA PHONE Terminals	$ 88	
10 mi	Leased Line $ 3.00/mi	$ 30	
		$ 2,367/mo	

* Installation Charges not included

** KSR 33 Type Lease approx. $ 35/mo.

*** Charge when installed on WU Premise. May be higher when installed at user location.

II. Leased Teletypewriter Vs. Purchased Teletypewriter

A. Introduction

Teletypewriters may be leased or they may be purchased from the Teletype Corporation (Western Electric Affiliate). If purchased, the costs of installation and maintenance are additional, but free employee training in the use of the teletypewriter is provided. Teletypewriters may be leased through the company providing communication services. The costs of installation and maintenance are included in the rental fee.

Significant features of two teletypewriter models (ASR 33 and ASR 35), and typical purchase and rental costs, obtained from Auerbach Corporation reports and from local telephone and Western Union offices, are outlined below.

B. Comparison of ASR 33 and ASR 35

1. Significant features
   a. Function buttons - fewer on model 33 but model 33 can be modified to include several offered by the model 35
   b. Operation - on model 35 typing on tape will not interfere with incoming messages; function buttons permit switching
   c. Size - model 35 is larger in physical dimensions than model 33
   d. Noise - model 35 is less noisy than model 33
   e. Output - model 35: 80 or 85 characters/line
      model 33: 72 or 74 characters/line
   f. Use factor - model 33 is lighter in construction; therefore, operation longer than 2 to 4 hours/day is not recommended

2. Rental per month

a.	Switchboard charges vary by type of switchboard connection	$1.35 to $2.35
b.	Data Set	$25.00
c.	Model 33	
	KSR	$35.00
	ASR	$50.00
d.	Model 35	
	KSR	$85.00
	ASR	$125.00
e.	Autoreader control	$4.00
f.	Installation charges (one time)	
	Data Set	$50.00
	Teletype	$50.00
	Autoreader control	$50.00

3. Purchase Price
   a. Model 33

KSR	$460.00
ASR	$650.00

   b. Model 35

KSR	$1,400.00
ASR	$2,500.00

4. Operation
   a. Model 33 TTY is used on a dial-up system rather than on private leased lines, because of its light construction.
   b. This practice is based on the assumption that a dial-up system implies light usage and a private leased line implies heavy usage.

III. Report Card Printing via Teletype

A. Problem Definition

Design a system which will print out student report cards in 24 hours for schools having 4000, 2000, 1000 and 500 students each.

B. Distribution

Although there will be other types of information flowing between the various points in the system, only one type is described in this case. This information is the report card data to be sent from the central processor to each of the schools.

C. Volume

Each report card can have up to 20 lines of information with each line having as many as 80 characters per line. Therefore, the total number of characters to be sent to each school is as follows:

School	Population	Characters/Report Card	Characters/School
A	4,000	1,600	6,400,000
B	2,000	1,600	3,200,000
C	1,000	1,600	1,600,000
D	500	1,600	800,000

D. Language

The information to be printed on report cards will be in the form of random alphanumeric digits, and must be accurately transcribed. Thus, for data transmission, error detection and correction equipment is justified. Errors can occur in two areas, the human error and the transmission error.

The human error would occur in the preparation of the data. Some tests have shown this to be about one undetected mistake for every 1,000 to 3,000 typed characters. For purposes of this example, it is assumed that these errors would be subsequently corrected through verification procedures prior to data transmission. The second type of error, transmission error, can be caused by faulty equipment or by a flaw in transmission. Errors introduced through faulty equipment are usually easily detected since this generally causes several errors. The errors as a result of transmission flaws are more difficult to detect. On dial-up calls, there can be an error rate of about one character for each 10,000 transmitted. With error detection and correction equipment, about one error in 10 million characters would be transmitted. Since the report card contains a series of random characters, an error could make a significant difference in grade.

In view of the above, 1 in 6 report cards can contain an error when printed through use of transmission equipment without error detection and correction equipment. Thus, transmission equipment with error detection and correction capabilities should be used.

E. Preparation Time

Two examples of report card preparation time are given below. One is with a low speed teletype and dial-up system; the second is with teletypes, dedicated line, and Western Union DALCODE (concentrator/deconcentrator).

1. Teletypes and Data Phone

This system has the capacity to transmit 10 characters per second. The total number of characters to be transmitted for each report card is 1620 (1600 characters/report card plus 20 characters for line control and buffering). Thus, the transmission time for one report card is 162 seconds.

To this must be added some operating time for dialing the call, waiting for the connection to be established, and in some cases coordinating the transaction with the personnel at the receiving end. Assume this to be 100 seconds per call. A call is considered to be an uninterrupted transaction, completed from beginning to end, be it one report card or 3,000 report cards.

Further, it is assumed that the paper used in the teletype comes in 200 foot rolls perforated at four inch intervals and contains 5 copies (original and 4 copies) with some preprinted material thereon. Thus, 600 report cards can be prepared per roll of teletype paper.

The time to prepare 600 report cards (600 report cards/roll) would be 600 (162) = 97,200 seconds or 27 hours, or three hours longer than the 24 hour period specified. To prepare 500 report cards would take 500 (162) = 81,000 seconds or 22.5 hours. This would allow 1.5 hours in the twenty-four hour period for stripping and final processing of report cards for distribution to the students.

The number of teletypes required to print report cards in 24 hours at each school of the size established previously is as follows:

School	Teletypes to Print Report Cards in 22.5 Hours			
	500	1,000	2,000	4,000
A	1			
B		2		
C			4	
D				8

It is to be noted that the above teletypes would not be available for any other use during the report card preparation time. Also, no time is allowed for equipment malfunction, paper jams, etc.

2. Example B - Teletype, DALCODE, Data Phone

A school with 21 teletypes using a Western Union DALCODE (concentrator/deconcentrator) on dedicated lines would take the following times for report card preparation.

School	Time for 21 Teletypes to Prepare Report Cards of Quantity (Hours)			
	500	1,000	2,000	4,000
A	1.07			
B		2.14		
C			4.28	
D				8.56

F. Cost

1. Single teletype on dedicated line. (It is clear that the tie-up time is too long to use a dial-up line.)

1 - Leased line $3/mi. for 30 mi.	$ 90/mo.
1 - 35 model teletypewriter ASR	$ 125/mo.
2 - 103 F Data Sets	$ 50/mo.
TOTAL	$ 265/mo. *
Monthly cost for 21 teletypewriters (for comparison with example 2 below)	$5,565/mo. *

2. 21 teletypes, DALCODE, Leased Line

1 - Leased line conditioned 2,400 bps for 30 mi.	$ 90/mo.
21 - 35 model teletypewriter ASR	$2,625/mo.
21 - Lines between TTY and WU Termination Rack $2.75 ea/mo.	$ 58/mo.
1 - WU Termination Rach	$ 111/mo.
1 - DALCODE (WU)	$ 295/mo.
2 - Data Phone Terminals	$ 88/mo.
TOTAL	$3,267/mo. *

* Does not include installation charges.