FEASIBILITY STUDY OF NETWORKS

Progress Report
for Period July 15, 1975-July 14, 1976

Elizabeth J. Campbell
Muriel Kannel
Kalman Saffran

Laboratory for Nuclear Science
Massachusetts Institute of Technology
Cambridge, Massachusetts

April 30, 1976

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product or process disclosed or represents that its use would not infringe privately owned rights.

NOTICE

PORIONS OF THIS REPORT ARE ILLIGIBLE. It has been reproduced from the best available copy to permit the broadest possible availability.

NOTICE
This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

Prepared for
THE U.S. ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION
UNDER CONTRACT E(11-1) 2487

Jy

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

ABSTRACT

The Laboratory for Nuclear Science has continued its study into the feasibility of networks, changing the focus of the investigation from the possible ways of collaborating with the MIT Information Processing Center in implementing a connection to the ARPA net to a wider consideration of the implications of computer resource sharing via networks. This report describes experiments performed on the ARPA net in collaboration with the Lawrence Berkeley Laboratory and experiments in more conventional forms of remote job entry to that site and others in ERDA. It discusses the use of the Multics mail as set up for ERDA, comparing it to other mail systems. Plans for future experiments in the ARPA net and for investigation of networks other than the ARPA net are proposed.

The conclusion was reached that resource sharing via networks is both feasible and desirable provided certain interface problems are solved. Recommendations are made for more investigation into the human factors involved in network use, methods of providing consulting help to users, and determination of the most cost effective use of remote resource sharing.

For the Laboratory for Nuclear Science, we have concluded that access to other ERDA computers via networks will meet some of our computing needs now and more in the future. However, remote resource sharing can be used only to supplement the on site computer, not replace it. At the present time, the best method of accessing the ARPA net for our users is via the MIT Multics System. However, as a permanent solution, a mini-host located in our computer center may provide more capability. In the next year, we plan to work out methods of attacking the problems of interfacing to the ARPA net and to use it in a limited manner for actual production work.

MASTER

TABLE OF CONTENTS

	Page
I. INTRODUCTION	1
1. Current Status of the Study	
2. Education of the LNS Staff	
3. Administrative Functions on Multics by ERDA	
4. Goals realized by the Study	
II. MAIL SYSTEMS	6
1. Characteristics of an Ideal System	
2. Multics Mail	
3. MSG System on Tenex	
4. TDA System under Development by CCA	
III. NETWORK EXPERIMENT	17
1. Remote Job Entry	
2. Experiments on Using the ARPA net to Access LBL	
3. Development of a General Scenario for Use of the LBL Facilities	
IV. INVESTIGATION INTO NETWORKS OTHER THAN THE ARPA NET	28
V. CONCLUSIONS AND RECOMMENDATIONS	30
VI. RESEARCHERS' ALLOCATION OF TIME	33
APPENDIX A. DRAFT OF USER MEMO 1 INTRODUCTION TO MAIL FACILITY ON MULTICS	
APPENDIX B. INITIAL DRAFT OF LABORATORY FOR NUCLEAR SCIENCE USERS GUIDE TO REMOTE BATCH SUBMITTAL TO LAWRENCE BERKELEY LABORATORY VIA MULTICS AND THE ARPA NET	

FEASIBILITY STUDY OF NETWORKS

Elizabeth J. Campbell
Muriel Kannel
Kalman Saffran

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
CAMBRIDGE, MASSACHUSETTS

I. INTRODUCTION

1. Current Status of the Study

During the past year our study on the feasibility of networks has concentrated on the investigation of the applicability and ease of use of network functions rather than on methods of implementing a network connection. We have enlarged the scope of the study to consider the broad implications of resource sharing via networks. Originally we had undertaken the study as an investigation of the feasibility of a collaboration between the Laboratory for Nuclear Science (LNS) and the Information Processing Center (IPC) at the Massachusetts Institute of Technology in joining the ARPA network. The research into implementation was carried on jointly by the LNS and the MIT Programming Development Office (PDO), principally by Harvey Baumel who was supported financially by both groups. A year ago we reached a tentative conclusion of the best methods of implementing an ARPA connection based on current hardware and software (Progress Report, C00-2487-1). At the present time the

IPC has no plans for connecting its IBM-370/168 to the ARPA net. The main interests in networking of both the PDO and the IPC lie in working with EDUCOM, the Planning Council on Computing and Research.

