

1979

# An evaluation and emulation of resource distribution in distributed processing networks

Russell Dale Anderson  
*Iowa State University*

Follow this and additional works at: <https://lib.dr.iastate.edu/rtd>

 Part of the [Electrical and Electronics Commons](#)

## Recommended Citation

Anderson, Russell Dale, "An evaluation and emulation of resource distribution in distributed processing networks" (1979).  
*Retrospective Theses and Dissertations*. 6434.  
<https://lib.dr.iastate.edu/rtd/6434>

This Dissertation is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact [digirep@iastate.edu](mailto:digirep@iastate.edu).

## INFORMATION TO USERS

This was produced from a copy of a document sent to us for microfilming. While the most advanced technological means to photograph and reproduce this document have been used, the quality is heavily dependent upon the quality of the material submitted.

The following explanation of techniques is provided to help you understand markings or notations which may appear on this reproduction.

1. The sign or "target" for pages apparently lacking from the document photographed is "Missing Page(s)". If it was possible to obtain the missing page(s) or section, they are spliced into the film along with adjacent pages. This may have necessitated cutting through an image and duplicating adjacent pages to assure you of complete continuity.
2. When an image on the film is obliterated with a round black mark it is an indication that the film inspector noticed either blurred copy because of movement during exposure, or duplicate copy. Unless we meant to delete copyrighted materials that should not have been filmed, you will find a good image of the page in the adjacent frame.
3. When a map, drawing or chart, etc., is part of the material being photographed the photographer has followed a definite method in "sectioning" the material. It is customary to begin filming at the upper left hand corner of a large sheet and to continue from left to right in equal sections with small overlaps. If necessary, sectioning is continued again—beginning below the first row and continuing on until complete.
4. For any illustrations that cannot be reproduced satisfactorily by xerography, photographic prints can be purchased at additional cost and tipped into your xerographic copy. Requests can be made to our Dissertations Customer Services Department.
5. Some pages in any document may have indistinct print. In all cases we have filmed the best available copy.

University  
Microfilms  
International

300 N. ZEEB ROAD, ANN ARBOR, MI 48106  
18 BEDFORD ROW, LONDON WC1R 4EJ, ENGLAND

7916184

ANDERSON, RUSSELL DALE  
AN EVALUATION AND EMULATION OF RESOURCE  
DISTRIBUTION IN DISTRIBUTED PROCESSING  
NETWORKS.

IOWA STATE UNIVERSITY, PH.D., 1979

University  
Microfilms  
International

300 N. ZEEB ROAD, ANN ARBOR, MI 48106

**An evaluation and emulation of resource distribution  
in distributed processing networks**

by

**Russell Dale Anderson**

**A Dissertation Submitted to the  
Graduate Faculty in Partial Fulfillment of  
The Requirements for the Degree of  
DOCTOR OF PHILOSOPHY**

**Major: Electrical Engineering**

**Approved:**

Signature was redacted for privacy.

**In Charge of Major Work**

Signature was redacted for privacy.

**~~For the Major~~ Department**

Signature was redacted for privacy.

**For the Graduate College**

**Iowa State University  
Ames, Iowa**

1979

## TABLE OF CONTENTS

	Page
I. INTRODUCTION	1
A. Preamble	1
B. Statement of the Problem	3
II. LITERATURE REVIEW	9
A. Computer Network Structure	9
B. Network Components	13
1. Computing hardware	13
2. Communication	17
3. Software	19
C. Existing Networks	22
D. Network Optimization	27
E. Workload Description	29
III. PROPOSED PROBLEM SOLUTION	32
IV. THE MODEL	39
A. General Description	39
B. Data Configuration	41
1. Hardware	46
2. Software	49
3. Workload	51
4. Network	53
5. Control	55
C. Program Description	56
1. HONE	56
2. JSTOP	57
3. PTOS	59
4. OSASS	63
5. ANLYZ	65

V.	AN EXAMPLE	67
	A. System Application Description	67
	B. Results and Discussion	71
VI.	CONCLUSIONS	81
VII.	LITERATURE CITED	86
VIII.	ACKNOWLEDGMENTS	95
IX.	APPENDIX A: LISTING FOR THE ROUTINE HONE	96
X.	APPENDIX B: LISTING FOR THE SUBROUTINE JSTOP	107
XI.	APPENDIX C: LISTING FOR THE SUBROUTINE PTOS	115
XII.	APPENDIX D: LISTING FOR THE SUBROUTINE OSASS	131
XIII.	APPENDIX E: LISTING FOR THE SUBROUTINE ANALYZ	139
XIV.	APPENDIX F: INPUT DATA SET FOR THE CENTRALIZED CONFIGURATION	150
XV.	APPENDIX G: EXECUTION OUTPUT FOR THE CENTRALIZED CONFIGURATION	160
XVI.	APPENDIX H: OUTPUT DATA SET FOR THE CENTRALIZED CONFIGURATION	174
XVII.	APPENDIX I: INPUT DATA SET FOR THE DISTRIBUTED CONFIGURATION	184
XVIII.	APPENDIX J: EXECUTION OUTPUT FOR THE DISTRIBUTED CONFIGURATION	194
XIX.	APPENDIX K: OUTPUT DATA SET FOR THE DISTRIBUTED CONFIGURATION	207

## I. INTRODUCTION

### A. Preamble

In the most general sense, a computer system consists of one or more central processing units, associated input/output devices, and other peripheral hardware that are related and interconnected, either by direct wiring or through communications links. A computer system is a utility (54) which has the function of providing to its user-customers the commodities of information collection, processing, and distribution. Like other utilities, the computer system must be available when required, responsive to demand, reliable in function, expandable for growth, convenient to use, and, above all, it must offer cost-effective service to widely varying user requirements. The typical user views the computer system as an instrument to be employed to perform a particular task. The computer system is used because it offers the most economical implementation of the task in terms of the interrelated factors of user effort, time required for the task, and the cost of performing the task.

As seen by the user, the total cost of performing a task is given by Abrams (1) as

$$C_t = C_h + C_c + C_u + C_d \quad (1)$$

where

$C_t$  = Total cost of performing a task

$C_h$  = Cost of use of computing facilities

$C_c$  = Cost of use of communication facilities

$C_u$  = Cost of user's time

$C_d$  = Cost of delay in obtaining the result.

The first cost, that of the computing facilities, is the classical batch processing cost. It is related to the portion of the computer system physical facilities used by the task and to the amount of time the task requires. Included in this cost is the rental or amortization of the purchase cost of the computers, peripherals, and systems software. The staff costs for systems engineers and analysts, operators, and other personnel, as well as space and energy costs are also included. The cost of the communications facilities is included for those cases where the user's site is separated from the computer center by a distance which requires data communication facilities. The cost of the communication is a function of the rate of data transmission required, the amount of data, and the distance of the transmission.

The last two terms of the total cost equation lack the objectivity of the first two and are difficult to quantify. However, some observations can be made regarding these terms. The user's time to prepare the task for submittal to the system is obviously a function of knowledge and skill. The effort required to attain this knowledge and skill is a function of the design of the computer system, the system support provided to the user, and the available programming lan-



guages. The turnaround time and accessibility of the system also affect the user's time requirement, as does the reliability. The cost of an unexpected delay in obtaining a satisfactory result is a function of the time-value of the solution; i.e., the decreasing value or increasing cost incurred as a result of a delayed response from the computer system. The delay in obtaining the response may be due to computer system overload or failure, poor system design, or communication system error or interruption. The ultimate goal in the design of a computer system utility is to provide task performance at a minimum cost to each system user.

#### B. Statement of the Problem

The minimization of the total cost of performing a task on a computer system requires the minimization of the sum of the four interrelated cost terms in equation (1). Before attempting to examine the sum of the costs, some trends may be observed in each of the four individual costs. First, considering  $C_h$ , the well-documented progress (18, 27, 58) in semiconductor technology during the last twenty years has resulted in significant increases in component performance and reliability, while simultaneously reducing size and cost. During this time period, the central processor and main memory components of computer systems have attained an increase in speed of more than two orders of magnitude while decreasing in cost by a similar factor. The combined result has

been a cost/performance increase of more than 10,000. Equally important is the fact that the cost/performance improvement has been accompanied by a thousand-fold increase in reliability. Improvements have also been made in cost/bit of secondary storage, primarily disk, due to lower cost components in the controllers and interfaces, but largely due to increased density of recording. Unfortunately, the remainder of the computer system hardware, the input/output devices and sequential secondary storage devices, are subject to cost/performance decreases related chiefly to the economy of scale; i.e., a device with twice the performance costs more, but not twice as much. The remainder of the costs in Ch, the people costs, the software costs, and space and energy costs have been steadily increasing.

The cost of communications facilities, Cc, has decreased in the past two decades by an order of magnitude (24, 88) on a cost/bit/mile basis considering long terrestrial lines. This decrease has been achieved mainly through improved use of the lines rather than a decrease in the cost of the lines. The effect of satellite communications on data transmission cost appears to be quite variable depending on the distance of the computer systems from the ground stations. The total number of ground stations is limited by high cost of each station and terrestrial lines must be employed between the ground stations and the computer systems. An anticipated de-

velopment which will have a far-reaching impact on data communication in the local calling area is the possible adoption of usage-sensitive billing. In the past, local data communication telephone line charges were negligible compared to other costs. Each local line had a fixed monthly charge with unlimited duration of usage. With usage-sensitive billing, the fixed monthly charge will cover a nominal amount of connection time and any time exceeding that amount will be billed on a cost/hour basis. This cost/hour has been predicted to be \$0.60 to \$1.20 per hour (58). For a centralized system supporting one hundred terminals operating eight hours per day, the increased communication charge would be \$10,000 to \$20,000 per month. At the present time, in some local areas a form of usage sensitive billing is being used. In other areas fixed monthly surcharges are being added for lines used for data communication.

The user's rate of pay, and therefore, cost per hour, is certainly an increasing value for a particular user. The only practical means of lowering the cost of the user's time is to reduce the amount of time required. In the past, a primary design criterion for centralized computer systems was minimum turnaround time. A user could walk to the computation center, load a deck in the high-speed card reader, and within thirty seconds could be tearing the resulting printout from the high-speed printer. The fact that the user's walk

to the computation center required five minutes each way and that the processing time was less than five percent of the user's time to run the program had, until recently, little impact on the system design. The recent proliferation of remote job entry stations, remote terminals, and interactive computing systems is an attempt to reduce the cost of the user's time by increasing the accessibility of the computer system. The system reliability will also influence the user time required to obtain a satisfactory result. The overall system reliability is highly variable; not only from one system to the next but also on any given system from one day to the next. The fact that semiconductor components are very reliable and are increasing in reliability has been mentioned previously. In small, monoprogrammed, component intensive systems, with well-debugged software, a very high reliability is exhibited. In large, multiprogrammed, software intensive, centralized systems, subject to constant modification and upgrading, the reliability may be less than that which can be reasonably tolerated; particularly, during the first few days following the installation of a major system modification.

The increasing use of computer systems in real-time operation to interact with users, processes, and equipment has increased the significance of Cd. The cost of delay in obtaining a satisfactory result is likely to be much higher in such cases than in the traditional batch processing environ-

ment. For example, in an airline reservation system, the cost of a slow response to a request to show available space on a given flight is the lost revenue from the prospective passenger. A more critical case would be a slow response to a request for retro-rocket firing time as a lunar lander approaches the surface of the moon. Obviously, as long as the system response time is less than the maximum tolerable delay, this cost can be zero. The fact that system components are becoming faster and more reliable tends to reduce the difficulty in maintaining a given response time. However, new applications will invariably appear which require the fastest response available. The ability of the computer system to achieve and maintain a given response time is strongly dependent on the hardware and software design of the system.

The design of a computer system to minimize the cost of performing a single task for a single user located at the computer site can be satisfactorily performed. The manufacturer's catalogs can be used to obtain the price of the hardware and any software to be purchased. Reasonable estimates can be made of the effect of different systems features on the user's time. Since the system is only performing a single task, the speed required from the hardware is dictated by the maximum allowable response time. The general design problem is much less tractable and can be stated as follows:

Given a myriad of tasks to be performed for a multitude of users at many diverse locations, design a computer system that will minimize the total cost for each task performed.

As desirable as the solution to this problem may be, its generality prohibits any definitive solution. The problem which is to be addressed is a subset of the general problem.

Namely:

For a typical task mix to be performed for a number of users grouped at local and remote sites, determine a system design which meets stated performance criteria and is most likely to minimize the total cost for each task performed in view of present cost trends.

The intent of this dissertation is to propose and justify a general configuration for such a system and to present a computer model of this system. The model facilitates the rapid and economical determination of the optimum location for processing within a distributed processing network to achieve near minimum cost computing for a given set of cost parameters.

## II. LITERATURE REVIEW

The design of a computer system which is capable of performing tasks for both local and remote users involves an examination of the general concepts of computer networking. A computer network consists of one or more central processing units, associated input/output devices, and other peripheral hardware distributed among a number of sites which are interconnected by communication links. This collection of computer and communication hardware will be under the direction of the users' programs through the control of the system software of the network. The determination of a likely configuration for a computer system to process a mixture of tasks requires an investigation of extant and proposed networks to observe strengths and weaknesses for handling particular types of tasks. The areas of network performance evaluation and workload description must be considered to determine if a proposed system design meets stated performance criteria for the given task mix.

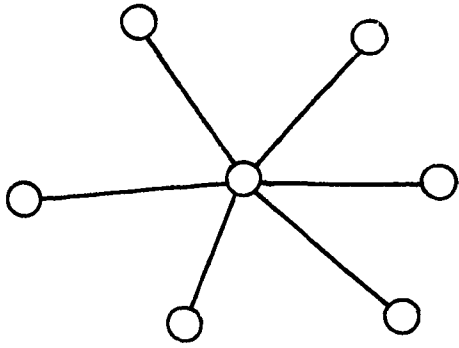
### A. Computer Network Structure

All computer networks can be described as a set of nodes and links. The nodes contain computing hardware which may range from a small amount of fixed hardware to a large-scale computer system. The function of some nodes can be to simply provide routing and switching between the links, other nodes

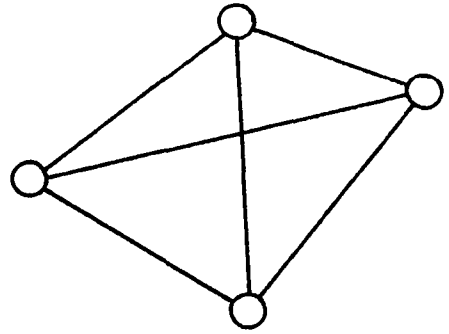
provide connection points for user terminals or computer systems (83). One of the basic attributes of any network is its topology or node-link organization. If, in an  $n$  node network, each of  $(n-1)$  nodes is connected to the  $n$ th node by one of  $(n-1)$  links, the network is called a star or centralized network. In a fully-connected distributed network, each node of an  $n$  node network has exactly one link to each of the other  $(n-1)$  nodes. This requires  $0.5(n(n-1))$  links. A partially-connected distributed network reduces the number of links by allowing two nodes to be linked through intermediary nodes. Such a network is said to be  $m$ -connected if at least  $m$  distinct paths exist in the network between any two nodes. In the particular case of a one-connected set of nodes, the network is called a ring. Various combinations of the above configurations are possible. For example, the tree configured network is formed by connecting the central node of a star network to the central nodes of two or more other star networks which are in turn connected to other stars (93). Figure 1 summarizes the above configurations.

Network composition is another descriptive attribute. If all nodes of the network are essentially identical in terms of computation and communication facilities, the network is called homogeneous. The term heterogeneous is applied to networks with dissimilar nodes. The type and rate of link communication between the nodes is referred to as the

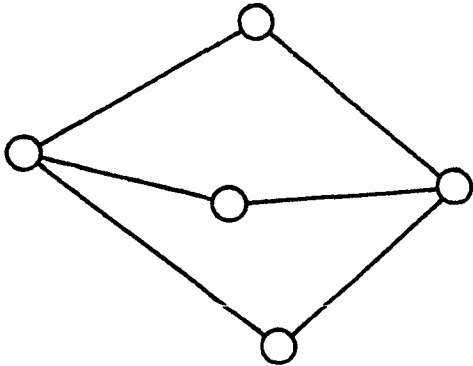




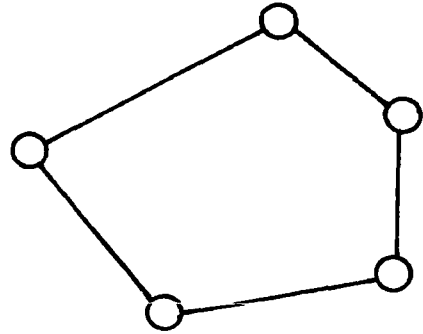
(a) Star



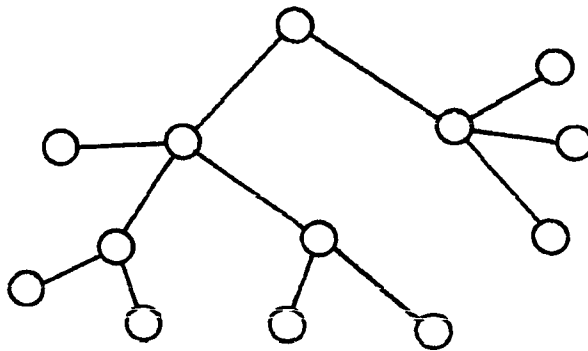
(b) Fully-connected distributed



(c) Partially-connected distributed



(d) Ring



(e) Tree

Figure 1. Network Topologies

coupling of the network. In loosely coupled networks, the communication is serial-by-bit at rates of fifty kilobits per second or less. In moderately coupled nets, higher speed serial lines, parallel busses or shared rotating memories are used for communication. Tight coupling is used to describe systems with shared direct access memory.

Computer networks have been designed or proposed to serve many diverse functions. Because of this, more specific terminology has been presented by several authors in an attempt to convey more information about the function of the network. The first efforts in computer networking and the majority of the large-scale networks in operation today are primarily data communication facilitators. As such, they provide remote users an access to one or more localized computers. The user views the network as a collection of computers with different capabilities. The trend among authors (29, 39) is to refer to networks of this type as computer-communications networks. The more general class of computer network which distributes not only information but also processing power is referred to as a distributed processing or distributed function network (64, 67, 94). The user views this network as one large computing system, even though the processing power of the network may be dispersed over many nodes to locate the power where it is most useful (75). If the dispersal is accomplished with computers of relatively

equal logical function (16), the processing is said to be horizontally distributed. If the dispersal is through computers of varying degrees of a hierarchy of logical function, the processing is said to be vertically distributed (20). Attempts have been made to set up a unified description technique for the many ways in which processors may be interconnected (6, 12, 40). Classification has also been proposed based on functional performance from the user's viewpoint (63).

#### B. Network Components

A computer network consists of a combination of three basic components. The computing hardware component includes the central processing units, the secondary storage units, input/output devices, and other peripherals. The communication component includes the communication medium and the necessary interface between the medium and the computing hardware to appropriately transform the data and control the data flow. The last component is the network software which is the network operating system consisting of the data base distribution system, the communication handlers, and resource and job allocation procedures.

##### 1. Computing hardware

Before 1950, the few computers which were designed had special purpose functions of solving specific problems. In

the decade of the fifties general purpose computers were developed and put into operation. Because of the electronic component technology, these were physically large machines and quite often were located in their own building, the computation center. The decade of the sixties saw the emergence of large scale processors due to the advent and extensive use of transistors and core memories. These processors were much faster, logically more powerful, and much smaller than the previous computers which they replaced. This meant that there was room at the computation center for many more peripherals; tape units, disks, line printers, card reader/punches, and others. The large-scale centralized computing system thus evolved. At the same time, another class of computer was being developed which ran counter to the large-scale centralization. It was not bigger and faster, but smaller and slower, and much less expensive. The minicomputer offered an alternative to remote users who had poor accessibility to the centralized systems. One of the strong points of the minicomputer was the convenient input/output structure which made interfacing to real time processes very convenient. With the advent of the decade of the seventies, the semiconductor manufacturers had developed integrated circuits with more logic functions and more speed for a relatively cheaper price, thus reducing the cost of "intelligence", i.e., the central processor and main memory in computer systems. The

impact of this cost reduction on the cost of computing performed in large scale systems is minimal, since it affects only a very small part of the overall cost of the system. The movement in large scale systems was primarily toward increasing the accessibility to more users by adding communication lines to remote terminals to form computer-communication networks. Typically, a minicomputer was used to interface the communication lines to the large computer. The decreasing cost of intelligence has had a much greater impact on minicomputers than on large scale computers, since a larger portion of the minicomputer system cost is devoted to the processing unit and main memory. In the early years of the seventies, another class of computers was developed. This time not by computer manufacturers but instead by the integrated circuit producers. The microprocessor is a single integrated circuit, which when combined with memory and control integrated circuits becomes a microcomputer. It is smaller and slower than a minicomputer but much less expensive. A new class of problems was now capable of being economically solved by use of a properly programmed microcomputer.

Through the years a hierarchy of computer classes has developed offering widely varying capabilities. In keeping with the micro- and mini- prefixes, the spectrum of available classes ranges over microcomputers, minicomputers, midicomputers, maxicomputers, and super-maxicomputers. The midicom-

puters are simply minicomputers with an extended word length to increase their capabilities. Maxicomputer is used to refer to conventional large scale systems and super-maxicomputer refers to a large scale system with parallel processing and interleaving to achieve high processing speeds.

Unfortunately, the decreasing cost of intelligence in computer systems has not been accompanied by a corresponding decrease in cost of peripheral devices. The rotating storage devices have decreased in cost per bit stored, particularly in disks with large storage capability. However, these decreases in cost are typically less than an order of magnitude (86) compared to the multiple orders of magnitude decrease in the cost of intelligence. A possible breakthrough may take place in this area if rotating storage can be replaced by MOS shift registers, magnetic bubbles, or charge-coupled devices (82). The line and character printers, the card reader/punch, paper tape equipment, and magnetic tape units are all strongly dependent on mechanics which restricts any drastic price decreases. The purchaser of a microcomputer can expect to pay more than the cost of the microcomputer to get a hard copy terminal. Since computing hardware for networks is virtually the same as stand alone computing hardware, a number of books on computers are available which cover various aspects of this hardware (14, 21, 55). Specif-

ic information about terminals which interface the computing system to the user has been presented in two very comprehensive papers by Hobbs (56, 57). Further information on available microcomputers and minicomputers including specifications and comparisons can be found in Davis (35) and Williman and Jelinek (102).

## 2. Communication

The communication component can be subdivided into the communication link and the communication interface. The network designer has few options in the choice of a communication link or path. A hard-wired circuit installed by the user may be an economical option for relatively short distances, particularly where the path is on property controlled by the user. A radio circuit, such as a microwave link, can be used but is not economically justifiable except in cases where no other options exist. Most computer network communication takes place via the common carriers' facilities. Based on the user's maximum data rate, required connection time, and location, the designer can choose a switched or leased circuit, or a value-added network connection. The characteristics of these services have been extensively covered in the literature (38, 99). The common carriers are government regulated and their tariffs must be approved. These tariffs (50) in combination with the channel data rate, error characteristics, and delay have been used as the basis

for many communication network design techniques and models (2, 28, 69, 76).

The interface between the digital device and the communication link has several possible functions. When channels which were designed for voice transmission are used for digital data, a transformation is required to convert the ones and zeros to frequencies between 300 and 3300 hertz. These tones must be transformed back to digital data at the receiver. The portion of the interface devoted to this modulation and demodulation is called a modem. Data may be transmitted either asynchronously, one character at a time, or synchronously, a continuous stream of bits. Synchronous transmission offers a higher data rate but requires protocols for bit, character, and message synchronization. Multiplexing is another function which may be performed by the interface. High capacity channels have a cost advantage over those of lower capacity. If a number of users can share a single high capacity channel, the communication cost for each user can be decreased. The multiplexor gathers the data from each user and interleaves it onto the communication channel. The interface is also responsible for all communication channel error detection and treatment. Many different schemes are possible which either detect the error and request retransmission or provide enough redundancy in the message to allow correction of those errors detected. The use of mini-



computers to perform many of the roles of the interface has been described in several papers (5, 80, 88).

The design strategy for most data communication networks must be strongly dependent on the rate structure of the common carriers. Location sensitive, distance sensitive, or usage sensitive changes in rates may force major changes in data communications. The overall data communications problem is examined in Chou (29) and Hopewell, Chou, and Frank (59). The attempts which are being made in trying to standardize data communication are discussed in Shutz and Clark (91).

### 3. Software

The functions of the network operating system are to provide for interprocess communication, intersystem resource sharing, intersystem task allocation, and intersystem data base distribution. The function of interprocess communication allows processes executing at one node of the network to exchange data in real time with processes executing at another node of the network. Intersystem resource sharing enables processes at one node to use peripheral devices which are physically connected to another node. The allocation of tasks to various nodes is done by the operating system to achieve load leveling, to unload a failing node, or to provide more computing resources to perform a task. Data base distribution allocates the files required by a process over those nodes of the system which will in some sense result in

a minimum access and storage cost. L'Archeveque and Yan (72) discuss many of these operating system requirements for a real-time environment.

All of the proposed functions of the network operating system will require communication handling programs to implement the internode transfers. Mills (79) and Willner and Lindinger (103) discuss the function and construction of these programs. A unified communication structure based on explicit data exchanges is presented by Wecker (100). The papers by Gray (51, 52) discuss a commercial communications product. A description of the communication handling programs on several different networks is given in Akkoyunla, Bernstein, and Schantz (4). Comparisons are made between these implementations based on routing, spurious data transmission, rendezvous conventions, role of the supervisor, and the interface to the user.

The objectives of data base distribution are to improve the data base availability and integrity while controlling communication costs by minimizing the number and size of messages and their paths when retrieving or updating files in the data base (22, 74). Booth (19) and Merrill (78) present an overview of the concepts involved in creating and using distributed data bases as well as specific techniques and problems. The paper by Zemrowski (104) explores the difficulties in managing a distributed data base and examines

future trends. Hunter (62) concludes that distributed data base management is less difficult if the design is compatible with all systems in the network and the data base organization is standardized. The problem of where to store files that are used by a number of different nodal computers is one which has been investigated by Chu (31), Segall (95), and Stubblefield (98). Chu also considered the performance of three classes of file directory systems (32) and provisions for avoiding deadlock when two processes compete for the same resource (33). The mathematical tools required for these analyses are presented in Hu (60) and Raeside (85).

The problem of job allocation in a computer-communications network is relatively simple since the user specifies the node where the job is to be executed. In some cases, if duplicate resources exist at other nodes, the operating system requests bids from all of the nodes which can run the job. The low bid indicates the least loaded system and the job is sent to that site. This achieves a degree of load leveling throughout the network and also improves overall network reliability since a system which has failed cannot submit a bid. The job allocation procedure in a fully specified network which has functionally similar nodes has been studied in detail by Balachandran, McCredie, and Mikhail (9). The investigation of criteria and techniques of job allocation in a general distributed processing network is an

area which will require further research.

The operating system for a network of minicomputers which was developed for the sharing of expensive peripherals is described in Fulton, Overstreet, and Thomas (49). The publication (36) contains the description of a very interesting set of commercial software products which extend existing computer operating systems to create a network operating system for a variety of configurations.

### C. Existing Networks

Virtually all of the commercial computer networks presently in service are computer-communications networks. The remote user has a terminal which is linked by a communication channel to a central computing site. The terminal provides a remote job entry station or interactive computing capability. Descriptions of the configuration and operation of many of these networks are available as follows: TYMNET (13), CYBERNET and TSS (41), GE Information Services and INFONET (92). Other commercial network descriptions may be found in (34, 71, 96).

The ARPA (Advanced Research Projects Agency) network has far more coverage in the literature (66, 70, 89, 87, 88) than any other computer network. This system was designed specifically to explore network technology as well as to interconnect and service ARPA-sponsored research centers from coast-to-coast and Hawaii. The ARPAnet is topologically a

two-connected distributed system of heterogeneous computers but with homogeneous minicomputer communications interfaces. Basically, ARPAnet is a computer-communications network but the network operating system does support user initiated file transfers between nodes. A homogeneous subset of the network supports resource sharing and distributed file access.

The MERIT network is a joint effort between three Michigan universities to interconnect their computer centers (8). The MERIT configuration is distributed, fully connected, with heterogeneous computers and homogeneous microcomputer communications interfaces. MERIT functions as a computer-communications network.

The SPIDER network at Bell Laboratories' Murray Hill, New Jersey location is a ring configured network of heterogeneous computers which are connected to the communications ring by means of homogeneous microprocessor interface units (48). The interface units provide for speed control between computers of different internal processing rates, implement all error control procedures, and support data transmission rates of 125,000 bits per second. The ring is hard-wired to the interfaces and digital data transmission is used. This system functions as a computer-communications network but does support resource sharing with respect to user files which are located at a node remote from that of the user.

The DCS (Distributed Computer System) network is an experimental network under development at the University of California at Irvine (42, 43). A heterogeneous set of processors is connected to a hard-wired ring communications channel through ring interfaces. The ring interface decides whether a message passing by on the ring is for a process located at this site. If so, it transfers the message to the process. This enables process-to-process communication to take place rather than computer-to-computer or terminal-to-computer. The software system is process oriented; all activities in the system are implemented as processes. The location of the process and the message transmission are transparent to the user who views the network as a computing system rather than a system of computers. A computer network which functions in a similar way is being developed at the University of Maryland. It is called DCN (Distributed Computing Network) and consists of a number of homogeneous mini-computers (39).

The OCTOPUS system at Lawrence Livermore Laboratory is a heterogeneous computer-communications network where all communication is hard-wired. A combination of topologies is employed with the large-scale computers, called workers, connected as a distributed network for file transfer and the terminals connected as centralized, or star, to the workers (45, 77). The file structure has been designed to handle se-

curity materials (46). A very candid paper presenting the user's view of this laboratory network points out many problem areas in designing computer networks which support research projects and at the same time are research projects (81). The HYDRA system at Los Alamos Scientific Laboratory supports functions similar to OCTOPUS. The configuration is unusual, since one of the group of large-scale computers is devoted exclusively to network control. The network control computer handles all terminal communication, controls the data base devices, and distributes tasks to the appropriate large-scale worker computer. A system description of HYDRA along with a forthright discussion of the technical and planning problems associated with the system appears in (30). Another system which is configured much like HYDRA is RIG (Rochester Intelligent Gateway) at the University of Rochester. The network control computer in RIG is a minicomputer which handles the communication to terminals, controls shared peripheral allocation, and distributes tasks to a local set of minicomputer workers, to one of three different remotely located large-scale computers, or to an ARPAnet connection (10).

Several hierarchical systems have been developed. At Bell Laboratories' Naperville, Illinois location, a number of minicomputers are used as real time data collectors and are interfaced to a more powerful minicomputer for access to a

line printer, card reader/punch, magnetic tape, and large disk. The interface and network software handle the communications and peripheral protocol in order to provide transparent operation to the user (11). A similar facility has been implemented at Northwestern University which provides, in addition to peripheral sharing, access to a large-scale computer. A modification to this network is being planned to allow off-loading of programs from a saturated minicomputer to one which is idle; thus, achieving a degree of distributed processing (73). A three level hierarchical system has been implemented at Purdue University; however it is primarily a computer-communications network which has evolved over a number of years (90). One of the most ambitious hierarchical networks is a laboratory support system at the University of Chicago. This network has three levels with the lowest level consisting of minicomputers, the second level is a more powerful minicomputer, and the top level is a large-scale system. At the lowest level, the minicomputers are dedicated to laboratory and other on-line functions. The intermediate level supports the minicomputers by providing an enhanced operating system for each of the minicomputers. All operating systems functions except interrupt and trapping are supplied at the intermediate level. The intermediate level also provides the interface to the batch and interactive services offered by the highest level. The system provides for distrib-



uted processing which at the lowest level is transparent to the user (7).

#### D. Network Optimization

In the early years of computer network development, as in the early years of computers, the primary performance criterion was whether or not it worked. Once it worked, the designers began to be concerned with how long it would be before it would quit again. Once the reliability improved to the point where the network became useful, the designers became concerned about measuring performance, locating areas for improvement, and optimizing the network or at least parts of it.

The foundation of the communication system which supports large computer-communication networks was in existence and had been extensively studied prior to its connection to computers. A classic paper by Frank and Chou (47) on the topological optimization of computer networks discusses much of this accumulated knowledge. Modeling, analysis, and design problems and techniques for both centralized and distributed computer-communication networks are discussed. The design objective is to provide a low-cost communications network which satisfies constraints on response time, throughput, reliability, and other parameters. Several fundamental communications network models for queuing and reliability analysis are described. Most of the material presented as-

sumes that network nodes are at fixed sites. Communication system factors for performance rating are discussed in (53). These factors are transfer rate, availability, reliability, accuracy, channel establishment time, network delay, line turn-around delay, transparency, and system security.

Kleiarock, Naylor, and Opderbeck (70) present a detailed description of the communication protocol on ARPAnet and the communications overhead costs incurred by using the message as the carrier for the network supervisor.

The CACTOS model (24) was developed to investigate the computer-communications trade-offs in computer-communication networks. The model starts with a number of fixed sites and first allocates a single computer to service all sites through communication channels. The cost of this configuration is found and the number of computers allocated is increased and the cost is again found. The last configuration has one computer per site. The model was put into operation in 1972 and many of the assumptions and conclusions are invalid for today's technology. A very general model for investigating computer-communication trade-offs is presented in (23). This model is applicable to star configurations with one central site and  $n$  local sites. The number of local sites,  $n$ , is used as the independent variable upon which all costs are based. Thus, by varying  $n$ , an optimum number of nodes may be found.

A rather recent, and certainly belated development in computer network performance optimization is the realization that the quality of service provided to the user is a major factor. As was stated in the introduction, the total cost of performing a task must include not only the computation and communication costs, but costs related to the user's time and the value of obtaining a correct result without unexpected delay (1). Several factors necessary to assure user service quality in a computer network have been discussed by Pyke (84). These factors include functional fidelity, level of performance, ease of use, reliability, availability, upgradability, maintainability, and confidentiality of data. The problem of user satisfaction with service as well as several other cost and performance factors, both tangible and intangible are discussed by the corporate director of computing for a large manufacturer in (86).

#### E. Workload Description

A major problem in predicting the performance of a proposed computer network or in evaluating the performance of an existing network is the development of an accurate description of the normal workload. As important as this problem appears, very few techniques are available for workload characterization. Three basic categories of workload representation have been suggested; the natural, artificial, and hybrid (44). The natural representation uses the real workload to

form the characterization, but may be strongly influenced by when and for how long the workload is observed. The artificial representation uses a characteristic workload which is designed and implemented independent of the real workload. These representations consist of particular instruction mixes, kernel-problems, or benchmark programs used to synthesize a characteristic workload. The hybrid representation is formed by assembling a characteristic workload by manipulating parts of the real workload. A group at the University of Maryland has verified the validity of the hybrid representation by formulating a workload characterization for a general computing environment in which they present three types of models for the workload (3). One of the models was used to construct a workload model for the large-scale computer at the university's computation center. The model treats the computer system as a device which executes processing tasks in response to user's requests. Each user request requires a certain amount of the hardware and software resources of the system. A vector  $X$  characterizes the resources requested with the  $j$ th component indicating the amount of the  $j$ th resource which is required to service the request. In addition to  $X$ , the request is characterized by  $T$ , the time the request was made;  $L$ , the location from which the request originated; and  $P$ , the type of request (batch, time-share, etc.). Every request for service is then charac-

terized by a quadruple  $(T,L,X,F)$  and the workload is the total of all requests. Since the model depends on time, it can be treated as a stochastic process which is, at the very least, extremely complex and may, in fact, be unmanageable. The most interesting result of the study occurred when the system logs of the actual computer workload were analyzed. It was found that the resource requirement for the actual jobs was not distributed randomly, but instead was clustered. That is, one group of jobs had similar resource requirements; another group had another set of resource requirements. The initial results indicate that a hybrid workload characterization consisting of a certain mix of jobs having the requirements of each of the clusters may be a viable approach for workload models.

### III. PROPOSED PROBLEM SOLUTION

The problem being addressed, as stated previously, is to determine a system design which meets required performance criteria for a general mix of tasks to be performed for a number of users grouped at local and remote sites and which is most likely to minimize the total cost for each task performed in view of present cost trends. The system configuration proposed for the solution of this problem is a hierarchically organized distributed processing system.

The hierarchical system configuration is chosen for a number of reasons. Computers are available in a hierarchy of capabilities, from the low powered microcomputers to the high powered super maxicomputers. A circumstance similar to that which has existed for many years in memory components (101). A study by a group at Carnegie-Mellon University headed by Bell has found that simple tasks can be performed most economically on simple computers and complex tasks are performed most economically on large, complex computers (15). Thus, given a general hierarchy of tasks, it is reasonable to provide a hierarchy of computers on which to perform these tasks. The present cost trends in computers and communications favors decreasing communication requirements by providing increased computing power located closer to the point of task generation. Moving the computation closer to the user results in improved response for the user. Since a portion

of the computer system is located local to the user, a degree of local control is possible which makes it more likely that local needs will be met. The hierarchical structure also results in improved availability. A given component failure will generally affect only a few users rather than the entire user community as can be the case in a centralized system. The modularity of the hierarchical structure provides more economical expansion and increased flexibility over other structures since the components being proliferated are generally the lower level minicomputers and microcomputers.

In order to achieve the advantages enumerated for the hierarchical configuration, certain constraints are placed on the system design. To enhance the modularity and maintainability, the computing hardware must be homogeneous on each level. The operating system and communication handlers must be the same in all computers on a given level in order to minimize the systems software design and maintenance effort. Each level should communicate with adjacent levels as if the adjacent level were a peripheral device. The communication should be process-to-process not computer-to-computer. This avoids the problem of full dependence, or master/slave relationship, between levels which can reduce reliability. The relationship between processes is more like producer/consumer. The communications software must be symmetrical, permitting any combination of producer and consumer

programs at either end of the communication link. In this way any level can provide a virtual peripheral or processor to another level. These constraints on the hierarchical network will make the system easier to implement, operate, and maintain while increasing the reliability and expansibility. The penalties which are incurred by these constraints must also be examined. The fact that the computing hardware is homogeneous on each level with common systems software will prohibit the tailoring of each node to optimally perform a specific task. However, the task can be moved up or down a level to provide more or less resources. The fact that communication control is decentralized will possibly increase the software complexity due to the protocol required to set up the link. The fact that this software can be replicated over many processors will reduce the burden of its implementation.

With the general network structure determined, a means must be found for examining the effects of moving the intelligence of the distributed processing system closer to the user. Techniques must be devised for rapid and economical determination of the optimum location for the processing of various types of tasks to achieve near minimum cost computing given a set of system parameters. The number of parameters required to describe a hierarchical system and its workload is of such a magnitude that the only feasible means of solu-



tion is to employ a computer model for the problem solution.

A number of network models exist for computer-communications systems (23, 24, 27) but very little has been done with distributed processing systems and, in particular, hierarchical systems. Bentley and Friedman (17) have devised algorithms for rapidly determining the shortest connecting network for a tree configuration which can be applied to a hierarchical system. However, simply minimizing the link length will not necessarily provide minimum cost computing. The work which has been carried out by Chang and Tang (25), and Chang (26) deals specifically with the modeling of hierarchical networks and includes many concepts which are applicable to the present problem.

The overall design goal of Chang's model of a distributed computer system is to produce an integrated hardware and software system which satisfies certain performance requirements and design constraints. The inclusion of both hardware and software is mandatory in any distributed processing system since the system performance is a function of both the hardware configuration and the workload applied to the system. The hardware portion of Chang's model consists of collection of processors, located at various stations, interconnected by a communication network. The software portion consists of defining a collection of transactions representing the workload which is to be processed by the system. The

system configuration is limited to a hierarchical structure with processing performed on a number of different levels. The model employs a three step design process to develop a feasible configuration. First, the transaction steps are assigned to different processor levels in the system hierarchy. Second, processors are allocated geographically to meet the workload requirements. Third, communication lines are allocated and the exact locations of the processors are determined. Chang has incorporated this design procedure into a computer program called ICON which was first written in APL (68) and then translated to FORTRAN (37). The system designer uses ICON interactively by providing system parameters to the program which then produces a feasible configuration which satisfies the designer's performance requirements. The interactive facilities enable the designer to change the system parameters based on the results so far and the program will then form a new result. The goal of ICON, like that of the model upon which it is based, is to find a feasible solution satisfying performance requirements. The minimization of the cost of processing within the proposed system was left to the user. System parameters were modified by the user, who then determined a new configuration for the next trial. The means of obtaining the modifications for the next trial were unspecified. The trials were continued until a least cost solution was produced.

While some of the concepts used in Chang's model will be required in the model to solve the present problem, the goal of the required model is the determination of a configuration for near minimum cost computing. The fundamental observation to be made is that any near minimum cost solution must be feasible, but any feasible solution will most likely not be near minimum cost. Some of the specific shortcomings in ICON which prevent its application to the present problem will be enumerated. First, in the assignment of transaction steps to different processor levels, an arbitrary choice is made for the particular processor to be used on each level. If a particular jobstep is incompatible with the processor chosen for a given level, the jobstep is assigned to a higher level, even though a different processor choice might allow processing at the lower level with less cost. Another shortcoming occurs in the geographical allocation of processors. Jobsteps assigned to a given level are grouped together in clusters based on the originating and terminating site of the jobs. An attempt is then made to merge these clusters of jobsteps and assign them to a single processor. ICON chooses an arbitrary cluster and then attempts to merge that cluster with the one whose initial site is the closest. This may result in minimum line lengths but most likely will not. Minimum line lengths, and resulting minimum cost, will be possible only if mergers are attempted starting with the ab-

absolute minimum distance between initial sites and proceeding to the next longest in sequence. Once two clusters are merged a new set of distances must be found before the next merger is attempted. Another major problem in ICON occurs because the line cost is assumed to be a function only of distance. The optimal site locating algorithm is predicated on the use of this line cost function. If a more realistic function is used which has a fixed termination charge in addition to the distance sensitive cost, the site locating algorithm does not work. For these major reasons and numerous minor difficulties, ICON was deemed inadequate as a model to be employed in the analysis required to determine the optimum location of processing in the context of the problem being addressed. Thus, the conclusion was reached that a new model must be developed.

#### IV. THE MODEL

##### A. General Description

The model which has been developed for investigating the optimum location for processing within a hierarchical distributed processing system is called HONE (Hierarchically Organized Network Emulator). The model is programmed in PL/I which was chosen primarily because data structures are supported by PL/I (61). Unlike arrays, data structure elements may have mixed attributes. An element of a data structure may be an array and arrays of structures are possible where each element in the array is a structure. This immense flexibility of storing data items which have a particular order and logical relationship to each other simplifies the handling of the large number of parameters involved in the description of a distributed processing network. The set of structures containing the network variables and the control parameters for running the program are maintained on a disk data set which is accessed by the program. The results of running the model are stored in a similar disk data set which can be used to replace the input data set for partial runs or making additions to a previously analyzed network. Modification of the disk data sets is accomplished from a remote terminal through the facilities of WYLBUR (97). The WYLBUR system provides on-line interactive text editing capabilities that allow the user to create text, change and correct it,

search and display it, and save and retrieve disk data sets. In addition to these features WYLBUR has facilities for remote job entry and retrieval from a remote terminal.

To run HONE, the user logs on to WYLBUR from a terminal and requests the input data set which is copied from disk into the WYLBUR active file. If a few parameters are to be changed within the existing configuration, the user employs the edit feature of WYLBUR by specifying the line number of the data to be changed. The present line is then listed for the user who responds by typing the new values below the existing values. Only the changes need be typed, all other data in the line will be unaffected. If changes are to be made to the existing configuration such as increasing the number of jobs to be included in the system workload, the edit feature of WYLBUR is used to indicate the new number of jobs and jobsteps, and to set the HONE program control parameter indicating a new configuration is desired. When HONE is then executed, the output data set contains the complete input data set with additional lines for the new parameters to be added. These additional lines contain all zeros but in the proper format to be edited. After editing this output data set to indicate the desired parameters, the user stores it into the input disk data set. The execution of HONE is initiated from the terminal through the use of the remote job entry facilities of WYLBUR. The output from HONE and the

output data set can also be accessed by WYLBUR and examined at the terminal. If a CRT terminal is being used and a hard copy of the output is desired, WYLBUR commands can be executed to list the results on a line printer.

#### B. Data Configuration

The set of data structures used for the input to and output from HONE contains the hardware and software description of the network, the network performance parameters, and the control variables for HONE. These data structures are given the attribute EXTERNAL in the HONE program and its major subroutines. The variable names used in the discussion below are those used in the declared structures of the programs as listed in appendixes A through E. Table 1 shows the format of the actual input/output data set employed to communicate data values from the user to HONE and from HONE to the user. The data names used in the input/output data set have, in some cases, been expanded for clarity and, in other cases, abbreviated to save space. Where the name used in the input/output data set is not identical to that used in the structure, the input/output data set name is given in square brackets, [ ], following the name used in the structure.

Table 1. Format of the input/output data set

FUNCTION SELECTION FOR THIS RUN

USE <1> TO INDICATE FUNCTION TO BE SELECTED

INITIALIZE	CALL	CALL	CALL	CALL	NEW NET
TO DEFAULTS	JSTOP	PTOS	OSASS	ANLYZ	CONFIG
<1 >	<0 >	<0 >	<0 >	<0 >	<1 >

UPBOUND-VALUES AND ALLOWABLE RANGES

NSTEP	NLINE	NSTEP	NFCN	NFILE	NJOB	NLINE	MPROS	NLEVEL
1/25	1/99	1/250	1/99	1/99	1/50	1/10	1/10	1/4
3	2	6	3	3	2	4	10	2

CTLPARM-VALUES AND ALLOWABLE RANGES

ISEL	XN	THRS	KCENT	TRAC	PLIST (1)	(2)	(3)	(4)
BIT	<9999	<9999	BIT	BIT	1/10	1/10	1/10	1/10
0	00	00	0	0	0	0	0	0

PROCESSOR COMPATABILITY MATRICES

PROS		1	2	3	4	5	6	7	8	9	10
CFILE	1	0	0	0	0	0	0	0	0	0	0
CFILE	2	0	0	0	0	0	0	0	0	0	0
CFILE	3	0	0	0	0	0	0	0	0	0	0

PROS		1	2	3	4	5	6	7	8	9	10
CFCN	1	0	0	0	0	0	0	0	0	0	0
CFCN	2	0	0	0	0	0	0	0	0	0	0
CFCN	3	0	0	0	0	0	0	0	0	0	0



Table 1 (Continued)

SITE DESCRIPTORS						
	STATUS	XDIST	YDIST	PROS	PROS LOAD	PROS UF
	0/2	<9999.9	<9999.9	1/10	M.MMMESNN	M.MMMESNN
SITE 1	0	0.0	0.0	0	0.0000E+00	0.0000E+00
SITE 2	0	0.0	0.0	0	0.0000E+00	0.0000E+00
SITE 3	0	0.0	0.0	0	0.0000E+00	0.0000E+00

LINKING LINE DESCRIPTORS					
	INSITE	FINSITE	L TYPE	L LOAD	L UP
	0/25	0/25	0/10	M.MMMESNN	M.MMMESNN
LINE 1	00	00	00	0.0000E+00	0.0000E+00
LINE 2	00	00	00	0.0000E+00	0.0000E+00

CHARACTERISTICS OF AVAILABLE PROCESSORS							
	LVL#	I RATE	PRIMSTR	SCNDSTR	MEDIA	CHAN RATE	PCOST
	1/10	M.MMMESNN	M.MMMESNN	M.MMMESNN	CODE	M.MMMESNN	<999999.99
PType 1	0	0.0000E+00	0.0000E+00	0.0000E+00	0	0.0000E+00	00.00
PType 2	0	0.0000E+00	0.0000E+00	0.0000E+00	0	0.0000E+00	00.00
PType 3	0	0.0000E+00	0.0000E+00	0.0000E+00	0	0.0000E+00	00.00
PType 4	0	0.0000E+00	0.0000E+00	0.0000E+00	0	0.0000E+00	00.00
PType 5	0	0.0000E+00	0.0000E+00	0.0000E+00	0	0.0000E+00	00.00
PType 6	0	0.0000E+00	0.0000E+00	0.0000E+00	0	0.0000E+00	00.00
PType 7	0	0.0000E+00	0.0000E+00	0.0000E+00	0	0.0000E+00	00.00
PType 8	0	0.0000E+00	0.0000E+00	0.0000E+00	0	0.0000E+00	00.00
PType 9	0	0.0000E+00	0.0000E+00	0.0000E+00	0	0.0000E+00	00.00
PType 10	0	0.0000E+00	0.0000E+00	0.0000E+00	0	0.0000E+00	00.00

Table 1 (Continued)

CHARACTERISTICS OF AVAILABLE LINES								
		XMSN RATE	LFCOST	LVCST1	LVCST2	LVCST3	LVCST4	LVCST5
		M.MMMESNN	<9999.9	<99.99	<99.99	<99.99	<99.99	<99.99
LTYPE	1	0.0000E+00	00.0	0.00	0.00	0.00	0.00	0.00
LTYPE	2	0.0000E+00	00.0	0.00	0.00	0.00	0.00	0.00
LTYPE	3	0.0000E+00	00.0	0.00	0.00	0.00	0.00	0.00
LTYPE	4	0.0000E+00	00.0	0.00	0.00	0.00	0.00	0.00

SOFTWARE FUNCTIONS				
		I STEPS	FCNTSIZE	FCNFSIZE
		M.MMMESNN	M.MMMESNN	M.MMMESNN
FCNTYPE	1	0.0000E+00	0.0000E+00	0.0000E+00
FCNTYPE	2	0.0000E+00	0.0000E+00	0.0000E+00
FCNTYPE	3	0.0000E+00	0.0000E+00	0.0000E+00

SOFTWARE FILES			
		MEDIUM	FILE SIZE
		1/10	M.MMMESNN
FILETYPE	1	0	0.0000E+00
FILETYPE	2	0	0.0000E+00
FILETYPE	3	0	0.0000E+00

JOBS IN WORKLOAD						
		#JSTPS	ARVL RATE	RSPNS TIME	FRSTEP#	LASTEP#
		1/99	M.MMMESNN	M.MMMESNN	1/250	1/250
JOB	1	00	0.0000E+00	0.0000E+00	00	00
JOB	2	00	0.0000E+00	0.0000E+00	00	00

Table 1 (Continued)

---

JOBSTEPS IN JOBS																	
	REQ	REQ	MSG	SP	RQ	OWN	STP	ASND		PROS	TIME	LINE	DLY	LVL	ASGND		
	FCN	FIL	LENGTH	LV	LV	JOB	#	SITE		M.MMMESNN	M.MMMESNN	M.MMMESNN	M.MMMESNN	1	2	3	4
	<99	<99	M.MMMESNN	<4	<4	<50	<99	1/25		M.MMMESNN	M.MMMESNN	M.MMMESNN	M.MMMESNN	B	B	B	B
JSTP	1	0	0	0.0000E+00	0	0	0	0	0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	2	0	0	0.0000E+00	0	0	0	0	0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	3	0	0	0.0000E+00	0	0	0	0	0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	4	0	0	0.0000E+00	0	0	0	0	0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	5	0	0	0.0000E+00	0	0	0	0	0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	6	0	0	0.0000E+00	0	0	0	0	0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0

---

## 1. Hardware

The hardware to be used in the network implementation is described by the major structure called `HRDWARE`. This structure contains two minor structures which are arrays of structures describing the available processors (`PTYPE`) and communication lines (`LTYPE`). A description of the elementary items follows:

In `PTYPE` [`CHARACTERISTICS OF AVAILABLE PROCESSORS`], the elements are:

`PROS_LVL#` [`LVL#`] - the processor level number indicates the level of the hierarchy where this processor is used. A one indicates the lowest level.

`I_RATE` - the instruction execution rate for this processor. The value used is normalized mips (millions of instructions per second). An eight bit byte is used as the basis for normalizing and a linear relationship is assumed based on processor word length; e.g., the normalized mips for a processor having a 16 bit word length would be two times the actual mips.

`PRIMSTR` - the net amount of primary random access storage, in kilobytes, available to the user. The net is found by deducting from the total random access memory, the amount occupied by the

resident operating system.

SCNDSTR - the amount of available on-line secondary storage in megabytes. An order of precedence is used to determine this value. If a system contains non-removable disk and/or drum storage, the amount of this storage is assigned to SCNDSTR. If a system contains removable disks, the value given to SCNDSTR is the amount of on-line storage available on the removable disks. If magnetic tape is the only secondary storage available, then SCNDSTR will be equal to the maximum amount of on-line tape storage. This order of precedence is necessary to realistically determine secondary storage capacity since, in the case of removable disks and tapes, the amount of storage can be expanded by simply loading another disk or tape.

MDSPRT [MEDIA CODE] - a code representing the secondary storage media supported by this system. The code values are assigned as follows:

- 1 - Removable disk
- 2 - Fixed disk or drum
- 3 - Magnetic Tape
- 4 - Removable disk and fixed disk or drum
- 5 - Removable disk and magnetic tape

6 - Fixed disk or drum and magnetic tape

7 - Removable disk, fixed disk or drum,  
and magnetic tape

CHAN\_RATE - the maximum input/output data transfer rate, in bytes per second, for this processor.

PCOST - the monthly cost for this computing system.

If the system is leased this is the monthly rental. If the system is purchased, the purchase price is amortized over a fifty month period.

In LTYPE [CHARACTERISTICS OF AVAILABLE LINES], the elements are:

XMSN\_RATE - the transmission rate of this line type in bytes/second. Asynchronous transmission is assumed with each eight bit byte preceded by a single start bit and followed by a single stop bit.

LFCOST - the fixed cost associated with the use of this line. Generally, this is the cost of the common carrier's termination, the modems required, and any other hardware necessary to interface the line to the computer. The cost is expressed in dollars/month considering monthly rental or 50 month amortization of purchased equipment.

LVCOSTN - the variable charge for line use based on

the cost/mile charged by the common carrier.  
 There are five rates representing varying distances:

LVCOST1 [LVCST1] - for distances beyond the  
   local calling area to 25  
   miles

LVCOST2 [LVCST2] - for over 25 miles to 100  
   miles

LVCOST3 [LVCST3] - for over 100 miles to 250  
   miles

LVCOST4 [LVCST4] - for over 250 miles to 500  
   miles

LVCOST5 [LVCST5] - for over 500 miles

## 2. Software

The software which is used by the jobs in the network is described by the major structure called SFTWARE. This structure contains two minor structures which are arrays of structures describing the functions (FCNTYPE) and files (FILETYPE) which can be called by the user. The functions may be user written application programs or system library programs. The files, likewise, may be user created or system library files. A description of the elementary items follows:

In FCNTYPE [SOFTWARE FUNCTIONS], the elements are:

I\_STEPS - the number of normalized instruction steps  
   executed in the performance of this function.

The value used should include the system overhead instruction steps as well as the actual function steps. An eight bit byte is used as the basis for normalizing and a linear relationship is assumed based on processor word length.

**FCNTSIZE** - the amount of main memory, in kilobytes, which must be temporarily allocated to this function. This storage may be for data accumulation or buffering, or for the function itself which is brought in from secondary storage for execution. This temporary allocation exists only while the function is executing.

**FCNFSIZE** - the amount of main memory, in kilobytes, which is fixed in main memory for the execution of this function. This storage is dedicated to this function and will be resident in main memory at all times. It is assumed that only one copy need exist in the processor being used and any job requiring this function may use that copy.

In FILETYPE [SOFTWARE FILES] the elements are:

**MEDIUM** - a code representing the medium upon which this file is stored. The code values are assigned as follows:

- 1 - Removable disk
- 2 - Fixed disk



### 3 - Magnetic tape

**FILE\_SIZE** - the length of this file, in megabytes. It is assumed that only one copy of a given file need exist on each processor where a function executes which requires this file.

In addition to the structure, **SFTWARE**, one other structure is used in the description of the software files and functions. This structure is an array of arrays indicating the compatibility of each function with each processor type and each file with each processor type. The major structure is called **PROCOMP** [**PROCESSOR COMPATABILITY MATRICES**], and the minor structures are **CFILE** and **CFCN**. The arrays contain bit data, 1's and 0's, indicating compatibility and incompatibility respectively. The user has the responsibility of determining the function-processor compatibilities since **HONE** has no way of determining the peripheral support or data generating devices at each processor. For example, a given function may require the reading of a card deck and will only be compatible with those processors having a card reader. The file-processor compatibilities are determined by **HONE** since they are dependent only upon media support and file size.

### 3. Workload

The workload of the distributed processing network is described by two major structures, **JOB** and **JOBSTEP**. Each of these is an array of structures. The workload of the system

consists of a number of jobs with each job performing a number of jobsteps. A description of the elementary members follows:

In JOB [JOBS IN WORKLOAD], the elements are:

NO\_JOBSTEPS [#JSTPS] - the total number of jobsteps included in this job.

ARVL\_RATE - the arrival rate of this job, expressed as jobs per second.

RSPNS\_TIME - the amount of time required to execute the job in seconds. HONE determines this value and places it in the output data set.

FRSTEP# - the identification number of the first jobstep in this job.

LASTSTEP# - the identification number of the last jobstep in this job.

In JOBSTEP [JOBSTEPS IN JOBS], the elements are:

REQFCN [REQ FCN] - the identification number of the function required by this jobstep.

REQFILE [REQ FIL] - the identification number of the file required by this jobstep.

MSG\_LENGTH - the number of bytes of data to be transferred from this jobstep to the next.

LVL\_SPEC [SP LV] - a code used to control the level to which this jobstep is assigned. 0 signifies that there is no preference, 1 signifies that there is

a preferred level, and 2 is used to indicate that a particular level assignment is required.

REQ\_LVL [RQ LV] - the requested level number for a jobstep whose level specification code is 1 or 2.

OWNING\_JOB [OWN JOB] - the identification number of the job owning this jobstep.

STEP# [STP #] - the sequential number of this jobstep within the owning job's jobsteps.

ASGND\_SITE [ASND SITE] - the site to which HONE assigned this jobstep.

PROS\_TIME - the time, in seconds, calculated by HONE for the execution of this jobstep.

LINE\_DELAY - the line delay, in seconds, calculated by HCNE for the data to be transferred from this jobstep to the next.

LVL\_ASGND - a bit array indicating the level to which the jobstep was assigned by HONE. A 0 indicates that this jobstep is not assigned to the particular level, a 1 indicates the assigned level.

#### 4. Network

The description of the network configuration which is developed by HCNE is contained in a major structure called NETWORK. This structure includes two minor structures, SITE and LINE, each of which are arrays of structures. The only information the user is required to provide in these arrays

are the locations of the fixed sites of the network. Any sites not initially specified will be optimally located by HONE. Any new sites created by HONE will be added to the existing sites. HONE will provide all of the line information. The elementary members of these two arrays are:

In SITE [SITE DESCRIPTORS], the elements are:

STATUS - a code representing the specificity of the site location. A 0 indicates a site which has not been assigned to any location. A 1 is used to indicate a site, which while given a location, can be moved. A 2 is used to indicate a fixed site.

XDIST - the X coordinate, in miles, of this site in a two dimensional plane.

YDIST - the Y coordinate, in miles, of this site in a two dimensional plane.

ASGND\_PROS [PROS] - the type number of the processor assigned to this site.

PROS\_LOAD - the processing load carried by this processor, expressed in normalized mips.

PROS\_UF - the utilization factor of this processor. This factor gives a measure of a processor's actual use compared to its capability.

In LINE [LINKING LINE DESCRIPTORS], the elements are:

INSITE - the initial site for this line.

FINSITE - the final site for this line.

L\_TYPE - the type number of the line assigned to this link.

L\_LOAD - the communication load carried by this line expressed in bytes/second.

L\_UF - the utilization factor of this line. This factor gives a measure of a line's present load compared to that which it is capable of handling.

### 5. Control

The structure UPBOUND [UPBOUND-VALUES AND ALLOWABLE RANGES] is used to convey to HONE the number of parameter sets the user is supplying in the input data set. These elements are:

NSITE - the number of sites included in SITE.

NLINE - the number of lines included in LINE.

NSTEP - the number of jobsteps included in JOBSTEP.

NFCN - the number of functions included in FCNTYPE.

NFILE - the number of files included in FILETYPE.

NJOB - the number of jobs included in JOB.

NLINE - the number of line types in LTYPE.

MPROS - the number of processor types in PTYPE.

These values are all determined by the user. However, HONE can increase both NSITE and NLINE when new sites are required.

The portion of the input data set which has not yet been examined is required for the control of the execution of HONE and will be discussed later during the description of the operation of HONE.

### C. Program Description

The program structure of HONE consists of a main routine, HONE, and four major subroutines, JSTOP, PTOS, OSASS, and ANLYZ, which are external to HONE and are called by HONE using an overlay technique. The data structures which contain the network descriptors are identically declared and given the EXTERNAL attribute in the main procedure and the major subroutines. This allows all of the routines to have access to all of the network parameters with the name of each parameter remaining the same for all references to that parameter. A general description of the functions performed by each of the routines is given below. The source code listings for the FI/I implementations of the routines can be found in appendixes A through E.

#### 1. HONE

The main routine HONE is the resident supervisor for all of the subroutines. HONE initially clears all of the data structures containing the system parameters and then loads the input data set from disk into the structures. Thus, any structure element not defined by the user has a default value

of zero. The next action of HONE is determined by the function selection parameters [FUNCTION SELECTION FOR THIS RUN] in the input data set. These parameters allow the user to initialize the subroutine control parameters to the default values, to instruct HONE to sequentially call one or more of the subroutines, JSTOP, PTOS, OSASS, and ANALYZ, and to use HONE to create an output data set which can be used as an input data set for a new network configuration [NEW NET CONFIG]. If a new configuration is requested, all other function selection parameters are ignored. HONE calls the requested subroutines in the sequential order of JSTOP, PTOS, OSASS, and ANALYZ. After each requested subroutine has finished executing, control is returned to HONE and that subroutine is transferred from main storage back to disk and the next requested subroutine is loaded into main memory for execution. After the last subroutine completes, HONE stores the output data structures into the output data set on disk and then terminates.

## 2. JSTOP

The main function of the subroutine, JSTOP, is to select an optimum processor for each level of the hierarchy and to assign each jobstep to a level. This has the effect of determining the processor type on which each jobstep will be executed. The first action of JSTOP is to determine the compatibility of each file with each processor type. If file J

is compatible with a processor I, the structure element, `PROCOMP.CFILE(I,J)`, receives a bit value of 1; otherwise, it is set to 0. The compatibility decision is based on the medium for the file and the media supported by the processor along with the file size. JSTOP then determines the values for the local structure, CJSTOP, which contains the Z array which indicates jobstep and processor compatibility and the LCL element which holds the number of the lowest compatible level on which each jobstep will execute. A jobstep J is compatible with a processor I if the function and file required by the jobstep are both compatible with that processor. The lowest compatible level for each jobstep is determined by scanning the list of processor types with which the jobstep is compatible and retaining the minimum level number among the level numbers for these processors. JSTOP next constructs an array which lists the possible processor types which can be used on each level in the order of ascending processor cost.

The choice of a processor for each level is made by selecting the first processor on the list (the least expensive) and then scanning all jobsteps whose lowest compatible level is this level. If all of these jobsteps are compatible with this processor, then this processor type number is assigned to be used on this level. If all of these jobsteps are not compatible with this processor, the next processor on the



list is chosen and the jobstep scan is repeated. At the conclusion of this procedure, every level will have an assigned processor which is the least expensive processor capable of executing all of the jobsteps for which that level is the lowest compatible level. The control parameter, KCENT, in the input data set determines the assignment of jobsteps to the various levels. If KCENT is a 0, a decentralized assignment is made with each jobstep being assigned to its lowest compatible level. If KCENT is a 1, a centralized assignment is made with each jobstep assigned to the highest level with which it is compatible. At the conclusion of JSTOP, a processor type has been chosen for each level and each jobstep has been assigned to a particular level.

### 3. PTOS

The primary function of PTOS is to determine the number of processors, and thus, the number of sites required on each level. The algorithm which is used is an extension of the Partition-Merge Algorithm presented in (25). The jobsteps have been sorted onto assigned levels by JSTOP and the next step is to group the jobsteps into clusters on each level. Each cluster will ultimately contain the workload of a single processor of the type assigned to the level on which that cluster is located. The operation of the clustering algorithm can best be illustrated by considering each jobstep to be represented by a 4-tuple,  $(i, j, k, l)$  where:

**i** = jobstep identification number

**j** = originating site of the job which owns the job-  
step

**k** = preassigned site for this jobstep, 0 indicates  
that the jobstep does not have a preassigned  
site

**l** = higher level site for this jobstep, 0 indicates  
that the jobstep does not link to a higher level  
site

Clustering starts on the highest level and proceeds one level at a time until the lowest level is reached. At each level the three stages of the clustering algorithm are implemented before moving to the next level. The first stage of the algorithm creates clusters with the same **j** value. One cluster is made for each set of jobsteps which all belong to jobs having the same originating site. Stage two of the clustering algorithm combines clusters on the basis of **k** value; that is, any two or more clusters with the same preassigned site on this level are combined into a single new cluster. Before stage three is started, a structure, called MERGE, is created which lists those cluster pairs which are compatible based on the fact that they are **k**- and **l**-compatible. Two clusters are **k**-compatible if neither has a preassigned site or if one has a preassigned site, the other has the same preassigned site or no preassigned site. Two clusters are **l**-compatible if the

higher level sites satisfy conditions like those for k compatibility. For each cluster pair placed in the MERGE structure a distance value is determined and stored in the DIST element of the structure. A representative location value is determined by summing the X coordinates and the Y coordinates of the originating sites for the owning jobs of each jobstep in one of the clusters. The total X and total Y are then divided by the number of jobsteps within the cluster. This gives a weighted average X and Y; weighted by the number of jobsteps having the same owning job originating site and averaged over the total number of jobsteps. The representative location  $(X', Y')$  for the other cluster of the pair is found in the same way. The value placed in DIST is the distance, in miles, between the points  $(X, Y)$  and  $(X', Y')$ . After all compatible pairs and their corresponding distances have been placed in MERGE, the structure is sorted on the variable DIST to place the cluster pairs in the order of increasing distance. This ordered MERGE structure is used in stage three of the clustering. In stage three, the resource requirements of the jobsteps included in the first cluster pair are evaluated to determine if the total requirement is less than the capacity of the processor assigned to this level. If so, the first cluster of the pair is moved into the second by reassigning the jobsteps which were in the first cluster. If the cluster pair would overload the processor assigned to

this level, the merger is not made, and the next possible cluster pair is taken from the MERGE structure to determine if merger is possible. Whenever a merger takes place in stage three, a new MERGE structure is created and stage three is reentered. This process continues until no mergers are made in stage three. A new site is then created for each of the remaining clusters which does not have a preassigned site. The created site has a status of unassigned-floating, a location of  $X=Y=0$ , and is assigned the processor type for this level. If the cluster for which the new site is created has a specified higher level site, a new line is added having an initial site corresponding to the new site and a final site at the higher level site. After all new sites and lines have been added, PTOS returns to stage one with the next lower level. This process continues until the clustering is completed for the lowest level.

At this point in PTOS, the jobsteps have all been assigned to sites and interconnecting lines have been assigned between the sites. The line loading can now be calculated. For each jobstep which is followed by a jobstep belonging to the same job but located at a different site, a path is found which leads to the succeeding site. The load on the line or lines in this path is increased by the required bytes per second found by taking the product of the message length between the jobsteps and the arrival rate of the job which owns

these jobsteps. After all the line loads have been accumulated, a line type assignment is made for each line. The criterion used for choosing a line type is that the line load should be no more than 50% of the line capacity.

The hierarchical network configuration is now specified. The total number of sites required has been determined. A processor has been assigned to each site and a site has been determined for each jobstep. Each required line has been specified and a line type for each line has been assigned.

#### 4. OSASS

The function of OSASS is to optimally locate all floating sites to minimize line costs. Two options are available based on the control parameter, ISEL. If the user specifies ISEL to be a 1, OSASS restricts the relocation of the floating sites to the locations of the specified fixed sites. An unassigned site is simply given the coordinates of each of the assigned sites, retaining the coordinates and connection cost of the least expensive location so far. After all possible relocations have been tried, the unassigned site is given the coordinates of the assigned site where the lowest connection costs occurred. When all sites have been optimally assigned under the constraints of the mode of relocation, OSASS terminates and returns control to HONE.

If the control parameter, ISEL, is a 0, OSASS employs an exhaustive, differential iterative technique to relocate

floating sites to an optimal location. This iterative technique sets up a grid of coordinate values and the connection cost of the least expensive point so far are retained. The grid is created by searching the coordinates of all sites connected to the site being relocated, in order to determine the largest and smallest X values and the largest and smallest Y values. These values determine the boundaries of the grid. The fineness of the grid is determined by the control parameter,  $XN$ , which is set by the user to indicate the desired number of intervals to be placed between the minimum and maximum X and Y values. The application of this iterative technique to each unassigned site takes place in two stages. The first stage is a preliminary locating stage which is required to distribute the new sites created by PTOS which were all given coordinates of  $X=Y=0$ . The preliminary locating stage uses only those connecting sites which are fixed or have been preliminarily located to determine the location of a floating site. If all sites connected to a given site are floating, the site is not relocated at this time. The preliminary stage continues cycling through all sites, relocating the floating sites as possible until all floating sites have been preliminarily located. The second stage of the application of the iterative technique consists of a sequential relocation of each floating site to a point of relative lowest cost. OSASS continues to cycle through the sites

until the cost improvement is less than the preset control parameter, THRSE, or until the number of iterations exceeds a built-in maximum, presently one hundred. Control is then returned to HONE.

An internal subroutine is used to calculate the connection cost of a given site. The present cost calculation assigns a cost of zero and a line type of zero to any line which is less than 0.1 miles long under the assumption that this will be a hard-wired communication link. This assignment allows HONE to be used for a system which includes a hierarchy of processors at a single location. Any line which is longer than 0.1 miles but shorter than 7.5 miles and is type 1 or 2, is given a mileage dependent charge of zero based on the assumption that this will be a local line.

At the conclusion of OSASS, every site will have an assigned location based on the relocation algorithm selected by the user.

## 5. ANLYZ

The main function of the subroutine, ANLYZ, is to determine the performance of the distributed processing network which has been developed. The load and utilization factor for the processing unit, main memory, and secondary storage of each processor are determined. The utilization factor of each line is found. An estimated processing time and line delay is determined for each jobstep assuming a Poisson ar-

rival and exponential time distribution. The estimated job response time is determined from the accumulated time of all jobsteps contained in the job. Total communication costs and processor costs are calculated and control is returned to HONE.



## V. AN EXAMPLE

### A. System Application Description

The utility of the HONE program can best be illustrated by means of a specific example. The application chosen for the example consists of a system for the remote updating of files which are maintained at a centralized location. The workload of the system is composed of a number of similar jobs which originate from different locations. Table 2 summarizes the jobstep composition of each of these jobs.

The first jobstep of each job must be initiated by the operator of the terminal which is the origin for that job. The terminals employ a cathode ray tube and keyboard for communication with the operator. Each terminal has the facilities for local cursor control, backspace/replacement correction, protected fields, and block transmission of keyboard generated data. The operator uses the keyboard to enter data to the screen and corrections can be made as necessary. When the desired message has been composed, the operator uses the 'SEND' key to transmit the entire block of data. The first block of data, jobstep 1, is a message having an average size of 100 bytes. This message identifies the file which is to be modified and the type of modification which is to be performed.

The message is received by the system's input-output handler and buffer routine, in jobstep 2, which collects the

Table 2. Composition of jobsteps for jobs in the example workload

JOBSTEP	FCN	ISTEPS	FCN TSIZE Kbytes	FCN FSIZE Kbytes	MSG LENGTH bytes	FILE	FILE SIZE Mbytes
1	1	0.0E+0	0	0	100	0	0
2	2	7.5E+3	0	8	250	0	0
3	3	6.0E+4	8	8	1000	1	0.05
4	1	0.0E+0	0	0	250	0	0
5	2	7.5E+3	0	8	250	0	0
6	4	1.0E+5	16	2	1600	2	0.03
7	5	6.0E+5	128	64	1600	3	400
8	2	7.5E+3	0	8	1600	0	0
9	1	0.0E+0	0	0	25	0	0
10	6	7.5E+2	0	8	25	0	0
11	7	1.0E+5	128	64	1000	3	400
12	3	6.0E+4	8	8	1000	1	0.05
13	1	0.0E+0	0	0	0	0	0

block of data and makes an initial interpretation to determine the next task which is to be performed. In this case it is jobstep 3, which prepares an output to assist the operator in the formatting of the updated data. This output routine further interprets the operator's initial message to determine the type of file modification which is required. Based

on this determination, one of a number of standard forms is retrieved from secondary storage and is transmitted to the terminal where it is displayed to the operator. The keyboard is then used to enter the required data in the appropriate unprotected fields of the displayed form. After the data has been entered and corrected as necessary, the operator sends this data block to the system, which completes jobstep 4. The input-output handler and buffer routine receives this transmission, in jobstep 5, and determines the next function which must be performed. This next function, jobstep 6, is a routine which checks the validity of the format of the data against that required by the file update which has been requested. After the verification, the data is reformatted as required by the file and a request is made for access to the file. This request initiates jobstep 7 which locates the file in secondary storage, reads the file into primary storage, and performs the necessary modifications. The modified version of the file is then written into a temporary file in secondary storage and also passed to the input-output handler to be transmitted to the terminal in jobstep 8. When the updated version of the file is displayed on the screen, the operator visually verifies the correctness of the updated file. A short message is then transmitted to the system by the operator, in jobstep 9, to acknowledge the validity of the updated file. The input monitor and buffer receives this

message, in jobstep 10, and calls the file replacement routine. The function of this routine, during jobstep 11, is to retrieve the updated version of the file from the temporary location in secondary storage and to release that location for future use. The routine then replaces the original copy of the file with the updated version. At the successful conclusion of this replacement, a completion message is passed to the output routine. In jobstep 12, the output routine retrieves the proper message from secondary storage and transmits it to the terminal where it is displayed to the operator in jobstep 13. The job is thus concluded and the system is prepared to accept a new job when one is initiated by the terminal operator. The procedure described is that of normal operation and contingency responses must be provided in case of malfunction or operator error. However, the assumption must be made that the bulk of the transactions will be normal in any acceptable system implementation. This example is directed toward the examination of that normal operation.

Two different hardware configurations will be analyzed by the HONE program, a centralized system and a distributed system. In both cases, twelve input/output terminals are provided with three terminals at each of four locations. The master files to be updated are stored at a central location in both configurations. The computer system at this central

location performs many other tasks in addition to the file maintenance. However, this example is concerned only with the workload on the central system created by the requests for file modification. Appendix F shows the input data set used for the centralized configuration. The input data set for the distributed configuration is given in appendix I. Note that the value of the control parameter, KCENT, is a one for the centralized version and a zero for the decentralized, or distributed, version. In both cases the average job arrival rate is twenty jobs per hour at each terminal. Thus, each operator will be initiating a new job every three minutes. The keyboard entry time for the messages which the operator must supply to the system will require approximately two minutes. Consequently, a satisfactory implementation must have a job response time of less than one minute.

#### B. Results and Discussion

Appendixes G and H show the output of the program HONE for the centralized configuration resulting from the input of the data set of appendix F. The output which occurs during the execution of HONE with the trace disabled (TRAC=0) is given in appendix G. In this mode, the subroutines, JSTOP, PTOS, and OSASS, provide only completion messages. The subroutine, ANALYZ, lists the details of the line and processor loading, response time, utilization, and cost. At the conclusion of the execution of the subroutines of HONE, the out-

put data set is stored on disk by the major program. The output data set for this run is presented in appendix H. The format of the output data set is identical to that of the input data set in order to facilitate iterative runs which have only minor modifications from one run to the next. Appendixes J and K are the results of the execution of HONE for the distributed configuration using the input data set of appendix I.

The hardware configurations devised by HONE for the centralized and distributed solutions are schematically summarized in figures 2 and 3. In the centralized version, only two levels of processor hierarchy occur since the centralization algorithm moves all processing to the highest compatible level, in this case, level 4. The input/output terminals on the first level are each connected by a 120 byte/second (type 2) line to the central location. In the distributed version, three levels of processor hierarchy are employed. The first level consists of eight of the twelve input/output terminals arranged with two at each of the four remote locations. Each of these pairs of terminals is connected directly (type 0 line) to a level 2 computer system containing an input/output terminal, a microcomputer, and floppy disk secondary storage. These resources are sufficient to provide the third input/output terminal for that location and to execute all of the software functions which require processing power except

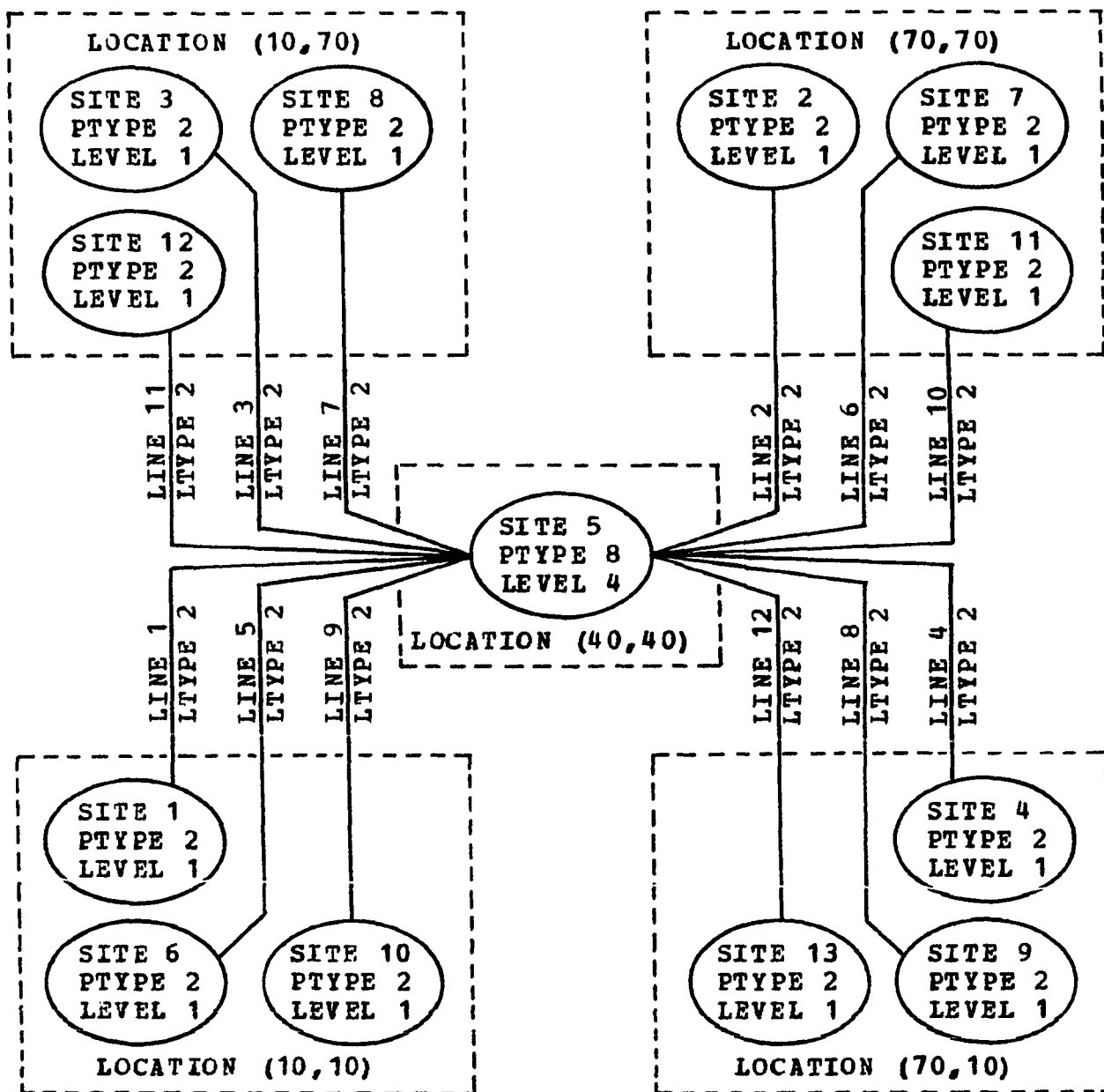


Figure 2. Centralized hardware configuration

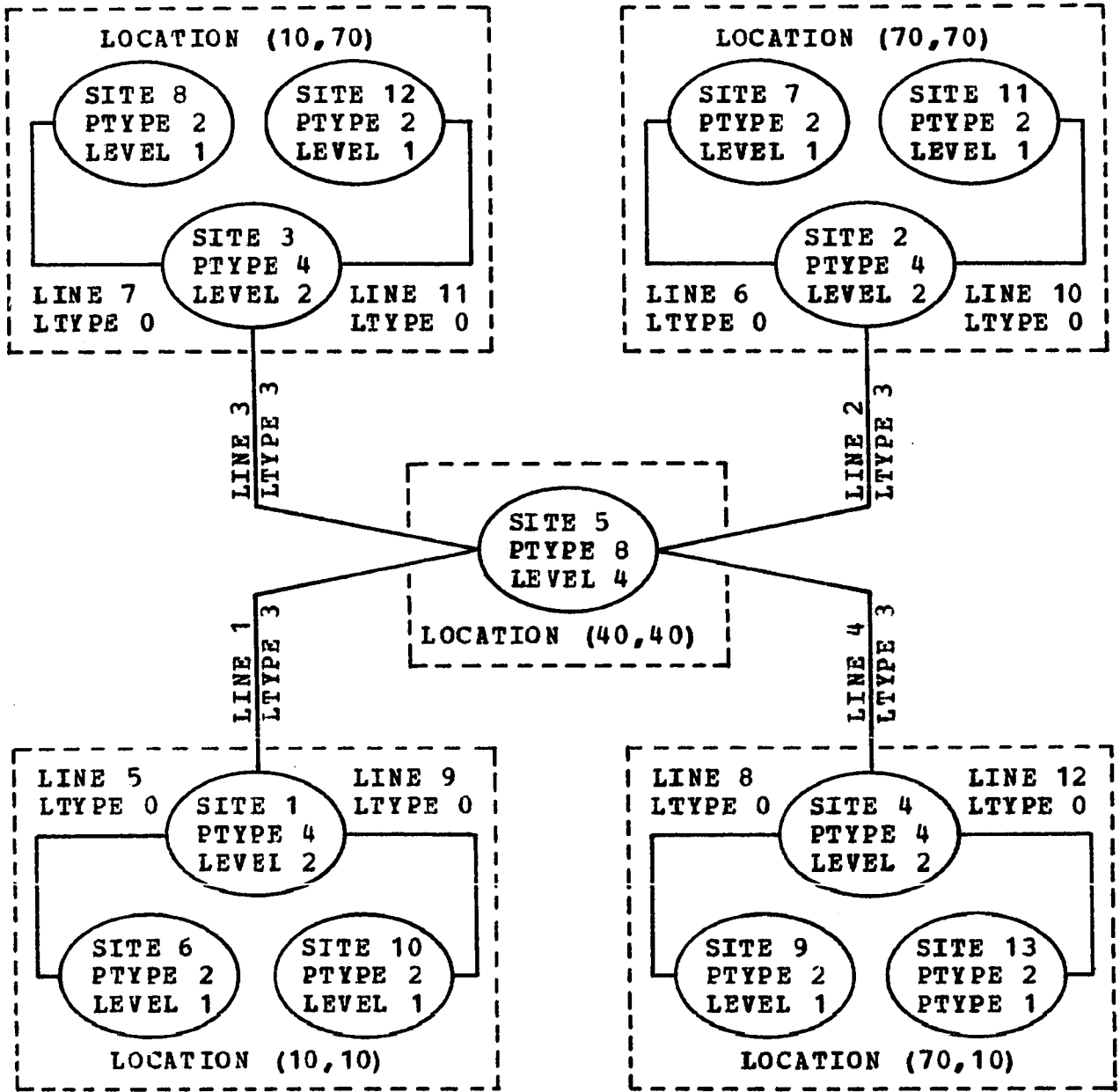


Figure 3. Distributed hardware configuration



for those which must access the centrally located master file. The level 2 processors are each connected to the central location with a 240 byte/second (type 3) line. Each level 2 processor provides local service to the three input/output terminals at a remote location and handles the necessary communication with the central system for all three terminals.