To satisfy the needs of the LNS for networking, we have decided that the LNS should join whatever networks ERDA supports, currently the ARPA net. However, at this time the LNS does not have the funding or volume of work to justify implementing a connection to the ARPA net by the LNS alone. It is more feasible for us to continue to access the network through the Multics system on the Honeywell computer at the IPC. When the volume of our traffic becomes large enough to make the use of Multics less cost effective and less convenient than a network station at the LNS, we shall implement a connection. If the ARPA net still appears to be the most desirable solution, we will most probably set up a mini-host using a PDP 11/45 with an ELF system as described in Progress Report C00-2487-1. Funding for a mini-host, a port on an IMP and communications costs have been requested in the LNS budget submission for FY 78.

During the past year the network research has been performed mainly by members of the LNS systems programming group with the support of interested users of the LNS Computer Facility. It has been in the areas of education of our staff, introduction of Multics to the ERDA community, investigation into network mail facilities and experiments in using the network with problems typical of the computations required by physics

research.

The members of the network team are Elizabeth Campbell, manager of the LNS Computer Facility, Muriel Kannel and Kalman Saffran, members of the LNS systems programming group. Ms. Campbell has been directing the study and representing the LNS on the Network Objectives Panel. Ms. Kannel is the liaison person for members of the ERDA community using Multics. She has also been performing network experiments and is the official representative from the LNS to the Network Investigators Panel. Mr. Saffran has been representing LNS at the Network Implementation Panel meetings and has recently begun a collaborative effort with members of the Lawrence Berkeley Laboratory (LBL) staff to set up procedures and guides for the use of their computers via the ARPA net by the LNS research staff. Harvey Baumel left MIT to join the Telenet Corporation and as yet no one has replaced him as the liaison between the PDD and the LNS.

2. Education of the LNS Staff

Much of our effort this year was spent in familiarizing ourselves with the inner workings of the Multics system, its intricacies and peculiarities, and in investigating what it and other systems at MIT had to offer. This was accomplished by attending classes offered by the IPC and working with others at MIT who have extensive experience in using both Multics and the ARPA net. Planned for May is attendance at a OODCI course on

telecommunications. Also, we have investigated the activities of other ERDA installations on the network by attendance at network meetings, and by visits to the Lawrence Berkeley Laboratory (LBL) and to the Argonne National Laboratory (ANL). We have made one visit to the Computer Corporation of America (CCA) to look at their mail system and the Datacomputer, a shared large scale data utility system designed to serve computers on the ARPA network. We plan further investigations of this nature.

3. Administrative Functions on Multics for ERDA.

During the year, we have helped to introduce others in the ERDA networking community to the Multics system and to facilitate their use of it. We set up an account named "ERDA" for all members of the ERDA network research groups to use, primarily as a mail service, but also for small experiments in the use of Multics. Original funding was provided by the LNS, the LBL and the UCLA. As the use became heavier, the above groups set up individual accounts of their own. The ERDA account is now administered by the LNS with funds provided by ERDA for that purpose. Ms. Kannel acts as project administrator of the ERDA account, performing functions such as setting up or modifying user profiles for system access. For the other Multics account maintained by ERDA installations, we have assisted with the administrative details of obtaining project and user identifications, etc. Each account is given an identification beginning with ERDA, followed by the abbreviation for their site.

The existing projects are ERDALNS, ERDAUCLA, ERDALBL, ERDALASL, and ERDAANL. With the exception of the ANL account, Ms. Kannel acts as an assistant project administrator. For some of the inexperienced users she has set up a simple start up procedure and modified user profiles to allow them to execute commands we have created. The principal endeavor has been the mail system which is described in the section of this report entitled 'Mail Systems'. In as much as we were able to, we have served as a resource for ERDA users experiencing difficulties with Multics. A public file of information has been set up for all the users of the ERDA project to give them the project mailing list, information on monies spent, and messages we or they put into the file for the purposes of sharing information.

4. Goals Realized by the Study.

During the year, we have developed an increased awareness both of the potential of using networks for resource sharing and of the problems encountered. We have especially come to realize the nature of the concerns of the user who is not a professional programmer. Therefore, we are in the process of setting up and defining procedures for users to help them in the learning process and to minimize the frustration that occurs in trying experimental systems. In other words, we recognize the need to make network use attractive to the user.

In performing the experiments we have learned the characteristics and requirements of a program that make it

suitable for running on a remote computer via the network. This information will enable us to set up criteria for jobs to be selected for experiments on the network. Our ultimate goal is determining those applications in the LNS that are the most suitable to be done remotely and those which should be done locally on the basis of cost effectiveness and ease of use.

Another goal realized this year has been the establishment of effective working relationships with other ERDA sites on the ARPA net, particularly with members of the LBL staff.