The results of the HONE runs show that the distributed configuration provides a shorter job response time and lower system cost than the centralized configuration. Each job requires 41.5 seconds of computing and communication time on the centralized system and only 27.3 seconds on the distributed system. This represents a 34% reduction in the response time. Both configurations require the same central computer system for storage of the master files. Thus, the costs which are significant in a comparison between the two configurations are those for all processors and lines except the central computer system. For the centralized configuration these processor and line costs total \$2507.28 per month. These costs for the distributed system are \$1563.76 per month, which represents a 37% savings.

The hardware and software maintenance costs, the system reliability, and system expansion capabilities are more difficult to quantify but can be examined subjectively. The distributed system executes the same software functions as

the centralized system. The primary difference in the software between the two configurations is the replication of those functions which are distributed to level 2. The additional software maintenance required by having four copies of the same program at four remote locations will be due primarily to the fact that the software is remote relative to the location of the systems personnel. However, the communication links between level 2 and level 4 can be used during maintenance for uploading and downloading of software. The hardware maintenance costs could possibly be slightly higher for the distributed system. The four microcomputer systems used on level 2 of the distributed configuration may require more maintenance than the additional eight communication links of the centralized system. This can not, though, be predicted with much certainty.

The reliability of these systems can best be examined by observing the effect of a system component failure on the availability of the system to the users. The centralized configuration has only two modes of failure. If the central computer system is out of service, all of the users are idled. If any communication link or any input/output terminal is out of service, only the user of that specific component will be affected. The distributed configuration has more modes of failure. However, with some additional software at level 2, the system availability to the users can

actually be increased. This additional software would use the secondary storage available on level 2 to accumulate the file updates from the users whenever the central computer system or a communication link to the central location is out of service. When the central system or link comes back on-line, the file updates and verifications are processed using these accumulated inputs. The failure of a level 2 computer system will idle only the three users who are connected to that system and the failure of an input/output terminal will affect only the user of that terminal. With the suggested software modification on level 2 of the distributed system, all four of the level 2 computer systems or all twelve of the input/output terminals must be simultaneously out of service for all twelve users to be idled. These conditions are far less probable than the failure of the central computer system in the centralized configuration.

The system expansion capabilities will be examined by considering the costs incurred in modifying each of the configurations in order to respond to an increase in the workload which overloads the present system. Such an increase corresponds to an average system job arrival rate which exceeds that which can be handled by the twelve operators. The expansion of the centralized configuration requires the addition of a terminal and communication link for each new remote station to be added. The cost of each new station at

any of the present remote locations is \$208.94 per month. The distributed configuration can be expanded by at least one more terminal at each remote location with only the expense of the terminal itself, \$30.00 per month. The expansion would take advantage of the excess capacity in the communication link from level 2 to level 4 which is only 30% utilized in the present configuration. A slight increase in job response time would accompany such an addition. The level 2 processor can easily handle the increased computing load. Further expansion of the distributed system would require communication links with more capacity and more primary memory in the level 2 processors. However, the communication link capacity can be doubled for \$60.00 per month and the memory capacity can be doubled for \$30.00 per month. Thus, the job handling capability of a remote station can be doubled to six input/output terminals for \$180.00 per month in the distributed configuration. This is less than the cost of adding a single terminal to the centralized configuration.

The analysis of the HONE runs for this example indicates that the distributed configuration has a lower initial cost, provides better job response times, is more economically expanded, and can have higher availability to the user than the centralized configuration. The hardware and software maintenance costs may be somewhat less for the centralized configuration.

Two further observations of the HONE runs can be made. First, the central computer system was assumed to be performing many other tasks in addition to the file updating required in this example. The centralized configuration requires 4.2% of the central computer system's processing capacity. The distributed configuration requires 3.1% since some of the processing takes place in the level 2 processors. While the difference is small, it makes available 1.1% of the processing capacity of a computer system which costs \$13,103.40 per month. The value of this capacity can be relatively significant in further demonstrating the superiority of the distributed configuration. The second observation is that less than 1% of the available processing capability is utilized in the four level 2 processors in the distributed configuration. However, the addition of these processors results in a reduction of more than 30% in both the remote system cost and the job response time compared to the centralized configuration. This result points out the significant advantage of designing a distributed processing network as a total system rather than a collection of individual processors connected by communication links. The commonly held idea that any computer must be utilized to nearly its full capability is obsolete in view of present cost trends. To justify the purchase of the four level 2 processors which have over 99% idle time would be next to impossible without

noting the effect on the total system performance and cost.

This particular example was constrained in order to illustrate not only the use of HONE, but also, the superiority of the distributed configuration over the centralized configuration for a commonly occurring workload. The primary constraint was that all sites were initially fixed and not relocatable by HONE. Therefore, the full power of the HONE program was not invoked. This example does illustrate that the HONE program provides a means for designing and analyzing hierarchically organized distributed processing systems and of performing these functions rapidly. The HONE run for the centralized configuration required 7.55 seconds of computer time and the run for the distributed system required 14.27 seconds.

## VI. CONCLUSIONS

This investigation has resulted in the development of a powerful tool for the design and analysis of hierarchically organized distributed processing networks. This tool, a computer model called HONE, is a significant extension of previous work due to its capability for resource representation, its ability to minimize hardware and communication costs, and its flexibility in representing system workloads. The model provides not only the system configuration, but also analyzes the performance of the system. Once a configuration has been determined, HONE can be used to study the sensitivity of the system to adjustments in workload, availability of sites, or other system component changes. The model provides for total distributed processing system design. The importance of designing the total system was illustrated by an example which showed that in a particular case, the total system performance could be increased while dramatically reducing costs. These results were obtained by adding a level of processors which were only 1% utilized.

The HONE model was the result of an investigation directed toward the examination of distributed processing networks in order to evaluate the performance of such networks as a function of the network configuration. The goal was to determine a system design which would be most likely to minimize the total cost for performing each task of a general

system workload in view of present cost trends. A hierarchically organized network structure was found to provide the flexibility necessary to optimally execute the hierarchy of tasks in a general workload. The hierarchical structure can employ the extant hierarchy of computing hardware to decrease task response time and total task cost relative to other network structures while increasing availability and expansibility.

With the general network structure determined, a model, called HONE, was developed to serve as an instrument to assist in configuring a network and to analyze the performance of the resulting network. The initial data presented to the model consists of the parameters used to describe the processor and line types, which can be selected to configure the network, and the workload description, including the locations of the originating and terminating sites. The HONE program selects an optimum processor for each level of the hierarchy and assigns each jobstep to a level. For a decentralized, distributed system each jobstep will be assigned to the lowest level with which it is compatible. HONE then groups the jobsteps into clusters at each level. Each cluster will ultimately contain the workload of a single processor of the type assigned to the level where the cluster is located. The jobsteps are combined into clusters by means of an algorithm which will minimize communication costs. The



message traffic between clusters is determined and a communication line type is assigned for each link.

The hierarchical network configuration has now been specified by HONE. The total number of required sites has been determined and a processor has been assigned to each site. The jobsteps have been distributed to specific sites and communication line requirements have been found. The HONE program then optimally locates all of the sites which do not have a specified location. The performance of the network is then determined by HONE. The loading of the processors and lines is found and the system costs are calculated. The model provides the system designer with a tool for the rapid and economical determination of the optimum location for processing within a distributed processing network in order to achieve near minimum cost computing for a given set of cost parameters.

The tacit assumption throughout this investigation was that each user would be charged only for the cost of the system resources required to perform that user's task. This implies that if a small number of the users of a system require a high level processor, that the cost of that processor is divided among those users. If the processor is underutilized, the task cost for these users may be high, even though it is minimized for the given workload. The system designer should recognize cases such as this as representing an

incomplete hierarchy from the results of the HONE program. To complete the hierarchy, the designer should consider sharing the required high level processors with other incomplete hierarchies.

The utility of the HONE program was demonstrated by the use of a specific example. The model was applied to the commonly occurring application of updating master files from several remote locations. Two different hardware configurations were developed and analyzed by HONE, a centralized, two level network and a distributed, three level network. The analysis showed that distributing the processing power of the system resulted in a savings of approximately one-third in system cost while decreasing the job response time by a similar amount when compared to the centralized system.

The HONE program has been found to be a valuable instrument for the investigation of hierarchically organized distributed processing networks. However, the use of HONE and the results achieved point to many areas where future study might be undertaken in order to advance the understanding of such networks. One area is in the characterization of workloads. The development of an accurate description of the normal workload of a proposed system is a necessary part of the design of that system. The extraction of essential features of the workloads for specific applications should be attempted. The model, HONE, could be further refined to in-

clude other factors which are part of the total job response time. The input/output time delays are presently considered by the user of HONE as additional to the job response time given by the model. The average delay in transferring data to and from secondary storage is considered in the present model as an additional instruction step load. The input/output and secondary storage delays could be considered as separate parameters in a refined version of the model. The inclusion of reliability and availability information in the model should be investigated. If data were supplied on the mean time between failures and the mean time to repair for the various system components, the model could possibly provide reliability predictions to the system designer. The predictions might include the modes of failure and the effect of each mode on system availability.

A system design has been proposed which can meet stated performance criteria and is most likely to minimize the total cost for each task of a general task mix for a number of users grouped at both local and remote sites. A model of this system design has been developed which provides a general framework to enable the rapid and economical determination of the optimum configuration and the analysis of the performance of this configuration.

## VII. LITERATURE CITED

1. Abrams, Marshall D. A new approach to performance evaluation of computer networks. Proceedings of the IEEE-NBS Symposium on Computer Networks (1974): 15-20.
2. Abramson, Norman and Kuo, Franklin. Computer-communication networks. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1974.
3. Agrawala, A. K.; Mohr, J. M.; and Bryant, R. M. An approach to the workload characterization problem. Computer 9 (June 1976): 18-32.
4. Akkoyunlu, Eralp; Bernstein, Arthur; and Schantz, Richard. Interprocess communication facilities for network operating systems. Computer 7 (June 1974): 46-55.
5. Amstutz, Stanford R. Distributed intelligence in data communications networks. Computer 4 (November/December 1971): 27-32.
6. Anderson, E. A. and Jensen, E. D. Computer interconnection structures: taxonomy, characteristics, and examples. ACM Computing Surveys 7, No. 4 (1975): 197-213.
7. Ashenhurst, P. L. and Vonderohe, R. H. A hierarchical network. Datamation 21 (February 1975): 40-44.
8. Aupperle, Eric M. Merit network re-examined. Digest of Papers, COMPCON 73 (Spring 1973): 25-30.
9. Balachandran, V.; McCredie, J. W.; and Mikhail, O. I. Models of the job allocation problem in computer networks. Digest of Papers, COMPCON 73 (Spring 1973): 211-214.
10. Ball, J. Eugene; Feldman, Jerome; Low, James R.; Rashid, Richard; and Rovner, Paul. RIG, Rochester's intelligent gateway: system overview. IEEE Trans. on Software Engineering SE-2, NO. 4 (1976): 321-328.
11. Barkauskas, B. J.; Rezac, R. R.; and Trlica, C. A. A computer network for peripheral time sharing. Digest of Papers, COMPCON 73 (Spring 1973): 227-229.

12. Becker, Hal B. A structured approach to information networks. Digest of Papers, COMPCON 73 (Spring 1973): 35-38.
13. Beere, Max and Sullivan, Neil. TYMNET - A serendipitous evolution. IEEE Trans. on Communications COM-20, No. 3 (1972): 511-515.
14. Bell, C. G.; Habermann, A. N.; McCredie, J.; Rutledge, R.; and Wulf, W. Computer networks. Computer 3 (September/October 1970): 13-23.
15. Bell, C. Gordon and Newell, Allen. Computer structures: readings and examples. New York: McGraw-Hill, 1971.
16. Bell, C. Gordon; Chen, Robert C.; Fuller, Samuel H.; Grason, John; Rege, Satish; and Siewiorek, Daniel P. The architecture and applications of computer modules: a set of components for digital systems design. Digest of Papers, COMPCON 73 (Spring 1973): 177-180.
17. Bentley, J. L. and Friedman, J. H. Fast algorithm for constructing minimum spanning trees in coordinate spaces. IEEE Trans. on Computers C-27, No. 2 (1978): 97-105.
18. Bloch, Erich and Galage, Dom. Component progress: its effect on high-speed computer architecture and machine organization. Computer 11 (April 1978): 64-75.
19. Booth, G. M. The use of distributed data bases in information networks. Proceedings of the First International Conference on Computer Communications (1972): 371-376.
20. Booth, G. M. Distributed information systems. AFIPS Conference Proceedings 45 (1976): 789-794.
21. Booth, Taylor L. Digital networks and computer systems. 2nd edition. New York: John Wiley and Sons, Inc., 1978.
22. Bray, O. H. Distributed data base design considerations. Trends and Applications: Computer Networks, IEEE Computer Society (1976): 121-127.

23. Buhr, R. J. A. and Woodside, C. M. The optimum number of nodes in a star configured distributed computer system. IEEE Trans. on Systems, Man, and Cybernetics SMC-8, No. 1 (1978): 76-79.
24. Cady, George M. and Luther, Gunther. Trade-off studies in computer networks. Digest of Papers, COMPCON 73 (Spring 1973): 147-150.
25. Chang, S. K. and Tang, D. T. Processor allocation in a distributed computer system. IBM Research Report. White Plains, New York: IBM Corporation, 1974.
26. Chang, Shi-Kuo. A model for distributed computer system design. IEEE Trans. on Systems, Man, and Cybernetics SMC-5, No. 6 (1976): 344-359.
27. Chiba, Tsuneyo. Impact of the LSI on high-speed computer packaging. Trans. of IEEE Computer Society C-27, No. 4 (1978): 319-325.
28. Chou, Wushow. Planning and design of data communications networks. AFIPS Conference Proceedings 43 (1974): 553-559.
29. Chou, Wushow. Computer communication networks - the parts make up the whole. AFIPS Conference Proceedings 44 (1975): 119-128.
30. Christman, Ronald D. Development of the LASL computer network. Digest of Papers, COMPCON 73 (Spring 1973): 239-242.
31. Chu, W. W. Optimal file allocation in a multiple computer system. IEEE Trans. on Computers C-18, No. 10 (1969): 885-889.
32. Chu, W. W. Performance of file directory systems for data bases in STAR and distributed networks. AFIPS Conference Proceedings 45 (1976): 577-588.
33. Chu, W. W. and Ohlmacher, G. Avoiding deadlock in distributed data bases. ACM National Symposium 1 (1974): 156-160.
34. Cuccio, A. B. J. Microprocessor utilization in transaction terminal nets. Digest of Papers, COMPCON 73 (Spring 1973): 161-164.

35. Davis, S. A fresh view of mini and micro computers. *Computer Design* 13 (May 1974): 67-79.
36. DECNET Technical Summary. Maynard, Massachusetts: Digital Equipment Corporation, 1976.
37. Dimitry, Donald and Mott, Thomas, Jr. Introduction to FORTRAN IV programming. New York: Holt, Rinehart, and Winston, Inc., 1966.
38. Doll, Dixon R. Telecommunications turbulence and the computer network evolution. *Computer* 7 (February 1974): 13-22.
39. Elovitz, H. S. and Heitmeyer, C. L. What is a computer network? National Telecommunications Conference Record (1974): 1007-1014.
40. Enslow, P. H., Jr. Multiprocessor organization - a survey. *ACM Computing Surveys* 9, No. 1 (1977): 103-129.
41. Farber, David J. Networks: an introduction. *Datamation* 18 (April 1972): 36-39.
42. Farber, David J. A ring network. *Datamation* 21 (February 1975): 44-46.
43. Farber, David J.; Feldman, Julian; Heinrich, Frank F.; Hopwood, Marsha D.; Larson, Kenneth C.; Loomis, Donald; and Rowe, Lawrence A. The distributed computing system. *Digest of Papers, COMPCON 73* (Spring 1973): 31-34.
44. Ferrari, Domenico. Workload characterization and selection in computer performance measurement. *Computer* 5 (July/August 1972): 18-24.
45. Fletcher, John G. Octopus communications structure. *Digest of Papers, COMPCON 73* (Spring 1973): 21-23.
46. Fletcher, John G. Octopus software security. *Digest of Papers, COMPCON 73* (Spring 1973): 61-62.
47. Frank, Howard and Chou, Wushow. Topological optimization of computer networks. *Proceedings of the IEEE* 60, No. 11 (1972): 1385-1397.
48. Fraser, A. G. A virtual channel network. *Datamation* 21 (February 1975): 51-56.

49. Fulton, D. L.; Overstreet, C.; and Thomas, R. T. The design of a minicomputer. Trends and Applications: Computer Networks, IEEE Computer Society (1976): 6-10.
50. Gerla, Mario. New line tariffs and their impact on network design. AFIPS Conference Proceedings 43 (1974): 577-582.
51. Gray, James P. Line control procedures. Proceedings of the IEEE 60, No. 11 (1972): 1301-1312.
52. Gray, James P. IBM's system network architecture. Datamation 21 (April 1975): 51-56.
53. Grubb, Dana S. and Cotton, Ira W. Rating performance. Data Communications 4 (September/October 1975): 41-47.
54. Hassett, F. C. and Silverman, H. W. Networks for computer utilities. Digest of Papers, COMPCON 73 (Spring 1973): 195-197.
55. Hill, Frederick J. and Peterson, Gerald R. Digital systems: hardware organization and design. 2nd edition. New York: John Wiley and Sons, Inc., 1978.
56. Hobbs, L. C. The rationale for smart terminals. Computer 4 (November/December 1971): 33-35.
57. Hobbs, L. C. Terminals. Proceedings of the IEEE 60, No. 11 (1972): 1273-1284.
58. Hodges, David A. Trends in computer hardware. Computer Design 15 (February 1976): 77-85.
59. Hopewell, Lynn; Chou, Wushow; and Frank, Howard. Analysis of architectural strategies for a large message-switching network: a case study. Computer 6 (April 1973): 26-34.
60. Hu, I. C. Integer programming and network flows. Reading, Massachusetts: Addison-Wesley Publishing Company, 1969.
61. Hughes, Joan K. PL/I programming. New York: John Wiley and Sons, Inc., 1973.
62. Hunter, John J. Distributing a database. Computer Decisions 8 (June 1976): 36-40.



63. Jasper, David P. A definition of networks. Digest of Papers, COMPCON 73 (Spring 1973): 67-69.
64. Joseph, E. C. Distributed function computer systems: innovative trends. Digest of Papers, COMPCON 74 (Spring 1974): 71-73.
65. Jutila, Sakari T. and Baram, Giora. A user-oriented evaluation of a time-shared computer system. IEEE Trans. on Systems, Man, and Cybernetics SMC-1, No. 4 (1971): 344-349.
66. Kahn, R. Resource-sharing computer communications networks. Proceedings of the IEEE 60, No. 11 (1972): 1397-1408.
67. Kallis, Stephen A., Jr., Networks and distributed processing. Mini-Micro Systems 10 (March 1977): 32-40.
68. Katzan, Harry, Jr. APL programming and computer techniques. New York: Van Nostrand Reinhold Company, 1970.
69. Kershenbaum, Aaron. Tools for planning and designing data communications networks. AFIPS Conference Proceedings 43 (1974): 583-591.
70. Kleinrock, Leonard; Naylor, Wm. E.; and Opderbeck, Holger. A study of line overhead in the ARPAnet. Communications of ACM 19, No. 1 (1976): 3-12.
71. Knight, John R. A case study: airlines reservation system. Proceedings of the IEEE 60, No. 11 (1972): 1423-1481.
72. L'Archeveque, J. V. R. and Yan, G. On the selection of architectures for distributed computer systems in real-time applications. IEEE Trans. on Nuclear Science NS-24, No. 1 (1977): 454-459.
73. Lennon, William J.; Barrett, Ronald C.; and Spies, John T. A mini-computer research network. Digest of Papers, COMPCON 73 (Spring 1973): 191-194.
74. Loomis, M. S. and Popek, G. J. A model for data base distribution. Trends and Applications: Computer Networks, IEEE Computer Society (1976): 162-169.

75. Lynch, Arthur. Distributed processing solves mainframe problems. *Data Communications* 5 (November/December 1976): 17-22.
76. McGregor, Patrick. Effective use of data communications hardware. *AFIPS Conference Proceedings* 43 (1974): 565-575.
77. Mendicino, Sam F. and Sutherland, George G. Performance measurements in LLL octopus computer network. *Digest of Papers, COMPCON 73* (Spring 1973): 109-112.
78. Merrill, Paul K. Computer netting and distributed data: a tutorial. *IBM Research Report*. White Plains, New York: IBM Corporation, 1974.
79. Mills, David L. Communication software. *Proceedings of the IEEE* 60, No. 11 (1972): 1333-1341.
80. Newport, C. B. and Ryzlak, Jan. Communication processors. *Proceedings of the IEEE* 60, No. 11 (1972): 1321-1332.
81. Owens, Jerry L. A user's view of the Lawrence Livermore Laboratory's computer networks. *Digest of Papers, COMPCON 73* (Spring 1973): 75-78.
82. Pohm, Arthur V. Cost/performance perspectives of paging with electronic and electromechanical backstores. *Proceedings of the IEEE* 63, No. 8 (1975): 1123-1128.
83. Pyke, T. N. and Blanc, R. P. Computer networking technology - a state of the art review. *Computer* 6 (August 1973): 13-19.
84. Pyke, T. N., Jr. Assuring user service quality in a distributed computer network. *Digest of Papers, COMPCON 76* (Spring 1976): 73-76.
85. Raeside, D. E. Bayesian statistics: mathematical tools for computers. *Computer* 8 (July 1975): 65-73.
86. Reynolds, Carl H. Issues in centralization. *Datamation* 23 (March 1977): 91-93, 97.
87. Roberts, Lawrence G. Network rationale - a 5 year reevaluation. *Digest of Papers, COMPCON 73* (Spring 1973): 3-5.

88. Roberts, Lawrence G. Data by the packet. IEEE Spectrum 11 (February 1974): 46-51.
89. Roberts, Lawrence G. and Wessler, Barry D. Computer network development to achieve resource sharing. AFIPS Spring Joint Computer Conference 36 (1970): 543-549.
90. Rosen, Saul and Steele, John M. A local computer network. Digest of Papers, COMPCON 73 (Spring 1973): 129-132.
91. Schutz, Gerald C. and Clark, George E. Data communication standards. Computer 7 (February 1974): 32-37.
92. Schwartz, M.; Boorstyn, R.; and Pickholtz, R. Terminal-oriented computer-communications networks. Proceedings of the IEEE 60, No. 11 (1972): 1408-1423.
93. Schwartz, Mischa. Computer-communication network design and analysis. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1977.
94. Scrupski, S. E. Distributed processing grows as its hardware and software develop. Electronics 49 (May 1976): 91-97.
95. Segall, Adrian. Dynamic file assignment in a computer network. IEEE Trans. on Automatic Control AC-21, No. 2 (1976): 161-173.
96. Shatz, Vernon L. Computer networks for retail stores. Computer 6 (April 1973): 21-25.
97. Stanford WYLBUR Reference Manual. Computation Center, Iowa State University, Ames, Iowa, 1976.
98. Stubblefield, F. W. A file management for experiment control. IEEE Trans. on Nuclear Science NS-24, NO. 1 (1977): 460-468.
99. Tenkhoff, P. A. and Collard, J. C. The common carriers' uncommon offerings. Datamation 21 (April 1975): 45-47.
100. Wecker, Stuart. A design for a multiple processor operating environment. Digest of Papers, COMPCON 73 (Spring 1973): 143-146.

101. Welch, Terry A. Memory hierarchy configuration analysis. Trans. on Computers C-27, No. 5 (1978): 408-413.
102. Williman, A. O. and Jelinek, H. J. Introduction to LSI microprocessor developments. Computer 9 (June 1976): 34-46.
103. Willner, Erich R. and Lindinger, Joseph L. COS: a portable communications oriented system. Computer 6 (February 1973): 20-26.
104. Zemrowski, K. M. Problems of data base use in distributed networks. 15th Annual Technical Symposium, Washington, D. C. Chapter of ACM (June 1976): 51-58.

## VIII. ACKNOWLEDGMENTS

The author wishes to thank R. M. Stewart for his encouragement and support during the course of this endeavor. The helpful suggestions, criticisms, and guidance provided to the author by A. V. Pohm were greatly appreciated. The author also thanks J. O. Kopplin for his generous support of the required computer time for this project. Finally, the author desires to state a special expression of gratitude to Jo for her selfless forbearance, to Kim for her steadfast inspiration, to Scott for his assumption of responsibilities beyond his years, and to all of them for accepting the sacrifices and showing the patience which made this effort possible.

This work was partially supported by the Ames Laboratory of the United States Energy Research and Development Administration and the Iowa State University Electronic Affiliates program.

IX. APPENDIX A:  
LISTING FOR THE ROUTINE HONE

```

HCNE:  PROC OPTIONS(MAIN);
DCL 1  NETWGRK EXTERNAL,
      2  SITE(25),
      3  STATUS DEC FIXED(1) INIT((25)0),
      3  XDIST DEC FIXED(5,1) INIT((25)0),
      3  YDIST DEC FIXED(5,1) INIT((25)0),
      3  ASGND_PROS DEC FIXED(2) INIT((25)0),
      3  PROS_LOAD DEC FLOAT(5) INIT((25)0),
      3  PROS_UF DEC FLOAT(5) INIT((25)0),
      2  LINE(99),
      3  INSITE DEC FIXED(2) INIT((99)0),
      3  FINSITE DEC FIXED(2) INIT((99)0),
      3  L_TYPE DEC FIXED(2) INIT((99)0),
      3  L_LOAD DEC FLOAT(5) INIT((99)0),
      3  L_UF DEC FLOAT(5) INIT((99)0);
DCL 1  HRDWARE EXTERNAL,
      2  PTYPE(10),
      3  PROS_LVL# DEC FIXED(1) INIT((10)0),
      3  I_RATE DEC FLOAT(5) INIT((10)0),
      3  PRIMSTR DEC FLOAT(5) INIT((10)0),
      3  SCNDSTR DEC FLOAT(5) INIT((10)0),
      3  MDSPET DEC FIXED(2) INIT((10)0),
      3  CHAN_RATE DEC FLOAT(5) INIT((10)0),
      3  PCOST DEC FIXED(8,2) INIT((10)0),
      2  LTYPE(10),
      3  XMSN_RATE DEC FLOAT(5) INIT((10)0),
      3  LFCOST DEC FIXED(5,1) INIT((10)0),
      3  LVCST1 DEC FIXED(4,2) INIT((10)0),
      3  LVCST2 DEC FIXED(4,2) INIT((10)0),
      3  LVCST3 DEC FIXED(4,2) INIT((10)0),
      3  LVCST4 DEC FIXED(4,2) INIT((10)0),
      3  LVCST5 DEC FIXED(4,2) INIT((10)0);
DCL 1  SFTWARE EXTERNAL,
      2  PCNTYPE(99),
      3  I_STEPS DEC FLOAT(5) INIT((99)0),

```

```

        3 FCNTSIZE DEC FLOAT(5) INIT((99)0),
        3 FCNFSIZE DEC FLOAT(5) INIT((99)0),
    2 FILETYPE(99),
        3 MEDIUM DEC FIXED(2) INIT((99)0),
        3 FILE_SIZE DEC FLOAT(5) INIT((99)0);
DCL 1 JOB(50) EXTERNAL,
    2 NO_JOBSTEPS DEC FIXED(2) INIT((50)0),
    2 ARVL_RATE DEC FLOAT(5) INIT((50)0),
    2 RSPNS_TIME DEC FLOAT(5) INIT((50)0),
    2 FRSTEP# DEC FIXED(3) INIT((50)0),
    2 LASTEP# DEC FIXED(3) INIT((50)0);
DCL 1 JOBSTEP(250) EXTERNAL,
    2 REQFCN DEC FIXED(3) INIT((250)0),
    2 REQFILE DEC FIXED(3) INIT((250)0),
    2 MSG_LENGTH DEC FLOAT(5) INIT((250)0),
    2 LVL_SPEC DEC FIXED(1) INIT((250)0),
    2 REQ_LVL DEC FIXED(1) INIT((250)0),
    2 OWNING_JOB DEC FIXED(2) INIT((250)0),
    2 STEP# DEC FIXED(2) INIT((250)0),
    2 ASGND_SITE DEC FIXED(2) INIT((250)0),
    2 PROS_TIME DEC FLOAT(5) INIT((250)0),
    2 LINE_DIY DEC FLOAT(5) INIT((250)0),
    2 LVL_ASGND(4) BIT(1) INIT((1000) (1) '0'B);
DCL 1 PROCOMP(10) EXTERNAL,
    2 CFILE(99) BIT(1) INIT((990) (1) '0'B),
    2 CFCN(99) BIT(1) INIT((990) (1) '0'B);
DCL 1 UPBOUND EXTERNAL,
    2 NSITE DEC FIXED(2),
    2 NLINE DEC FIXED(2),
    2 NSTEP DEC FIXED(3),
    2 NFCN DEC FIXED(2),
    2 NFILE DEC FIXED(2),
    2 NJOB DEC FIXED(2),
    2 NLINE DEC FIXED(2),
    2 MPROS DEC FIXED(2),

```



```

      2 NLEVEL DEC FIXED (1);
DCL 1 CTLPARM EXTERNAL,
      2 ISEL BIT (1),
      2 XN DEC FIXED (4),
      2 THRSO DEC FIXED (4),
      2 KCENT BIT (1),
      2 TRAC BIT (1),
      2 PLIST (4) DEC FIXED (2);
DCL 1 NUPBOUND,
      2 NNSITE DEC FIXED (2),
      2 NNLINE DEC FIXED (2),
      2 NNSTEP DEC FIXED (3),
      2 NNFCN DEC FIXED (2),
      2 NNFILE DEC FIXED (2),
      2 NNJOB DEC FIXED (2),
      2 NNLINE DEC FIXED (2),
      2 NNPROS DEC FIXED (2),
      2 NNLEVEL DEC FIXED (1);
DCL HDRPS (4) CHAR (60);
DCL INDEF BIT (1);
DCL UJSTOP BIT (1);
DCL UPTOS BIT (1);
DCL UOSASS BIT (1);
DCL UANLYZ BIT (1);
DCL NUNET BIT (1);
DCL HDUP (3) CHAR (64);
DCL HDOP (3) CHAR (65);
DCL HDPC1 CHAR (48);
DCL HDPC2 CHAR (9);
DCL CLID CHAR (7);
DCL PID (10) FIXED DEC (2);
DCL PCID CHAR (7);
DCL CFLID CHAR (7);
DCL CFNID CHAR (7);
DCL HDNS (3) CHAR (62);

```

```

DCL NSID CHAR(5);
DCL HDNL(3) CHAR(62);
DCL NLID CHAR(5) INIT('LINE ');
DCL HDHP(3) CHAR(80);
DCL HPID CHAR(6);
DCL HDHL(3) CHAR(80);
DCL HLID CHAR(6);
DCL HDPN(3) CHAR(55);
DCL SPID CHAR(8);
DCL HDPL(3) CHAR(36);
DCL SFLID CHAR(9);
DCL HDJB(3) CHAR(60);
DCL JBID CHAR(4);
DCL HDJS(4) CHAR(80);
DCL JSID CHAR(5);
DCL INFILE FILE INPUT STREAM;
DCL OUTFILE FILE OUTPUT STREAM;
PUT PAGE;
/* GET RUNPARMS */
GET FILE(INFILE) EDIT
(HDRPS) (R(FHDRP))
(INDEF,UJSTOP,UPTOS,UOSASS,UANLYZ,NUNET)
(R(FIRPM));

/* GET UPBOUND */
GET FILE(INFILE) SKIP(3) EDIT
(HDUP) (R(FHDUP))
(UPBOUND) (R(FUPBD))
(NUPBOUND) (R(FUPBD));

/* GET CTLPARM */
GET FILE(INFILE) SKIP(3) EDIT
(HDCP) (R(FHDCP))
(CTLPARM) (R(FCLPM));

/* GET PROCOMP.CFILE */
GET FILE(INFILE) SKIP(3) EDIT
(HDPC1) (R(FHDP1))

```

```

                (HDPC2, (PID(I) DO I=1 TO MPROS)) (R(FHDP2))
                ((CFLID, JJ, (PROCOMP.CFILE(I, J) DO I=1 TO MPROS)
                DO J=1 TO NFILE)) (R(FPCP));

/* GET PROCOMP.CFCN */
    GET FILE(INFILE) SKIP(4) EDIT
        ((CFNID, JJ, (PROCOMP.CFCN(I, J) DO I=1 TO MPROS)
        DO J=1 TO NFCN)) (R(FPCP));

/* GET NETWORK.SITE */
    GET FILE(INFILE) SKIP(3) EDIT
        (HDNS) (R(FHDN))
        ((NSID, JJ, NETWORK.SITE(J) DO J=1 TO NSITE))
        (R(FNTWS));

/* GET NETWORK.LINE */
    GET FILE(INFILE) SKIP(3) EDIT
        (HDNL) (R(FHDN))
        ((NLID, JJ, NETWORK.LINE(J) DO J=1 TO NLINE))
        (R(FNTWL));

/* GET HRDWARE.PTYPE */
    GET FILE(INFILE) SKIP(3) EDIT
        (HDHP) (R(FHDHP))
        ((HPID, JJ, HRDWARE.PTYPE(J) DO J=1 TO MPROS)) (R(FHDWP));

/* GET HRDWARE.LTYPE */
    GET FILE(INFILE) SKIP(3) EDIT
        (HDHL) (R(FHDHL))
        ((HLID, JJ, HRDWARE.LTYPE(J) DO J=1 TO MLINE)) (R(FHDWL));

/* GET SFTWARE.FCNTYPE */
    GET FILE(INFILE) SKIP(3) EDIT
        (HDFN) (R(FHDFN))
        ((SFNID, JJ, SFTWARE.FCNTYPE(J) DO J=1 TO NFCN))
        (R(FSWFN));

/* GET SFTWARE.FILETYPE */
    GET FILE(INFILE) SKIP(3) EDIT
        (HDFL) (R(FHDFL))
        ((SFLID, JJ, SFTWARE.FILETYPE(J) DO J=1 TO NFILE))
        (R(FSWFL));

```

```

/* GET JOBS */
  GET FILE(INFILE) SKIP(3) EDIT
    (HDJB) (R(FHDJB))
    ((JBID,JJ,JOB(J) DO J=1 TO NJOB)) (R(FJOB));
/* GET JOBSTEPS */
  GET FILE(INFILE) SKIP(3) EDIT
    (HDJS) (R(FHDJS))
    ((JSID,JJ,JOBSTEP(J) DO J=1 TO NSTEP)) (R(FJBSP));
/* SET DATA FORMAT FOR NEW NETWORK CONFIGURATION */
  IF NUNET THEN DO;
    NSITE=NNSITE;
    NLINE=NNLINE;
    NSTEP=NNSTEP;
    NFCN=NNFCN;
    NFILE=NNFILE;
    NJOB=NNJOB;
    MLINE=NMLINE;
    MPROS=NMPROS;
    NLEVEL=NNLEVEL;
    GOTO OUTPUT;
  END;
/* INITIALIZE DEFAULTS */
  IF INDEF THEN DO;
    KCENT='0'B;
    ISEL='0'B;
    XN=100;
    THRS=1;
    TRAC='0'B;
    PLIST(1)=0;
    PLIST(2)=0;
    PLIST(3)=0;
    PLIST(4)=0;
  END;
/* CALL JOBSTEP TO PROCESSOR ASSIGNMENT */
  IF UJSTOP THEN CALL JSTOP;

```

```

/* CALL PROCESSOR TO SITE ASSIGNMENT */
  IF UPTCS THEN CALL PTOS;
/* CALL OPTIMAL SITE LOCATOR */
  IF UOSASS THEN CALL OSASS;
/* CALL ANALYZER */
  IF UANLYZ THEN CALL ANLYZ;
/* PUT RUNPARMS */
OUTPUT: PUT FILE(OUTFILE) EDIT
      (HDRPS) (R(FHDRP))
      ('<', INDEF, '>', '<', UJSTOP, '>', '<', UPTOS, '>',
      '<', UOSASS, '>', '<', UANLYZ, '>', '<', NUNET, '>')
      (R(FRPRM));

/* PUT UPBOUND */
  PUT FILE(OUTFILE) SKIP(3) EDIT
      (HDUP) (R(FHDUP))
      (UPBOUND) (R(FUPBD));

/* PUT CTLPARM */
  PUT FILE(OUTFILE) SKIP(4) EDIT
      (HDCP) (R(FHDCP))
      (CTLPARM) (R(FCLPM));

/* PUT PROCOMP.CFILE */
  PUT FILE(OUTFILE) SKIP(3) EDIT
      (HDPC1) (R(FHDP1))
      (HDPC2, (PID(I) DO I=1 TO MPROS)) (R(FHDP2))
      ((CFLID, J, (PROCOMP.CFILE(I, J) DO I=1 TO MPROS)
      DO J=1 TO NFILE)) (R(FPCP));

/* PUT PROCOMP.CFCN */
  PUT FILE(OUTFILE) SKIP(3) EDIT
      (HDPC2, (PID(I) DO I=1 TO MPROS)) (R(FHDP2))
      ((CFNID, J, (PROCOMP.CFCN(I, J) DO I=1 TO MPROS)
      DO J=1 TO NFCN)) (R(FPCP));

/* PUT NETWORK.SITE */
  PUT FILE(OUTFILE) SKIP(3) EDIT
      (HDNS) (R(FHDN))
      ((NSID, J, NETWORK.SITE(J) DO J=1 TO NSITE))

```

```

(R (FNTWS)) ;

/* PUT NETWORK.LINE */
  PUT FILE(OUTFILE) SKIP(3) EDIT
    (HDNL) (R (PHDN))
    ((NLID,J,NETWORK.LINE(J) DO J=1 TO NLINE))
    (R (FNTWL)) ;

/* PUT HRDWARE.PTYPE */
  PUT FILE(OUTFILE) SKIP(3) EDIT
    (HDHP) (R (FHDHP))
    ((HPID,J,HRDWARE.PTYPE(J) DO J=1 TO MPROS)) (R (FHDWP)) ;

/* PUT HRDWARE.LTYPE */
  PUT FILE(OUTFILE) SKIP(3) EDIT
    (HLHL) (R (FHDHL))
    ((HLID,J,HRDWARE.LTYPE(J) DO J=1 TO MLINE)) (R (FHDWL)) ;

/* PUT SFTWARE.FCNTYPE */
  PUT FILE(OUTFILE) SKIP(3) EDIT
    (HDFN) (R (FHDFN))
    ((SFNID,J,SFTWARE.FCNTYPE(J) DO J=1 TO NFCN))
    (R (FSWFN)) ;

/* PUT SFTWARE.FILETYPE */
  PUT FILE(OUTFILE) SKIP(3) EDIT
    (HDFL) (R (FHDFL))
    ((SFLID,J,SFTWARE.FILETYPE(J) DO J=1 TO NFILE))
    (R (FSWFL)) ;

/* PUT JOBS */
  PUT FILE(OUTFILE) SKIP(3) EDIT
    (HDJB) (R (FHDJB))
    ((JBID,J,JOB(J) DO J=1 TO NJOB)) (R (FJOB)) ;

/* PUT JOBSTEPS */
  PUT FILE(OUTFILE) SKIP(3) EDIT
    (HDJS) (R (FHDJS))
    ((JSID,J,JOBSTEP(J) DO J=1 TO NSTEP)) (R (FJBSP)) ;

/* FORMATS FOR RUNPARMS */
FIRPM:FORMAT (COL(1),X(7),B(2),X(11),B(2),X(6),B(2),3(X(7),B(2))) ;
FHERP:FORMAT (4(COL(1),A(60))) ;

```

```

FRFRM: FORMAT (COL (1), X (6), A (1), B (2), A (1), X (9), A (1), B (2), A (1), X (4),
              A (1), B (2), A (1), 3 (X (5), A (1), B (2), A (1))) ;
/* FORMATS FOR UPBOUND */
FHDUP: FORMAT (3 (COL (1), A (64))) ;
FUPBD: FORMAT (COL (1), 9 (X (3), F (4))) ;
/* FORMATS FOR CTLPARM */
FHDCP: FORMAT (3 (COL (1), A (65))) ;
FCLPM: FORMAT (COL (1), X (3), B (3), 2 (X (2), F (5)),
              X (1), 2 (X (3), B (3)), X (5), 4 (X (2), F (3))) ;
/* FORMATS FOR PROCOMP */
FHDP1: FORMAT (COL (1), A (45)) ;
FHDP2: FORMAT (COL (1), A (9), (MPROS) (X (2), F (2))) ;
FPCP:  FORMAT (COL (1), A (7), F (2), (MPROS) (X (3), B (1))) ;
/* FORMATS FOR NETWORK */
FHDN:  FORMAT (3 (COL (1), A (62))) ;
FNTWS: FORMAT (COL (1), A (5), F (2), X (2), F (3), 2 (X (3), F (6, 1)), X (2),
              F (4), X (1), E (11, 4), X (2), E (11, 4)) ;
FNTWL: FORMAT (COL (1), A (5), F (2), 3 (X (4), F (4)), 2 (X (3), E (11, 4))) ;
/* FORMATS FOR HARDWARE */
FHDHP: FORMAT (3 (COL (1), A (80))) ;
FHDWP: FORMAT (COL (1), A (6), F (2), X (1), F (3), 3 (E (11, 4)),
              X (3), F (2), X (1), E (11, 4), X (2), F (9, 2)) ;
FHDHL: FORMAT (3 (COL (1), A (80))) ;
FHDWL: FORMAT (COL (1), A (6), F (2), X (1), E (11, 4),
              X (3), F (6, 1), 5 (X (3), F (5, 2))) ;
/* FORMATS FOR SOFTWARE */
FHDFN: FORMAT (3 (COL (1), A (55))) ;
FSWFN: FORMAT (COL (1), A (8), F (2), X (2), 3 (X (3), E (11, 4))) ;
FHDPL: FORMAT (3 (COL (1), A (36))) ;
FSWFL: FORMAT (COL (1), A (9), F (2), X (6), F (4), X (3), E (11, 4)) ;
/* FORMATS FOR JOB */
FHDJB: FORMAT (3 (COL (1), A (60))) ;
FJCB:  FORMAT (COL (1), A (4), F (2), X (5), F (2), 2 (X (3), E (11, 4)), X (5),
              F (3), X (5), F (3)) ;
/* FORMATS FOR JOBSTEP */