II. MAIL SYSTEMS

1. Characteristics of an Ideal Mail System

In setting up a Multics Account to enable members of the ERDA network groups to use the Multics mail system, we became interested in the use of networks solely for communication purposes. After investigating what some of the various existing mail systems have to offer, we developed a set of characteristics we would like in an ideal mail system. We shall first describe these characteristics and then produce a critique of the systems of whose existence we are aware. We would like to point out that since these systems are being updated continuously and since our

experience with them may not be extensive enough for us to have discovered all the hidden "goodies", we may overlook, inadvertently, in our critique some of the features of a system and thus fail to give complete credit to that system.

a. General Characteristics

First in importance, a mail system should be designed for the layman, i.e. it should speak English, not computer jargon. It should be easy to use without extensive reading of or reference to manuals. Good help facilities should be provided. The user should be able to obtain assistance in a variety of ways and at any point in the process, such as an on line description of any command, or a list of possible steps to take if encountering difficulty in the middle of a command. The system should prompt the user with questions as to what action he wishes to take next.

Any mail system must be secure. Users must be assured that their mail is private, to be read only by themselves, unless they grant permission to someone else.

A mail system should be inexpensive. This means careful design of the use of file space.

b. Read Mail Features

The person receiving a message should be able to do the following:

- Examine a list of messages received classified by subject, author, date and time, others to whom it was sent and

length of message.

- Read selectively any message in the list.
- Dispose of the message by deleting it, leaving it where it is, filing it in a variety of ways such as by subject, author, chronologically, or any combination of these.
- Be able to copy it to someone else.
- Quit reading at any point.
- Respond to it at the time received, with the system supplying the information to whom it is going, the date, reference to the subject, and the date the message was received.
- Search the list by author, subject, date received, or combination of such.
- Mark a message as read.

c. Send Mail Feature

In sending mail, the system should provide the following:

- Automatic prompting of the user by typing headings such as "To:", "Subject:", and automatic insertion of the date, time and name of sender.
- Ability to file it in a variety of ways, subject, author, etc.
- In line simple editing abilities such as erasing letters or lines.
- Ability to enter a more sophisticated editor to edit a letter once it is written.

- A message to the writer that the letter has been sent.

d. Message Facility

In addition to the reading and sending of mail in the above manner, which corresponds to letter writing, the system should provide a message facility for communicating with other users on line. It should give the user the ability to:

- Query the system to see who is logged in.
- Send a message in the middle of other processes.
- Decide whether or not to accept messages.
- Defer messages so as not be interrupted.
- Cancel the defer message option.
- Save and print defined messages.

e. Editing Capabilities

As mentioned above, the system should allow simple in line editing while writing and more sophisticated editing on a letter already created in a file. The editor should be easy to use and provide the user the ability to do the following.

- Search for and change words, sentences, etc.
- Insert material
- Rearrange paragraphs, etc.
- Reformat and realign.

f. Filing Capabilities

Most of the filing capabilities have been mentioned under the sections on reading and sending mail, but in

general we can summarize them as giving the user the abilities to:

- Have files automatically created for him such as for messages received or sent.
- File by subject, author, date or any combination of these and in whatever order appropriate as alphabetical, chronological, etc.
- Search files by the same categories as above.
- Remove any messages temporarily or permanently from a file.
- Find and read, copy or forward any message.
- Have only one copy in the systems even though referenced in a variety of files.

g. Additional Features

These are some features which do not fall into the above categories, such as:

- Uniform method of identifying users in a system.
- Ability to scan a list of users, arranged alphabetically by last name to locate their user identification, i.e. much like using a phone book. This list could also contain other information the user might care to make public, such as phone number, site address, etc.
- Ability to obtain members of any particular public distribution list.
- Method of obtaining costs such as cost of sending a message.

2. Multics Mail

The Multics mail system at the moment does not contain all of the desirable characteristics, but it would not be difficult to implement most, if not all of them, by collecting routines already written by various users and supplementing them with whatever others would be needed. An extensive command structure of handling files, etc., already exists. The main difficulty with Multics is that it is a highly sophisticated system designed more for the systems programmer to use than for a layman. However, this can be overcome by writing upper level commands that translate 'English commands' into systems commands. Guides, written in terms that the layman can understand, would also alleviate the problem. A draft of such a guide prepared by LNS for use by members of the ERDA project on Multics is included as Appendix A to this report. There are two advantages of considering developing a mail system on Multics. One is that the Multics mail facility is secure, and that security feature is hardware enforced so no amount of 'hacking' is going to let someone who is unauthorized read another user's mail. The second is that for the sophisticated user, Multics offers the more sophisticated environment, while at the same time, a 'use English' approach to Multics can be developed to give easy access to the system by a novice.

At the present time, the Multics mail system as set up for ERDA users (see Appendix A) contains the following features

of the Ideal system. The user can read and send mail with author, date and time provided by the system. Mail can be kept or deleted. The user can send mail to more than one user at a time and to a distribution list of ERDA members. He can request that other distribution lists be set up for him. He does not have the ability to examine the contents of his mail by subject, author, etc., to read messages selectively or to answer without going into the send message mode. Neither can he easily forward the message to someone else. Letters that a user sends are not filed automatically unless the user creates the file. Automatic prompting of what the user wishes to do next when receiving or sending mail is not provided. The only prompt expression for the user is "Input". Excellent editing facilities, both in line and through various editor, are provided. The message facility for on line use is excellent, providing all the features required except for a file of received messages. None of the filing capabilities are provided automatically, but all are possible by the creation and manipulation of the users own files. In Multics each user can choose his own personid so a uniform method of addressing users does not exist. The personids can be difficult to remember as they are combinations of surnames, initials, etc., with upper and lower case intermixed. This problem could be helped by giving the user access to a phone directory type of list of user s. For ERDA, a list of all people with ERDA mailboxes may be accessed.

As mentioned above, programs can be added to the ERDA

mail system on Multics to provide some of the missing capabilities. One such program is read_mail written by Ken Pogran at the Laboratory for Computer Science at MIT. It provides the following features: a file consisting of a numbered list of messages received categorized by number of lines, date, time, sender and subject; ability to delete or undelete specified messages, forward a specified message to a person or particular mail file, print a specified message on a terminal, print a summary of specified messages, and quit the program at any point. Help is received by using the explain command or typing a ?. The user is prompted by the message "ACTION:". The read_mail program is already in use by some of the ERDA users of Multics. We plan to give the ability to use it to all ERDA users and to issue a guide on its use. We plan to investigate also what other hidden features may also exist in Multics, make a list of what is missing and assign the necessary programming tasks to a student we plan to hire.

At this point it would be appropriate to mention the use of network mail. Essentially, net_mail is the interface between the ARPA net and a system such as Multics that is running on the host computer. It allows communication between users of various hosts without the user having to be registered on any particular system on either host. Access to net_mail is granted by the network representative at each host. Each host can decide whether or not to allow communication through net_mail. All the ERDA users of Multics have been given access to net_mail.

3. MSG System on TENEX

MSG is a program available on TENEX Systems for reading, writing and subsectioning files which have a "message file format". Thus, it affords a very powerful and useful mail system. It provides most of the features we would like to find in a system. For reading messages, it provides the ability to examine them in a list containing author, date and time, subject, and length of message and to read them selectively. The user may forward messages and mark or unmark them as having been examined, i.e. read. For sending an answer to the author of the message that has been received, the user can reply without having to type the address of the sender and may copy to all the recipients of the original message if desirable. In sending messages, the user is prompted for the necessary information by "To", "From", etc. Messages may be sent to one or more people. Messages that are being forwarded can be edited or have comments added to them. The user can choose whether or not to receive copies of messages he sends. All of the normal editing capabilities exist.

The filing capabilities are very extensive, satisfying all requirements. A search can be made by subject and author. So far, this is the only system of which we are aware containing this ability. Messages may be deleted, undeleted or permanently expunged from files.

The main difficulty with the TENEX System is that it does not seem to be written for the layman. The documentation is

excellent but programmer oriented. Use is made of expressions like string and of symbols such as ">" or "<", etc.

4. TDA System under Development by CCA

The Computer Corporation of America has under development a message system, TDA (Telecommunications Data Access), which will shortly be available commercially. It can be used as an in house system or via Telenet. It has interesting attributes. It was designed as a message system from the beginning, not placed together from parts of an existing system. As such it is a dedicated system which provides good security. It is mini computer based and thus probably inexpensive. It is capable of upward expansion to a very large number of users. To us, most desirable is that it was designed for laymen, executives, secretaries, scientists, etc., not systems programmers. After a demonstration lasting about an hour, I felt very comfortable with the system. Commands are English words that are self-explanatory. No use is made of special abbreviations or symbols such as 'udd', '>', '<', or '_'. Upper and lower case are available but one does not have to discriminate in their use. The preliminary documentation is quite readable by the non programmer user. In essence, TDA is similar to SNDMSG on the TENEX system but with more and improved features either implemented or planned. It has excellent help facilities. The user can ask for the responses valid at any point in the system. If more help is needed extensive

Information can be obtained, applicable to the point at which the user is in the system, with instructions on how to get more help. Documentation about a particular command can be requested or only a list of the possible arguments for a command. Extensive prompting of the user is done by the system asking what function is desired next. At any time except at the prompt for the edit command the user can ask for informations such as the time, etc. The help command can be used while editing.