```

```
FHDJS:FORMAT (4 (COL (1) , A (80) ) ) ;  
FJESP:FORMAT (COL (1) , A (5) , F (3) , X (1) , 2 (X (1) , F (3) ) , E (11,4) ,  
2 (X (1) , F (2) ) , 2 (X (1) , F (3) ) , X (1) , F (4) , 2 (E (11,4) ) ,  
X (1) , 4 (X (1) , B (1) ) ) ;  
END HONE;
```



K. APPENDIX B:  
LISTING FOR THE SUBROUTINE JSTOP

```

JSTOP:   PROC;
DCL 1 NETWORKRK EXTERNAL,
  2 SITE(25),
    3 STATUS DEC FIXED(1),
    3 XDIST DEC FIXED(5,1),
    3 YDIST DEC FIXED(5,1),
    3 ASGND_PROS DEC FIXED(2),
    3 PROS_LOAD DEC FLOAT(5),
    3 PROS_UF DEC FLOAT(5),
  2 LINE(99),
    3 INSITE DEC FIXED(2),
    3 FINSITE DEC FIXED(2),
    3 L_TYPE DEC FIXED(2),
    3 L_LOAD DEC FLOAT(5),
    3 L_UF DEC FLOAT(5);
DCL 1 HARDWARE EXTERNAL,
  2 PTYPE(10),
    3 PROS_LVL# DEC FIXED(1),
    3 I_RATE DEC FLOAT(5),
    3 PRIMSTR DEC FLOAT(5),
    3 SCNDSTR DEC FLOAT(5),
    3 MDSPT DEC FIXED(2),
    3 CHAN_RATE DEC FLOAT(5),
    3 PCOST DEC FIXED(8,2),
  2 LTYPE(10),
    3 XMSN_RATE DEC FLOAT(5),
    3 LFCOST DEC FIXED(5,1),
    3 LVCST1 DEC FIXED(4,2),
    3 LVCST2 DEC FIXED(4,2),
    3 LVCST3 DEC FIXED(4,2),
    3 LVCST4 DEC FIXED(4,2),
    3 LVCST5 DEC FIXED(4,2);
DCL 1 SOFTWARE EXTERNAL,
  2 FCNTYPE(99),
    3 I_STEPS DEC FLOAT(5),

```

```

        3 FCNTSIZE DEC FLOAT(5),
        3 FCNFSIZE DEC FLOAT(5),
    2 FILETYPE(99),
        3 MEDIUM DEC FIXED(2),
        3 FILE_SIZE DEC FLOAT(5);
DCL 1 JOB(50) EXTERNAL,
    2 NO_JOBSTEPS DEC FIXED(2),
    2 ARVL_RATE DEC FLOAT(5),
    2 RSPNS_TIME DEC FLOAT(5),
    2 FRSTEP# DEC FIXED(3),
    2 LASTEP# DEC FIXED(3);
DCL 1 JOBSTEP(250) EXTERNAL,
    2 REQFCN DEC FIXED(3),
    2 REQFILE DEC FIXED(3),
    2 MSG_LENGTH DEC FLOAT(5),
    2 LVL_SPEC DEC FIXED(1),
    2 REQ_LVL DEC FIXED(1),
    2 OWNING_JOB DEC FIXED(2),
    2 STEP# DEC FIXED(2),
    2 ASGND_SITE DEC FIXED(2),
    2 PROS_TIME DEC FLOAT(5),
    2 LINE_DIY DEC FLOAT(5),
    2 LVL_ASGND(4) BIT(1);
DCL 1 PROCOMP(10) EXTERNAL,
    2 CFILE(99) BIT(1),
    2 CFCN(99) BIT(1);
DCL 1 UPBOUND EXTERNAL,
    2 NSITE DEC FIXED(2),
    2 NLINE DEC FIXED(2),
    2 NSTEP DEC FIXED(3),
    2 NFCN DEC FIXED(2),
    2 NFILE DEC FIXED(2),
    2 NJOB DEC FIXED(2),
    2 MLINE DEC FIXED(2),
    2 MPROS DEC FIXED(2),

```

```

      2 NLEVEL DEC FIXED(1);
DCL 1 CTLPARM EXTERNAL,
      2 ISEL BIT(1),
      2 XN DEC FIXED(4),
      2 THRSO DEC FIXED(4),
      2 KCENT BIT(1),
      2 TRAC BIT(1),
      2 PLIST(4) DEC FIXED(2);
/* ROUTINE TO ASSIGN JOBSTEPS TO PROCESSOR TYPES */
DCL 1 CJSTOP(250),
      2 Z(10) BIT(1) INIT((2500) (1)'0'B),
      2 LCL DEC FIXED(1) INIT((48) 0);
DCL TABLE(4,12) DEC FIXED(2) INIT((48) 0);
DCL IASM BIT(1);
/* DETERMINE FILE-PROCESSOR COMPATIBILITIES */
KFLOOP: DO KF=1 TO NFILE;
KFLOOP: DO KP=1 TO MPROS;
      CFILE(KP,KF)='0'B;
      IF MEDIUM(KF)=1 &
        ((MDSPRT(KP)=1|MDSPRT(KP)=5) &
          (FILE_SIZE(KF)<=SCNDSTR(KP))) |
        (MDSPRT(KP)=4|MDSPRT(KP)=7)) THEN DO;
        CFILE(KP,KF)='1'B;
        GOTO FKLOOP;
      END;
      IF MEDIUM(KF)=2 &
        ((MDSPRT(KP)=2) | (MDSPRT(KP)=4)
          | (MDSPRT(KP)=6) | (MDSPRT(KP)=7)) &
          (FILE_SIZE(KF)<=SCNDSTR(KP))) THEN DO;
        CFILE(KP,KF)='1'B;
        GOTO FKLOOP;
      END;
      IF MEDIUM(KF)=3 &
        ((MDSPRT(KP)=3) & (FILE_SIZE(KF)<=SCNDSTR(KP))) |
        (MDSPRT(KP)=5|MDSPRT(KP)=6|MDSPRT(KP)=7)) THEN

```

```

                                CFILE(KP,KF)='1'B;
FKPLOOP:   END KLOOP;
FKFLOOP:   END KFLOOP;
/* CREATION OF Z MAP */
ILOOP: DO I=1 TO NSTEP;
        IFCN=REQFCN(I);
        IFILE=REQFILE(I);
JLOOP: DO J=1 TO MPROS;
        Z(I,J)='0'B;
        IF ~CFCN(J,IFCN) THEN GOTO FJLOOP;
        IF IFILE~=0 THEN
            IF ~CFILE(J,IFILE) THEN GOTO FJLOOP;
        IF LVL_SPEC(I)=2 THEN
            IF REQ_LVL(I)~=PROS_LVL#(J) THEN GOTO FJLOOP;
        IF ASGND_SITE(I)=0 THEN GOTO IJCMP;
            ELSE II=ASGND_SITE(I);
        IF ASGND_PROS(II)=0 THEN GOTO IJCMP;
        IF ASGND_PROS(II)~=J THEN GOTO FJLOOP;
/* JOBSTEP I IS COMPATIBLE WITH PROCESSOR J */
IJCMP:   Z(I,J)='1'B;
FJLOOP:   END JLOOP;
        END ILOCP;
/* CREATION OF TABLE - PCOST ORDERED */
ILOOP2: DO I=1 TO NLEVEL;
        K=2;
JLOOP2: DO J=1 TO MPROS;
        IF PROS_LVL#(J)~=I THEN GOTO FJLOOP2;
        IF K=2 THEN DO;
            TABLE(I,3)=J;
            GOTO INCK;
        END;
        TABLE(I,K+1)=J;
IKLOOP: DO IK=K TO 3 BY -1;
        JK=TABLE(I,IK+1);
        JJ=TABLE(I,IK);

```

```

                IF PCOST(JJ)>PCOST(JK) THEN DO;
                    TABLE(I,IK+1)=JJ;
                    TABLE(I,IK)=JK;
                    END;
                ELSE GOTO INCK;
FIKLOOP:      END IKLOOP;
INCK:        K=K+1;
FJLOOP2:     END JLOOP2;
            TABLE(I,1)=1;
            TABLE(I,2)=K-2;
            END ILCOP2;
/* FIND LOWEST COMPATIBLE LEVEL OF EACH JSTEP */
ILOOP3: DO I=1 TO NSTEP;
        MINL=9;
        JLOOP3: DO J=1 TO MPROS;
                IF -Z(I,J) THEN GOTO FJLOOP3;
                ELSE DO;
                    IL=PROS_LVL*(J);
                    IF IL<MINL THEN MINL=IL;
                    END;
                FJLOOP3: END JLOOP3;
                IF MINL=9 THEN DO;
                    JSTP#=0;
                    GOTO FAIL;
                    END;
                LCL(I)=MINL;
        FILOOP3: END ILOOP3;
/* CHOOSE PROCESSOR FOR EACH LEVEL NOT PREVIOUSLY SPECIFIED */
LLOOP: DO L=1 TO NLEVEL;
        IF PLIST(L)-=0 THEN GOTO FLLOOP;
        II=TABLE(L,1)+2;
        PROS=TABLE(L,II);
        ILOOP4: DO I=1 TO NSTEP;
        TRYP: IF LCL(I)=L THEN
                IF Z(I,PROS) THEN GOTO FILOOP4;

```

```

ELSE DO;
  TABLE(L,1)=TABLE(L,1)+1;
  IF TABLE(L,1)>TABLE(L,2) THEN DO;
    JSTP#=I;
    GOTO FAIL;
  END;
ELSE DO;
  II=TABLE(L,1)+2;
  PROS=TABLE(L,II);
  GOTO ILOOP4;
END;

FILOOP4:  END ILOOP4;
FLLOOP:  END LLOOP;
/* RECORD CHOSEN PROCESSORS IN PLIST */
JLCOP4:  DO J=1 TO NLEVEL;
  IF PLIST(J)≠0 THEN GOTO FJLOOP4;
  II=TABLE(J,1)+2;
  PLIST(J)=TABLE(J,II);
FJLOOP4:  END JLOOP4;
  IF KCENT THEN GOTO CENT;
/* DECENTRALIZED JOBSTEP ASSIGNMENT */
DECENT:  DO I=1 TO NSTEP;
JLOOP5:  DO J=1 TO NLEVEL;
  IF LCL(I)=J THEN LVL_ASGND(I,J)='1'B;
  ELSE LVL_ASGND(I,J)='0'B;
  END JLOOP5;
END DECENT;
GOTO FINISH;
/* CENTRALIZED JOBSTEP ASSIGNMENT */
CENT:    DO I=1 TO NSTEP;
  IASM='0'B;
  IJOB=OWNING_JOB(I);
  IF(I=FRSTEP#(IJOB)|I=LASTEP#(IJOB)) THEN DO;
JLOOP6:  DO J=1 TO NLEVEL;
  IF J=LCL(I) THEN LVL_ASGND(I,J)='1'B;

```

```

                ELSE LVL_ASGND(I,J)='0'B;
            END JLOOP6;

            END;
        ELSE DO;
JJLOOP:    DO JJ=1 TO NLEVEL;
            J=NLEVEL-JJ+1;
            II=PLIST(J);
            IF IASM THEN LVL_ASGND(I,J)='0'B;
                ELSE DO;
                    LVL_ASGND(I,J)=Z(I,II);
                    IASM=LVL_ASGND(I,J);
                END;
            END JJLOOP;
        END CENT;
        GOTO FINISH;
FAIL:     PUT SKIP EDIT('TRANSACTION ALLOCATION FAILED,JSTP# ',JSTP#)
            (COL(1),A(36),F(3));
FINISH:IF TRAC THEN DO;
/* PRINT Z ARRAY */
    PUT SKIP EDIT('Z MAP','PROCESSOR',(J DO J=1 TO MPROS),
        'MINLV',('JOBSTEP',I,(Z(I,J) DO J=1 TO MPROS),LCL(I)
        DO I=1 TO NSTEP)) (COL(4),A(5),COL(7),A(9),
        (MPROS) (X(2),F(2)),X(2),A(5),SKIP,(NSTEP) (COL(5),A(7),
        X(1),F(3)),(MPROS) (X(3),B(1)),X(4),F(1)));
/* PRINT TABLE */
    PUT SKIP(4) EDIT('TABLE','PTR','TYPES','PROCESSOR LIST',
        ('LEVEL',I,(TABLE(I,J) DO J=1 TO (MPROS+2))
        DO I=1 TO NLEVEL)) (COL(4),A(5),SKIP,COL(18),
        A(3),X(1),A(5),X(2),A(14),SKIP,(NLEVEL)
        (COL(9),A(5),X(1),F(1),(MPROS+2) (X(3),F(2))));
    PUT SKIP(3) EDIT(('LEVEL',J,': PROCESSOR IS ',PLIST(J)
        DO J=1 TO NLEVEL)) ((NLEVEL) (COL(1),A,F(1),A,F(2)));
    END;
    PUT SKIP(3) LIST('END OF JOBSTEP TO PROCESSOR ASSIGNMENT');
    END JSTOP;

```



XI. APPENDIX C:  
LISTING FOR THE SUBROUTINE PTOS

```

PTOS: PROC;
DCL 1 NETWORK EXTERNAL,
  2 SITE(25),
    3 STATUS DEC FIXED(1),
    3 XDIST DEC FIXED(5,1),
    3 YDIST DEC FIXED(5,1),
    3 ASGND_PROS DEC FIXED(2),
    3 PROS_LOAD DEC FLOAT(5),
    3 PROS_UP DEC FLOAT(5),
  2 LINE(99),
    3 INSITE DEC FIXED(2),
    3 FINSITE DEC FIXED(2),
    3 L_TYPE DEC FIXED(2),
    3 L_LOAD DEC FLOAT(5),
    3 L_UP DEC FLOAT(5);
DCL 1 HARDWARE EXTERNAL,
  2 PTYPE(10),
    3 PROS_LVL# DEC FIXED(1),
    3 I_RATE DEC FLOAT(5),
    3 PRIMSTR DEC FLOAT(5),
    3 SCNDSTR DEC FLOAT(5),
    3 MDSPFT DEC FIXED(2),
    3 CHAN_RATE DEC FLOAT(5),
    3 PCOST DEC FIXED(8,2),
  2 LTYPE(10),
    3 XMSN_RATE DEC FLOAT(5),
    3 LFCOST DEC FIXED(5,1),
    3 LVCST1 DEC FIXED(4,2),
    3 LVCST2 DEC FIXED(4,2),
    3 LVCST3 DEC FIXED(4,2),
    3 LVCST4 DEC FIXED(4,2),
    3 LVCST5 DEC FIXED(4,2);
DCL 1 SOFTWARE EXTERNAL,
  2 FCNTYPE(99),
    3 I_STEPS DEC FLOAT(5),

```

```

        3 FCNTSIZE DEC FLOAT(5),
        3 FCNFSIZE DEC FLOAT(5),
    2 FILETYPE(99),
        3 MEDIUM DEC FIXED(2),
        3 FILE_SIZE DEC FLOAT(5);
DCL 1 JOB(50) EXTERNAL,
    2 NO_JOBSTEPS DEC FIXED(2),
    2 ARVL_RATE DEC FLOAT(5),
    2 RSPNS_TIME DEC FLOAT(5),
    2 FRSTEP# DEC FIXED(3),
    2 LASTEP# DEC FIXED(3);
DCL 1 JOBSTEP(250) EXTERNAL,
    2 REQFCN DEC FIXED(3),
    2 REQFILE DEC FIXED(3),
    2 MSG_LENGTH DEC FLOAT(5),
    2 LVL_SPEC DEC FIXED(1),
    2 REQ_LVI DEC FIXED(1),
    2 OWNING_JOB DEC FIXED(2),
    2 STEP# DEC FIXED(2),
    2 ASGND_SITE DEC FIXED(2),
    2 PROS_TIME DEC FLOAT(5),
    2 LINE_DIY DEC FLOAT(5),
    2 LVL_ASGND(4) BIT(1);
DCL 1 PROCOMP(10) EXTERNAL,
    2 CFILE(99) BIT(1),
    2 CFCN(99) BIT(1);
DCL 1 UPBOUND EXTERNAL,
    2 NSITE DEC FIXED(2),
    2 NLINE DEC FIXED(2),
    2 NSTEP DEC FIXED(3),
    2 NFCN DEC FIXED(2),
    2 NFILE DEC FIXED(2),
    2 NJOB DEC FIXED(2),
    2 MLINE DEC FIXED(2),
    2 MPROS DEC FIXED(2),

```

```

      2 NLEVEL DEC FIXED(1);
DCL 1 CTLPARM EXTERNAL,
      2 ISEL BIT(1),
      2 XN DEC FIXED(4),
      2 THRSO DEC FIXED(4),
      2 KCENT BIT(1),
      2 TRAC BIT(1),
      2 PLIST(4) DEC FIXED(2);
/* ROUTINE TO FORM SITE TREE AND ASSIGN PROCESSORS */
DCL JBSITE(50) DEC FIXED(2) INIT((50) 0);
DCL A(250,4) DEC FIXED(2) INIT((1000) 0);
DCL 1 CLUSTER(100),
      2 AVAILC BIT(1) INIT ((100) (1) '0'B),
      2 B(3) DEC FIXED(2) INIT((300) 0);
DCL 1 MERGE(100),
      2 NCC1 DEC FIXED(2),
      2 NCC2 DEC FIXED(2),
      2 DIST DEC FIXED(5,1);
DCL 1 TEMP1,
      2 T1 DEC FIXED(2),
      2 T2 DEC FIXED(2),
      2 T3 DEC FIXED(5,1);
DCL 1 TEMP2,
      2 T1 DEC FIXED(2),
      2 T2 DEC FIXED(2),
      2 T3 DEC FIXED(5,1);
DCL C(100,2) BIT(1) INIT((200) (1) '0'B);
DCL D(5) DEC FIXED(2) INIT((5) 0);
DCL (XNC,YNC,SNC,XJ,YJ,SJ,DS,DIT) DEC FIXED(5,1);
DCL COST DEC FIXED(6,1);
DCL (MIPS,RAM,FRAM,TRAM,DASD,TRAF) DEC FLOAT(5);
/* PRINT LEVEL ASSIGNMENT MAP FOR JOBSTEPS */
      IF TRAC THEN
          PUT SKIP(3) EDIT('JOBSTEP-ASSIGNED LEVEL MAP',
                          'LEVEL',(J DO J=1 TO NLEVEL),

```

```

('JOBSTEP',I,(LVL_ASGND(I,J)
DO J=1 TO NLEVEL) DO I=1 TO NSTEP)
(COL(4),A(26),COL(12),A(5),(NLEVEL)
(X(3),F(1)),SKIP,(NSTEP)(COL(6),A(7),
X(1),F(3),(NLEVEL)(X(3),B(1)))));
/* CLUSTER EACH LEVEL BEGINNING AT THE HIGHEST */
LVLLP: DO LVM=1 TO NLEVEL;
      LVL=NLEVEL-LVM+1;
      IP=PLIST(LVL);
      NC=0;
      IF TRAC THEN
        PUT SKIP(2) EDIT('CLUSTERING LEVEL ',LVL)
          (COL(1),A(17),F(2));
/* STAGE1 OF CLUSTERING - ONE CLUSTER FOR EACH ORIGINATING SITE */
STAGE1: DO I=1 TO NSTEP;
        IF ~LVL_ASGND(I,LVL) THEN GOTO FSTAGE1;
        ID=OWNING_JOB(I);
        IO=I-STEP#(I)+1;
        IF NC=0 THEN DO;
/* NEW ORIGINATING SITE - START NEW CLUSTER */
NCSTP:   NC=NC+1;
        AVAILC(NC)='1'B;
        B(NC,1)=ASGND_SITE(IO);
        B(NC,2)=ASGND_SITE(I);
        B(NC,3)=JBSITE(ID);
        A(I,LVL)=NC;
        GOTO FSTAGE1;
        END;
        ELSE DO;
JLOOP1: DO J=1 TO NC;
        IF ASGND_SITE(IO)=B(J,1) THEN DO;
/* PREVIOUSLY ENCOUNTERED ORIGINATING SITE */
        ISS=ASGND_SITE(I);
        B(J,2)=MAX(B(J,2),ISS);
        B(J,3)=MAX(B(J,3),JBSITE(ID));

```

```

        A(I,LVL)=J;
        GOTO FSTAGE1;
    END;
    END JLOOP1;
    END;
    GOTO NCSTR;
FSTAGE1:  END STAGE1;
/* STAGE2 OF CLUSTERING - COMBINE CLUSTERS WITH THE SAME
   PREASSIGNED SITE */
STAGE2:  IF NC=0 THEN GOTO FLVLLP;
/* ENTERING STAGE2 B AND A MAPS */
    IF TRAC THEN DO;
        PUT SKIP(3) EDIT('ENTER STAGE2 - B MAP', 'LEVEL', LVL,
            'CLUSTER', 'INITIAL', 'PREASGND', 'HIGHER-LVL', 'AVAILABLE',
            'SITE', 'SITE', 'SITE', ('CLUSTER', I, AVAILC(I),
            (B(I,J) DO J=1 TO 3) DO I=1 TO NC))
            (COL(4), A(20), COL(30), A(5), F(2),
            COL(20), 2(A(7), X(3)), A(8), X(1), A(10),
            COL(19), A(9), X(4), A(4), X(6), A(4), X(6), A(4),
            (NC) (COL(5), A(7), X(1), F(2), X(8), B(1), X(1),
            3(X(8), F(2)))));
        PUT SKIP(3) EDIT('ENTER STAGE2 - A MAP', ('LEVEL', I
            DO I=1 TO 4), 'CLUSTER', 'CLUSTER', 'CLUSTER',
            'CLUSTER', ('JOBSTEP', I, (A(I,J) DO J=1 TO 4)
            DO I=1 TO NSTEP)) (COL(4), A(20), COL(18),
            4(A(5), F(2), X(2)), COL(18), 4(A(7), X(2)),
            (NSTEP) (COL(4), A(7), X(1), F(3), X(5),
            4(F(3), X(6)))));
    END;
    MC=NC;
    MC1=NC+1;
JLOOP2:  DO J=1 TO NC;
        IF B(J,2)=0 THEN GOTO NCSTR2;
        IF MC<=NC THEN GOTO NCSTR2;
JJLOOP:  DO JJ=MC1 TO MC;

```

```

                IF B(JJ,2)=B(J,2) THEN DO;
/* PREVIOUSLY ENCOUNTERED PREASSIGNED SITE */
                B(JJ,2)=MAX(B(JJ,2),B(J,2));
                B(JJ,3)=MAX(B(JJ,3),B(J,3));
                GOTO JTOJJ;
                END;
                END JJLOOP;
/* NEW PREASSIGNED SITE - START NEW CLUSTER */
NCSTR2:        MC=MC+1;
                AVAILC(J)='0'B;
                AVAILC(MC)='1'B;
                B(MC,1)=0;
                B(MC,2)=B(J,2);
                B(MC,3)=B(J,3);
                JJ=MC;
/* MERGE CLUSTER J INTO CLUSTER JJ */
JTOJJ:        DO I=1 TO NSTEP;
                IF LVL_ASGND(I,LVL) THEN
                    IF A(I,LVL)=J THEN A(I,LVL)=JJ;
                END JTOJJ;
                END JLOOP2;
IF TRAC THEN DO;
    PUT SKIP(3) EDIT('LEAVE STAGE2 - B MAP','LEVEL',LVL,
        'CLUSTER','INITIAL','PREASGND','HIGHER-LVL','AVAILABLE',
        'SITE','SITE','SITE',('CLUSTER',I,AVAILC(I),
        (B(I,J) DO J=1 TO 3) DO I=1 TO MC)
        (COL(4),A(20),COL(30),A(5),F(2),
        COL(20),2(A(7),X(3)),A(8),X(1),A(10),
        COL(19),A(9),X(4),A(4),X(6),A(4),X(6),A(4),
        (MC) (COL(5),A(7),X(1),F(2),X(8),B(1),X(1),
        3(X(8),F(2)))));
    PUT SKIP(3) EDIT('LEAVE STAGE2 - A MAP',('LEVEL',I
        DO I=1 TO 4), 'CLUSTER', 'CLUSTER', 'CLUSTER',
        'CLUSTER', ('JOBSTEP',I, (A(I,J) DO J=1 TO 4)
        DO I=1 TO NSTEP)) (COL(4),A(20),COL(18),

```

```

4 (A (5) , F (2) , X (2) ) , COL (18) , 4 (A (7) , X (2) ) ,
(NSTEP) (COL (4) , A (7) , X (1) , F (3) , X (5) ,
4 (F (3) , X (6) ) ) ;

END;
/* STAGE3 CLUSTERING */
IF MC=NC+1 THEN GOTO SITEUD;
/* CREATE MERGE STRUCTURE FOR ALL PAS & HLS COMPATIBLE CLUSTERS */
CLRMRG: DO I=1 TO 50;
MERGE (I)=0;
END;
XJ=0; YJ=0; SJ=0;
KIC=0; YIC=0; SIC=0;
L=0; IC=0;
NEXTIC: IC=IC+1;
IF -AVAILC (IC) THEN GOTO NEXTIC;
JLOOP4: DO J= (IC+1) TO MC;
IF -AVAILC (J) THEN GOTO FJLOOP4;
IF B (J, 2) >0 & B (IC, 2) >0 THEN
IF B (J, 2) -B (IC, 2) THEN GOTO FJLOOP4;
IF B (J, 3) >0 & B (IC, 3) >0 THEN
IF B (J, 3) -B (IC, 3) THEN GOTO FJLOOP4;
ILOOP5: DO I=1 TO NSTEP;
IF -LVL_ASGND (I, LVL) THEN GOTO FILOOP5;
IO=I-STEP# (I) +1;
ICITE=ASGND_SITE (IO) ;
IF A (I, LVL) -J THEN
IF A (I, LVL) -IC THEN GOTO FILOOP5;
ELSE DO;
XIC=XIC+XDIST (ICITE) ;
YIC=YIC+YDIST (ICITE) ;
SIC=SIC+1;
GOTO FILOOP5;
END;
ELSE DO;
XJ=XJ+XDIST (ICITE) ;

```



```

                YJ=YJ+YDIST(ICITE);
                SJ=SJ+1;
            END;
FILOOP5:      END ILOOP5;
              L=L+1;
              XJ=XJ/SJ;
              YJ=YJ/SJ;
              XIC=XIC/SIC;
              YIC=YIC/SIC;
              DIST(L)=SQRT((XJ-XIC)**2+(YJ-YIC)**2);
              NCC1(L)=IC;
              NCC2(L)=J;
FJLOOP4:      END JLOOP4;
              IF IC<=MC-1 THEN GOTO NEXTIC;
/* SORT MERGE STRUCTURE IN ORDER OF INCREASING DISTANCE */
              IF L=0 THEN GOTO SITHUD;
              IF TRAC THEN
                  PUT SKIP(2) EDIT ('MERGER POSSIBILITIES','CLUSTERS',
                      'DISTANCE',('COMBINATION',I,NCC1(I),'&',NCC2(I),
                      DIST(I) DO I=1 TO L)) (COL(2),A(20),COL(18),A(8),
                      X(3),A(8),(L) (COL(2),A(11),X(1),F(2),X(3),F(2),
                      X(1),A(1),X(1),F(2),X(4),F(7,1)));
              ISRT=0;
SORTCH:      ISRT=ISRT+1;
              IF ISRT>=L THEN GOTO STAGE3;
ILOOP6:      DO I=1 TO (L-ISRT);
              TEMP1=MERGE(I);
              TEMP2=MERGE(I+1);
              IF TEMP1.T3>=TEMP2.T3 THEN DO;
                  MERGE(I)=TEMP2;
                  MERGE(I+1)=TEMP1;
              END;
              END ILOOP6;
              GOTO SORTCH;
STAGE3:      IF TRAC THEN

```

```

                PUT SKIP(2) EDIT ('MERGER POSSIBILITIES', 'CLUSTERS',
                'DISTANCE', ('COMBINATION', I, NCC1(I), '&', NCC2(I),
                DIST(I) DO I=1 TO L)) (COL(2), A(20), COL(18), A(8),
                X(3), A(8), (L) (COL(2), A(11), X(1), F(2), X(3), F(2),
                X(1), A(1), X(1), F(2), X(4), F(7, 1)));
KLCOR:   DO K=1 TO L;
/* FIND RESOURCE REQUIREMENTS FOR THIS CLUSTER PAIR */
CLRC1:   DO I=1 TO NFILE;
          C(I, 1)='0'B;
          END CLRC1;
CLRC2:   DO I=1 TO NPCN;
          C(I, 2)='0'B;
          END CLRC2;
MIPS=0; TRAF=0; RAM=0; FRAM=0; TRAM=0; CHAN=0; DASD=0;
ILOOP1:  DO I=1 TO NSTEP;
          IF ~LVL_ASGND(I, LVL) THEN GOTO FILOOP1;
          IF A(I, LVL) ~NCC1(K) & A(I, LVL) ~NCC2(K) THEN GOTO FILOOP1;
          IO=I-STEP#(I)+1;
          ICITE=ASGND_SITE(IO);
          ID=OWNING_JOB(I);
          IFCN=REQFCN(I);
          IFILE=REQFILE(I);
FNDMIP:  MIPS=MIPS+10**(-6*ARVL_RATE(ID)*I_STEPS(IFCN));
          IF I_RATE(IP)=0 THEN TRAM=0;
          ELSE
            TRAM=TRAM+ARVL_RATE(ID)*FCNTSIZE(IFCN)*
              (10**(-6*I_STEPS(IFCN)/I_RATE(IP)));
          IF I<=1 THEN GOTO CONT1;
          IDD=OWNING_JOB(I-1);
          IF ID~IDD THEN GOTO CONT1;
          IF LVL_ASGND(I-1, LVL) THEN GOTO CONT1;
          TRAF=TRAF+ARVL_RATE(ID)*MSG_LENGTH(I-1);
CONT1:   IF I=NSTEP THEN GOTO CONT2;
          IDD=OWNING_JOB(I+1);
          IF ID~IDD THEN GOTO CONT2;

```

```

IF LVL_ASGND(I+1,LVL) THEN GOTO CONT2;
TRAF=TRAF+ARVL_RATE(ID)*MSG_LENGTH(I+1);
CONT2: IF IFILE≠0 THEN
        IF (¬C(IFILE,1)) & (MEDIUM(IFILE)=2) THEN DO;
            DASD=DASD+FILE_SIZE(IFILE);
            C(IFILE,1)='1'B;
            END;
        IF ¬C(IFCN,2) THEN DO;
            FRAM=FRAM+FCNFSIZE(IFCN);
            C(IFCN,2)='1'B;
            END;
FILOOP1: END ILOOP1;
RAM=FRAM+TRAM;
/* PROCESSOR LOAD CHECK - CAN MERGED CLUSTERS BE SUPPORTED? */
IF TRAC THEN
    PUT SKIP(2) EDIT('COMBINING CLUSTERS',NCC1(K),'AND',NCC2(K),
        'LEVEL',LVL,'PROS= ',IP,'MIPS= ',MIPS,'AVRAM= ',
            RAM,'FDASD= ',DASD,'TRAF= ',TRAF)
        (COL(1),A(18),F(4),X(1),A(3),F(4),X(3),
            A(5),F(2),COL(2),A(6),F(3),X(2),A(6),
            E(11,4),X(2),A(7),E(11,4),COL(13),A(7),
            E(11,4),X(2),A(6),E(11,4));
    IF MIPS>I_RATE(IP) |
        RAM>PRIMSTR(IP) |
        DASD>SCNDSTR(IP) |
        TRAF>CHAN_RATE(IP) THEN DO;
        IF TRAC THEN
            PUT SKIP(2) LIST('COMBINATION CAUSES OVERLOAD');
            GOTO FKLOOP;
        END;
/* LOAD OK - MERGE CLUSTER IC INTO CLUSTER J */
IC=NCC1(K);
J=NCC2(K);
AVAILC(IC)='0'B;
B(J,2)=MAX(B(IC,2),B(J,2));

```

```

      B(J,3)=MAX(B(IC,3),B(J,3));
/* ASSIGN JOBSTEPS IN CLUSTER IC TO CLUSTER J */
ILOOP2: DO I=1 TO NSTEP;
      IF ~LVL_ASGND(I,LVL) THEN GOTO FILOOP2;
      IF A(I,LVL)~=IC THEN GOTO FILOOP2;
      A(I,LVL)=J;
FILOOP2: END ILOOP2;
      GOTO CLRMRG;
FKLOOP: END KLOOP;
SITEUD: DO II=1 TO MC;
      IF ~AVAILC(II) THEN GOTO FSITEUD;
      IF B(II,2)=0 THEN DO;
/* START A NEW SITE FOR EACH CLUSTER WHICH DOES NOT HAVE
   A PREASSIGNED SITE */
          NSITE=NSITE+1;
          XDIST(NSITE)=0;
          YDIST(NSITE)=0;
          PROS_LOAD(NSITE)=0;
          PROS_UP(NSITE)=0;
          ASGND_PROS(NSITE)=PLIST(LVL);
          JJ=NSITE;
          END;
        ELSE DO;
          JJ=B(II,2);
          IF ASGND_PROS(JJ)<=0 THEN
            ASGND_PROS(JJ)=PLIST(LVL);
          END;
        IF B(II,3)<=0 THEN GOTO ILOOP4;
/* IF CLUSTER HAS A HIGHER LEVEL SITE,CREATE A LINK TO IT */
        NLINE=NLINE+1;
        INSITE(NLINE)=JJ;
        FINSITE(NLINE)=B(II,3);
ILOOP4: DO I=1 TO NSTEP;
      IF LVL_ASGND(I,LVL) THEN
        IF A(I,LVL)=II THEN DO;

```

```

/* ASSIGN EACH JOBSTEP IN THIS CLUSTER TO SITE JJ */
      ASGND_SITE(I) =JJ;
/* HIGHER LEVEL SITE FOR THIS JOB IS SET TO JJ */
      ID=OWNING_JOB(I);
      JBSITE(ID)=JJ;
      END;
END ILOOP4;
PSITEUD:  END SITEUD;
          IF TRAC THEN DO;
          PUT SKIP(3) EDIT('LEAVE STAGE3 - B MAP', 'LEVEL', LVL,
            'CLUSTER', 'INITIAL', 'PREASGND', 'HIGHER-LVL', 'AVAILABLE',
            'SITE', 'SITE', 'SITE', ('CLUSTER', I, AVAILC(I),
            (B(I,J) DO J=1 TO 3) DO I=1 TO MC))
            (COL(4), A(20), COL(30), A(5), F(2),
            COL(20), 2(A(7), X(3)), A(8), X(1), A(10),
            COL(19), A(9), X(4), A(4), X(6), A(4), X(6), A(4),
            (MC) (COL(5), A(7), X(1), F(2), X(8), B(1), X(1),
            3(X(8), F(2)))));
          PUT SKIP(3) EDIT('LEAVE STAGE3 - A MAP', ('LEVEL', I
            DO I=1 TO 4), 'CLUSTER', 'CLUSTER', 'CLUSTER',
            'CLUSTER', ('JOBSTEP', I, (A(I,J) DO J=1 TO 4)
            DO I=1 TO NSTEP)) (COL(4), A(20), COL(18),
            4(A(5), F(2), X(2)), COL(18), 4(A(7), X(2)),
            (NSTEP) (COL(4), A(7), X(1), F(3), X(5),
            4(F(3), X(6)))));
          END;
FLVLLP:  END LVLLP;
/* DETERMINE LOADING ON ALL LINKS */
ILOOP7:  DO I=1 TO NLINE;
          L_LOAD(I)=0;
          END ILOOP7;
JLOOP3:  DO J=1 TO NSTEP;
          IF ASGND_SITE(J)=0 THEN GOTO FJLOOP3;
          I=ASGND_SITE(J);
          II=OWNING_JOB(J);

```

```

        IF OWNING_JOB(J+1) /= OWNING_JOB(J) THEN GOTO FJLOOP3;
        IF ASGND_SITE(J+1) = 0 THEN GOTO FJLOOP3;
        IOUT=ASGND_SITE(J+1);
        IF I=IOUT THEN GOTO FJLOOP3;
        CALL PATH(I,K);
        IF K<=0 THEN GOTO FJLOOP3;
IKLOOP:   DO IK=1 TO K;
          LI=D(IK);
          L_LOAD(LL)=L_LOAD(LL) +
            ARVL_RATE(LI)*MSG_LENGTH(J);
        ENE IKLOOP;
FJLOOP3: END JLOOP3;
/* ASSIGN A LINE TYPE TO EACH LINK */
ILLOOP:  DO IL=1 TO NLINE;
          LLD=L_LOAD(IL);
LTPLP:   DO LTP=1 TO MLINE;
          IF LLD>=0.50*XMSN_RATE(LTP) THEN GOTO FLTPLP;
          ELSE DO;
            L_TYPE(IL) =LTP;
            GOTO FILLOOP;
          END;
FLTPLP:  END LTPLP;
FILLOOP: END ILLOOP;
/* SUBROUTINE PATH FINDS LINKS USED IN EACH MESSAGE PATH */
PATH: PROC(IN,K);
        DCL E(100,5) DEC FIXED(2) INIT((500) 0);
        DCL EE(100,5) DEC FIXED(2) INIT((500) 0);
/* PATH ANALYSIS */
        IF TRAC THEN
          PUT SKIP(2) EDIT('FIND A PATH FROM ',IN,' TO ',IOUT)
            (COL(1),A(17),F(2),A(4),F(2));
          IF NLINE=0 THEN GOTO OUT;
          K1=1;
          K2=1;
          E(K1,K2)=IN;

```

```

AGAIN:      K11=0;
            K22=K2+1;
ILOOP8:    DO I=1 TO K1;
            J1=E(I,K2);
KLOOP1:    DO K=1 TO NLINE;
            IF INSITE(K)=J1 THEN J2=FINSITE(K);
            ELSE
            IF FINSITE(K)=J1 THEN J2=INSITE(K);
            ELSE GOTO FKLOOP1;
            IF K11=0 THEN GOTO NODUP;
            DO K111=1 TO K11;
            IF J2=EE(K111,K22) THEN GOTO FKLOOP1;
            END;
NODUP:     K11=K11+1;
            DO L=1 TO K2;
            EE(K11,L)=E(I,L);
            END;
            EE(K11,K2)=K;
            EE(K11,K22)=J2;
            IF TPAC THEN
            PUT SKIP EDIT('PATH# ',K11,'LINES# ',(EE(K11,M) DO M=1
            TO K22-1),'TO NODE ',EE(K11,K22))
            (COL(1),A(6),F(3),X(4),A(6),(K22-1)(X(2),F(2)),
            X(2),A(8),F(3));
FKLOOP1:  END KLOOP1;
FILOOP8:  END ILOOP8;
            IF K11=0 THEN GOTO OUT;
            DO I=1 TO K11;
            DO M=1 TO K22;
            E(I,M)=EE(I,M);
            END;
            IF E(I,K22)=IOUT THEN DO;
            ND=I;
            GOTO SUCES;
            END;

```

```

      END;
      GOTO UPDATE;
SUCES:  K=K22-1;
        DO M=1 TO K;
        D(M)=E(ND,M);
        END;
        IF TRAC THEN
        PUT SKIP EDIT('SUCCESS,LINES= ',(D(M) DO M=1 TO K),
                      'TO NODE ',E(ND,K22))
                      (COL(1),A(14),(K)(X(2),F(2)),
                      X(2),A(8),F(3));

      RETURN;
UPDATE: K1=K11;
        K2=K22;
        IF K2<5 THEN GOTO AGAIN;
OUT:    K=0;
        IF TRAC THEN
        PUT SKIP LIST('PATH FAILURE');
        RETURN;
END PATH;
PUT SKIP (3) LIST('END OF PROCESSOR ASSIGNMENT');
END PTCS;

```



XII. APPENDIX D:  
LISTING FOR THE SUBROUTINE OSASS

```

OSASS:  PROC;
DCL 1 NETWORK EXTERNAL,
  2 SITE(25),
    3 STATUS DEC FIXED(1),
    3 XDIST DEC FIXED(5,1),
    3 YDIST DEC FIXED(5,1),
    3 ASGND_PROS DEC FIXED(2),
    3 PROS_LOAD DEC FLOAT(5),
    3 PROS_UF DEC FLOAT(5),
  2 LINE(99),
    3 INSITE DEC FIXED(2),
    3 FINSITE DEC FIXED(2),
    3 L_TYPE DEC FIXED(2),
    3 L_LOAD DEC FLOAT(5),
    3 L_UF DEC FLOAT(5);
DCL 1 HARDWARE EXTERNAL,
  2 PTYPE(10),
    3 PROS_LVL# DEC FIXED(1),
    3 I_RATE DEC FLOAT(5),
    3 PRIMSTR DEC FLOAT(5),
    3 SCNDSTR DEC FLOAT(5),
    3 MDSPET DEC FIXED(2),
    3 CHAN_RATE DEC FLOAT(5),
    3 PCOST DEC FIXED(8,2),
  2 LTYPE(10),
    3 KMSN_RATE DEC FLOAT(5),
    3 LFCOST DEC FIXED(5,1),
    3 LVCST1 DEC FIXED(4,2),
    3 LVCST2 DEC FIXED(4,2),
    3 LVCST3 DEC FIXED(4,2),
    3 LVCST4 DEC FIXED(4,2),
    3 LVCST5 DEC FIXED(4,2);
DCL 1 SOFTWARE EXTERNAL,
  2 FCNTYPE(99),
    3 I_STEPS DEC FLOAT(5),