The user has files created automatically by the system and may create others. The user may examine these files which contain numbered messages indicating date, time, sender, receiver, subject and message length. After reading a message, the user may take actions such as deleting a message or responding to it. Accidental deletion is prevented as the system saves deleted messages giving the user the opportunity to retrieve them or permanently delete them. In answering a message, much of the sending information is supplied by the system from the message being answered such as author, date, reference to subject, etc. Messages may be forwarded. In sending the message, the system prompts the user with all the necessary format. The message may be sent to one or more persons or to a distribution list. When finished writing a message, the user is asked what he wants to do next such as editing, sending now, etc. The editing facilities are excellent. The editor is written from the laymen point of view in the choice of terms such as LOCATE etc.

Filing capabilities as mentioned above are excellent. Messages may be filed into several files simultaneously. No matter how many files a message is in, or how many receivers there are of the message, only one copy is kept in the system. Messages sent are automatically filed for the sender of the message.

There are probably a host of other features as yet not known to us as we have not used the system but plan to do so in order to discover its full potential.

III. NETWORK EXPERIMENTS

During the year we have tried various experiments in accessing computers at other ERDA sites either by remote job entry or via the ARPA net. Our purpose has been threefold; to assess the capabilities of each method, to determine the problems the user will face, and to decide what applications are best suited from the point of view of cost effectiveness and ease of use. Using this information we plan to set up guides to the use of other facilities especially designed for the research staff at LNS. The following describes the major experiments done and the learning gained from them.

1. Remote Job Entry

One of our first experiments was an attempt to use the CDC 6600 at the Courant Institute at New York University (NYU). We had obtained a Control Data 200 User terminal to use for remote job entry. Initially the experiment involved running a small sample FORTRAN program. As resource material we began with a Short Guide to NYU facilities and with Peter MacLean of the NYU staff available for phone consultation. The first few tries were wrought with frustration.

It would have been helpful if we had a sample deck and instructions for running it, or a listing of the same, sent to us. We finally did develop these ourselves with the help of Peter MacLean. Some of the snags we ran into before getting this small job to run were:

- a. Our job card had a comma in it rather than a period because the smudged printing in the NYU short guide made the period look like a comma.
- b. An extra blank card in our deck made NYU's Remote Job Entry feature crash. The error was corrected at NYU, and we in turn no longer put blank cards in our decks.
- c. Our account was inadvertently removed from the NYU accounting files.
- d. Difficulty occurred with our card reader which was repaired by CDC.
- e. Ostensibly there was much down time at NYU. This is always difficult to verify, since NYU's Machine Information

Extension is usually busy at these times.

f. There were difficulties getting a data line to NYU on some days and constant telephone disconnects on other days.

By mid-July we finally did get our sample deck to run. Peter MacLean arranged for an account for us for the new fiscal year which began July 1. Another difficulty arose for us during the summer. The 200 User Terminal controller can have one of sixteen site codes set by hardware switches. The allowable site codes were changed for us by NYU and it took us a while to figure out why we could no longer reach NYU. This period included servicing by CDC and the phone company since we were baffled as to why we could not run. After this was settled, we once again got our sample to run.

At that point we had hoped to try to run COCHASE, a theoretical physics program to solve coupled channel equations, at NYU. The program was coded in double precision 360 FORTRAN and was fairly long (2000 cards). Since there are no double precision complex functions in CDC FORTRAN, we edited out all those calls and converted the program to single precision (which is accurate enough in CDC FORTRAN). The editing was done using the Honeywell Multics system. The resultant deck had to be converted from EBCDIC to BOC (IBM character code to that of CDC).

However, we were not able to consistently run the sample deck. There were some days that we just cycled around the input, other days when we could not reach NYU at all, or the cards could not be read in. We had hoped to run the sample deck

and follow it immediately by the COCHASE deck using the same set of control cards. However, the difficulties we had with both the 200 user terminal and the NYU operating system, made us stop this experiment in favor of more fruitful ones.

One of our basic difficulties at NYU is that batch remote job entry allows very little communication with the computer personnel on the other end, the only communication being via telephone. A time sharing interactive environment is much easier to operate in for a networking experiment if debugging is necessary.

Our experience at NYU could be contrasted with that of running on the CDC 7600 at Brookhaven via the CDC user terminal. Last summer when the analysis of data by Professor Ting in his J particle experiments at Brookhaven reached major proportions, we decided that the LNS systems programming group should give him some assistance. A major part of the analysis was performed on the CDC 7600 at Brookhaven through his collaborators on the experiment there. Three of the LNS programmers spent a day at Brookhaven with the intent of studying their operating system as it applied to Professor Ting's work. After that they assisted one of Professor Ting's colleagues in running some of the analysis programs on the CDC 7600 via the CDC user terminal located at the LNS. It was very successful.

The difference between this success and the problems with NYU can be accounted for by the following. Although members of our staff had visited NYU, it had not been for the express

purpose of gaining information for running a particular program. Professor Ting's program was already operational at BNL and did not require debugging. It was handled by someone very familiar with the program and with the Brookhaven system and who had a real need to make it work. All that was required by him was assistance from the LNS systems group in using the CDC user terminal. Also a factor of luck was involved. The Brookhaven system was up on the days we tried to use it. It was not in as heavy use as the one at NYU. Also our CDC user terminal did not choose to malfunction on those days. We think it is safe to assume that if familiarity with a system is gained, and if the computer is not overloaded, remote job entry is fairly straightforward and at the present time easier to use than access via the ARPA net. However, it does not provide as many desirable features or as effective communication costs for a large volume of work. At the moment we are using the terminal over regular phone lines, as we do not have enough volume to justify either a dedicated phone line or FTS service. Cost to New York during the day is thus typically \$.29 per minute after the initial period.

2. Experiments in Using the ARPA net to Access LBL.

One of the main experiments in progress on the ARPA net is an attempt to get operational at LBL a program called VARR II for Professor Golay in the MIT Nuclear Engineering Department. The VARR code is a large heat flow program for nuclear reactors written for and particularly well suited for a CDC 7600 or 6600.

The program was originally written for Westinghouse by Science Applications, Incorporated in LaJolla. Prof. Golay obtained a copy from the Chicago Operations Office and turned it over to his student Yi Bin Chen to run. Mr. Chen tried unsuccessfully to convert it to run on the 370/168 at the MIT Information Processing Center. When he came to us for help, we suggested that we would get it operational at LBL on their CDC 7600 or 6600 as a network experiment, using Multics to get into the ARPA net and then turn it over to him to run production at LBL. The experiment would fund the debugging and Prof. Golay would pay LBL for the production runs. Dennis Hall and Don Austin of the LBL staff agreed to help us on the LBL end. The experiment proceeded very slowly. At first we tried sending the file using the LBL editor. However, the file was long (4000 cards) and had to be divided into segments to be concatenated at LBL. This tedious process seemed not to work. When the LBL User FTP became operational, LBL used it to take the program file from MIT to LBL. It took about 20 minutes to transfer 4000 cards. The program still continued to bomb out due to compilation and run time errors. Although we were told originally we had a running deck, we discovered after talking with Mr. Chen that the deck he was using was not the same deck as the listing he had given us from Westinghouse. The deck which he obtained from Chicago was an earlier version of the program which he had tried to update, thus introducing errors into program.

Further consultation with Westinghouse by Professor

Golay revealed that there was a current version of the VARRII program in the LBL libraries. We no longer needed the obviously incorrect file that was transferred from MIT to LBL. Instead the LBL library version could be used. An execution procedure for running this version of VARRII on data transferred from MIT-Multics to LBL was developed. The scenario for running the program for the MIT-Multics user was:

- a. Log in to LBL using the ARPA network.
- b. Invoke UFTP (User File Transfer Protocol) at LBL.
- c. Log in to MIT-Multics for file transfer of the input data to be sent from MIT-Multics to LBL.
- d. Run the program on the transferred data.
- e. Scan the results and to see if the run is completed.
- f. Transfer the results back to MIT-Multics if they look correct. This involves invoking UFTP again and logging back into MIT for a second data transfer.

When this scenario was last tried in December, 1975, the program had some run time errors. It was decided to have Professor Golay contact Westinghouse once again before any more money was used on the project.

Westinghouse insists that this is a running program. They have made some suggestions on changing the structure of the data and have given Mr. Chen a list of the Command Control Cards used to make the program operable on the Westinghouse CDC machine. Professor Golay and Mr. Chen now have their own account on Multics. Mr. Chen, with the help of Muriel Kannel at the LNS

and Dennis Hall at the LBL, is trying to run VARRII on a new set of data but using the same scenario. Hopefully we should have some results soon. We, the investigators both here at MIT (Muriel Kannel) and at LBL (Dennis Hall and Don Austin), have concluded that this is a reasonable network experiment. However, some of the networking issues have been obscured by problems with the VARR program itself. We all agree it is necessary to have an expert on the actual program available, or else too much time is caught up in unnecessary frustration.

In contrast to the VARR program, we were able to convert and run, using the same networking scenario, a program of about 200 cards in a relatively short period of time. The program called Fermi, consisted of a main program (in IBM FORTRAN IV) and a subroutine in CDC FORTRAN, which formed Hartree-Fock wave functions with given BCM coefficients.

The problems we encountered were:

- a. The syntax for literal and integer data is different for IBM and CDC FORTRAN and so some changes were needed.
- b. Blank cards are treated as zeros in IBM data but not in CDC data.
- c. CDC FORTRAN can handle a program completely in BCD code or completely in EBCDIC. Fermi, a mixed deck, had to have one subroutine converted.