```

```

        3 FCNTSIZE DEC FLOAT(5),
        3 FCNFSIZE DEC FLOAT(5),
    2 FILETYPE(99),
        3 MEDIUM DEC FIXED(2),
        3 FILE_SIZE DEC FLOAT(5);
DCL 1 JOB(50) EXTERNAL,
    2 NO_JOBSTEPS DEC FIXED(2),
    2 ARVL_RATE DEC FLOAT(5),
    2 RSPNS_TIME DEC FLOAT(5),
    2 FRSTEP# DEC FIXED(3),
    2 LASTEP# DEC FIXED(3);
DCL 1 JOBSTEP(250) EXTERNAL,
    2 REQFCN DEC FIXED(3),
    2 REQFILE DEC FIXED(3),
    2 MSG_LENGTH DEC FLOAT(5),
    2 LVL_SPEC DEC FIXED(1),
    2 REQ_LVL DEC FIXED(1),
    2 OWNING_JOB DEC FIXED(2),
    2 STEP# DEC FIXED(2),
    2 ASGND_SITE DEC FIXED(2),
    2 PROS_TIME DEC FLOAT(5),
    2 LINE_DLY DEC FLOAT(5),
    2 LVL_ASGND(4) BIT(1);
DCL 1 PROCOMP(10) EXTERNAL,
    2 CFILE(99) BIT(1),
    2 CFCN(99) BIT(1);
DCL 1 UPBOUND EXTERNAL,
    2 NSITE DEC FIXED(2),
    2 NLINE DEC FIXED(2),
    2 NSTEP DEC FIXED(3),
    2 NPCN DEC FIXED(2),
    2 NFILE DEC FIXED(2),
    2 NJOB DEC FIXED(2),
    2 MLINE DEC FIXED(2),
    2 MPROS DEC FIXED(2),

```

```

      2 NLEVEL DEC FIXED(1);
DCL 1 CTLPARM EXTERNAL,
      2 ISEL BIT(1),
      2 XN DEC FIXED(4),
      2 THRSDEC DEC FIXED(4),
      2 KCENT BIT(1),
      2 TRAC BIT(1),
      2 PLIST(4) DEC FIXED(2);
/* ROUTINE TO OPTIMALLY LOCATE FLOATING SITES */
      DCL IGO BIT(1);
      DCL (COST,CC,CM) DEC FIXED (8,2);
      DCL (X,Y,XX,YY,XMIN,XMAX,YMIN,YMAX) DEC FIXED (5,1);
      DCL CIM DEC FIXED (5);
      DCL PRELIM BIT (1);
      DCL FF BIT(1);
          MAXL=0;
          PRELIM='1'B;
RELOC:  CIM=0;
ISLOOP: DO IS=1 TO NSITE;
          IF STATUS(IS)=2 THEN GOTO FISLOOP;
          IGO='1'B;X=0;Y=0;
          CALL FNDLC(COST,X,Y,FF);
/* PRINT CURRENT CONNECTION COST */
          IF TRAC THEN
              PUT SKIP(2) EDIT('OSASS CURRENT CONNECTION COST',
                  'SITE= ',IS,'COST= ',COST)
                  (COL(1),A(29),X(4),A(6),F(3),
                  X(4),A(6),F(10,2));
          IF COST<=0 THEN GOTO FISLOOP;
          X=XDIST(IS);
          Y=YDIST(IS);
          CC=COST;
          IF ISEL THEN GOTO FINSET;
/* ITERATIVE TECHNIQUE FOR FINDING OPTIMUM LOCATION */
          XMIN=9999.9;

```

```

YMIN=9999.9;
XMAX=0;
YMAX=0;
JLOOP1: DO J=1 TO NLINE;
        IF INSITE(J)=IS THEN I=FINSITE(J);
            ELSE
                IF FINSITE(J)=IS THEN I=INSITE(J);
                    ELSE GOTO FJLOOP1;
                IF XMIN>XDIST(I) THEN XMIN=XDIST(I);
                IF XMAX<XDIST(I) THEN XMAX=XDIST(I);
                IF YMIN>YDIST(I) THEN YMIN=YDIST(I);
                IF YMAX<YDIST(I) THEN YMAX=YDIST(I);
FJLOOP1: END JLOOP1;
ILOOP1: DO I=1 TO (XN+1);
        XX=(( (XN-I+1)*XMIN)+((I-1)*XMAX))/XN;
JLOOP2: DO J=1 TO (XN+1);
        YY=(( (XN-J+1)*YMIN)+((J-1)*YMAX))/XN;
        IGO='0'B;
        CALL FNDLC(CM,XX,YY,PF);
        IF CM>=CC THEN GOTO FJLOOP2;
        CC=CM;
        X=XX;
        Y=YY;
/* PRINT ITERATE NEW CONNECTION COST */
        IF TRAC THEN
            PUT SKIP EDIT('OSASS ITERATE CONNECTION COST',
                'NEW SITE(',X,Y,') ','COST= ',CC)
                (COL(1),A(29),X(4),A(9),F(7,1),X(2),
                F(7,1),A(1),X(4),A(6),F(10,2));
FJLOOP2: END JLOOP2;
        END ILOOP1;
        GOTO MOVEIS;
/* FIND OPTIMUM SITE AMONG THE GIVEN FIXED SITES */
FINSET: DO I=1 TO NSITE;
        IF STATUS(I)≠2 THEN GOTO FFINSET;

```

```

      IGO='0'B;
      CALL FNDLC (CM,XDIST (I),YDIST (I),FF);
      IF CM>=CC THEN GOTO FFINSET;
      CC=CM;
      X=XDIST (I);
      Y=YDIST (I);
/* PRINT FINSET NEW CONNECTION COST */
      IF TRAC THEN
        PUT SKIP EDIT('OSASS FINSET CONNECTION COST',
                     'SITE= ',I,'COST= ',CC)
                     (COL (1),A (28),X (4),A (6),F (3),
                      X (4),A (6),F (10,2));

FFINSET:  END FINSET;
MOVEIS:  IF ~FF THEN DO;
          XDIST (IS)=X;
          YDIST (IS)=Y;
          STATUS (IS)=1;
          IF CIM<COST-CC THEN CIM=COST-CC;
          END;

FISLOOP:END ISLOOP;
/* CHECK TO SEE IF ALL SITES HAVE BEEN PRELIMINARILY LOCATED */
      IF PRELIM THEN DO;
        PRELIM='0'B;
ILCOP3:  DO I=1 TO NSITE;
          IF STATUS (I)=0 THEN PRELIM='1'B;
        END ILOOP3;
        END;
      IF PRELIM THEN GOTO RELOC;
      MAXL=MAXL+1;
      IF MAXL<=100 THEN
        IF CIM>THRS D THEN GOTO RELOC;
ILOOP2:  DO I=1 TO NLINE;
          IS=INSITE (I);
          IK=FINSITE (I);
          DD=SQRT ( (XDIST (IS) -XDIST (IK)) **2

```

```

                                + (YDIST(IS) - YDIST(IK)) ** 2);
                                IF DD <= 0.1 THEN L_TYPE(I) = 0;
FLOOP2:      END ILOOP2;
              PUT SKIP (3) LIST('END OF OPTIMUM LOCATING');
              RETURN;
/* SUBROUTINE TO FIND CONNECTION COST OF THIS SITE */
FNDLC: PROC (SCOST, SX, SY, FF);
          DCL (SCOST, UNIT) DEC FIXED (8, 2);
          DCL (SX, SY, DD) DEC FIXED (5, 1);
          DCL FF BIT (1);
          SCOST = 0;
          FF = '1'B;
          IF NLINE <= 0 THEN RETURN;
KLOOP1:   DO K = 1 TO NLINE;
          IF INSITE(K) = IS THEN IK = FINSITE(K);
          ELSE
            IF FINSITE(K) = IS THEN IK = INSITE(K);
            ELSE GOTO FKLOOP1;
/* UNASSIGNED SITES ARE NOT INCLUDED */
          IF STATUS(IK) ^= 0 THEN FF = '0'B;
          ELSE GOTO FKLOOP1;
          IF IGO THEN
            DD = SQRT((XDIST(IS) - XDIST(IK)) ** 2
                    + (YDIST(IS) - YDIST(IK)) ** 2);
            ELSE
              DD = SQRT((XDIST(IK) - SX) ** 2 + (YDIST(IK) - SY) ** 2);
/* CHECK IF LINE CAN BE HARDWIRED */
          IF DD <= 0.1 THEN GOTO FKLOOP1;
          LINTYP = L_TYPE(K);
/* CHECK IF LOCAL LINE CAN BE USED */
          IF (LINTYP = 1 | LINTYP = 2) & (DD <= 7.5) THEN DO;
            UNIT = 0;
            GOTO DCOST;
            END;
          IF DD <= 25 THEN UNIT = LVCST1(LINTYP);

```

```
        ELSE IF DD<=100 THEN UNIT=LVCST2(LINTYP);
        ELSE IF DD<=250 THEN UNIT=LVCST3(LINTYP);
        ELSE IF DD<=500 THEN UNIT=LVCST4(LINTYP);
        ELSE UNIT=LVCST5(LINTYP);
DCOST:   SCOST=SCOST+0.5*(LFCOST(LINTYP)+DD*UNIT);
FKLOOP1: END KLOOP1;
        RETURN;
        END FNDLC;
        END OSASS;
```



XIII. APPENDIX E:  
LISTING FOR THE SUBROUTINE ANALYZ

```

ANLYZ: PROC;
DCL 1 NETWORK EXTERNAL,
  2 SITE(25),
    3 STATUS DEC FIXED(1),
    3 XDIST DEC FIXED(5,1),
    3 YDIST DEC FIXED(5,1),
    3 ASGND_PROS DEC FIXED(2),
    3 PROS_LOAD DEC FLOAT(5),
    3 PROS_UP DEC FLOAT(5),
  2 LINE(99),
    3 INSITE DEC FIXED(2),
    3 FINSITE DEC FIXED(2),
    3 L_TYPE DEC FIXED(2),
    3 L_LOAD DEC FLOAT(5),
    3 L_UP DEC FLOAT(5);
DCL 1 HARDWARE EXTERNAL,
  2 PTYPE(10),
    3 PROS_LVL# DEC FIXED(1),
    3 I_RATE DEC FLOAT(5),
    3 PRIMSTR DEC FLOAT(5),
    3 SCNDSTR DEC FLOAT(5),
    3 MDSPBT DEC FIXED(2),
    3 CHAN_RATE DEC FLOAT(5),
    3 PCOST DEC FIXED(8,2),
  2 LTYPE(10),
    3 XMSN_RATE DEC FLOAT(5),
    3 LFCOST DEC FIXED(5,1),
    3 LVCST1 DEC FIXED(4,2),
    3 LVCST2 DEC FIXED(4,2),
    3 LVCST3 DEC FIXED(4,2),
    3 LVCST4 DEC FIXED(4,2),
    3 LVCST5 DEC FIXED(4,2);
DCL 1 SOFTWARE EXTERNAL,
  2 FCNTYPE(99),
    3 I_STEPS DEC FLOAT(5),

```

```

        3 FCNTSIZE DEC FLOAT(5) ,
        3 FCNFSIZE DEC FLOAT(5) ,
    2 FILETYPE(99) ,
        3 MEDIUM DEC FIXED(2) ,
        3 FILE_SIZE DEC FLOAT(5) ;
DCL 1 JOB(50) EXTERNAL,
    2 NO_JOBSTEPS DEC FIXED(2) ,
    2 ARVL_RATE DEC FLOAT(5) ,
    2 RSPNS_TIME DEC FLOAT(5) ,
    2 FRSTEP# DEC FIXED(3) ,
    2 LASTEP# DEC FIXED(3) ;
DCL 1 JOBSTEP(250) EXTERNAL,
    2 REQFCN DEC FIXED(3) ,
    2 REQFILE DEC FIXED(3) ,
    2 MSG_LENGTH DEC FLOAT(5) ,
    2 LVL_SPEC DEC FIXED(1) ,
    2 REQ_LVL DEC FIXED(1) ,
    2 OWNING_JOB DEC FIXED(2) ,
    2 STEP# DEC FIXED(2) ,
    2 ASGND_SITE DEC FIXED(2) ,
    2 PROS_TIME DEC FLOAT(5) ,
    2 LINE_DLY DEC FLOAT(5) ,
    2 LVL_ASGND(4) BIT(1) ;
DCL 1 PROCOMP(10) EXTERNAL,
    2 CFILE(99) BIT(1) ,
    2 CFCW(99) BIT(1) ;
DCL 1 UPBOUND EXTERNAL,
    2 NSITE DEC FIXED(2) ,
    2 NLINE DEC FIXED(2) ,
    2 NSTEP DEC FIXED(3) ,
    2 NFCW DEC FIXED(2) ,
    2 NFILE DEC FIXED(2) ,
    2 NJOB DEC FIXED(2) ,
    2 MLINE DEC FIXED(2) ,
    2 MPROS DEC FIXED(2) ,

```

```

      2 NLEVEL DEC FIXED(1);
DCL 1 CTLPARM EXTERNAL,
      2 ISEL BIT(1),
      2 XN DEC FIXED(4),
      2 THRSO DEC FIXED(4),
      2 KCENT BIT(1),
      2 TRAC BIT(1),
      2 PLIST(4) DEC FIXED(2);
/* PROCEDURE TO ANALYZE PRESENT NETWORK TO DETERMINE
   RESPONSE TIMES AND COSTS */
      DCL B(5) DEC FIXED(2) INIT((5) 0);
      DCL (CL,CIL,CP,CC,UNIT) DEC FIXED(10,2);
      DCL DD DEC FIXED(5,1);
      DCL RAM_LOAD(25) DEC FLOAT(5) INIT((25) 0);
      DCL RAM_UP(25) DEC FLOAT(5) INIT((25) 0);
      DCL STR2_LOAD(25) DEC FLOAT(5) INIT((25) 0);
      DCL STR2_UP(25) DEC FLOAT(5) INIT((25) 0);
      DCL DEGRAD DEC FLOAT(5);
      PUT SKIP(2) LIST('START NETWORK ANALYSIS');
ILOOP1: DO I=1 TO NSTEP;
          PROS_TIME(I)=0;
          LINE_DLY(I)=0;
        END ILOOP1;
ILOOP2: DO I=1 TO NSITE;
          PROS_LOAD(I)=0;
          PROS_UP(I)=0;
        END ILOOP2;
ILOOP3: DO I=1 TO NLINE;
          L_UP(I)=0;
        END ILOOP3;
/* FIND PROCESSOR LOAD AND DYNAMIC RAM LOAD FOR EACH PROCESSOR */
JLOOP1: DO J=1 TO NSTEP;
          IF ASGND_SITE(J)=0 THEN GOTO FJLOOP1;
          I=ASGND_SITE(J);
          IF ASGND_PROS(I)=0 THEN GOTO FJLOOP1;

```

```

        IP=ASGND_PROS(I);
        II=OWNING_JOB(J);
        JJ=REQFCN(J);
        PROS_LOAD(I)=PROS_LOAD(I)+ARVL_RATE(II)*I_STEPS(JJ)*10**-6;
        IF I_RATE(IP)=0 THEN RAM_LOAD(I)=0;
        ELSE
            RAM_LOAD(I)=RAM_LOAD(I)+ARVL_RATE(II)*FCNTSIZE(JJ)*
                (10**-6*I_STEPS(JJ)/I_RATE(IP));
FJLOOP1: END JLOOP1;
        PUT SKIP(3);
/* INCREASE RAM LOAD BY RESIDENT RAM FOR EACH REQUIRED FUNCTION */
ILOOP4: DO I=1 TO NSITE;
        IF ASGND_PROS(I)=0 THEN GOTO FILOOP4;
        IP=ASGND_PROS(I);
IFNLOOP: DO IFN=1 TO NFCN;
IJSLOOP1: DO IJS=1 TO NSTEP;
        IF REQFCN(IJS)=IFN THEN
            IF ASGND_SITE(IJS)=I THEN DO;
                RAM_LOAD(I)=RAM_LOAD(I)+FCNFSIZE(IFN);
                GOTO FIFNLOOP;
            END;
        END IJSLOOP1;
FIFNLOOP: END IFNLOOP;
/* FIND FIXED SECONDARY STORAGE REQUIREMENTS */
IFLLOOP: DO IFL=1 TO NFILE;
IJSLOOP2: DO IJS=1 TO NSTEP;
        IF REQFILE(IJS)=IFL THEN
            IF MEDIUM(IFL)=2 THEN
                IF ASGND_SITE(IJS)=I THEN DO;
                    STR2_LOAD(I)=STR2_LOAD(I)+FILE_SIZE(IFL);
                    GOTO FIFLLOOP;
                END;
            END IJSLOOP2;
FIFLLOOP: END IFLLOOP;
/* FIND RAM,STR2,& PROS UTILIZATION FACTORS */

```

```

IF PRIMSTR(IP) =0 THEN RAM_UF(I) =0;
ELSE
  RAM_UF(I) =RAM_LOAD(I) /PRIMSTR(IP);
IF SCNDSTR(IP) =0 THEN STR2_UF(I) =0;
ELSE
  STR2_UF(I) =STR2_LOAD(I) /SCNDSTR(IP);
IF I_RATE(IP) =0 THEN PROS_UF(I) =0;
ELSE
  PROS_UF(I) =PROS_LOAD(I) /I_RATE(IP);
IF (PROS_UF(I) <1) & (RAM_UF(I) <1) & (STR2_UF(I) <1) THEN
  PUT SKIP(2) EDIT ('PROCESSOR ',IP,' SITE ',I,
  ' MIPS= ',PROS_LOAD(I),' UTILIZED=',PROS_UF(I),
  ' AVRAM= ',RAM_LOAD(I),' UTILIZED=',RAM_UF(I),
  ' FIXED SCNDSTR= ',STR2_LOAD(I),' UTILIZED=',
  STR2_UF(I)) (COL(1),2(A,F(2)),3(COL(3),2(A,E(11,4)))));
ELSE
  PUT SKIP(2) EDIT ('OVERLOAD.PROCESSOR ',IP,' SITE ',I,
  ' MIPS= ',PROS_LOAD(I),' UTILIZED=',PROS_UF(I),
  ' AVRAM= ',RAM_LOAD(I),' UTILIZED=',RAM_UF(I),
  ' FIXED SCNDSTR= ',STR2_LOAD(I),' UTILIZED=',
  STR2_UF(I)) (COL(1),2(A,F(2)),3(COL(3),2(A,E(11,4)))));
FILOOP4: END ILOOP4;
PUT SKIP(3);
/* FIND LINE UTILIZATION FACTORS */
ILOOP5: DO I=1 TO NLINE;
  IF L_TYPE(I) =0 THEN DO;
    L_UF(I) =0;
    GOTO LOU;T;
  END;
  LINTYP=L_TYPE(I);
  L_UF(I) =L_LOAD(I) /XMSN_RATE(LINTYP);
LOU:T: IF L_UF(I) <1 THEN
  PUT SKIP EDIT ('INSITE ',INSITE(I),
  ', FINSITE ',FINSITE(I),', ',LINE TYPE IS ',L_TYPE(I),
  ', LOAD= ',L_LOAD(I),', ',UTILIZE= ',

```

```

L_UP(I) (3 (A,F(2)), 2 (A,E(11,4)));
ELSE
PUT SKIP EDIT('OVERLOAD.INSITE ', INSITE(I),
', FINSITE ', FINSITE(I), ', LINE TYPE IS ', L_TYPE(I),
', LOAD= ', L_LOAD(I), ', UTILIZE= ',
L_UP(I) (3 (A,F(2)), 2 (A,E(11,4)));
END ILOOP5;
/* CALCULATE JOBSTEP PROCESSING TIME AND LINE DELAY.
ACCUMULATE JOB RESPONSE TIME */
ID=0;
JLOOP2: DO J=1 TO NSTEP;
IF ASGND_SITE(J)=0 THEN GOTO FJLOOP2;
I=ASGND_SITE(J);
IF ASGND_PROS(I)=0 THEN GOTO FJLOOP2;
IP=ASGND_PROS(I);
IF PROS_UP(I)>=1 THEN GOTO FJLOOP2;
JJ=REQFCN(J);
II=OWNING_JOB(J);
DEGRAD=MAX(PROS_UP(I), RAM_UP(I));
IF I_RATE(IP)=0 THEN PROS_TIME(J)=0;
ELSE
PROS_TIME(J)=10**(-6*I_STEPS(JJ)/(I_RATE(IP)*(1-DEGRAD)));
IF II=ID THEN
RSPNS_TIME(II)=RSPNS_TIME(II)+PROS_TIME(J);
ELSE DO;
RSPNS_TIME(II)=PROS_TIME(J);
ID=II;
PUT SKIP(3) EDIT('JOB= ', ID) (A,F(3));
END;
IF OWNING_JOB(J+1) /= OWNING_JOB(J) THEN GOTO DONE;
IF ASGND_SITE(J+1)=0 THEN GOTO FJLOOP2;
IOUT=ASGND_SITE(J+1);
IF I=IOUT THEN GOTO OUTPUT;
CALL PATH(I,K);
IF K=0 THEN GOTO FJLOOP2;

```

```

IKLOOP2:   DO IK=1 TO K;
           LL=B(IK);
           LINTYP=L_TYPE(LL);
           IF LINTYP=0 THEN GOTO FIKLOOP2;
           LINE_DLY(J)=LINE_DLY(J)+
             MSG_LENGTH(J)/(XMSN_RATE(LINTYP)*(1-L_UF(LL)));
FIKLOOP2:  END IKLOOP2;
DONE:     RSPNS_TIME(II)=RSPNS_TIME(II)+LINE_DLY(J);
OUTPUT:   PUT SKIP EDIT('STEP= ',STEP#(J),
                       ' ,SITE= ',ASGND_SITE(J),
                       ' ,PROS TIME= ',PROS_TIME(J),
                       ' ,LINE DELAY= ',LINE_DLY(J),
                       ' ,JOB TIME ACCUM= ',RSPNS_TIME(II))
          (2(A,P(2)),SKIP,3(A,E(11,4)));
FJLOOP2:  END JLOOP2;
/* FIND TOTAL LINE COST AND PROCESSOR COST */
CL=0;
CP=0;
PUT SKIP (3);
ILCOP6:  DO I=1 TO NLINE;
         II=L_TYPE(I);
         IF II=0 THEN GOTO FILOOP6;
         L1=INSITE(I);
         L2=FINSITE(I);
         DD=SQRT((XDIST(L1)-XDIST(L2))**2+(YDIST(L1)-YDIST(L2))**2);
         IF (II=1|II=2) & (DD<=7.5) THEN DO;
             UNIT=0;
             GOTO DCOST;
         END;
         IF DD<=25 THEN UNIT=LVCST1(II);
         ELSE IF DD<=100 THEN UNIT=LVCST2(II);
         ELSE IF DD<=250 THEN UNIT=LVCST3(II);
         ELSE IF DD<=500 THEN UNIT=LVCST4(II);
         ELSE UNIT=LVCST5(II);
DCOST:   CLL=LFCOST(II)+UNIT*DD;

```



```

        PUT SKIP EDIT('LINE ',I,' COST',CLL)
                (A,F(2),A,F(11,2));
        CL=CL+CLL;
FILOOP6: END ILOOP6;
        PUT SKIP (3);
ILOOP7: DO I=1 TO NSITE;
        IF ASGND_PROS(I)=0 THEN GOTO FILOOP7;
        II=ASGND_PROS(I);
        PUT SKIP EDIT('SITE ',I,' PROCESSOR COST ',PCOST(II))
                (A,F(2),A,F(11,2));
        CP=CP+PCOST(II);
FILOOP7: END ILOOP7;
        CC=CL+CP;
        PUT SKIP(3) EDIT('TOTAL LINE COST= ',CL,
                ' ,TOTAL PROS COST= ',CP,
                ' ,TOTAL COST= ',CC) (3(A,F(11,2)));
PATH: PROC(IN,K);
        DCL A(100,5) DEC FIXED(2) INIT((500) 0);
        DCL AA(100,5) DEC FIXED(2) INIT((500) 0);
/* PATH ANALYSIS */
        IF TRAC THEN
                PUT SKIP(2) EDIT('FIND A PATH FROM ',IN,' TO ',IOUT)
                        (COL(1),A(17),F(2),A(4),F(2));
                IF NLINE=0 THEN GOTO OUT;
                K1=1;
                K2=1;
                A(K1,K2)=IN;
AGAIN:   K11=0;
                K22=K2+1;
ILOOP8: DO I=1 TO K1;
                J1=A(I,K2);
KLOOP1: DO K=1 TO NLINE;
                IF INSITE(K)=J1 THEN J2=FINSITE(K);
                        ELSE
                                IF FINSITE(K)=J1 THEN J2=INSITE(K);

```

```

ELSE GOTO FKLOOF1;
IF K11=0 THEN GOTO NODUP;
DO K111=1 TO K11;
  IF J2=AA(K111,K22) THEN GOTO FKLOOP1;
  END;
NODUP: K11=K11+1;
DO L=1 TO K2;
  AA(K11,L)=A(I,L);
  END;
  AA(K11,K2)=K;
  AA(K11,K22)=J2;
IF TRAC THEN
PUT SKIP EDIT('PATH# ',K11,'LINES# ',(AA(K11,M) DO M=1
  TO K22-1),'TO NODE ',AA(K11,K22))
  (COL(1),A(6),F(3),X(4),A(6),(K22-1)(X(2),F(2)),
  X(2),A(8),F(3));
FKLOOP1: END KLOOP1;
FILOOP8: END ILOOP8;
IF K11=0 THEN GOTO OUT;
DO I=1 TO K11;
  DO M=1 TO K22;
    A(I,M)=AA(I,M);
  END;
  IF A(I,K22)=IOUT THEN DO;
    ND=I;
    GOTO SUCES;
  END;

END;
GOTO UPDATE;
SUCES: K=K22-1;
DO M=1 TO K;
  B(M)=A(ND,M);
  END;
IF TRAC THEN
PUT SKIP EDIT('SUCCESS,LINES= ',(B(M) DO M=1 TO K),

```

```
      ' TO NODE ', A (ND, K22)
      (COL (1), A (14), (K) (X (2), F (2)),
      X (2), A (8), F (3)) ;
```

```
UPDATE:  RETURN;
          K1=K11;
          K2=K22;
          IF K2<5 THEN GOTO AGAIN;
OUT:     K=0;
          IF TRAC THEN
          PUT SKIP LIST ('PATH FAILURE');
          RETURN;
          END PATH;
          PUT SKIP (2) LIST ('END OF NETWORK ANALYSIS');
END ANALYZ;
```

XIV. APPENDIX F:

INPUT DATA SET FOR THE CENTRALIZED CONFIGURATION

FUNCTION SELECTION FOR THIS RUN

USE <1> TO INDICATE FUNCTION TO BE SELECTED

INITIALIZE	CALL	CALL	CALL	CALL	NEW NET
TO DEFAULTS	JSTOP	PTOS	OSASS	ANLYZ	CONFIG
<0 >	<1 >	<1 >	<1 >	<1 >	<0 >

UPBOUND-VALUES AND ALLOWABLE RANGES

NSITE	NLINE	NSTEP	NFCN	NFILE	NJOB	MLINE	MPROS	NLEVL
1/25	1/99	1/250	1/99	1/99	1/50	1/10	1/10	1/4
13	0	156	7	3	12	8	9	4

CTLPRM-VALUES AND ALLOWABLE RANGES

ISEL	XN	THRS	KCENT	TRAC	PLIST(1)	(2)	(3)	(4)
BIT	<9999	<9999	BIT	BIT	1/10	1/10	1/10	1/10
0	10	5	1	0	0	0	0	0

PROCESSOR COMPATABILITY MATRICES

PROS	1	2	3	4	5	6	7	8	9
CFILE 1	0	0	0	0	0	0	0	0	0
CFILE 2	0	0	0	0	0	0	0	0	0
CFILE 3	0	0	0	0	0	0	0	0	0

PROS	1	2	3	4	5	6	7	8	9
CFCN 1	0	1	0	0	0	0	0	0	0
CFCN 2	0	0	0	1	1	1	1	1	1
CFCN 3	0	0	0	1	1	1	1	1	1
CFCN 4	0	0	0	1	1	1	1	1	1
CFCN 5	0	0	0	0	0	0	0	1	1
CFCN 6	0	0	0	1	1	1	1	1	1
CFCN 7	0	0	0	0	0	0	0	1	1

SITE DESCRIPTORS		STATUS		XDIST	YDIST	PROS	PROS LOAD	PROS UP
SITE 1	2	0/2	<9999.9	10.0	10.0	1/10	0.0000E+00	0.0000E+00
SITE 2	2			70.0	70.0		0.0000E+00	0.0000E+00
SITE 3	2			10.0	70.0		0.0000E+00	0.0000E+00
SITE 4	2			70.0	10.0		0.0000E+00	0.0000E+00
SITE 5	2			40.0	40.0		0.0000E+00	0.0000E+00
SITE 6	2			10.0	10.0		0.0000E+00	0.0000E+00
SITE 7	2			70.0	70.0		0.0000E+00	0.0000E+00
SITE 8	2			10.0	70.0		0.0000E+00	0.0000E+00
SITE 9	2			70.0	10.0		0.0000E+00	0.0000E+00
SITE 10	2			10.0	10.0		0.0000E+00	0.0000E+00
SITE 11	2			70.0	70.0		0.0000E+00	0.0000E+00
SITE 12	2			10.0	70.0		0.0000E+00	0.0000E+00
SITE 13	2			70.0	10.0		0.0000E+00	0.0000E+00

LINKING LINE DESCRIPTORS

INSITE	FINSITE	L TYPE	L LOAD	L UP
0/25	0/25	0/10	M.MMMNESNN	M.MMMNESNN

CHARACTERISTICS OF AVAILABLE PROCESSORS

PTYPE	LVL#	I RATE	PRMSTR	SCNDSTR	MEDIA CODE	CHAN RATE	PCOST
PTYPE 1	1/10	M.MMMNESNN	M.MMMNESNN	M.MMMNESNN	0	3.0000E+01	<999999.99
PTYPE 2	1	0.0000E+00	0.0000E+00	0.0000E+00	0	2.4000E+02	20.00
PTYPE 3	2	5.3000E-01	1.0000E+00	0.0000E+00	0	1.0000E+06	30.00
PTYPE 4	2	7.6000E-01	3.2000E+01	1.5000E-01	1	2.0000E+06	34.70
PTYPE 5	2	5.7000E-01	3.2000E+01	5.1200E-01	1	1.6600E+06	80.00
PTYPE 6	3	9.8000E-01	6.4000E+01	5.0000E+00	1	2.0000E+06	200.00
PTYPE 7	3	1.0000E+01	1.2800E+02	2.8000E+01	4	1.5000E+06	550.00
							2931.40

FTYPE	8	4	1.5000E+00	5.2400E+02	8.0000E+02	7	1.3000E+06	13103.40
PTYPE	9	4	2.0000E+01	4.0000E+03	1.6000E+03	7	1.5000E+06	84172.00

CHARACTERISTICS OF AVAILABLE LINES

		XMSN RATE	LFCOST	LVCST1	LVCST2	LVCST3	LVCST4	LVCST5
		M.MMMESNN	<9999.9	<99.99	<99.99	<99.99	<99.99	<99.99
LTYPE	1	3.0000E+01	73.0	3.30	2.31	1.65	1.16	0.83
LTYPE	2	1.2000E+02	81.0	3.30	2.31	1.65	1.16	0.83
LTYPE	3	2.4000E+02	153.0	3.30	2.31	1.65	1.16	0.83
LTYPE	4	4.8000E+02	213.0	3.30	2.31	1.65	1.16	0.83
LTYPE	5	9.6000E+02	433.0	3.30	2.31	1.65	1.16	0.83
LTYPE	6	1.9200E+03	850.0	3.00	3.00	3.00	3.00	3.00
LTYPE	7	5.0000E+03	850.0	6.00	6.00	6.00	6.00	6.00
LTYPE	8	2.3000E+04	1250.0	30.00	30.00	30.00	30.00	30.00

SOFTWARE FUNCTIONS

		I STEPS	FCNTSIZE	FCNFSIZE
		M.MMMESNN	M.MMMESNN	M.MMMESNN
FCNTYPE	1	0.0000E+00	0.0000E+00	0.0000E+00
FCNTYPE	2	7.5000E+03	0.0000E+00	8.0000E+00
FCNTYPE	3	6.0000E+04	8.0000E+00	8.0000E+00
FCNTYPE	4	1.0000E+05	1.6000E+01	2.0000E+00
FCNTYPE	5	6.0000E+05	1.2800E+02	6.4000E+01
FCNTYPE	6	7.5000E+02	0.0000E+00	8.0000E+00
FCNTYPE	7	1.0000E+05	1.2800E+02	6.4000E+01

SOFTWARE FILES

		MEDIUM	FILE SIZE
		1/10	M.MMMESNN
FILETYPE	1	1	5.0000E-02
FILETYPE	2	1	3.0000E-02
FILETYPE	3	2	4.0000E+02

JOB IN WORKLOAD

	#JSTPS	ARVL RATE	RSPNS TIME	FRSTEP#	LASTEP#
	1/99	M.MMMESNN	M.MMMESNN	1/250	1/250
JOB 1	13	5.5555E-03	0.0000E+00	1	13
JOB 2	13	5.5555E-03	0.0000E+00	14	26
JOB 3	13	5.5555E-03	0.0000E+00	27	39
JOB 4	13	5.5555E-03	0.0000E+00	40	52
JOB 5	13	5.5555E-03	0.0000E+00	53	65
JOB 6	13	5.5555E-03	0.0000E+00	66	78
JOB 7	13	5.5555E-03	0.0000E+00	79	91
JOB 8	13	5.5555E-03	0.0000E+00	92	104
JOB 9	13	5.5555E-03	0.0000E+00	105	117
JOB 10	13	5.5555E-03	0.0000E+00	118	130
JOB 11	13	5.5555E-03	0.0000E+00	131	143
JOB 12	13	5.5555E-03	0.0000E+00	144	156

JOBSTEPS IN JOBS

	REQ	REQ	MSG	SP	RQ	OWN	STP	ASND		PROS	TIME	LINE	DLY	LVL	ASGND		
	FCN	FIL	LENGTH	LV	LV	JOB	#	SITE	1/25	M.MMMESNN	M.MMMESNN			1	2	3	4
	<99	<99	M.MMMESNN	<4	<4	<50	<99	1/25						B	B	B	B
JSTP 1	1	0	1.0000E+02	2	1	1	1	1	0.0000E+00	0.0000E+00	0	0	0	0	0	0	0
JSTP 2	2	0	2.5000E+02	0	0	1	2	0	0.0000E+00	0.0000E+00	0	0	0	0	0	0	0
JSTP 3	3	1	1.0000E+03	0	0	1	3	0	0.0000E+00	0.0000E+00	0	0	0	0	0	0	0
JSTP 4	1	0	2.5000E+02	0	0	1	4	0	0.0000E+00	0.0000E+00	0	0	0	0	0	0	0
JSTP 5	2	0	2.5000E+02	0	0	1	5	0	0.0000E+00	0.0000E+00	0	0	0	0	0	0	0
JSTP 6	4	2	1.6000E+03	0	0	1	6	0	0.0000E+00	0.0000E+00	0	0	0	0	0	0	0
JSTP 7	5	3	1.6000E+03	2	4	1	7	5	0.0000E+00	0.0000E+00	0	0	0	0	0	0	0
JSTP 8	2	0	1.6000E+03	0	0	1	8	0	0.0000E+00	0.0000E+00	0	0	0	0	0	0	0
JSTP 9	1	0	2.5000E+01	0	0	1	9	0	0.0000E+00	0.0000E+00	0	0	0	0	0	0	0
JSTP 10	6	0	2.5000E+01	0	0	1	10	0	0.0000E+00	0.0000E+00	0	0	0	0	0	0	0
JSTP 11	7	3	1.0000E+03	2	4	1	11	5	0.0000E+00	0.0000E+00	0	0	0	0	0	0	0
JSTP 12	3	1	1.0000E+03	0	0	1	12	0	0.0000E+00	0.0000E+00	0	0	0	0	0	0	0



JSTP	13	1	0	0.0000E+00	2	1	1	13	1	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	14	1	0	1.0000E+02	2	1	2	1	2	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	15	2	0	2.5000E+02	0	0	2	2	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	16	3	1	1.0000E+03	0	0	2	3	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	17	1	0	2.5000E+02	0	0	2	4	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	18	2	0	2.5000E+02	0	0	2	5	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	19	4	2	1.6000E+03	0	0	2	6	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	20	5	3	1.6000E+03	2	4	2	7	5	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	21	2	0	1.6000E+03	0	0	2	8	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	22	1	0	2.5000E+01	0	0	2	9	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	23	6	0	2.5000E+01	0	0	2	10	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	24	7	3	1.0000E+03	2	4	2	11	5	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	25	3	1	1.0000E+03	0	0	2	12	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	26	1	0	0.0000E+00	2	1	2	13	2	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	27	1	0	1.0000E+02	2	1	3	1	3	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	28	2	0	2.5000E+02	0	0	3	2	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	29	3	1	1.0000E+03	0	0	3	3	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	30	1	0	2.5000E+02	0	0	3	4	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	31	2	0	2.5000E+02	0	0	3	5	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	32	4	2	1.6000E+03	0	0	3	6	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	33	5	3	1.6000E+03	2	4	3	7	5	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	34	2	0	1.6000E+03	0	0	3	8	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	35	1	0	2.5000E+01	0	0	3	9	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	36	6	0	2.5000E+01	0	0	3	10	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	37	7	3	1.0000E+03	2	4	3	11	5	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	38	3	1	1.0000E+03	0	0	3	12	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	39	1	0	0.0000E+00	2	1	3	13	3	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	40	1	0	1.0000E+02	2	1	4	1	4	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	41	2	0	2.5000E+02	0	0	4	2	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	42	3	1	1.0000E+03	0	0	4	3	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	43	1	0	2.5000E+02	0	0	4	4	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	44	2	0	2.5000E+02	0	0	4	5	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	45	4	2	1.6000E+03	0	0	4	6	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	46	5	3	1.6000E+03	2	4	4	7	5	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	47	2	0	1.6000E+03	0	0	4	8	0	0.0000E+00	0.0000E+00	0	0	0	0



JSTP 83	2	0	2.5000E+02	0	0	7	5	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 84	4	2	1.6000E+03	0	0	7	6	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 85	5	3	1.6000E+03	2	4	7	7	5	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 86	2	0	1.6000E+03	0	0	7	8	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 87	1	0	2.5000E+01	0	0	7	9	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 88	6	0	2.5000E+01	0	0	7	10	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 89	7	3	1.0000E+03	2	4	7	11	5	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 90	3	1	1.0000E+03	0	0	7	12	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 91	1	0	0.0000E+00	2	1	7	13	8	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 92	1	0	1.0000E+02	2	1	8	1	9	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 93	2	0	2.5000E+02	0	0	8	2	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 94	3	1	1.0000E+03	0	0	8	3	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 95	1	0	2.5000E+02	0	0	8	4	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 96	2	0	2.5000E+02	0	0	8	5	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 97	4	2	1.6000E+03	0	0	8	6	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 98	5	3	1.6000E+03	2	4	8	7	5	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 99	2	0	1.6000E+03	0	0	8	8	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 100	1	0	2.5000E+01	0	0	8	9	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 101	6	0	2.5000E+01	0	0	8	10	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 102	7	3	1.0000E+03	2	4	8	11	5	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 103	3	1	1.0000E+03	0	0	8	12	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 104	1	0	0.0000E+00	2	1	8	13	9	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 105	1	0	1.0000E+02	2	1	9	1	10	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 106	2	0	2.5000E+02	0	0	9	2	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 107	3	1	1.0000E+03	0	0	9	3	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 108	1	0	2.5000E+02	0	0	9	4	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 109	2	0	2.5000E+02	0	0	9	5	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 110	4	2	1.6000E+03	0	0	9	6	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 111	5	3	1.6000E+03	2	4	9	7	5	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 112	2	0	1.6000E+03	0	0	9	8	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 113	1	0	2.5000E+01	0	0	9	9	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 114	6	0	2.5000E+01	0	0	9	10	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 115	7	3	1.0000E+03	2	4	9	11	5	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 116	3	1	1.0000E+03	0	0	9	12	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 117	1	0	0.0000E+00	2	1	9	13	10	0.0000E+00	0.0000E+00	0	0	0	0



JSTP 153	6	0	2.5000E+01	0	0	12	10	0	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 154	7	3	1.0000E+03	2	4	12	11	5	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 155	3	1	1.0000E+03	0	0	12	12	0	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 156	1	0	0.0000E+00	2	1	12	13	13	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0