These problems were all solved relatively easily. It was particularly helpful to have the programmer available when the one run time error occurred. She knew immediately that the

zero data was not handled properly. An investigator not familiar with the program would have had more trouble "zeroing" in on the problem. The same user also wanted the source code of two rather short programs developed at LBL. Dennis Hall put them in a Multics file on his account and we were able to have them punched here at MIT. This was much quicker than using the U.S. Mail to send the cards across the country.

3. Development of a General Scenario for Use of the LBL Facilities

After our attempts at running jobs both at NYU and the LBL, we decided to facilitate our network experiments by developing a series of guides for use of the computers at various remote sites by LNS personnel. Our experiments will have much more relevance if conducted by "real" users running "real" programs. This is difficult at the present time with the state of the network implementation and current documentation. Since we are trying to run a program at LBL and working closely with the staff involved in network research there, we decided to make our first guide one on describing access to the LBL facilities. The initial draft of the guide is Appendix B of this report. At this time it describes only the scenario for batch submittal of a FORTRAN program with the data included as part of the program deck. Future additions to the guide will include the following: examples and explanations concerning the use of auxiliary storage devices (disk, tape, data cell, etc.) to store programs

and data at LBL, additional automation of the batch job submittal procedure and additional reference material as it is acquired. Future guides to the use of the LBL facilities will include an "LNS User Guide to Remote Batch Submittal to LBL via a CDC 200 UT Remote Job Entry Station" and an "LNS Users Guide to Running an Interactive Job on the CDC 6600 at LBL".

The development of the scenario was a complex task as it required learning to use several computer systems and software subsystems. The systems involved in the remote submittal process are: MIT Multics, LBL Sesame, the ARPA net user telnet, user file transfer and server file transfer protocols, the Multics QEDX editor, the Sesame NETED editor,, and the CDC FORTRAN IV FTN4. However, this was a valuable learning experience for Mr. Saffran, the author of the guide.

Remote consulting with LBL on the running of the sample program presented some problems. Two sources of information at LBL were required. Since the ARPA net implementation and use is still under development, questions about the network functions had to be directed to Dennis Hall, the LBL network investigator. Questions about the use of the LBL computers and systems, were more appropriately addressed to the consulting office in the LBL computer center. There was no way of exchanging written or printed information such as console or computer printout. Telephone lines always seem to be busy when you need them most.

Terminal adaptability presented a minor problem in that an ASCII upper and lower case terminal would have been ideally

sulted for the development of the scenario. The terminals we had available for use were an ASCII upper case only terminal and an EBCDIC upper and lower case terminal. Since the terminals are on lease, we plan to change them for ones more suitable for use in the experiments we have planned.

A software error in a Multics FORTRAN IV routine and a problem with the LBL user FTP delayed development of the scenario for a few days. The problem with the LBL user FTP has been corrected. The Multics problem still exists. We plan to correct it by writing a program which will make the LBL carriage control characters in a job print output compatible with those of Multics.

We made an estimate of the cost of running the sample job at LBL versus running it at the LNS on the IBM 360/65. The batch job cost at LNS was \$.65 and at LBL about \$2.07. However, on short running jobs it is difficult to compare costs in a fair manner. The estimate of the communication costs is more relevant. The Multics terminal session cost, file transfer and retrieval, and printing plus the LBL terminal session costs amounted to \$6.20. This does not include Multics data storage charges and any Multics or LBL ARPA net use charges.

We also tried running the same program at LBL by submitting it as a batch job via remote job entry on the CDC 200 UT user terminal. In this case the procedure was much simpler, involving only a few trial runs. Communication costs were approximately \$3.39 for six minutes on the phone line. A

change we plan in phone service will lower this cost. When we begin to do experiments with large size and longer running jobs, we will obtain more accurate cost comparisons.

IV. INVESTIGATION INTO NETWORKS OTHER THAN THE ARPA NET

Although no actual experiments have been performed, we have looked into the use of networks other than the ARPA net. We have decided that the most logical one to try first is Telenet for the following reasons. Telenet has installed a TAC (Telenet Asynchronous Controller) at MIT. The IBM 370/168 is linked to it via the MEMOREX 1270 and the Honeywell computer via a 355, both on 1200 and 300 baud lines. There are also two terminals directly linked to the TAC, one on a 300 baud line and one on a 1200 baud. We could experiment with the use of the IBM 370/168 via Telenet. A computer which is highly suitable for Telenet experimentation, because of the design of both its hardware and software, is the Prime. LNS owns a PRIME 300, used for research in heavy ion physics. We have been thinking how we could experiment with Telenet, using it as an interface or host. Conversations have been held with members of the Heavy Ion Group to assure their willingness to allow us the use of the Prime and to collaborate to some extent in the experiment. To use it on

Telenet will mean updating the software to the Prime network interface to the latest version of the operating system and acquiring a synchronous line adapter. Estimates of the time required for modifying the software run from two weeks to a few months. If experimenting with the Prime proves feasible, we might be able to make some arrangement with the Prime vendor on the hardware, such as a loan.