XV. APPENDIX G:

EXECUTION OUTPUT FOR THE CENTRALIZED CONFIGURATION

END OF JOBSTEP TO PROCESSOR ASSIGNMENT  
END OF PROCESSOR ASSIGNMENT  
END OF OPTIMUM LOCATING  
START NETWORK ANALYSIS

PROCESSOR 2 SITE 1  
MIPS= 0.0000E+00 UTILIZED= 0.0000E+00  
AVRAM= 0.0000E+00 UTILIZED= 0.0000E+00  
FIXED SCNDSTR= 0.0000E+00 UTILIZED= 0.0000E+00  
PROCESSOR 2 SITE 2  
MIPS= 0.0000E+00 UTILIZED= 0.0000E+00  
AVRAM= 0.0000E+00 UTILIZED= 0.0000E+00  
FIXED SCNDSTR= 0.0000E+00 UTILIZED= 0.0000E+00  
PROCESSOR 2 SITE 3  
MIPS= 0.0000E+00 UTILIZED= 0.0000E+00  
AVRAM= 0.0000E+00 UTILIZED= 0.0000E+00  
FIXED SCNDSTR= 0.0000E+00 UTILIZED= 0.0000E+00  
PROCESSOR 2 SITE 4  
MIPS= 0.0000E+00 UTILIZED= 0.0000E+00  
AVRAM= 0.0000E+00 UTILIZED= 0.0000E+00  
FIXED SCNDSTR= 0.0000E+00 UTILIZED= 0.0000E+00  
PROCESSOR 8 SITE 5  
MIPS= 6.2882E-02 UTILIZED= 4.1922E-02  
AVRAM= 1.5810E+02 UTILIZED= 3.0171E-01  
FIXED SCNDSTR= 4.0000E+02 UTILIZED= 5.0000E-01  
PROCESSOR 2 SITE 6  
MIPS= 0.0000E+00 UTILIZED= 0.0000E+00  
AVRAM= 0.0000E+00 UTILIZED= 0.0000E+00  
FIXED SCNDSTR= 0.0000E+00 UTILIZED= 0.0000E+00  
PROCESSOR 2 SITE 7  
MIPS= 0.0000E+00 UTILIZED= 0.0000E+00  
AVRAM= 0.0000E+00 UTILIZED= 0.0000E+00  
FIXED SCNDSTR= 0.0000E+00 UTILIZED= 0.0000E+00  
PROCESSOR 2 SITE 8  
MIPS= 0.0000E+00 UTILIZED= 0.0000E+00

AVRAM= 0.0000E+00 UTILIZED= 0.0000E+00  
 FIXED SCNDSTR= 0.0000E+00 UTILIZED= 0.0000E+00  
 PROCESSOR 2 SITE 9  
 MIPS= 0.0000E+00 UTILIZED= 0.0000E+00  
 AVRAM= 0.0000E+00 UTILIZED= 0.0000E+00  
 FIXED SCNDSTR= 0.0000E+00 UTILIZED= 0.0000E+00  
 PROCESSOR 2 SITE 10  
 MIPS= 0.0000E+00 UTILIZED= 0.0000E+00  
 AVRAM= 0.0000E+00 UTILIZED= 0.0000E+00  
 FIXED SCNDSTR= 0.0000E+00 UTILIZED= 0.0000E+00  
 PROCESSOR 2 SITE 11  
 MIPS= 0.0000E+00 UTILIZED= 0.0000E+00  
 AVRAM= 0.0000E+00 UTILIZED= 0.0000E+00  
 FIXED SCNDSTR= 0.0000E+00 UTILIZED= 0.0000E+00  
 PROCESSOR 2 SITE 12  
 MIPS= 0.0000E+00 UTILIZED= 0.0000E+00  
 AVRAM= 0.0000E+00 UTILIZED= 0.0000E+00  
 FIXED SCNDSTR= 0.0000E+00 UTILIZED= 0.0000E+00  
 PROCESSOR 2 SITE 13  
 MIPS= 0.0000E+00 UTILIZED= 0.0000E+00  
 AVRAM= 0.0000E+00 UTILIZED= 0.0000E+00  
 FIXED SCNDSTR= 0.0000E+00 UTILIZED= 0.0000E+00

INSITE 1	, FINSITE 5	, LINE TYPE IS 2	, LOAD= 2.2083E+01	, UTILIZE= 1.8403E-01
INSITE 2	, FINSITE 5	, LINE TYPE IS 2	, LOAD= 2.2083E+01	, UTILIZE= 1.8403E-01
INSITE 3	, FINSITE 5	, LINE TYPE IS 2	, LOAD= 2.2083E+01	, UTILIZE= 1.8403E-01
INSITE 4	, FINSITE 5	, LINE TYPE IS 2	, LOAD= 2.2083E+01	, UTILIZE= 1.8403E-01
INSITE 6	, FINSITE 5	, LINE TYPE IS 2	, LOAD= 2.2083E+01	, UTILIZE= 1.8403E-01
INSITE 7	, FINSITE 5	, LINE TYPE IS 2	, LOAD= 2.2083E+01	, UTILIZE= 1.8403E-01
INSITE 8	, FINSITE 5	, LINE TYPE IS 2	, LOAD= 2.2083E+01	, UTILIZE= 1.8403E-01
INSITE 9	, FINSITE 5	, LINE TYPE IS 2	, LOAD= 2.2083E+01	, UTILIZE= 1.8403E-01
INSITE 10	, FINSITE 5	, LINE TYPE IS 2	, LOAD= 2.2083E+01	, UTILIZE= 1.8403E-01
INSITE 11	, FINSITE 5	, LINE TYPE IS 2	, LOAD= 2.2083E+01	, UTILIZE= 1.8403E-01
INSITE 12	, FINSITE 5	, LINE TYPE IS 2	, LOAD= 2.2083E+01	, UTILIZE= 1.8403E-01
INSITE 13	, FINSITE 5	, LINE TYPE IS 2	, LOAD= 2.2083E+01	, UTILIZE= 1.8403E-01



```

JOB= 1
STEP= 1 ,SITE= 1
  PROS TIME= 0.0000E+00 ,LINE DELAY= 1.0213E+00 ,JOB TIME ACCUM= 1.0213E+00
STEP= 2 ,SITE= 5
  PROS TIME= 7.1603E-03 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 1.0284E+00
STEP= 3 ,SITE= 5
  PROS TIME= 5.7283E-02 ,LINE DELAY= 1.0213E+01 ,JOB TIME ACCUM= 1.1298E+01
STEP= 4 ,SITE= 1
  PROS TIME= 0.0000E+00 ,LINE DELAY= 2.5532E+00 ,JOB TIME ACCUM= 1.3852E+01
STEP= 5 ,SITE= 5
  PROS TIME= 7.1603E-03 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 1.3859E+01
STEP= 6 ,SITE= 5
  PROS TIME= 9.5471E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 1.3954E+01
STEP= 7 ,SITE= 5
  PROS TIME= 5.7283E-01 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 1.4527E+01
STEP= 8 ,SITE= 5
  PROS TIME= 7.1603E-03 ,LINE DELAY= 1.6340E+01 ,JOB TIME ACCUM= 3.0875E+01
STEP= 9 ,SITE= 1
  PROS TIME= 0.0000E+00 ,LINE DELAY= 2.5532E-01 ,JOB TIME ACCUM= 3.1130E+01
STEP= 10 ,SITE= 5
  PROS TIME= 7.1603E-04 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 3.1131E+01
STEP= 11 ,SITE= 5
  PROS TIME= 9.5471E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 3.1226E+01
STEP= 12 ,SITE= 5
  PROS TIME= 5.7283E-02 ,LINE DELAY= 1.0213E+01 ,JOB TIME ACCUM= 4.1496E+01
STEP= 13 ,SITE= 1
  PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 4.1496E+01
JOB= 2
STEP= 1 ,SITE= 2
  PROS TIME= 0.0000E+00 ,LINE DELAY= 1.0213E+00 ,JOB TIME ACCUM= 1.0213E+00
STEP= 2 ,SITE= 5
  PROS TIME= 7.1603E-03 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 1.0284E+00
STEP= 3 ,SITE= 5
  PROS TIME= 5.7283E-02 ,LINE DELAY= 1.0213E+01 ,JOB TIME ACCUM= 1.1298E+01
STEP= 4 ,SITE= 2

```

PROS TIME=	0.0000E+00	,LINE DELAY=	2.5532E+00	,JOB TIME ACCUM=	1.3852E+01
STEP=	5	,SITE=	5		
PROS TIME=	7.1603E-03	,LINE DELAY=	0.0000E+00	,JOB TIME ACCUM=	1.3859E+01
STEP=	6	,SITE=	5		
PROS TIME=	9.5471E-02	,LINE DELAY=	0.0000E+00	,JOB TIME ACCUM=	1.3954E+01
STEP=	7	,SITE=	5		
PROS TIME=	5.7283E-01	,LINE DELAY=	0.0000E+00	,JOB TIME ACCUM=	1.4527E+01
STEP=	8	,SITE=	5		
PROS TIME=	7.1603E-03	,LINE DELAY=	1.6340E+01	,JOB TIME ACCUM=	3.0875E+01
STEP=	9	,SITE=	2		
PROS TIME=	0.0000E+00	,LINE DELAY=	2.5532E-01	,JOB TIME ACCUM=	3.1130E+01
STEP=	10	,SITE=	5		
PROS TIME=	7.1603E-04	,LINE DELAY=	0.0000E+00	,JOB TIME ACCUM=	3.1131E+01
STEP=	11	,SITE=	5		
PROS TIME=	9.5471E-02	,LINE DELAY=	0.0000E+00	,JOB TIME ACCUM=	3.1226E+01
STEP=	12	,SITE=	5		
PROS TIME=	5.7283E-02	,LINE DELAY=	1.0213E+01	,JOB TIME ACCUM=	4.1496E+01
STEP=	13	,SITE=	2		
PROS TIME=	0.0000E+00	,LINE DELAY=	0.0000E+00	,JOB TIME ACCUM=	4.1496E+01
JCB=	3				
STEP=	1	,SITE=	3		
PROS TIME=	0.0000E+00	,LINE DELAY=	1.0213E+00	,JOB TIME ACCUM=	1.0213E+00
STEP=	2	,SITE=	5		
PROS TIME=	7.1603E-03	,LINE DELAY=	0.0000E+00	,JOB TIME ACCUM=	1.0284E+00
STEP=	3	,SITE=	5		
PROS TIME=	5.7283E-02	,LINE DELAY=	1.0213E+01	,JOB TIME ACCUM=	1.1298E+01
STEP=	4	,SITE=	3		
PROS TIME=	0.0000E+00	,LINE DELAY=	2.5532E+00	,JOB TIME ACCUM=	1.3852E+01
STEP=	5	,SITE=	5		
PROS TIME=	7.1603E-03	,LINE DELAY=	0.0000E+00	,JOB TIME ACCUM=	1.3859E+01
STEP=	6	,SITE=	5		
PROS TIME=	9.5471E-02	,LINE DELAY=	0.0000E+00	,JOB TIME ACCUM=	1.3954E+01
STEP=	7	,SITE=	5		
PROS TIME=	5.7283E-01	,LINE DELAY=	0.0000E+00	,JOB TIME ACCUM=	1.4527E+01
STEP=	8	,SITE=	5		

PROS TIME= 7.1603E-03 ,LINE DELAY= 1.6340E+01 ,JOB TIME ACCUM= 3.0875E+01  
 STEP= 9 ,SITE= 3  
 PROS TIME= 0.0000E+00 ,LINE DELAY= 2.5532E-01 ,JOB TIME ACCUM= 3.1130E+01  
 STEP= 10 ,SITE= 5  
 PROS TIME= 7.1603E-04 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 3.1131E+01  
 STEP= 11 ,SITE= 5  
 PROS TIME= 9.5471E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 3.1226E+01  
 STEP= 12 ,SITE= 5  
 PROS TIME= 5.7283E-02 ,LINE DELAY= 1.0213E+01 ,JOB TIME ACCUM= 4.1496E+01  
 STEP= 13 ,SITE= 3  
 PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 4.1496E+01  
 JOB= 4  
 STEP= 1 ,SITE= 4  
 PROS TIME= 0.0000E+00 ,LINE DELAY= 1.0213E+00 ,JOB TIME ACCUM= 1.0213E+00  
 STEP= 2 ,SITE= 5  
 PROS TIME= 7.1603E-03 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 1.0284E+00  
 STEP= 3 ,SITE= 5  
 PROS TIME= 5.7283E-02 ,LINE DELAY= 1.0213E+01 ,JOB TIME ACCUM= 1.1298E+01  
 STEP= 4 ,SITE= 4  
 PROS TIME= 0.0000E+00 ,LINE DELAY= 2.5532E+00 ,JOB TIME ACCUM= 1.3852E+01  
 STEP= 5 ,SITE= 5  
 PROS TIME= 7.1603E-03 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 1.3859E+01  
 STEP= 6 ,SITE= 5  
 PROS TIME= 9.5471E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 1.3954E+01  
 STEP= 7 ,SITE= 5  
 PROS TIME= 5.7283E-01 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 1.4527E+01  
 STEP= 8 ,SITE= 5  
 PROS TIME= 7.1603E-03 ,LINE DELAY= 1.6340E+01 ,JOB TIME ACCUM= 3.0875E+01  
 STEP= 9 ,SITE= 4  
 PROS TIME= 0.0000E+00 ,LINE DELAY= 2.5532E-01 ,JOB TIME ACCUM= 3.1130E+01  
 STEP= 10 ,SITE= 5  
 PROS TIME= 7.1603E-04 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 3.1131E+01  
 STEP= 11 ,SITE= 5  
 PROS TIME= 9.5471E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 3.1226E+01  
 STEP= 12 ,SITE= 5

PROS TIME= 5.7283E-02 ,LINE DELAY= 1.0213E+01 ,JOB TIME ACCUM= 4.1496E+01  
 STEP= 13 ,SITE= 4  
 PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 4.1496E+01  
 JOB= 5  
 STEP= 1 ,SITE= 6  
 PROS TIME= 0.0000E+00 ,LINE DELAY= 1.0213E+00 ,JOB TIME ACCUM= 1.0213E+00  
 STEP= 2 ,SITE= 5  
 PROS TIME= 7.1603E-03 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 1.0284E+00  
 STEP= 3 ,SITE= 5  
 PROS TIME= 5.7283E-02 ,LINE DELAY= 1.0213E+01 ,JOB TIME ACCUM= 1.1298E+01  
 STEP= 4 ,SITE= 6  
 PROS TIME= 0.0000E+00 ,LINE DELAY= 2.5532E+00 ,JOB TIME ACCUM= 1.3852E+01  
 STEP= 5 ,SITE= 5  
 PROS TIME= 7.1603E-03 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 1.3859E+01  
 STEP= 6 ,SITE= 5  
 PROS TIME= 9.5471E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 1.3954E+01  
 STEP= 7 ,SITE= 5  
 PROS TIME= 5.7283E-01 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 1.4527E+01  
 STEP= 8 ,SITE= 5  
 PROS TIME= 7.1603E-03 ,LINE DELAY= 1.6340E+01 ,JOB TIME ACCUM= 3.0875E+01  
 STEP= 9 ,SITE= 6  
 PROS TIME= 0.0000E+00 ,LINE DELAY= 2.5532E-01 ,JOB TIME ACCUM= 3.1130E+01  
 STEP= 10 ,SITE= 5  
 PROS TIME= 7.1603E-04 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 3.1131E+01  
 STEP= 11 ,SITE= 5  
 PROS TIME= 9.5471E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 3.1226E+01  
 STEP= 12 ,SITE= 5  
 PROS TIME= 5.7283E-02 ,LINE DELAY= 1.0213E+01 ,JOB TIME ACCUM= 4.1496E+01  
 STEP= 13 ,SITE= 6  
 PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 4.1496E+01  
 JOB= 6  
 STEP= 1 ,SITE= 7  
 PROS TIME= 0.0000E+00 ,LINE DELAY= 1.0213E+00 ,JOB TIME ACCUM= 1.0213E+00  
 STEP= 2 ,SITE= 5  
 PROS TIME= 7.1603E-03 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 1.0284E+00

STEP= 3 ,SITE= 5	PROS TIME= 5.7283E-02 ,LINE DELAY= 1.0213E+01 ,JOB TIME ACCUM= 1.1298E+01
STEP= 4 ,SITE= 7	PROS TIME= 0.0000E+00 ,LINE DELAY= 2.5532E+00 ,JOB TIME ACCUM= 1.3852E+01
STEP= 5 ,SITE= 5	PROS TIME= 7.1603E-03 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 1.3859E+01
STEP= 6 ,SITE= 5	PROS TIME= 9.5471E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 1.3954E+01
STEP= 7 ,SITE= 5	PROS TIME= 5.7283E-01 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 1.4527E+01
STEP= 8 ,SITE= 5	PROS TIME= 7.1603E-03 ,LINE DELAY= 1.6340E+01 ,JOB TIME ACCUM= 3.0875E+01
STEP= 9 ,SITE= 7	PROS TIME= 0.0000E+00 ,LINE DELAY= 2.5532E-01 ,JOB TIME ACCUM= 3.1130E+01
STEP= 10 ,SITE= 5	PROS TIME= 7.1603E-04 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 3.1131E+01
STEP= 11 ,SITE= 5	PROS TIME= 9.5471E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 3.1226E+01
STEP= 12 ,SITE= 5	PROS TIME= 5.7283E-02 ,LINE DELAY= 1.0213E+01 ,JOB TIME ACCUM= 4.1496E+01
STEP= 13 ,SITE= 7	PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 4.1496E+01
JOB= 7	
STEP= 1 ,SITE= 8	PROS TIME= 0.0000E+00 ,LINE DELAY= 1.0213E+00 ,JOB TIME ACCUM= 1.0213E+00
STEP= 2 ,SITE= 5	PROS TIME= 7.1603E-03 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 1.0284E+00
STEP= 3 ,SITE= 5	PROS TIME= 5.7283E-02 ,LINE DELAY= 1.0213E+01 ,JOB TIME ACCUM= 1.1298E+01
STEP= 4 ,SITE= 8	PROS TIME= 0.0000E+00 ,LINE DELAY= 2.5532E+00 ,JOB TIME ACCUM= 1.3852E+01
STEP= 5 ,SITE= 5	PROS TIME= 7.1603E-03 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 1.3859E+01
STEP= 6 ,SITE= 5	PROS TIME= 9.5471E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 1.3954E+01

STEP= 7 ,SITE= 5	PROS TIME= 5.7283E-01 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 1.4527E+01
STEP= 8 ,SITE= 5	PROS TIME= 7.1603E-03 ,LINE DELAY= 1.6340E+01 ,JOB TIME ACCUM= 3.0875E+01
STEP= 9 ,SITE= 8	PROS TIME= 0.0000E+00 ,LINE DELAY= 2.5532E-01 ,JOB TIME ACCUM= 3.1130E+01
STEP= 10 ,SITE= 5	PROS TIME= 7.1603E-04 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 3.1131E+01
STEP= 11 ,SITE= 5	PROS TIME= 9.5471E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 3.1226E+01
STEP= 12 ,SITE= 5	PROS TIME= 5.7283E-02 ,LINE DELAY= 1.0213E+01 ,JOB TIME ACCUM= 4.1496E+01
STEP= 13 ,SITE= 8	PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 4.1496E+01
JOB= 8	
STEP= 1 ,SITE= 9	PROS TIME= 0.0000E+00 ,LINE DELAY= 1.0213E+00 ,JOB TIME ACCUM= 1.0213E+00
STEP= 2 ,SITE= 5	PROS TIME= 7.1603E-03 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 1.0284E+00
STEP= 3 ,SITE= 5	PROS TIME= 5.7283E-02 ,LINE DELAY= 1.0213E+01 ,JOB TIME ACCUM= 1.1298E+01
STEP= 4 ,SITE= 9	PROS TIME= 0.0000E+00 ,LINE DELAY= 2.5532E+00 ,JOB TIME ACCUM= 1.3852E+01
STEP= 5 ,SITE= 5	PROS TIME= 7.1603E-03 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 1.3859E+01
STEP= 6 ,SITE= 5	PROS TIME= 9.5471E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 1.3954E+01
STEP= 7 ,SITE= 5	PROS TIME= 5.7283E-01 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 1.4527E+01
STEP= 8 ,SITE= 5	PROS TIME= 7.1603E-03 ,LINE DELAY= 1.6340E+01 ,JOB TIME ACCUM= 3.0875E+01
STEP= 9 ,SITE= 9	PROS TIME= 0.0000E+00 ,LINE DELAY= 2.5532E-01 ,JOB TIME ACCUM= 3.1130E+01
STEP= 10 ,SITE= 5	PROS TIME= 7.1603E-04 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 3.1131E+01

```

STEP= 11 ,SITE= 5
  PROS TIME= 9.5471E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 3.1226E+01
STEP= 12 ,SITE= 5
  PROS TIME= 5.7283E-02 ,LINE DELAY= 1.0213E+01 ,JOB TIME ACCUM= 4.1496E+01
STEP= 13 ,SITE= 9
  PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 4.1496E+01
JOB= 9
STEP= 1 ,SITE= 10
  PROS TIME= 0.0000E+00 ,LINE DELAY= 1.0213E+00 ,JOB TIME ACCUM= 1.0213E+00
STEP= 2 ,SITE= 5
  PROS TIME= 7.1603E-03 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 1.0284E+00
STEP= 3 ,SITE= 5
  PROS TIME= 5.7283E-02 ,LINE DELAY= 1.0213E+01 ,JOB TIME ACCUM= 1.1298E+01
STEP= 4 ,SITE= 10
  PROS TIME= 0.0000E+00 ,LINE DELAY= 2.5532E+00 ,JOB TIME ACCUM= 1.3852E+01
STEP= 5 ,SITE= 5
  PROS TIME= 7.1603E-03 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 1.3859E+01
STEP= 6 ,SITE= 5
  PROS TIME= 9.5471E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 1.3954E+01
STEP= 7 ,SITE= 5
  PROS TIME= 5.7283E-01 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 1.4527E+01
STEP= 8 ,SITE= 5
  PROS TIME= 7.1603E-03 ,LINE DELAY= 1.6340E+01 ,JOB TIME ACCUM= 3.0875E+01
STEP= 9 ,SITE= 10
  PROS TIME= 0.0000E+00 ,LINE DELAY= 2.5532E-01 ,JOB TIME ACCUM= 3.1130E+01
STEP= 10 ,SITE= 5
  PROS TIME= 7.1603E-04 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 3.1131E+01
STEP= 11 ,SITE= 5
  PROS TIME= 9.5471E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 3.1226E+01
STEP= 12 ,SITE= 5
  PROS TIME= 5.7283E-02 ,LINE DELAY= 1.0213E+01 ,JOB TIME ACCUM= 4.1496E+01
STEP= 13 ,SITE= 10
  PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 4.1496E+01
JOB= 10
STEP= 1 ,SITE= 11

```

```

PROS TIME= 0.0000E+00 ,LINE DELAY= 1.0213E+00 ,JOB TIME ACCUM= 1.0213E+00
STEP= 2 ,SITE= 5
PROS TIME= 7.1603E-03 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 1.0284E+00
STEP= 3 ,SITE= 5
PROS TIME= 5.7283E-02 ,LINE DELAY= 1.0213E+01 ,JOB TIME ACCUM= 1.1298E+01
STEP= 4 ,SITE= 11
PROS TIME= 0.0000E+00 ,LINE DELAY= 2.5532E+00 ,JOB TIME ACCUM= 1.3852E+01
STEP= 5 ,SITE= 5
PROS TIME= 7.1603E-03 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 1.3859E+01
STEP= 6 ,SITE= 5
PROS TIME= 9.5471E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 1.3954E+01
STEP= 7 ,SITE= 5
PROS TIME= 5.7283E-01 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 1.4527E+01
STEP= 8 ,SITE= 5
PROS TIME= 7.1603E-03 ,LINE DELAY= 1.6340E+01 ,JOB TIME ACCUM= 3.0875E+01
STEP= 9 ,SITE= 11
PROS TIME= 0.0000E+00 ,LINE DELAY= 2.5532E-01 ,JOB TIME ACCUM= 3.1130E+01
STEP= 10 ,SITE= 5
PROS TIME= 7.1603E-04 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 3.1131E+01
STEP= 11 ,SITE= 5
PROS TIME= 9.5471E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 3.1226E+01
STEP= 12 ,SITE= 5
PROS TIME= 5.7283E-02 ,LINE DELAY= 1.0213E+01 ,JOB TIME ACCUM= 4.1496E+01
STEP= 13 ,SITE= 11
PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 4.1496E+01
JOB= 11
STEP= 1 ,SITE= 12
PROS TIME= 0.0000E+00 ,LINE DELAY= 1.0213E+00 ,JOB TIME ACCUM= 1.0213E+00
STEP= 2 ,SITE= 5
PROS TIME= 7.1603E-03 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 1.0284E+00
STEP= 3 ,SITE= 5
PROS TIME= 5.7283E-02 ,LINE DELAY= 1.0213E+01 ,JOB TIME ACCUM= 1.1298E+01
STEP= 4 ,SITE= 12
PROS TIME= 0.0000E+00 ,LINE DELAY= 2.5532E+00 ,JOB TIME ACCUM= 1.3852E+01
STEP= 5 ,SITE= 5

```



PROS TIME= 7.1603E-03 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 1.3859E+01  
 STEP= 6 ,SITE= 5  
 PROS TIME= 9.5471E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 1.3954E+01  
 STEP= 7 ,SITE= 5  
 PROS TIME= 5.7283E-01 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 1.4527E+01  
 STEP= 8 ,SITE= 5  
 PROS TIME= 7.1603E-03 ,LINE DELAY= 1.6340E+01 ,JOB TIME ACCUM= 3.0875E+01  
 STEP= 9 ,SITE= 12  
 PROS TIME= 0.0000E+00 ,LINE DELAY= 2.5532E-01 ,JOB TIME ACCUM= 3.1130E+01  
 STEP= 10 ,SITE= 5  
 PROS TIME= 7.1603E-04 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 3.1131E+01  
 STEP= 11 ,SITE= 5  
 PROS TIME= 9.5471E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 3.1226E+01  
 STEP= 12 ,SITE= 5  
 PROS TIME= 5.7283E-02 ,LINE DELAY= 1.0213E+01 ,JOB TIME ACCUM= 4.1496E+01  
 STEP= 13 ,SITE= 12  
 PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 4.1496E+01  
 JOB= 12  
 STEP= 1 ,SITE= 13  
 PROS TIME= 0.0000E+00 ,LINE DELAY= 1.0213E+00 ,JOB TIME ACCUM= 1.0213E+00  
 STEP= 2 ,SITE= 5  
 PROS TIME= 7.1603E-03 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 1.0284E+00  
 STEP= 3 ,SITE= 5  
 PROS TIME= 5.7283E-02 ,LINE DELAY= 1.0213E+01 ,JOB TIME ACCUM= 1.1298E+01  
 STEP= 4 ,SITE= 13  
 PROS TIME= 0.0000E+00 ,LINE DELAY= 2.5532E+00 ,JOB TIME ACCUM= 1.3852E+01  
 STEP= 5 ,SITE= 5  
 PROS TIME= 7.1603E-03 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 1.3859E+01  
 STEP= 6 ,SITE= 5  
 PROS TIME= 9.5471E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 1.3954E+01  
 STEP= 7 ,SITE= 5  
 PROS TIME= 5.7283E-01 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 1.4527E+01  
 STEP= 8 ,SITE= 5  
 PROS TIME= 7.1603E-03 ,LINE DELAY= 1.6340E+01 ,JOB TIME ACCUM= 3.0875E+01  
 STEP= 9 ,SITE= 13

PROS TIME= 0.0000E+00 ,LINE DELAY= 2.5532E-01 ,JOB TIME ACCUM= 3.1130E+01  
 STEP= 10 ,SITE= 5  
 PROS TIME= 7.1603E-04 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 3.1131E+01  
 STEP= 11 ,SITE= 5  
 PROS TIME= 9.5471E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 3.1226E+01  
 STEP= 12 ,SITE= 5  
 PROS TIME= 5.7283E-02 ,LINE DELAY= 1.0213E+01 ,JOB TIME ACCUM= 4.1496E+01  
 STEP= 13 ,SITE= 13  
 PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 4.1496E+01

LINE 1 COST 178.94  
 LINE 2 COST 178.94  
 LINE 3 COST 178.94  
 LINE 4 COST 178.94  
 LINE 5 COST 178.94  
 LINE 6 COST 178.94  
 LINE 7 COST 178.94  
 LINE 8 COST 178.94  
 LINE 9 COST 178.94  
 LINE 10 COST 178.94  
 LINE 11 COST 178.94  
 LINE 12 COST 178.94

SITE 1 PROCESSOR COST 30.00  
 SITE 2 PROCESSOR COST 30.00  
 SITE 3 PROCESSOR COST 30.00  
 SITE 4 PROCESSOR COST 30.00  
 SITE 5 PROCESSOR COST 13103.40  
 SITE 6 PROCESSOR COST 30.00  
 SITE 7 PROCESSOR COST 30.00  
 SITE 8 PROCESSOR COST 30.00  
 SITE 9 PROCESSOR COST 30.00  
 SITE 10 PROCESSOR COST 30.00  
 SITE 11 PROCESSOR COST 30.00  
 SITE 12 PROCESSOR COST 30.00

SITE 13 PROCESSOR COST 30.00  
TOTAL LINE COST= 2147.28 ,TOTAL PROS COST= 13463.40 ,TOTAL COST= 15610  
END OF NETWORK ANALYSIS

XVI. APPENDIX H:

OUTPUT DATA SET FOR THE CENTRALIZED CONFIGURATION

FUNCTION SELECTION FOR THIS RUN

USE <1> TO INDICATE FUNCTION TO BE SELECTED

INITIALIZE	CALL	CALL	CALL	CALL	NEW NET
TO DEFAULTS	JSTOP	PTOS	OSASS	ANLYZ	CONFIG
<0 >	<1 >	<1 >	<1 >	<1 >	<0 >

UPBOUND-VALUES AND ALLOWABLE RANGES

NSITE	NLINE	NSTEP	NFCN	NFILE	NJOB	NLINE	MPROS	NLEVL
1/25	1/99	1/250	1/99	1/99	1/50	1/10	1/10	1/4
13	12	156	7	3	12	8	9	4

CTLPRM-VALUES AND ALLOWABLE RANGES

ISEL	XN	THRS	KCENT	TRAC	PLIST (1)	(2)	(3)	(4)
BIT	<9999	<9999	BIT	BIT	1/10	1/10	1/10	1/10
0	10	5	1	0	2	4	6	8

PROCESSOR COMPATABILITY MATRICES

PROS		1	2	3	4	5	6	7	8	9
CFILE	1	0	0	0	1	1	1	1	1	1
CFILE	2	0	0	0	1	1	1	1	1	1
CFILE	3	0	0	0	0	0	0	0	1	1

PROS		1	2	3	4	5	6	7	8	9
CFCN	1	0	1	0	0	0	0	0	0	0
CFCN	2	0	0	0	1	1	1	1	1	1
CFCN	3	0	0	0	1	1	1	1	1	1
CFCN	4	0	0	0	1	1	1	1	1	1
CFCN	5	0	0	0	0	0	0	0	1	1
CFCN	6	0	0	0	1	1	1	1	1	1
CFCN	7	0	0	0	0	0	0	0	1	1

SITE DESCRIPTORS

	STATUS	XDIST	YDIST	PROS	PROS LOAD	PROS UF
	0/2	<9999.9	<9999.9	1/10	M.MMMESNN	M.MMMESNN
SITE 1	2	10.0	10.0	2	0.0000E+00	0.0000E+00
SITE 2	2	70.0	70.0	2	0.0000E+00	0.0000E+00
SITE 3	2	10.0	70.0	2	0.0000E+00	0.0000E+00
SITE 4	2	70.0	10.0	2	0.0000E+00	0.0000E+00
SITE 5	2	40.0	40.0	8	6.2882E-02	4.1922E-02
SITE 6	2	10.0	10.0	2	0.0000E+00	0.0000E+00
SITE 7	2	70.0	70.0	2	0.0000E+00	0.0000E+00
SITE 8	2	10.0	70.0	2	0.0000E+00	0.0000E+00
SITE 9	2	70.0	10.0	2	0.0000E+00	0.0000E+00
SITE 10	2	10.0	10.0	2	0.0000E+00	0.0000E+00
SITE 11	2	70.0	70.0	2	0.0000E+00	0.0000E+00
SITE 12	2	10.0	70.0	2	0.0000E+00	0.0000E+00
SITE 13	2	70.0	10.0	2	0.0000E+00	0.0000E+00

LINKING LINE DESCRIPTORS

	INSITE	FINSITE	L TYPE	L LOAD	L UF
	0/25	0/25	0/10	M.MMMESNN	M.MMMESNN
LINE 1	1	5	2	2.2083E+01	1.8403E-01
LINE 2	2	5	2	2.2083E+01	1.8403E-01
LINE 3	3	5	2	2.2083E+01	1.8403E-01
LINE 4	4	5	2	2.2083E+01	1.8403E-01
LINE 5	6	5	2	2.2083E+01	1.8403E-01
LINE 6	7	5	2	2.2083E+01	1.8403E-01
LINE 7	8	5	2	2.2083E+01	1.8403E-01
LINE 8	9	5	2	2.2083E+01	1.8403E-01
LINE 9	10	5	2	2.2083E+01	1.8403E-01
LINE 10	11	5	2	2.2083E+01	1.8403E-01
LINE 11	12	5	2	2.2083E+01	1.8403E-01
LINE 12	13	5	2	2.2083E+01	1.8403E-01

CHARACTERISTICS OF AVAILABLE PROCESSORS

	LVL#	I RATE	PRIMSTR	SCNDSTR	MEDIA	CHAN RATE	PCOST
	1/10	M.MMMESNN	M.MMMESNN	M.MMMESNN	CODE	M.MMMESNN	<999999.99
PTYPE	1	0.0000E+00	0.0000E+00	0.0000E+00	0	3.0000E+01	20.00
PTYPE	2	0.0000E+00	0.0000E+00	0.0000E+00	0	2.4000E+02	30.00
PTYPE	3	5.3000E-01	1.0000E+00	0.0000E+00	0	1.0000E+06	34.70
PTYPE	4	7.6000E-01	3.2000E+01	1.5000E-01	1	2.0000E+06	80.00
PTYPE	5	5.7000E-01	3.2000E+01	5.1200E-01	1	1.6600E+06	200.00
PTYPE	6	9.8000E-01	6.4000E+01	5.0000E+00	1	2.0000E+06	550.00
PTYPE	7	1.0000E+01	1.2800E+02	2.8000E+01	4	1.5000E+06	2931.40
PTYPE	8	1.5000E+00	5.2400E+02	8.0000E+02	7	1.3000E+06	13103.40
PTYPE	9	2.0000E+01	4.0000E+03	1.6000E+03	7	1.5000E+06	84172.00

CHARACTERISTICS OF AVAILABLE LINES

	XMSN RATE	LFCOST	LVCST1	LVCST2	LVCST3	LVCST4	LVCST5	
	M.MMMESNN	<9999.9	<99.99	<99.99	<99.99	<99.99	<99.99	
LTYPE	1	3.0000E+01	73.0	3.30	2.31	1.65	1.16	0.83
LTYPE	2	1.2000E+02	81.0	3.30	2.31	1.65	1.16	0.83
LTYPE	3	2.4000E+02	153.0	3.30	2.31	1.65	1.16	0.83
LTYPE	4	4.8000E+02	213.0	3.30	2.31	1.65	1.16	0.83
LTYPE	5	9.6000E+02	433.0	3.30	2.31	1.65	1.16	0.83
LTYPE	6	1.9200E+03	850.0	3.00	3.00	3.00	3.00	3.00
LTYPE	7	5.0000E+03	850.0	6.00	6.00	6.00	6.00	6.00
LTYPE	8	2.3000E+04	1250.0	30.00	30.00	30.00	30.00	30.00

177

SOFTWARE FUNCTIONS

	I STEPS	FCNTSIZE	FCNFSIZE
	M.MMMESNN	M.MMMESNN	M.MMMESNN
FCNTYPE	1	0.0000E+00	0.0000E+00
FCNTYPE	2	7.5000E+03	8.0000E+00
FCNTYPE	3	6.0000E+04	8.0000E+00

FCNTYPE	4	1.0000E+05	1.6000E+01	2.0000E+00
FCNTYPE	5	6.0000E+05	1.2800E+02	6.4000E+01
FCNTYPE	6	7.5000E+02	0.0000E+00	8.0000E+00
FCNTYPE	7	1.0000E+05	1.2800E+02	6.4000E+01

SOFTWARE FILES

		MEDIUM	FILE SIZE
		1/10	M.MMMESNN
FILETYPE	1	1	5.0000E-02
FILETYPE	2	1	3.0000E-02
FILETYPE	3	2	4.0000E+02

JOBS IN WORKLOAD

		#JSTPS	ARVL RATE	RSPNS TIME	FRSTEP#	LASTEP#
		1/99	M.MMMESNN	M.MMMESNN	1/250	1/250
JOB	1	13	5.5555E-03	4.1496E+01	1	13
JOB	2	13	5.5555E-03	4.1496E+01	14	26
JOB	3	13	5.5555E-03	4.1496E+01	27	39
JOB	4	13	5.5555E-03	4.1496E+01	40	52
JOB	5	13	5.5555E-03	4.1496E+01	53	65
JOB	6	13	5.5555E-03	4.1496E+01	66	78
JOB	7	13	5.5555E-03	4.1496E+01	79	91
JOB	8	13	5.5555E-03	4.1496E+01	92	104
JOB	9	13	5.5555E-03	4.1496E+01	105	117
JOB	10	13	5.5555E-03	4.1496E+01	118	130
JOB	11	13	5.5555E-03	4.1496E+01	131	143
JOB	12	13	5.5555E-03	4.1496E+01	144	156

JOBSTEPS IN JOBS

REQ	REQ	MSG	SP	RQ	OWN	STP	ASND					LVL	ASGND		
FCN	FIL	LENGTH	LV	LV	JOB	#	SITE	PROS	TIME	LINE	DLY	1	2	3	4
<99	<99	M.MMMESNN	<4	<4	<50	<99	1/25	M.MMMESNN	M.MMMESNN	M.MMMESNN		B	B	B	B



JSTP	1	1	0	1.0000E+02	2	1	1	1	1	0.0000E+00	1.0213E+00	1	0	0	0
JSTP	2	2	0	2.5000E+02	0	0	1	2	5	7.1603E-03	0.0000E+00	0	0	0	1
JSTP	3	3	1	1.0000E+03	0	0	1	3	5	5.7283E-02	1.0213E+01	0	0	0	1
JSTP	4	1	0	2.5000E+02	0	0	1	4	1	0.0000E+00	2.5532E+00	1	0	0	0
JSTP	5	2	0	2.5000E+02	0	0	1	5	5	7.1603E-03	0.0000E+00	0	0	0	1
JSTP	6	4	2	1.6000E+03	0	0	1	6	5	9.5471E-02	0.0000E+00	0	0	0	1
JSTP	7	5	3	1.6000E+03	2	4	1	7	5	5.7283E-01	0.0000E+00	0	0	0	1
JSTP	8	2	0	1.6000E+03	0	0	1	8	5	7.1603E-03	1.6340E+01	0	0	0	1
JSTP	9	1	0	2.5000E+01	0	0	1	9	1	0.0000E+00	2.5532E-01	1	0	0	0
JSTP	10	6	0	2.5000E+01	0	0	1	10	5	7.1603E-04	0.0000E+00	0	0	0	1
JSTP	11	7	3	1.0000E+03	2	4	1	11	5	9.5471E-02	0.0000E+00	0	0	0	1
JSTP	12	3	1	1.0000E+03	0	0	1	12	5	5.7283E-02	1.0213E+01	0	0	0	1
JSTP	13	1	0	0.0000E+00	2	1	1	13	1	0.0000E+00	0.0000E+00	1	0	0	0
JSTP	14	1	0	1.0000E+02	2	1	2	1	2	0.0000E+00	1.0213E+00	1	0	0	0
JSTP	15	2	0	2.5000E+02	0	0	2	2	5	7.1603E-03	0.0000E+00	0	0	0	1
JSTP	16	3	1	1.0000E+03	0	0	2	3	5	5.7283E-02	1.0213E+01	0	0	0	1
JSTP	17	1	0	2.5000E+02	0	0	2	4	2	0.0000E+00	2.5532E+00	1	0	0	0
JSTP	18	2	0	2.5000E+02	0	0	2	5	5	7.1603E-03	0.0000E+00	0	0	0	1
JSTP	19	4	2	1.6000E+03	0	0	2	6	5	9.5471E-02	0.0000E+00	0	0	0	1
JSTP	20	5	3	1.6000E+03	2	4	2	7	5	5.7283E-01	0.0000E+00	0	0	0	1
JSTP	21	2	0	1.6000E+03	0	0	2	8	5	7.1603E-03	1.6340E+01	0	0	0	1
JSTP	22	1	0	2.5000E+01	0	0	2	9	2	0.0000E+00	2.5532E-01	1	0	0	0
JSTP	23	6	0	2.5000E+01	0	0	2	10	5	7.1603E-04	0.0000E+00	0	0	0	1
JSTP	24	7	3	1.0000E+03	2	4	2	11	5	9.5471E-02	0.0000E+00	0	0	0	1
JSTP	25	3	1	1.0000E+03	0	0	2	12	5	5.7283E-02	1.0213E+01	0	0	0	1
JSTP	26	1	0	0.0000E+00	2	1	2	13	2	0.0000E+00	0.0000E+00	1	0	0	0
JSTP	27	1	0	1.0000E+02	2	1	3	1	3	0.0000E+00	1.0213E+00	1	0	0	0
JSTP	28	2	0	2.5000E+02	0	0	3	2	5	7.1603E-03	0.0000E+00	0	0	0	1
JSTP	29	3	1	1.0000E+03	0	0	3	3	5	5.7283E-02	1.0213E+01	0	0	0	1
JSTP	30	1	0	2.5000E+02	0	0	3	4	3	0.0000E+00	2.5532E+00	1	0	0	0
JSTP	31	2	0	2.5000E+02	0	0	3	5	5	7.1603E-03	0.0000E+00	0	0	0	1
JSTP	32	4	2	1.6000E+03	0	0	3	6	5	9.5471E-02	0.0000E+00	0	0	0	1
JSTP	33	5	3	1.6000E+03	2	4	3	7	5	5.7283E-01	0.0000E+00	0	0	0	1
JSTP	34	2	0	1.6000E+03	0	0	3	8	5	7.1603E-03	1.6340E+01	0	0	0	1
JSTP	35	1	0	2.5000E+01	0	0	3	9	3	0.0000E+00	2.5532E-01	1	0	0	0

JSTP	36	6	0	2.5000E+01	0	0	3	10	5	7.1603E-04	0.0000E+00	0	0	0	1
JSTP	37	7	3	1.0000E+03	2	4	3	11	5	9.5471E-02	0.0000E+00	0	0	0	1
JSTP	38	3	1	1.0000E+03	0	0	3	12	5	5.7283E-02	1.0213E+01	0	0	0	1
JSTP	39	1	0	0.0000E+00	2	1	3	13	3	0.0000E+00	0.0000E+00	1	0	0	0
JSTP	40	1	0	1.0000E+02	2	1	4	1	4	0.0000E+00	1.0213E+00	1	0	0	0
JSTP	41	2	0	2.5000E+02	0	0	4	2	5	7.1603E-03	0.0000E+00	0	0	0	1
JSTP	42	3	1	1.0000E+03	0	0	4	3	5	5.7283E-02	1.0213E+01	0	0	0	1
JSTP	43	1	0	2.5000E+02	0	0	4	4	4	0.0000E+00	2.5532E+00	1	0	0	0
JSTP	44	2	0	2.5000E+02	0	0	4	5	5	7.1603E-03	0.0000E+00	0	0	0	1
JSTP	45	4	2	1.6000E+03	0	0	4	6	5	9.5471E-02	0.0000E+00	0	0	0	1
JSTP	46	5	3	1.6000E+03	2	4	4	7	5	5.7283E-01	0.0000E+00	0	0	0	1
JSTP	47	2	0	1.6000E+03	0	0	4	8	5	7.1603E-03	1.6340E+01	0	0	0	1
JSTP	48	1	0	2.5000E+01	0	0	4	9	4	0.0000E+00	2.5532E-01	1	0	0	0
JSTP	49	6	0	2.5000E+01	0	0	4	10	5	7.1603E-04	0.0000E+00	0	0	0	1
JSTP	50	7	3	1.0000E+03	2	4	4	11	5	9.5471E-02	0.0000E+00	0	0	0	1
JSTP	51	3	1	1.0000E+03	0	0	4	12	5	5.7283E-02	1.0213E+01	0	0	0	1
JSTP	52	1	0	0.0000E+00	2	1	4	13	4	0.0000E+00	0.0000E+00	1	0	0	0
JSTP	53	1	0	1.0000E+02	2	1	5	1	6	0.0000E+00	1.0213E+00	1	0	0	0
JSTP	54	2	0	2.5000E+02	0	0	5	2	5	7.1603E-03	0.0000E+00	0	0	0	1
JSTP	55	3	1	1.0000E+03	0	0	5	3	5	5.7283E-02	1.0213E+01	0	0	0	1
JSTP	56	1	0	2.5000E+02	0	0	5	4	6	0.0000E+00	2.5532E+00	1	0	0	0
JSTP	57	2	0	2.5000E+02	0	0	5	5	5	7.1603E-03	0.0000E+00	0	0	0	1
JSTP	58	4	2	1.6000E+03	0	0	5	6	5	9.5471E-02	0.0000E+00	0	0	0	1
JSTP	59	5	3	1.6000E+03	2	4	5	7	5	5.7283E-01	0.0000E+00	0	0	0	1
JSTP	60	2	0	1.6000E+03	0	0	5	8	5	7.1603E-03	1.6340E+01	0	0	0	1
JSTP	61	1	0	2.5000E+01	0	0	5	9	6	0.0000E+00	2.5532E-01	1	0	0	0
JSTP	62	6	0	2.5000E+01	0	0	5	10	5	7.1603E-04	0.0000E+00	0	0	0	1
JSTP	63	7	3	1.0000E+03	2	4	5	11	5	9.5471E-02	0.0000E+00	0	0	0	1
JSTP	64	3	1	1.0000E+03	0	0	5	12	5	5.7283E-02	1.0213E+01	0	0	0	1
JSTP	65	1	0	0.0000E+00	2	1	5	13	6	0.0000E+00	0.0000E+00	1	0	0	0
JSTP	66	1	0	1.0000E+02	2	1	6	1	7	0.0000E+00	1.0213E+00	1	0	0	0
JSTP	67	2	0	2.5000E+02	0	0	6	2	5	7.1603E-03	0.0000E+00	0	0	0	1
JSTP	68	3	1	1.0000E+03	0	0	6	3	5	5.7283E-02	1.0213E+01	0	0	0	1
JSTP	69	1	0	2.5000E+02	0	0	6	4	7	0.0000E+00	2.5532E+00	1	0	0	0
JSTP	70	2	0	2.5000E+02	0	0	6	5	5	7.1603E-03	0.0000E+00	0	0	0	1

JSTP	71	4	2	1.6000E+03	0	0	6	6	5	9.5471E-02	0.0000E+00	0	0	0	1
JSTP	72	5	3	1.6000E+03	2	4	6	7	5	5.7283E-01	0.0000E+00	0	0	0	1
JSTP	73	2	0	1.6000E+03	0	0	6	8	5	7.1603E-03	1.6340E+01	0	0	0	1
JSTP	74	1	0	2.5000E+01	0	0	6	9	7	0.0000E+00	2.5532E-01	1	0	0	0
JSTP	75	6	0	2.5000E+01	0	0	6	10	5	7.1603E-04	0.0000E+00	0	0	0	1
JSTP	76	7	3	1.0000E+03	2	4	6	11	5	9.5471E-02	0.0000E+00	0	0	0	1
JSTP	77	3	1	1.0000E+03	0	0	6	12	5	5.7283E-02	1.0213E+01	0	0	0	1
JSTP	78	1	0	0.0000E+00	2	1	6	13	7	0.0000E+00	0.0000E+00	1	0	0	0
JSTP	79	1	0	1.0000E+02	2	1	7	1	8	0.0000E+00	1.0213E+00	1	0	0	0
JSTP	80	2	0	2.5000E+02	0	0	7	2	5	7.1603E-03	0.0000E+00	0	0	0	1
JSTP	81	3	1	1.0000E+03	0	0	7	3	5	5.7283E-02	1.0213E+01	0	0	0	1
JSTP	82	1	0	2.5000E+02	0	0	7	4	8	0.0000E+00	2.5532E+00	1	0	0	0
JSTP	83	2	0	2.5000E+02	0	0	7	5	5	7.1603E-03	0.0000E+00	0	0	0	1
JSTP	84	4	2	1.6000E+03	0	0	7	6	5	9.5471E-02	0.0000E+00	0	0	0	1
JSTP	85	5	3	1.6000E+03	2	4	7	7	5	5.7283E-01	0.0000E+00	0	0	0	1
JSTP	86	2	0	1.6000E+03	0	0	7	8	5	7.1603E-03	1.6340E+01	0	0	0	1
JSTP	87	1	0	2.5000E+01	0	0	7	9	8	0.0000E+00	2.5532E-01	1	0	0	0
JSTP	88	6	0	2.5000E+01	0	0	7	10	5	7.1603E-04	0.0000E+00	0	0	0	1
JSTP	89	7	3	1.0000E+03	2	4	7	11	5	9.5471E-02	0.0000E+00	0	0	0	1
JSTP	90	3	1	1.0000E+03	0	0	7	12	5	5.7283E-02	1.0213E+01	0	0	0	1
JSTP	91	1	0	0.0000E+00	2	1	7	13	8	0.0000E+00	0.0000E+00	1	0	0	0
JSTP	92	1	0	1.0000E+02	2	1	8	1	9	0.0000E+00	1.0213E+00	1	0	0	0
JSTP	93	2	0	2.5000E+02	0	0	8	2	5	7.1603E-03	0.0000E+00	0	0	0	1
JSTP	94	3	1	1.0000E+03	0	0	8	3	5	5.7283E-02	1.0213E+01	0	0	0	1
JSTP	95	1	0	2.5000E+02	0	0	8	4	9	0.0000E+00	2.5532E+00	1	0	0	0
JSTP	96	2	0	2.5000E+02	0	0	8	5	5	7.1603E-03	0.0000E+00	0	0	0	1
JSTP	97	4	2	1.6000E+03	0	0	8	6	5	9.5471E-02	0.0000E+00	0	0	0	1
JSTP	98	5	3	1.6000E+03	2	4	8	7	5	5.7283E-01	0.0000E+00	0	0	0	1
JSTP	99	2	0	1.6000E+03	0	0	8	8	5	7.1603E-03	1.6340E+01	0	0	0	1
JSTP	100	1	0	2.5000E+01	0	0	8	9	9	0.0000E+00	2.5532E-01	1	0	0	0
JSTP	101	6	0	2.5000E+01	0	0	8	10	5	7.1603E-04	0.0000E+00	0	0	0	1
JSTP	102	7	3	1.0000E+03	2	4	8	11	5	9.5471E-02	0.0000E+00	0	0	0	1
JSTP	103	3	1	1.0000E+03	0	0	8	12	5	5.7283E-02	1.0213E+01	0	0	0	1
JSTP	104	1	0	0.0000E+00	2	1	8	13	9	0.0000E+00	0.0000E+00	1	0	0	0
JSTP	105	1	0	1.0000E+02	2	1	9	1	10	0.0000E+00	1.0213E+00	1	0	0	0

JSTP 106	2	0	2.5000E+02	0	0	9	2	5	7.1603E-03	0.0000E+00	0	0	0	1
JSTP 107	3	1	1.0000E+03	0	0	9	3	5	5.7283E-02	1.0213E+01	0	0	0	1
JSTP 108	1	0	2.5000E+02	0	0	9	4	10	0.0000E+00	2.5532E+00	1	0	0	0
JSTP 109	2	0	2.5000E+02	0	0	9	5	5	7.1603E-03	0.0000E+00	0	0	0	1
JSTP 110	4	2	1.6000E+03	0	0	9	6	5	9.5471E-02	0.0000E+00	0	0	0	1
JSTP 111	5	3	1.6000E+03	2	4	9	7	5	5.7283E-01	0.0000E+00	0	0	0	1
JSTP 112	2	0	1.6000E+03	0	0	9	8	5	7.1603E-03	1.6340E+01	0	0	0	1
JSTP 113	1	0	2.5000E+01	0	0	9	9	10	0.0000E+00	2.5532E-01	1	0	0	0
JSTP 114	6	0	2.5000E+01	0	0	9	10	5	7.1603E-04	0.0000E+00	0	0	0	1
JSTP 115	7	3	1.0000E+03	2	4	9	11	5	9.5471E-02	0.0000E+00	0	0	0	1
JSTP 116	3	1	1.0000E+03	0	0	9	12	5	5.7283E-02	1.0213E+01	0	0	0	1
JSTP 117	1	0	0.0000E+00	2	1	9	13	10	0.0000E+00	0.0000E+00	1	0	0	0
JSTP 118	1	0	1.0000E+02	2	1	10	1	11	0.0000E+00	1.0213E+00	1	0	0	0
JSTP 119	2	0	2.5000E+02	0	0	10	2	5	7.1603E-03	0.0000E+00	0	0	0	1
JSTP 120	3	1	1.0000E+03	0	0	10	3	5	5.7283E-02	1.0213E+01	0	0	0	1
JSTP 121	1	0	2.5000E+02	0	0	10	4	11	0.0000E+00	2.5532E+00	1	0	0	0
JSTP 122	2	0	2.5000E+02	0	0	10	5	5	7.1603E-03	0.0000E+00	0	0	0	1
JSTP 123	4	2	1.6000E+03	0	0	10	6	5	9.5471E-02	0.0000E+00	0	0	0	1
JSTP 124	5	3	1.6000E+03	2	4	10	7	5	5.7283E-01	0.0000E+00	0	0	0	1
JSTP 125	2	0	1.6000E+03	0	0	10	8	5	7.1603E-03	1.6340E+01	0	0	0	1
JSTP 126	1	0	2.5000E+01	0	0	10	9	11	0.0000E+00	2.5532E-01	1	0	0	0
JSTP 127	6	0	2.5000E+01	0	0	10	10	5	7.1603E-04	0.0000E+00	0	0	0	1
JSTP 128	7	3	1.0000E+03	2	4	10	11	5	9.5471E-02	0.0000E+00	0	0	0	1
JSTP 129	3	1	1.0000E+03	0	0	10	12	5	5.7283E-02	1.0213E+01	0	0	0	1
JSTP 130	1	0	0.0000E+00	2	1	10	13	11	0.0000E+00	0.0000E+00	1	0	0	0
JSTP 131	1	0	1.0000E+02	2	1	11	1	12	0.0000E+00	1.0213E+00	1	0	0	0
JSTP 132	2	0	2.5000E+02	0	0	11	2	5	7.1603E-03	0.0000E+00	0	0	0	1
JSTP 133	3	1	1.0000E+03	0	0	11	3	5	5.7283E-02	1.0213E+01	0	0	0	1
JSTP 134	1	0	2.5000E+02	0	0	11	4	12	0.0000E+00	2.5532E+00	1	0	0	0
JSTP 135	2	0	2.5000E+02	0	0	11	5	5	7.1603E-03	0.0000E+00	0	0	0	1
JSTP 136	4	2	1.6000E+03	0	0	11	6	5	9.5471E-02	0.0000E+00	0	0	0	1
JSTP 137	5	3	1.6000E+03	2	4	11	7	5	5.7283E-01	0.0000E+00	0	0	0	1
JSTP 138	2	0	1.6000E+03	0	0	11	8	5	7.1603E-03	1.6340E+01	0	0	0	1
JSTP 139	1	0	2.5000E+01	0	0	11	9	12	0.0000E+00	2.5532E-01	1	0	0	0
JSTP 140	6	0	2.5000E+01	0	0	11	10	5	7.1603E-04	0.0000E+00	0	0	0	1

JSTP 141	7	3	1.0000E+03	2	4	11	11	5	9.5471E-02	0.0000E+00	0	0	0	1
JSTP 142	3	1	1.0000E+03	0	0	11	12	5	5.7283E-02	1.0213E+01	0	0	0	1
JSTP 143	1	0	0.0000E+00	2	1	11	13	12	0.0000E+00	0.0000E+00	1	0	0	0
JSTP 144	1	0	1.0000E+02	2	1	12	1	13	0.0000E+00	1.0213E+00	1	0	0	0
JSTP 145	2	0	2.5000E+02	0	0	12	2	5	7.1603E-03	0.0000E+00	0	0	0	1
JSTF 146	3	1	1.0000E+03	0	0	12	3	5	5.7283E-02	1.0213E+01	0	0	0	1
JSTP 147	1	0	2.5000E+02	0	0	12	4	13	0.0000E+00	2.5532E+00	1	0	0	0
JSTP 148	2	0	2.5000E+02	0	0	12	5	5	7.1603E-03	0.0000E+00	0	0	0	1
JSTP 149	4	2	1.6000E+03	0	0	12	6	5	9.5471E-02	0.0000E+00	0	0	0	1
JSTF 150	5	3	1.6000E+03	2	4	12	7	5	5.7283E-01	0.0000E+00	0	0	0	1
JSTP 151	2	0	1.6000E+03	0	0	12	8	5	7.1603E-03	1.6340E+01	0	0	0	1
JSTP 152	1	0	2.5000E+01	0	0	12	9	13	0.0000E+00	2.5532E-01	1	0	0	0
JSTP 153	6	0	2.5000E+01	0	0	12	10	5	7.1603E-04	0.0000E+00	0	0	0	1
JSTP 154	7	3	1.0000E+03	2	4	12	11	5	9.5471E-02	0.0000E+00	0	0	0	1
JSTP 155	3	1	1.0000E+03	0	0	12	12	5	5.7283E-02	1.0213E+01	0	0	0	1
JSTP 156	1	0	0.0000E+00	2	1	12	13	13	0.0000E+00	0.0000E+00	1	0	0	0

XVII. APPENDIX I:

INPUT DATA SET FOR THE DISTRIBUTED CONFIGURATION

FUNCTION SELECTION FOR THIS RUN

USE <1> TO INDICATE FUNCTION TO BE SELECTED

INITIALIZE	CALL	CALL	CALL	CALL	NEW NET
TO DEFAULTS	JSTOP	PTOS	OSASS	ANLYZ	CONFIG
<0 >	<1 >	<1 >	<1 >	<1 >	<0 >

UPBOUND-VALUES AND ALLOWABLE RANGES

NSITE	NLINE	NSTEP	NFCN	NFILE	NJOB	MLINE	MPROS	NLEVL
1/25	1/99	1/250	1/99	1/99	1/50	1/10	1/10	1/4
13	0	156	7	3	12	8	9	4

CTLPARM-VALUES AND ALLOWABLE RANGES

ISEL	XN	THRS	KCENT	TRAC	PLIST (1)	(2)	(3)	(4)
BIT	<9999	<9999	BIT	BIT	1/10	1/10	1/10	1/10
0	10	5	0	0	0	0	0	0

PROCESSOR COMPATABILITY MATRICES

PROS		1	2	3	4	5	6	7	8	9
CFILE 1	1	0	0	0	0	0	0	0	0	0
CFILE 2	2	0	0	0	0	0	0	0	0	0
CFILE 3	3	0	0	0	0	0	0	0	0	0

PROS		1	2	3	4	5	6	7	8	9
CFCN 1	1	0	1	1	1	0	0	0	0	0
CFCN 2	2	0	0	0	1	1	1	1	1	1
CFCN 3	3	0	0	0	1	1	1	1	1	1
CFCN 4	4	0	0	0	1	1	1	1	1	1
CFCN 5	5	0	0	0	0	0	0	0	1	1
CFCN 6	6	0	0	0	1	1	1	1	1	1
CFCN 7	7	0	0	0	0	0	0	0	1	1

SITE DESCRIPTORS

		STATUS	XDIST	YDIST	PROS	PROS LOAD	PROS UF
		0/2	<9999.9	<9999.9	1/10	M.MMMESNN	M.MMMESNN
SITE 1	1	2	10.0	10.0	0	0.0000E+00	0.0000E+00
SITE 2	2	2	70.0	70.0	0	0.0000E+00	0.0000E+00
SITE 3	3	2	10.0	70.0	0	0.0000E+00	0.0000E+00
SITE 4	4	2	70.0	10.0	0	0.0000E+00	0.0000E+00
SITE 5	5	2	40.0	40.0	0	0.0000E+00	0.0000E+00
SITE 6	6	2	10.0	10.0	0	0.0000E+00	0.0000E+00
SITE 7	7	2	70.0	70.0	0	0.0000E+00	0.0000E+00
SITE 8	8	2	10.0	70.0	0	0.0000E+00	0.0000E+00
SITE 9	9	2	70.0	10.0	0	0.0000E+00	0.0000E+00
SITE 10	10	2	10.0	10.0	0	0.0000E+00	0.0000E+00
SITE 11	11	2	70.0	70.0	0	0.0000E+00	0.0000E+00
SITE 12	12	2	10.0	70.0	0	0.0000E+00	0.0000E+00
SITE 13	13	2	70.0	10.0	0	0.0000E+00	0.0000E+00

LINKING LINE DESCRIPTORS

	INSITE	FINSITE	L TYPE	L LOAD	L UF
	0/25	0/25	0/10	M.MMMESNN	M.MMMESNN

CHARACTERISTICS OF AVAILABLE PROCESSORS

	LVL#	I RATE	PRMSR	SCNDSTR	MEDIA	CHAN RATE	PCOST
	1/10	M.MMMESNN	M.MMMESNN	M.MMMESNN	CODE	M.MMMESNN	<999999.99
PTYPE 1	1	0.0000E+00	0.0000E+00	0.0000E+00	0	3.0000E+01	20.00
PTYPE 2	1	0.0000E+00	0.0000E+00	0.0000E+00	0	2.4000E+02	30.00
PTYPE 3	2	5.3000E-01	1.0000E+00	0.0000E+00	0	1.0000E+06	34.70
PTYPE 4	2	7.6000E-01	3.2000E+01	1.5000E-01	1	2.0000E+06	80.00
PTYPE 5	2	5.7000E-01	3.2000E+01	5.1200E-01	1	1.6600E+06	200.00
PTYPE 6	3	9.8000E-01	6.4000E+01	5.0000E+00	1	2.0000E+06	550.00
PTYPE 7	3	1.0000E+01	1.2800E+02	2.8000E+01	4	1.5000E+06	2931.40



PTYPE	8	4	1.5000E+00	5.2400E+02	8.0000E+02	7	1.3000E+06	13103.40
PTYPE	9	4	2.0000E+01	4.0000E+03	1.6000E+03	7	1.5000E+06	84172.00

CHARACTERISTICS OF AVAILABLE LINES

		XMSN RATE	LFCOST	LVCST1	LVCST2	LVCST3	LVCST4	LVCST5
		M.MMMESNN	<9999.9	<99.99	<99.99	<99.99	<99.99	<99.99
LTYPE	1	3.0000E+01	73.0	3.30	2.31	1.65	1.16	0.83
LTYPE	2	1.2000E+02	81.0	3.30	2.31	1.65	1.16	0.83
LTYPE	3	2.4000E+02	153.0	3.30	2.31	1.65	1.16	0.83
LTYPE	4	4.8000E+02	213.0	3.30	2.31	1.65	1.16	0.83
LTYPE	5	9.6000E+02	433.0	3.30	2.31	1.65	1.16	0.83
LTYPE	6	1.9200E+03	850.0	3.00	3.00	3.00	3.00	3.00
LTYPE	7	5.0000E+03	850.0	6.00	6.00	6.00	6.00	6.00
LTYPE	8	2.3000E+04	1250.0	30.00	30.00	30.00	30.00	30.00

SOFTWARE FUNCTIONS

		I STEPS	PCNTSIZE	FCNFSIZE
		M.MMMESNN	M.MMMESNN	M.MMMESNN
FCNTYPE	1	0.0000E+00	0.0000E+00	0.0000E+00
FCNTYPE	2	7.5000E+03	0.0000E+00	8.0000E+00
FCNTYPE	3	6.0000E+04	8.0000E+00	8.0000E+00
FCNTYPE	4	1.0000E+05	1.6000E+01	2.0000E+00
FCNTYPE	5	6.0000E+05	1.2800E+02	6.4000E+01
FCNTYPE	6	7.5000E+02	0.0000E+00	8.0000E+00
FCNTYPE	7	1.0000E+05	1.2800E+02	6.4000E+01

SOFTWARE FILES

		MEDIUM	FILE SIZE
		1/10	M.MMMESNN
FILETYPE	1	1	5.0000E-02
FILETYPE	2	1	3.0000E-02
FILETYPE	3	2	4.0000E+02

**JOBS IN WORKLOAD**

	#JSTPS	ARVL RATE	RSPNS TIME	FRSTEP#	LASTEP#
	1/99	M.MMMESNN	M.MMMESNN	1/250	1/250
JOB 1	13	5.5555E-03	0.0000E+00	1	13
JOB 2	13	5.5555E-03	0.0000E+00	14	26
JOB 3	13	5.5555E-03	0.0000E+00	27	39
JOB 4	13	5.5555E-03	0.0000E+00	40	52
JOB 5	13	5.5555E-03	0.0000E+00	53	65
JOB 6	13	5.5555E-03	0.0000E+00	66	78
JOB 7	13	5.5555E-03	0.0000E+00	79	91
JOB 8	13	5.5555E-03	0.0000E+00	92	104
JOB 9	13	5.5555E-03	0.0000E+00	105	117
JOB 10	13	5.5555E-03	0.0000E+00	118	130
JOB 11	13	5.5555E-03	0.0000E+00	131	143
JOB 12	13	5.5555E-03	0.0000E+00	144	156

**JOBSTEPS IN JOBS**

	REQ	REQ	MSG	SP	FQ	OWN	STP	ASND		PROS	TIME	LINE	DLY	LVL ASGND			
	FCN	FIL	LENGTH	LV	LV	JOB	#	SITE		M.MMMESNN	M.MMMESNN			1	2	3	4
	<99	<99	M.MMMESNN	<4	<4	<50	<99	1/25		M.MMMESNN	M.MMMESNN			B	B	B	B
JSTP 1	1	0	1.0000E+02	2	2	1	1	1	0.0000E+00	0.0000E+00	0.0000E+00			0	0	0	0
JSTP 2	2	0	2.5000E+02	0	0	1	2	0	0.0000E+00	0.0000E+00	0.0000E+00			0	0	0	0
JSTP 3	3	1	1.0000E+03	0	0	1	3	0	0.0000E+00	0.0000E+00	0.0000E+00			0	0	0	0
JSTP 4	1	0	2.5000E+02	2	2	1	4	0	0.0000E+00	0.0000E+00	0.0000E+00			0	0	0	0
JSTP 5	2	0	2.5000E+02	0	0	1	5	0	0.0000E+00	0.0000E+00	0.0000E+00			0	0	0	0
JSTP 6	4	2	1.6000E+03	0	0	1	6	0	0.0000E+00	0.0000E+00	0.0000E+00			0	0	0	0
JSTP 7	5	3	1.6000E+03	2	4	1	7	5	0.0000E+00	0.0000E+00	0.0000E+00			0	0	0	0
JSTP 8	2	0	1.6000E+03	0	0	1	8	0	0.0000E+00	0.0000E+00	0.0000E+00			0	0	0	0
JSTP 9	1	0	2.5000E+01	2	2	1	9	0	0.0000E+00	0.0000E+00	0.0000E+00			0	0	0	0
JSTP 10	6	0	2.5000E+01	0	0	1	10	0	0.0000E+00	0.0000E+00	0.0000E+00			0	0	0	0
JSTP 11	7	3	1.0000E+03	2	4	1	11	5	0.0000E+00	0.0000E+00	0.0000E+00			0	0	0	0
JSTP 12	3	1	1.0000E+03	0	0	1	12	0	0.0000E+00	0.0000E+00	0.0000E+00			0	0	0	0



JSTP	48	1	0	2.5000E+01	2	2	4	9	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	49	6	0	2.5000E+01	0	0	4	10	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	50	7	3	1.0000E+03	2	4	4	11	5	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	51	3	1	1.0000E+03	0	0	4	12	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	52	1	0	0.0000E+00	2	2	4	13	4	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	53	1	0	1.0000E+02	2	1	5	1	6	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	54	2	0	2.5000E+02	0	0	5	2	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	55	3	1	1.0000E+03	0	0	5	3	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	56	1	0	2.5000E+02	0	0	5	4	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	57	2	0	2.5000E+02	0	0	5	5	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	58	4	2	1.6000E+03	0	0	5	6	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	59	5	3	1.6000E+03	2	4	5	7	5	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	60	2	0	1.6000E+03	0	0	5	8	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	61	1	0	2.5000E+01	0	0	5	9	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	62	6	0	2.5000E+01	0	0	5	10	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	63	7	3	1.0000E+03	2	4	5	11	5	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	64	3	1	1.0000E+03	0	0	5	12	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	65	1	0	0.0000E+00	2	1	5	13	6	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	66	1	0	1.0000E+02	2	1	6	1	7	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	67	2	0	2.5000E+02	0	0	6	2	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	68	3	1	1.0000E+03	0	0	6	3	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	69	1	0	2.5000E+02	0	0	6	4	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	70	2	0	2.5000E+02	0	0	6	5	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	71	4	2	1.6000E+03	0	0	6	6	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	72	5	3	1.6000E+03	2	4	6	7	5	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	73	2	0	1.6000E+03	0	0	6	8	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	74	1	0	2.5000E+01	0	0	6	9	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	75	6	0	2.5000E+01	0	0	6	10	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	76	7	3	1.0000E+03	2	4	6	11	5	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	77	3	1	1.0000E+03	0	0	6	12	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	78	1	0	0.0000E+00	2	1	6	13	7	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	79	1	0	1.0000E+02	2	1	7	1	8	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	80	2	0	2.5000E+02	0	0	7	2	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	81	3	1	1.0000E+03	0	0	7	3	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP	82	1	0	2.5000E+02	0	0	7	4	0	0.0000E+00	0.0000E+00	0	0	0	0



JSTP 118	1	0	1.0000E+02	2	1	10	1	11	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 119	2	0	2.5000E+02	0	0	10	2	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 120	3	1	1.0000E+03	0	0	10	3	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 121	1	0	2.5000E+02	0	0	10	4	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 122	2	0	2.5000E+02	0	0	10	5	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 123	4	2	1.6000E+03	0	0	10	6	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 124	5	3	1.6000E+03	2	4	10	7	5	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 125	2	0	1.6000E+03	0	0	10	8	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 126	1	0	2.5000E+01	0	0	10	9	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 127	6	0	2.5000E+01	0	0	10	10	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 128	7	3	1.0000E+03	2	4	10	11	5	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 129	3	1	1.0000E+03	0	0	10	12	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 130	1	0	0.0000E+00	2	1	10	13	11	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 131	1	0	1.0000E+02	2	1	11	1	12	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 132	2	0	2.5000E+02	0	0	11	2	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 133	3	1	1.0000E+03	0	0	11	3	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 134	1	0	2.5000E+02	0	0	11	4	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 135	2	0	2.5000E+02	0	0	11	5	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 136	4	2	1.6000E+03	0	0	11	6	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 137	5	3	1.6000E+03	2	4	11	7	5	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 138	2	0	1.6000E+03	0	0	11	8	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 139	1	0	2.5000E+01	0	0	11	9	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 140	6	0	2.5000E+01	0	0	11	10	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 141	7	3	1.0000E+03	2	4	11	11	5	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 142	3	1	1.0000E+03	0	0	11	12	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 143	1	0	0.0000E+00	2	1	11	13	12	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 144	1	0	1.0000E+02	2	1	12	1	13	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 145	2	0	2.5000E+02	0	0	12	2	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 146	3	1	1.0000E+03	0	0	12	3	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 147	1	0	2.5000E+02	0	0	12	4	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 148	2	0	2.5000E+02	0	0	12	5	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 149	4	2	1.6000E+03	0	0	12	6	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 150	5	3	1.6000E+03	2	4	12	7	5	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 151	2	0	1.6000E+03	0	0	12	8	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 152	1	0	2.5000E+01	0	0	12	9	0	0.0000E+00	0.0000E+00	0	0	0	0

JSTP 153	6	0	2.5000E+01	0	0	12	10	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 154	7	3	1.0000E+03	2	4	12	11	5	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 155	3	1	1.0000E+03	0	0	12	12	0	0.0000E+00	0.0000E+00	0	0	0	0
JSTP 156	1	0	0.0000E+00	2	1	12	13	13	0.0000E+00	0.0000E+00	0	0	0	0

XVIII. APPENDIX J:

EXECUTION OUTPUT FOR THE DISTRIBUTED CONFIGURATION



END OF JOBSTEP TO PROCESSOR ASSIGNMENT  
END OF PROCESSOR ASSIGNMENT  
END OF OPTIMUM LOCATING  
START NETWORK ANALYSIS

PROCESSOR 4 SITE 1

MIPS= 4.0541E-03 UTILIZED= 5.3344E-03  
AVRAM= 2.6056E+01 UTILIZED= 8.1425E-01  
FIXED SCNDSTR= 0.0000E+00 UTILIZED= 0.0000E+00

PROCESSOR 4 SITE 2

MIPS= 4.0541E-03 UTILIZED= 5.3344E-03  
AVRAM= 2.6056E+01 UTILIZED= 8.1425E-01  
FIXED SCNDSTR= 0.0000E+00 UTILIZED= 0.0000E+00

PROCESSOR 4 SITE 3

MIPS= 4.0541E-03 UTILIZED= 5.3344E-03  
AVRAM= 2.6056E+01 UTILIZED= 8.1425E-01  
FIXED SCNDSTR= 0.0000E+00 UTILIZED= 0.0000E+00

PROCESSOR 4 SITE 4

MIPS= 4.0541E-03 UTILIZED= 5.3344E-03  
AVRAM= 2.6056E+01 UTILIZED= 8.1425E-01  
FIXED SCNDSTR= 0.0000E+00 UTILIZED= 0.0000E+00

PROCESSOR 8 SITE 5

MIPS= 4.6666E-02 UTILIZED= 3.1111E-02  
AVRAM= 1.3198E+02 UTILIZED= 2.5187E-01  
FIXED SCNDSTR= 4.0000E+02 UTILIZED= 5.0000E-01

PROCESSOR 2 SITE 6

MIPS= 0.0000E+00 UTILIZED= 0.0000E+00  
AVRAM= 0.0000E+00 UTILIZED= 0.0000E+00  
FIXED SCNDSTR= 0.0000E+00 UTILIZED= 0.0000E+00

PROCESSOR 2 SITE 7

MIPS= 0.0000E+00 UTILIZED= 0.0000E+00  
AVRAM= 0.0000E+00 UTILIZED= 0.0000E+00  
FIXED SCNDSTR= 0.0000E+00 UTILIZED= 0.0000E+00

PROCESSOR 2 SITE 8

MIPS= 0.0000E+00 UTILIZED= 0.0000E+00

AVRAM= 0.0000E+00 UTILIZED= 0.0000E+00  
 FIXED SCNDSTR= 0.0000E+00 UTILIZED= 0.0000E+00  
 PROCESSOR 2 SITE 9  
 MIPS= 0.0000E+00 UTILIZED= 0.0000E+00  
 AVRAM= 0.0000E+00 UTILIZED= 0.0000E+00  
 FIXED SCNDSTR= 0.0000E+00 UTILIZED= 0.0000E+00  
 PROCESSOR 2 SITE 10  
 MIPS= 0.0000E+00 UTILIZED= 0.0000E+00  
 AVRAM= 0.0000E+00 UTILIZED= 0.0000E+00  
 FIXED SCNDSTR= 0.0000E+00 UTILIZED= 0.0000E+00  
 PROCESSOR 2 SITE 11  
 MIPS= 0.0000E+00 UTILIZED= 0.0000E+00  
 AVRAM= 0.0000E+00 UTILIZED= 0.0000E+00  
 FIXED SCNDSTR= 0.0000E+00 UTILIZED= 0.0000E+00  
 PROCESSOR 2 SITE 12  
 MIPS= 0.0000E+00 UTILIZED= 0.0000E+00  
 AVRAM= 0.0000E+00 UTILIZED= 0.0000E+00  
 FIXED SCNDSTR= 0.0000E+00 UTILIZED= 0.0000E+00  
 PROCESSOR 2 SITE 13  
 MIPS= 0.0000E+00 UTILIZED= 0.0000E+00  
 AVRAM= 0.0000E+00 UTILIZED= 0.0000E+00  
 FIXED SCNDSTR= 0.0000E+00 UTILIZED= 0.0000E+00

INSITE 1	,PINSITE 5	,LINE TYPE IS 3	,LOAD= 7.0416E+01	,UTILIZE= 2.9340E-01
INSITE 2	,PINSITE 5	,LINE TYPE IS 3	,LOAD= 7.0416E+01	,UTILIZE= 2.9340E-01
INSITE 3	,PINSITE 5	,LINE TYPE IS 3	,LOAD= 7.0416E+01	,UTILIZE= 2.9340E-01
INSITE 4	,PINSITE 5	,LINE TYPE IS 3	,LOAD= 7.0416E+01	,UTILIZE= 2.9340E-01
INSITE 6	,PINSITE 1	,LINE TYPE IS 0	,LOAD= 2.2083E+01	,UTILIZE= 0.0000E+00
INSITE 7	,PINSITE 2	,LINE TYPE IS 0	,LOAD= 2.2083E+01	,UTILIZE= 0.0000E+00
INSITE 8	,PINSITE 3	,LINE TYPE IS 0	,LOAD= 2.2083E+01	,UTILIZE= 0.0000E+00
INSITE 9	,PINSITE 4	,LINE TYPE IS 0	,LOAD= 2.2083E+01	,UTILIZE= 0.0000E+00
INSITE 10	,PINSITE 1	,LINE TYPE IS 0	,LOAD= 2.2083E+01	,UTILIZE= 0.0000E+00
INSITE 11	,PINSITE 2	,LINE TYPE IS 0	,LOAD= 2.2083E+01	,UTILIZE= 0.0000E+00
INSITE 12	,PINSITE 3	,LINE TYPE IS 0	,LOAD= 2.2083E+01	,UTILIZE= 0.0000E+00
INSITE 13	,PINSITE 4	,LINE TYPE IS 0	,LOAD= 2.2083E+01	,UTILIZE= 0.0000E+00

```

JOB= 1
STEP= 1 ,SITE= 1
  PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 0.0000E+00
STEP= 2 ,SITE= 1
  PROS TIME= 5.3129E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 5.3129E-02
STEP= 3 ,SITE= 1
  PROS TIME= 4.2503E-01 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 4.7816E-01
STEP= 4 ,SITE= 1
  PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 4.7816E-01
STEP= 5 ,SITE= 1
  PROS TIME= 5.3129E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 5.3129E-01
STEP= 6 ,SITE= 1
  PROS TIME= 7.0838E-01 ,LINE DELAY= 9.4348E+00 ,JOB TIME ACCUM= 1.0675E+01
STEP= 7 ,SITE= 5
  PROS TIME= 5.3467E-01 ,LINE DELAY= 9.4348E+00 ,JOB TIME ACCUM= 2.0644E+01
STEP= 8 ,SITE= 1
  PROS TIME= 5.3129E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 2.0697E+01
STEP= 9 ,SITE= 1
  PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 2.0697E+01
STEP= 10 ,SITE= 1
  PROS TIME= 5.3129E-03 ,LINE DELAY= 1.4742E-01 ,JOB TIME ACCUM= 2.0850E+01
STEP= 11 ,SITE= 5
  PROS TIME= 8.9112E-02 ,LINE DELAY= 5.8968E+00 ,JOB TIME ACCUM= 2.6836E+01
STEP= 12 ,SITE= 1
  PROS TIME= 4.2503E-01 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 2.7261E+01
STEP= 13 ,SITE= 1
  PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 2.7261E+01
JOB= 2
STEP= 1 ,SITE= 2
  PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 0.0000E+00
STEP= 2 ,SITE= 2
  PROS TIME= 5.3129E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 5.3129E-02
STEP= 3 ,SITE= 2
  PROS TIME= 4.2503E-01 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 4.7816E-01
STEP= 4 ,SITE= 2

```

PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 4.7816E-01  
 STEP= 5 ,SITE= 2  
 PROS TIME= 5.3129E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 5.3129E-01  
 STEP= 6 ,SITE= 2  
 PROS TIME= 7.0838E-01 ,LINE DELAY= 9.4348E+00 ,JOB TIME ACCUM= 1.0675E+01  
 STEP= 7 ,SITE= 5  
 PROS TIME= 5.3467E-01 ,LINE DELAY= 9.4348E+00 ,JOB TIME ACCUM= 2.0644E+01  
 STEP= 8 ,SITE= 2  
 PROS TIME= 5.3129E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 2.0697E+01  
 STEP= 9 ,SITE= 2  
 PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 2.0697E+01  
 STEP= 10 ,SITE= 2  
 PROS TIME= 5.3129E-03 ,LINE DELAY= 1.4742E-01 ,JOB TIME ACCUM= 2.0850E+01  
 STEP= 11 ,SITE= 5  
 PROS TIME= 8.9112E-02 ,LINE DELAY= 5.8968E+00 ,JOB TIME ACCUM= 2.6836E+01  
 STEP= 12 ,SITE= 2  
 PROS TIME= 4.2503E-01 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 2.7261E+01  
 STEP= 13 ,SITE= 2  
 PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 2.7261E+01  
 JOB= 3  
 STEP= 1 ,SITE= 3  
 PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 0.0000E+00  
 STEP= 2 ,SITE= 3  
 PROS TIME= 5.3129E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 5.3129E-02  
 STEP= 3 ,SITE= 3  
 PROS TIME= 4.2503E-01 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 4.7816E-01  
 STEP= 4 ,SITE= 3  
 PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 4.7816E-01  
 STEP= 5 ,SITE= 3  
 PROS TIME= 5.3129E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 5.3129E-01  
 STEP= 6 ,SITE= 3  
 PROS TIME= 7.0838E-01 ,LINE DELAY= 9.4348E+00 ,JOB TIME ACCUM= 1.0675E+01  
 STEP= 7 ,SITE= 5  
 PROS TIME= 5.3467E-01 ,LINE DELAY= 9.4348E+00 ,JOB TIME ACCUM= 2.0644E+01  
 STEP= 8 ,SITE= 3

PROS TIME= 5.3129E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 2.0697E+01  
 STEP= 9 ,SITE= 3  
 PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 2.0697E+01  
 STEP= 10 ,SITE= 3  
 PROS TIME= 5.3129E-03 ,LINE DELAY= 1.4742E-01 ,JOB TIME ACCUM= 2.0850E+01  
 STEP= 11 ,SITE= 5  
 PROS TIME= 8.9112E-02 ,LINE DELAY= 5.8968E+00 ,JOB TIME ACCUM= 2.6836E+01  
 STEP= 12 ,SITE= 3  
 PROS TIME= 4.2503E-01 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 2.7261E+01  
 STEP= 13 ,SITE= 3  
 PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 2.7261E+01  
 JOB= 4  
 STEP= 1 ,SITE= 4  
 PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 0.0000E+00  
 STEP= 2 ,SITE= 4  
 PROS TIME= 5.3129E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 5.3129E-02  
 STEP= 3 ,SITE= 4  
 PROS TIME= 4.2503E-01 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 4.7816E-01  
 STEP= 4 ,SITE= 4  
 PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 4.7816E-01  
 STEP= 5 ,SITE= 4  
 PROS TIME= 5.3129E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 5.3129E-01  
 STEP= 6 ,SITE= 4  
 PROS TIME= 7.0838E-01 ,LINE DELAY= 9.4348E+00 ,JOB TIME ACCUM= 1.0675E+01  
 STEP= 7 ,SITE= 5  
 PROS TIME= 5.3467E-01 ,LINE DELAY= 9.4348E+00 ,JOB TIME ACCUM= 2.0644E+01  
 STEP= 8 ,SITE= 4  
 PROS TIME= 5.3129E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 2.0697E+01  
 STEP= 9 ,SITE= 4  
 PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 2.0697E+01  
 STEP= 10 ,SITE= 4  
 PROS TIME= 5.3129E-03 ,LINE DELAY= 1.4742E-01 ,JOB TIME ACCUM= 2.0850E+01  
 STEP= 11 ,SITE= 5  
 PROS TIME= 8.9112E-02 ,LINE DELAY= 5.8968E+00 ,JOB TIME ACCUM= 2.6836E+01  
 STEP= 12 ,SITE= 4

PROS TIME= 4.2503E-01 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 2.7261E+01  
 STEP= 13 ,SITE= 4  
 PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 2.7261E+01  
 JOB= 5  
 STEP= 1 ,SITE= 6  
 PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 0.0000E+00  
 STEP= 2 ,SITE= 1  
 PROS TIME= 5.3129E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 5.3129E-02  
 STEP= 3 ,SITE= 1  
 PROS TIME= 4.2503E-01 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 4.7816E-01  
 STEP= 4 ,SITE= 6  
 PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 4.7816E-01  
 STEP= 5 ,SITE= 1  
 PROS TIME= 5.3129E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 5.3129E-01  
 STEP= 6 ,SITE= 1  
 PROS TIME= 7.0838E-01 ,LINE DELAY= 9.4348E+00 ,JOB TIME ACCUM= 1.0675E+01  
 STEP= 7 ,SITE= 5  
 PROS TIME= 5.3467E-01 ,LINE DELAY= 9.4348E+00 ,JOB TIME ACCUM= 2.0644E+01  
 STEP= 8 ,SITE= 1  
 PROS TIME= 5.3129E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 2.0697E+01  
 STEP= 9 ,SITE= 6  
 PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 2.0697E+01  
 STEP= 10 ,SITE= 1  
 PROS TIME= 5.3129E-03 ,LINE DELAY= 1.4742E-01 ,JOB TIME ACCUM= 2.0850E+01  
 STEP= 11 ,SITE= 5  
 PROS TIME= 8.9112E-02 ,LINE DELAY= 5.8968E+00 ,JOB TIME ACCUM= 2.6836E+01  
 STEP= 12 ,SITE= 1  
 PROS TIME= 4.2503E-01 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 2.7261E+01  
 STEP= 13 ,SITE= 6  
 PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 2.7261E+01  
 JOB= 6  
 STEP= 1 ,SITE= 7  
 PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 0.0000E+00  
 STEP= 2 ,SITE= 2  
 PROS TIME= 5.3129E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 5.3129E-02

STEP= 3 ,SITE= 2	PROS TIME= 4.2503E-01 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 4.7816E-01
STEP= 4 ,SITE= 7	PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 4.7816E-01
STEP= 5 ,SITE= 2	PROS TIME= 5.3129E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 5.3129E-01
STEP= 6 ,SITE= 2	PROS TIME= 7.0838E-01 ,LINE DELAY= 9.4348E+00 ,JOB TIME ACCUM= 1.0675E+01
STEP= 7 ,SITE= 5	PROS TIME= 5.3467E-01 ,LINE DELAY= 9.4348E+00 ,JOB TIME ACCUM= 2.0644E+01
STEP= 8 ,SITE= 2	PROS TIME= 5.3129E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 2.0697E+01
STEP= 9 ,SITE= 7	PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 2.0697E+01
STEP= 10 ,SITE= 2	PROS TIME= 5.3129E-03 ,LINE DELAY= 1.4742E-01 ,JOB TIME ACCUM= 2.0850E+01
STEP= 11 ,SITE= 5	PROS TIME= 8.9112E-02 ,LINE DELAY= 5.8968E+00 ,JOB TIME ACCUM= 2.6836E+01
STEP= 12 ,SITE= 2	PROS TIME= 4.2503E-01 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 2.7261E+01
STEP= 13 ,SITE= 7	PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 2.7261E+01
JOB= 7	
STEP= 1 ,SITE= 8	PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 0.0000E+00
STEP= 2 ,SITE= 3	PROS TIME= 5.3129E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 5.3129E-02
STEP= 3 ,SITE= 3	PROS TIME= 4.2503E-01 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 4.7816E-01
STEP= 4 ,SITE= 8	PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 4.7816E-01
STEP= 5 ,SITE= 3	PROS TIME= 5.3129E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 5.3129E-01
STEP= 6 ,SITE= 3	PROS TIME= 7.0838E-01 ,LINE DELAY= 9.4348E+00 ,JOB TIME ACCUM= 1.0675E+01

STEP= 7 ,SITE= 5	PROS TIME= 5.3467E-01 ,LINE DELAY= 9.4348E+00 ,JOB TIME ACCUM= 2.0644E+01
STEP= 8 ,SITE= 3	PROS TIME= 5.3129E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 2.0697E+01
STEP= 9 ,SITE= 8	PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 2.0697E+01
STEP= 10 ,SITE= 3	PROS TIME= 5.3129E-03 ,LINE DELAY= 1.4742E-01 ,JOB TIME ACCUM= 2.0850E+01
STEP= 11 ,SITE= 5	PROS TIME= 8.9112E-02 ,LINE DELAY= 5.8968E+00 ,JOB TIME ACCUM= 2.6836E+01
STEP= 12 ,SITE= 3	PROS TIME= 4.2503E-01 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 2.7261E+01
STEP= 13 ,SITE= 8	PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 2.7261E+01
JOB= 8	
STEP= 1 ,SITE= 9	PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 0.0000E+00
STEP= 2 ,SITE= 4	PROS TIME= 5.3129E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 5.3129E-02
STEP= 3 ,SITE= 4	PROS TIME= 4.2503E-01 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 4.7816E-01
STEP= 4 ,SITE= 9	PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 4.7816E-01
STEP= 5 ,SITE= 4	PROS TIME= 5.3129E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 5.3129E-01
STEP= 6 ,SITE= 4	PROS TIME= 7.0838E-01 ,LINE DELAY= 9.4348E+00 ,JOB TIME ACCUM= 1.0675E+01
STEP= 7 ,SITE= 5	PROS TIME= 5.3467E-01 ,LINE DELAY= 9.4348E+00 ,JOB TIME ACCUM= 2.0644E+01
STEP= 8 ,SITE= 4	PROS TIME= 5.3129E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 2.0697E+01
STEP= 9 ,SITE= 9	PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 2.0697E+01
STEP= 10 ,SITE= 4	PROS TIME= 5.3129E-03 ,LINE DELAY= 1.4742E-01 ,JOB TIME ACCUM= 2.0850E+01



STEP= 11 ,SITE= 5  
 PROS TIME= 8.9112E-02 ,LINE DELAY= 5.8968E+00 ,JOB TIME ACCUM= 2.6836E+01  
 STEP= 12 ,SITE= 4  
 PROS TIME= 4.2503E-01 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 2.7261E+01  
 STEP= 13 ,SITE= 9  
 PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 2.7261E+01  
 JOB= 9  
 STEP= 1 ,SITE= 10  
 PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 0.0000E+00  
 STEP= 2 ,SITE= 1  
 PROS TIME= 5.3129E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 5.3129E-02  
 STEP= 3 ,SITE= 1  
 PROS TIME= 4.2503E-01 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 4.7816E-01  
 STEP= 4 ,SITE= 10  
 PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 4.7816E-01  
 STEP= 5 ,SITE= 1  
 PROS TIME= 5.3129E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 5.3129E-01  
 STEP= 6 ,SITE= 1  
 PROS TIME= 7.0838E-01 ,LINE DELAY= 9.4348E+00 ,JOB TIME ACCUM= 1.0675E+01  
 STEP= 7 ,SITE= 5  
 PROS TIME= 5.3467E-01 ,LINE DELAY= 9.4348E+00 ,JOB TIME ACCUM= 2.0644E+01  
 STEP= 8 ,SITE= 1  
 PROS TIME= 5.3129E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 2.0697E+01  
 STEP= 9 ,SITE= 10  
 PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 2.0697E+01  
 STEP= 10 ,SITE= 1  
 PROS TIME= 5.3129E-03 ,LINE DELAY= 1.4742E-01 ,JOB TIME ACCUM= 2.0850E+01  
 STEP= 11 ,SITE= 5  
 PROS TIME= 8.9112E-02 ,LINE DELAY= 5.8968E+00 ,JOB TIME ACCUM= 2.6836E+01  
 STEP= 12 ,SITE= 1  
 PROS TIME= 4.2503E-01 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 2.7261E+01  
 STEP= 13 ,SITE= 10  
 PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 2.7261E+01  
 JOB= 10  
 STEP= 1 ,SITE= 11

PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 0.0000E+00  
 STEP= 2 ,SITE= 2  
 PROS TIME= 5.3129E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 5.3129E-02  
 STEP= 3 ,SITE= 2  
 PROS TIME= 4.2503E-01 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 4.7816E-01  
 STEP= 4 ,SITE= 11  
 PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 4.7816E-01  
 STEP= 5 ,SITE= 2  
 PROS TIME= 5.3129E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 5.3129E-01  
 STEP= 6 ,SITE= 2  
 PROS TIME= 7.0838E-01 ,LINE DELAY= 9.4348E+00 ,JOB TIME ACCUM= 1.0675E+01  
 STEP= 7 ,SITE= 5  
 PROS TIME= 5.3467E-01 ,LINE DELAY= 9.4348E+00 ,JOB TIME ACCUM= 2.0644E+01  
 STEP= 8 ,SITE= 2  
 PROS TIME= 5.3129E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 2.0697E+01  
 STEP= 9 ,SITE= 11  
 PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 2.0697E+01  
 STEP= 10 ,SITE= 2  
 PROS TIME= 5.3129E-03 ,LINE DELAY= 1.4742E-01 ,JOB TIME ACCUM= 2.0850E+01  
 STEP= 11 ,SITE= 5  
 PROS TIME= 8.9112E-02 ,LINE DELAY= 5.8968E+00 ,JOB TIME ACCUM= 2.6836E+01  
 STEP= 12 ,SITE= 2  
 PROS TIME= 4.2503E-01 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 2.7261E+01  
 STEP= 13 ,SITE= 11  
 PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 2.7261E+01  
 JOB= 11  
 STEP= 1 ,SITE= 12  
 PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 0.0000E+00  
 STEP= 2 ,SITE= 3  
 PROS TIME= 5.3129E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 5.3129E-02  
 STEP= 3 ,SITE= 3  
 PROS TIME= 4.2503E-01 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 4.7816E-01  
 STEP= 4 ,SITE= 12  
 PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 4.7816E-01  
 STEP= 5 ,SITE= 3

PROS TIME= 5.3129E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 5.3129E-01  
 STEP= 6 ,SITE= 3  
 PROS TIME= 7.0838E-01 ,LINE DELAY= 9.4348E+00 ,JOB TIME ACCUM= 1.0675E+01  
 STEP= 7 ,SITE= 5  
 PROS TIME= 5.3467E-01 ,LINE DELAY= 9.4348E+00 ,JOB TIME ACCUM= 2.0644E+01  
 STEP= 8 ,SITE= 3  
 PROS TIME= 5.3129E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 2.0697E+01  
 STEP= 9 ,SITE= 12  
 PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 2.0697E+01  
 STEP= 10 ,SITE= 3  
 PROS TIME= 5.3129E-03 ,LINE DELAY= 1.4742E-01 ,JOB TIME ACCUM= 2.0850E+01  
 STEP= 11 ,SITE= 5  
 PROS TIME= 8.9112E-02 ,LINE DELAY= 5.8968E+00 ,JOB TIME ACCUM= 2.6836E+01  
 STEP= 12 ,SITE= 3  
 PROS TIME= 4.2503E-01 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 2.7261E+01  
 STEP= 13 ,SITE= 12  
 PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 2.7261E+01  
 JOB= 12  
 STEP= 1 ,SITE= 13  
 PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 0.0000E+00  
 STEP= 2 ,SITE= 4  
 PROS TIME= 5.3129E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 5.3129E-02  
 STEP= 3 ,SITE= 4  
 PROS TIME= 4.2503E-01 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 4.7816E-01  
 STEP= 4 ,SITE= 13  
 PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 4.7816E-01  
 STEP= 5 ,SITE= 4  
 PROS TIME= 5.3129E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 5.3129E-01  
 STEP= 6 ,SITE= 4  
 PROS TIME= 7.0838E-01 ,LINE DELAY= 9.4348E+00 ,JOB TIME ACCUM= 1.0675E+01  
 STEP= 7 ,SITE= 5  
 PROS TIME= 5.3467E-01 ,LINE DELAY= 9.4348E+00 ,JOB TIME ACCUM= 2.0644E+01  
 STEP= 8 ,SITE= 4  
 PROS TIME= 5.3129E-02 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 2.0697E+01  
 STEP= 9 ,SITE= 13

PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 2.0697E+01  
 STEP= 10 ,SITE= 4  
 PROS TIME= 5.3129E-03 ,LINE DELAY= 1.4742E-01 ,JOB TIME ACCUM= 2.0850E+01  
 STEP= 11 ,SITE= 5  
 PROS TIME= 8.9112E-02 ,LINE DELAY= 5.8968E+00 ,JOB TIME ACCUM= 2.6836E+01  
 STEP= 12 ,SITE= 4  
 PROS TIME= 4.2503E-01 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 2.7261E+01  
 STEP= 13 ,SITE= 13  
 PROS TIME= 0.0000E+00 ,LINE DELAY= 0.0000E+00 ,JOB TIME ACCUM= 2.7261E+01

LINE 1 COST 250.94  
 LINE 2 COST 250.94  
 LINE 3 COST 250.94  
 LINE 4 COST 250.94

SITE 1 PROCESSOR COST 80.00  
 SITE 2 PROCESSOR COST 80.00  
 SITE 3 PROCESSOR COST 80.00  
 SITE 4 PROCESSOR COST 80.00  
 SITE 5 PROCESSOR COST 13103.40  
 SITE 6 PROCESSOR COST 30.00  
 SITE 7 PROCESSOR COST 30.00  
 SITE 8 PROCESSOR COST 30.00  
 SITE 9 PROCESSOR COST 30.00  
 SITE 10 PROCESSOR COST 30.00  
 SITE 11 PROCESSOR COST 30.00  
 SITE 12 PROCESSOR COST 30.00  
 SITE 13 PROCESSOR COST 30.00

TOTAL LINE COST= 1003.76 ,TOTAL PROS COST= 13663.40 ,TOTAL COST= 14667  
 END OF NETWORK ANALYSIS

XIX. APPENDIX K:

OUTPUT DATA SET FOR THE DISTRIBUTED CONFIGURATION

FUNCTION SELECTION FOR THIS RUN

USE <1> TO INDICATE FUNCTION TO BE SELECTED

INITIALIZE	CALL	CALL	CALL	CALL	NEW NET
TO DEFAULTS	JSTOP	PTOS	OSASS	ANLYZ	CONFIG
<0 >	<1 >	<1 >	<1 >	<1 >	<0 >

UPBOUND-VALUES AND ALLOWABLE RANGES

NSITE	NLINE	NSTEP	NFCN	NFILE	NJOB	NLINE	MPROS	NLEVL
1/25	1/99	1/250	1/99	1/99	1/50	1/10	1/10	1/4
13	12	156	7	3	12	8	9	4

CTLPARM-VALUES AND ALLOWABLE RANGES

ISEL	XN	THRS	KCENT	TRAC	PLIST(1)	(2)	(3)	(4)
BIT	<9999	<9999	BIT	BIT	1/10	1/10	1/10	1/10
0	10	5	0	0	2	4	6	8

PROCESSOR COMPATABILITY MATRICES

PROS		1	2	3	4	5	6	7	8	9
CFILE	1	0	0	0	1	1	1	1	1	1
CFILE	2	0	0	0	1	1	1	1	1	1
CFILE	3	0	0	0	0	0	0	0	1	1

PROS		1	2	3	4	5	6	7	8	9
CFCN	1	0	1	1	1	0	0	0	0	0
CFCN	2	0	0	0	1	1	1	1	1	1
CFCN	3	0	0	0	1	1	1	1	1	1
CFCN	4	0	0	0	1	1	1	1	1	1
CFCN	5	0	0	0	0	0	0	0	1	1
CFCN	6	0	0	0	1	1	1	1	1	1
CFCN	7	0	0	0	0	0	0	0	1	1

SITE DESCRIPTORS

	STATUS	XDIST	YDIST	PROS	PROS LOAD	PROS UF
	0/2	<9999.9	<9999.9	1/10	M.MMMESNN	M.MMMESNN
SITE 1	2	10.0	10.0	4	4.0541E-03	5.3344E-03
SITE 2	2	70.0	70.0	4	4.0541E-03	5.3344E-03
SITE 3	2	10.0	70.0	4	4.0541E-03	5.3344E-03
SITE 4	2	70.0	10.0	4	4.0541E-03	5.3344E-03
SITE 5	2	40.0	40.0	8	4.6666E-02	3.1111E-02
SITE 6	2	10.0	10.0	2	0.0000E+00	0.0000E+00
SITE 7	2	70.0	70.0	2	0.0000E+00	0.0000E+00
SITE 8	2	10.0	70.0	2	0.0000E+00	0.0000E+00
SITE 9	2	70.0	10.0	2	0.0000E+00	0.0000E+00
SITE 10	2	10.0	10.0	2	0.0000E+00	0.0000E+00
SITE 11	2	70.0	70.0	2	0.0000E+00	0.0000E+00
SITE 12	2	10.0	70.0	2	0.0000E+00	0.0000E+00
SITE 13	2	70.0	10.0	2	0.0000E+00	0.0000E+00

LINKING LINE DESCRIPTORS

	INSITE	FINSITE	L TYPE	L LOAD	L UF
	0/25	0/25	0/10	M.MMMESNN	M.MMMESNN
LINE 1	1	5	3	7.0416E+01	2.9340E-01
LINE 2	2	5	3	7.0416E+01	2.9340E-01
LINE 3	3	5	3	7.0416E+01	2.9340E-01
LINE 4	4	5	3	7.0416E+01	2.9340E-01
LINE 5	6	1	0	2.2083E+01	0.0000E+00
LINE 6	7	2	0	2.2083E+01	0.0000E+00
LINE 7	8	3	0	2.2083E+01	0.0000E+00
LINE 8	9	4	0	2.2083E+01	0.0000E+00
LINE 9	10	1	0	2.2083E+01	0.0000E+00
LINE 10	11	2	0	2.2083E+01	0.0000E+00
LINE 11	12	3	0	2.2083E+01	0.0000E+00
LINE 12	13	4	0	2.2083E+01	0.0000E+00

CHARACTERISTICS OF AVAILABLE PROCESSORS

	LVL#	I RATE	PRMSTR	SCNDSTR	MEDIA	CHAN RATE	PCOST
	1/10	M.MMMESNN	M.MMMESNN	M.MMMESNN	CODE	M.MMMESNN	<999999.99
P	1	0.0000E+00	0.0000E+00	0.0000E+00	0	3.0000E+01	20.00
P	2	0.0000E+00	0.0000E+00	0.0000E+00	0	2.4000E+02	30.00
P	3	5.3000E-01	1.0000E+00	0.0000E+00	0	1.0000E+06	34.70
P	4	7.6000E-01	3.2000E+01	1.5000E-01	1	2.0000E+06	80.00
P	5	5.7000E-01	3.2000E+01	5.1200E-01	1	1.6600E+06	200.00
P	6	9.8000E-01	6.4000E+01	5.0000E+00	1	2.0000E+06	550.00
P	7	1.0000E+01	1.2800E+02	2.8000E+01	4	1.5000E+06	2931.40
P	8	1.5000E+00	5.2400E+02	8.0000E+02	7	1.3000E+06	13103.40
P	9	2.0000E+01	4.0000E+03	1.6000E+03	7	1.5000E+06	84172.00

CHARACTERISTICS OF AVAILABLE LINES

	XMSN RATE	LFCOST	LVCST1	LVCST2	LVCST3	LVCST4	LVCST5	
	M.MMMESNN	<9999.9	<99.99	<99.99	<99.99	<99.99	<99.99	
L	1	3.0000E+01	73.0	3.30	2.31	1.65	1.16	0.83
L	2	1.2000E+02	81.0	3.30	2.31	1.65	1.16	0.83
L	3	2.4000E+02	153.0	3.30	2.31	1.65	1.16	0.83
L	4	4.8000E+02	213.0	3.30	2.31	1.65	1.16	0.83
L	5	9.6000E+02	433.0	3.30	2.31	1.65	1.16	0.83
L	6	1.9200E+03	850.0	3.00	3.00	3.00	3.00	3.00
L	7	5.0000E+03	850.0	6.00	6.00	6.00	6.00	6.00
L	8	2.3000E+04	1250.0	30.00	30.00	30.00	30.00	30.00

SOFTWARE FUNCTIONS

	I STEPS	FCNTSIZE	FCNFSIZE
	M.MMMESNN	M.MMMESNN	M.MMMESNN
FC	1	0.0000E+00	0.0000E+00
FC	2	7.5000E+03	8.0000E+00
FC	3	6.0000E+04	8.0000E+00



FCNTYPE	4	1.0000E+05	1.6000E+01	2.0000E+00
FCNTYPE	5	6.0000E+05	1.2800E+02	6.4000E+01
FCNTYPE	6	7.5000E+02	0.0000E+00	8.0000E+00
FCNTYPE	7	1.0000E+05	1.2800E+02	6.4000E+01

SOFTWARE FILES

	MEDIUM	FILE SIZE
	1/10	M.MMMESNN
FILETYPE 1	1	5.0000E-02
FILETYPE 2	1	3.0000E-02
FILETYPE 3	2	4.0000E+02

JOBS IN WORKLOAD

	#JSTPS	ARVL RATE	RSPNS TIME	FRSTEP#	LASTEP#
	1/99	M.MMMESNN	M.MMMESNN	1/250	1/250
JOB 1	13	5.5555E-03	2.7261E+01	1	13
JOB 2	13	5.5555E-03	2.7261E+01	14	26
JOB 3	13	5.5555E-03	2.7261E+01	27	39
JOB 4	13	5.5555E-03	2.7261E+01	40	52
JOB 5	13	5.5555E-03	2.7261E+01	53	65
JOB 6	13	5.5555E-03	2.7261E+01	66	78
JOB 7	13	5.5555E-03	2.7261E+01	79	91
JOB 8	13	5.5555E-03	2.7261E+01	92	104
JOB 9	13	5.5555E-03	2.7261E+01	105	117
JOB 10	13	5.5555E-03	2.7261E+01	118	130
JOB 11	13	5.5555E-03	2.7261E+01	131	143
JOB 12	13	5.5555E-03	2.7261E+01	144	156

JOBSTEPS IN JOES

REQ	REQ	MSG	SP	RQ	OWN	STP	ASND		LVL	ASGND					
FCN	FIL	LENGTH	LV	LV	JOB	#	SITE	PROS	TIME	LINE	DLY	1	2	3	4
<99	<99	M.MMMESNN	<4	<4	<50	<99	1/25	M.MMMESNN	M.MMMESNN			B	B	B	B

JSTP	1	1	0	1.0000E+02	2	2	1	1	1	0.0000E+00	0.0000E+00	0	1	0	0
JSTP	2	2	0	2.5000E+02	0	0	1	2	1	5.3129E-02	0.0000E+00	0	1	0	0
JSTP	3	3	1	1.0000E+03	0	0	1	3	1	4.2503E-01	0.0000E+00	0	1	0	0
JSTP	4	1	0	2.5000E+02	2	2	1	4	1	0.0000E+00	0.0000E+00	0	1	0	0
JSTP	5	2	0	2.5000E+02	0	0	1	5	1	5.3129E-02	0.0000E+00	0	1	0	0
JSTP	6	4	2	1.6000E+03	0	0	1	6	1	7.0838E-01	9.4348E+00	0	1	0	0
JSTP	7	5	3	1.6000E+03	2	4	1	7	5	5.3467E-01	9.4348E+00	0	0	0	1
JSTP	8	2	0	1.6000E+03	0	0	1	8	1	5.3129E-02	0.0000E+00	0	1	0	0
JSTP	9	1	0	2.5000E+01	2	2	1	9	1	0.0000E+00	0.0000E+00	0	1	0	0
JSTP	10	6	0	2.5000E+01	0	0	1	10	1	5.3129E-03	1.4742E-01	0	1	0	0
JSTP	11	7	3	1.0000E+03	2	4	1	11	5	8.9112E-02	5.8968E+00	0	0	0	1
JSTP	12	3	1	1.0000E+03	0	0	1	12	1	4.2503E-01	0.0000E+00	0	1	0	0
JSTP	13	1	0	0.0000E+00	2	2	1	13	1	0.0000E+00	0.0000E+00	0	1	0	0
JSTP	14	1	0	1.0000E+02	2	2	2	1	2	0.0000E+00	0.0000E+00	0	1	0	0
JSTP	15	2	0	2.5000E+02	0	0	2	2	2	5.3129E-02	0.0000E+00	0	1	0	0
JSTP	16	3	1	1.0000E+03	0	0	2	3	2	4.2503E-01	0.0000E+00	0	1	0	0
JSTP	17	1	0	2.5000E+02	2	2	2	4	2	0.0000E+00	0.0000E+00	0	1	0	0
JSTP	18	2	0	2.5000E+02	0	0	2	5	2	5.3129E-02	0.0000E+00	0	1	0	0
JSTP	19	4	2	1.6000E+03	0	0	2	6	2	7.0838E-01	9.4348E+00	0	1	0	0
JSTP	20	5	3	1.6000E+03	2	4	2	7	5	5.3467E-01	9.4348E+00	0	0	0	1
JSTP	21	2	0	1.6000E+03	0	0	2	8	2	5.3129E-02	0.0000E+00	0	1	0	0
JSTP	22	1	0	2.5000E+01	2	2	2	9	2	0.0000E+00	0.0000E+00	0	1	0	0
JSTP	23	6	0	2.5000E+01	0	0	2	10	2	5.3129E-03	1.4742E-01	0	1	0	0
JSTP	24	7	3	1.0000E+03	2	4	2	11	5	8.9112E-02	5.8968E+00	0	0	0	1
JSTP	25	3	1	1.0000E+03	0	0	2	12	2	4.2503E-01	0.0000E+00	0	1	0	0
JSTP	26	1	0	0.0000E+00	2	2	2	13	2	0.0000E+00	0.0000E+00	0	1	0	0
JSTP	27	1	0	1.0000E+02	2	2	3	1	3	0.0000E+00	0.0000E+00	0	1	0	0
JSTP	28	2	0	2.5000E+02	0	0	3	2	3	5.3129E-02	0.0000E+00	0	1	0	0
JSTP	29	3	1	1.0000E+03	0	0	3	3	3	4.2503E-01	0.0000E+00	0	1	0	0
JSTP	30	1	0	2.5000E+02	2	2	3	4	3	0.0000E+00	0.0000E+00	0	1	0	0
JSTP	31	2	0	2.5000E+02	0	0	3	5	3	5.3129E-02	0.0000E+00	0	1	0	0
JSTP	32	4	2	1.6000E+03	0	0	3	6	3	7.0838E-01	9.4348E+00	0	1	0	0
JSTP	33	5	3	1.6000E+03	2	4	3	7	5	5.3467E-01	9.4348E+00	0	0	0	1
JSTP	34	2	0	1.6000E+03	0	0	3	8	3	5.3129E-02	0.0000E+00	0	1	0	0
JSTP	35	1	0	2.5000E+01	2	2	3	9	3	0.0000E+00	0.0000E+00	0	1	0	0

JSTP	36	6	0	2.5000E+01	0	0	3	10	3	5.3129E-03	1.4742E-01	0	1	0	0
JSTP	37	7	3	1.0000E+03	2	4	3	11	5	8.9112E-02	5.8968E+00	0	0	0	1
JSTP	38	3	1	1.0000E+03	0	0	3	12	3	4.2503E-01	0.0000E+00	0	1	0	0
JSTP	39	1	0	0.0000E+00	2	2	3	13	3	0.0000E+00	0.0000E+00	0	1	0	0
JSTP	40	1	0	1.0000E+02	2	2	4	1	4	0.0000E+00	0.0000E+00	0	1	0	0
JSTP	41	2	0	2.5000E+02	0	0	4	2	4	5.3129E-02	0.0000E+00	0	1	0	0
JSTP	42	3	1	1.0000E+03	0	0	4	3	4	4.2503E-01	0.0000E+00	0	1	0	0
JSTP	43	1	0	2.5000E+02	2	2	4	4	4	0.0000E+00	0.0000E+00	0	1	0	0
JSTP	44	2	0	2.5000E+02	0	0	4	5	4	5.3129E-02	0.0000E+00	0	1	0	0
JSTP	45	4	2	1.6000E+03	0	0	4	6	4	7.0838E-01	9.4348E+00	0	1	0	0
JSTP	46	5	3	1.6000E+03	2	4	4	7	5	5.3467E-01	9.4348E+00	0	0	0	1
JSTP	47	2	0	1.6000E+03	0	0	4	8	4	5.3129E-02	0.0000E+00	0	1	0	0
JSTP	48	1	0	2.5000E+01	2	2	4	9	4	0.0000E+00	0.0000E+00	0	1	0	0
JSTP	49	6	0	2.5000E+01	0	0	4	10	4	5.3129E-03	1.4742E-01	0	1	0	0
JSTP	50	7	3	1.0000E+03	2	4	4	11	5	8.9112E-02	5.8968E+00	0	0	0	1
JSTP	51	3	1	1.0000E+03	0	0	4	12	4	4.2503E-01	0.0000E+00	0	1	0	0
JSTP	52	1	0	0.0000E+00	2	2	4	13	4	0.0000E+00	0.0000E+00	0	1	0	0
JSTP	53	1	0	1.0000E+02	2	1	5	1	6	0.0000E+00	0.0000E+00	1	0	0	0
JSTP	54	2	0	2.5000E+02	0	0	5	2	1	5.3129E-02	0.0000E+00	0	1	0	0
JSTP	55	3	1	1.0000E+03	0	0	5	3	1	4.2503E-01	0.0000E+00	0	1	0	0
JSTP	56	1	0	2.5000E+02	0	0	5	4	6	0.0000E+00	0.0000E+00	1	0	0	0
JSTP	57	2	0	2.5000E+02	0	0	5	5	1	5.3129E-02	0.0000E+00	0	1	0	0
JSTP	58	4	2	1.6000E+03	0	0	5	6	1	7.0838E-01	9.4348E+00	0	1	0	0
JSTP	59	5	3	1.6000E+03	2	4	5	7	5	5.3467E-01	9.4348E+00	0	0	0	1
JSTP	60	2	0	1.6000E+03	0	0	5	8	1	5.3129E-02	0.0000E+00	0	1	0	0
JSTP	61	1	0	2.5000E+01	0	0	5	9	6	0.0000E+00	0.0000E+00	1	0	0	0
JSTP	62	6	0	2.5000E+01	0	0	5	10	1	5.3129E-03	1.4742E-01	0	1	0	0
JSTP	63	7	3	1.0000E+03	2	4	5	11	5	8.9112E-02	5.8968E+00	0	0	0	1
JSTP	64	3	1	1.0000E+03	0	0	5	12	1	4.2503E-01	0.0000E+00	0	1	0	0
JSTP	65	1	0	0.0000E+00	2	1	5	13	6	0.0000E+00	0.0000E+00	1	0	0	0
JSTP	66	1	0	1.0000E+02	2	1	6	1	7	0.0000E+00	0.0000E+00	1	0	0	0
JSTP	67	2	0	2.5000E+02	0	0	6	2	2	5.3129E-02	0.0000E+00	0	1	0	0
JSTP	68	3	1	1.0000E+03	0	0	6	3	2	4.2503E-01	0.0000E+00	0	1	0	0
JSTP	69	1	0	2.5000E+02	0	0	6	4	7	0.0000E+00	0.0000E+00	1	0	0	0
JSTP	70	2	0	2.5000E+02	0	0	6	5	2	5.3129E-02	0.0000E+00	0	1	0	0

JSTP	71	4	2	1.6000E+03	0	0	6	6	2	7.0838E-01	9.4348E+00	0	1	0	0
JSTP	72	5	3	1.6000E+03	2	4	6	7	5	5.3467E-01	9.4348E+00	0	0	0	1
JSTP	73	2	0	1.6000E+03	0	0	6	8	2	5.3129E-02	0.0000E+00	0	1	0	0
JSTP	74	1	0	2.5000E+01	0	0	6	9	7	0.0000E+00	0.0000E+00	1	0	0	0
JSTP	75	6	0	2.5000E+01	0	0	6	10	2	5.3129E-03	1.4742E-01	0	1	0	0
JSTP	76	7	3	1.0000E+03	2	4	6	11	5	8.9112E-02	5.8968E+00	0	0	0	1
JSTP	77	3	1	1.0000E+03	0	0	6	12	2	4.2503E-01	0.0000E+00	0	1	0	0
JSTP	78	1	0	0.0000E+00	2	1	6	13	7	0.0000E+00	0.0000E+00	1	0	0	0
JSTP	79	1	0	1.0000E+02	2	1	7	1	8	0.0000E+00	0.0000E+00	1	0	0	0
JSTP	80	2	0	2.5000E+02	0	0	7	2	3	5.3129E-02	0.0000E+00	0	1	0	0
JSTP	81	3	1	1.0000E+03	0	0	7	3	3	4.2503E-01	0.0000E+00	0	1	0	0
JSTP	82	1	0	2.5000E+02	0	0	7	4	8	0.0000E+00	0.0000E+00	1	0	0	0
JSTP	83	2	0	2.5000E+02	0	0	7	5	3	5.3129E-02	0.0000E+00	0	1	0	0
JSTP	84	4	2	1.6000E+03	0	0	7	6	3	7.0838E-01	9.4348E+00	0	1	0	0
JSTP	85	5	3	1.6000E+03	2	4	7	7	5	5.3467E-01	9.4348E+00	0	0	0	1
JSTP	86	2	0	1.6000E+03	0	0	7	8	3	5.3129E-02	0.0000E+00	0	1	0	0
JSTP	87	1	0	2.5000E+01	0	0	7	9	8	0.0000E+00	0.0000E+00	1	0	0	0
JSTP	88	6	0	2.5000E+01	0	0	7	10	3	5.3129E-03	1.4742E-01	0	1	0	0
JSTP	89	7	3	1.0000E+03	2	4	7	11	5	8.9112E-02	5.8968E+00	0	0	0	1
JSTP	90	3	1	1.0000E+03	0	0	7	12	3	4.2503E-01	0.0000E+00	0	1	0	0
JSTP	91	1	0	0.0000E+00	2	1	7	13	8	0.0000E+00	0.0000E+00	1	0	0	0
JSTP	92	1	0	1.0000E+02	2	1	8	1	9	0.0000E+00	0.0000E+00	1	0	0	0
JSTP	93	2	0	2.5000E+02	0	0	8	2	4	5.3129E-02	0.0000E+00	0	1	0	0
JSTP	94	3	1	1.0000E+03	0	0	8	3	4	4.2503E-01	0.0000E+00	0	1	0	0
JSTP	95	1	0	2.5000E+02	0	0	8	4	9	0.0000E+00	0.0000E+00	1	0	0	0
JSTP	96	2	0	2.5000E+02	0	0	8	5	4	5.3129E-02	0.0000E+00	0	1	0	0
JSTP	97	4	2	1.6000E+03	0	0	8	6	4	7.0838E-01	9.4348E+00	0	1	0	0
JSTP	98	5	3	1.6000E+03	2	4	8	7	5	5.3467E-01	9.4348E+00	0	0	0	1
JSTP	99	2	0	1.6000E+03	0	0	8	8	4	5.3129E-02	0.0000E+00	0	1	0	0
JSTP	100	1	0	2.5000E+01	0	0	8	9	9	0.0000E+00	0.0000E+00	1	0	0	0
JSTP	101	6	0	2.5000E+01	0	0	8	10	4	5.3129E-03	1.4742E-01	0	1	0	0
JSTP	102	7	3	1.0000E+03	2	4	8	11	5	8.9112E-02	5.8968E+00	0	0	0	1
JSTP	103	3	1	1.0000E+03	0	0	8	12	4	4.2503E-01	0.0000E+00	0	1	0	0
JSTP	104	1	0	0.0000E+00	2	1	8	13	9	0.0000E+00	0.0000E+00	1	0	0	0
JSTP	105	1	0	1.0000E+02	2	1	9	1	10	0.0000E+00	0.0000E+00	1	0	0	0

JSTP 106	2	0	2.5000E+02	0	0	9	2	1	5.3129E-02	0.0000E+00	0	1	0	0
JSTP 107	3	1	1.0000E+03	0	0	9	3	1	4.2503E-01	0.0000E+00	0	1	0	0
JSTP 108	1	0	2.5000E+02	0	0	9	4	10	0.0000E+00	0.0000E+00	1	0	0	0
JSTP 109	2	0	2.5000E+02	0	0	9	5	1	5.3129E-02	0.0000E+00	0	1	0	0
JSTP 110	4	2	1.6000E+03	0	0	9	6	1	7.0838E-01	9.4348E+00	0	1	0	0
JSTP 111	5	3	1.6000E+03	2	4	9	7	5	5.3467E-01	9.4348E+00	0	0	0	1
JSTP 112	2	0	1.6000E+03	0	0	9	8	1	5.3129E-02	0.0000E+00	0	1	0	0
JSTP 113	1	0	2.5000E+01	0	0	9	9	10	0.0000E+00	0.0000E+00	1	0	0	0
JSTP 114	6	0	2.5000E+01	0	0	9	10	1	5.3129E-03	1.4742E-01	0	1	0	0
JSTP 115	7	3	1.0000E+03	2	4	9	11	5	8.9112E-02	5.8968E+00	0	0	0	1
JSTP 116	3	1	1.0000E+03	0	0	9	12	1	4.2503E-01	0.0000E+00	0	1	0	0
JSTP 117	1	0	0.0000E+00	2	1	9	13	10	0.0000E+00	0.0000E+00	1	0	0	0
JSTP 118	1	0	1.0000E+02	2	1	10	1	11	0.0000E+00	0.0000E+00	1	0	0	0
JSTP 119	2	0	2.5000E+02	0	0	10	2	2	5.3129E-02	0.0000E+00	0	1	0	0
JSTP 120	3	1	1.0000E+03	0	0	10	3	2	4.2503E-01	0.0000E+00	0	1	0	0
JSTP 121	1	0	2.5000E+02	0	0	10	4	11	0.0000E+00	0.0000E+00	1	0	0	0
JSTP 122	2	0	2.5000E+02	0	0	10	5	2	5.3129E-02	0.0000E+00	0	1	0	0
JSTP 123	4	2	1.6000E+03	0	0	10	6	2	7.0838E-01	9.4348E+00	0	1	0	0
JSTP 124	5	3	1.6000E+03	2	4	10	7	5	5.3467E-01	9.4348E+00	0	0	0	1
JSTP 125	2	0	1.6000E+03	0	0	10	8	2	5.3129E-02	0.0000E+00	0	1	0	0
JSTP 126	1	0	2.5000E+01	0	0	10	9	11	0.0000E+00	0.0000E+00	1	0	0	0
JSTP 127	6	0	2.5000E+01	0	0	10	10	2	5.3129E-03	1.4742E-01	0	1	0	0
JSTP 128	7	3	1.0000E+03	2	4	10	11	5	8.9112E-02	5.8968E+00	0	0	0	1
JSTP 129	3	1	1.0000E+03	0	0	10	12	2	4.2503E-01	0.0000E+00	0	1	0	0
JSTP 130	1	0	0.0000E+00	2	1	10	13	11	0.0000E+00	0.0000E+00	1	0	0	0
JSTP 131	1	0	1.0000E+02	2	1	11	1	12	0.0000E+00	0.0000E+00	1	0	0	0
JSTP 132	2	0	2.5000E+02	0	0	11	2	3	5.3129E-02	0.0000E+00	0	1	0	0
JSTP 133	3	1	1.0000E+03	0	0	11	3	3	4.2503E-01	0.0000E+00	0	1	0	0
JSTP 134	1	0	2.5000E+02	0	0	11	4	12	0.0000E+00	0.0000E+00	1	0	0	0
JSTP 135	2	0	2.5000E+02	0	0	11	5	3	5.3129E-02	0.0000E+00	0	1	0	0
JSTP 136	4	2	1.6000E+03	0	0	11	6	3	7.0838E-01	9.4348E+00	0	1	0	0
JSTP 137	5	3	1.6000E+03	2	4	11	7	5	5.3467E-01	9.4348E+00	0	0	0	1
JSTP 138	2	0	1.6000E+03	0	0	11	8	3	5.3129E-02	0.0000E+00	0	1	0	0
JSTP 139	1	0	2.5000E+01	0	0	11	9	12	0.0000E+00	0.0000E+00	1	0	0	0
JSTP 140	6	0	2.5000E+01	0	0	11	10	3	5.3129E-03	1.4742E-01	0	1	0	0

JSTP 141	7	3	1.0000E+03	2	4	11	11	5	8.9112E-02	5.8968E+00	0	0	0	1
JSTP 142	3	1	1.0000E+03	0	0	11	12	3	4.2503E-01	0.0000E+00	0	1	0	0
JSTP 143	1	0	0.0000E+00	2	1	11	13	12	0.0000E+00	0.0000E+00	1	0	0	0
JSTP 144	1	0	1.0000E+02	2	1	12	1	13	0.0000E+00	0.0000E+00	1	0	0	0
JSTP 145	2	0	2.5000E+02	0	0	12	2	4	5.3129E-02	0.0000E+00	0	1	0	0
JSTP 146	3	1	1.0000E+03	0	0	12	3	4	4.2503E-01	0.0000E+00	0	1	0	0
JSTP 147	1	0	2.5000E+02	0	0	12	4	13	0.0000E+00	0.0000E+00	1	0	0	0
JSTP 148	2	0	2.5000E+02	0	0	12	5	4	5.3129E-02	0.0000E+00	0	1	0	0
JSTP 149	4	2	1.6000E+03	0	0	12	6	4	7.0838E-01	9.4348E+00	0	1	0	0
JSTP 150	5	3	1.6000E+03	2	4	12	7	5	5.3467E-01	9.4348E+00	0	0	0	1
JSTP 151	2	0	1.6000E+03	0	0	12	8	4	5.3129E-02	0.0000E+00	0	1	0	0
JSTP 152	1	0	2.5000E+01	0	0	12	9	13	0.0000E+00	0.0000E+00	1	0	0	0
JSTP 153	6	0	2.5000E+01	0	0	12	10	4	5.3129E-03	1.4742E-01	0	1	0	0
JSTP 154	7	3	1.0000E+03	2	4	12	11	5	8.9112E-02	5.8968E+00	0	0	0	1
JSTP 155	3	1	1.0000E+03	0	0	12	12	4	4.2503E-01	0.0000E+00	0	1	0	0
JSTP 156	1	0	0.0000E+00	2	1	12	13	13	0.0000E+00	0.0000E+00	1	0	0	0