Another approach to Telenet experimentation that we are considering is trying to go from one PDP 10 to another. We have discussed the possibility of a collaborative attempt with David Friesen of the First Data Corporation in Waltham. His company provides a time sharing service on three PDP 10's and uses Telenet for some of their communication. Mr. Friesen is an ideal person for us to work with as he was formerly associated with the Accelerator Physics Collaboration at the LNS. He wrote the major portion of the software for PEPR, the precision encoded pattern recognition system for reading bubble chamber film, which operates on the LNS PDP 10. Thus he is extremely familiar with the data handling and analysis requirements of the LNS physicists. The experiment we have in mind is the simulation of the type of file transfer of data that physicists would be likely to make between two computers--in this case the PDP 10 at LNS and the one at First Data.

We are just beginning to look into what research is being done at MIT on other networks. People here are working closely with both NerComp and EDUCOM. We plan to contact them to

see if there is any way in which we can share network experience with them.

V. CONCLUSIONS AND RECOMMENDATIONS

Though our experience with the ARPA net has been limited, it has led us to the conclusion that resource sharing via a network can provide a very feasible method of supplementing modest on site computer facilities in laboratories with computing needs similar to those of the LNS. Laboratories of the size of the LNS cannot support complex kinds of either hardware or software as can the larger national laboratories. Networks appear to offer the opportunity to access when needed a computer that can supply larger memory, more on line storage, a faster central processor and more sophisticated time sharing or graphic systems than available on the on site computer. It also provides a way of handling a peak load by being able to send programs to whatever computer has available the time and resources required. More sophisticated approaches to computing can be taken such as processing a job in one computer, accessing the necessary data stored in another.

However, there are still many problems to overcome in network use. Systems even on the same model computers differ

sufficiently that there must be on site a person familiar with the remote systems. It takes a long time to learn the intricacies and subtleties of each system. Although the ARPA net has been around for a number of years, it must still be considered to be in its infancy. This is due to the fact that very little work has been done to produce or distribute documentation on the use of the various computer facilities which are in the network community. A description of the use of the network itself is certainly available, but the situation at present might well be likened to giving a telephone to two people speaking different languages, explaining to both persons how to use the telephone, but not teaching either the other's language. At the present time, for batch processing it is obviously easier to access another computer via a remote job entry terminal over direct phone lines than via a network since the knowledge of the methods of accessing the network on either end is not required. Much of the implementation of putting ERDA computers into the ARPA net is still in the developmental stage. The situation will vastly improve as this implementation is improved and completed. At LNS we feel we have accomplished quite a lot in setting up a connection with at least one of the computer systems on the network, that at LBL. We have their documentation and are in the process of writing guides to supplement it. Through our systems programmers, we plan to provide an interface for our users to the consulting facilities at LBL. We have received the utmost cooperation from the LBL staff. We believe the approach we are

taking is the one which will help users of computers who are not programmers to take advantage of the network. We discovered that performing the experiments was more time consuming than we had anticipated. A standard method of describing and documenting experiments must be worked out between experimenters at the ERDA sites on the network. This will probably happen at the meeting of the Network Investigators Panel in May.

Our recommendations for future experimenting with networks are the following. More consideration should be given to the problem of the user interface to the network, the human factors involved, the amount and type of consulting help required, etc. More investigation needs to be done into the cost effectiveness of networking, load leveling to achieve the best use of resource sharing, what equipment to have on site, etc. We believe the network research is best carried on as it is now by a cooperative effort among the various ERDA installations.

For the LNS the best solution to the computing requirements of the research staff is an on site computer, larger than our present IBM 360/65, supplemented with access to the ARPA net. For the large fraction of our workload involving the analysis of large amounts of data read into the computer, analysed once and rarely used again, it is more cost effective to use the on site computer. However, for our large theoretical calculations with little input and output, the network is already a viable solution. These programs would be more efficient on computers with more processing power than the 360/65 and in many

cases with greater single precision accuracy (such as on CDC computers.) Access to more sophisticated systems would allow our users to develop improved and more effective computing techniques. Access to other computers on the ARPA net is best done via Multics until the volume of our load justifies a most cost effective and convenient method.

Therefore, we plan to continue our experiments on the ARPA net working on as many other computers as possible. It is our hope that we can involve our users and provide them with an interface sufficiently easy to use, so that they will start to do some of their production computing at sites where they can obtain funding. We plan to continue use of Multics for functions such as the mail system, both to increase our knowledge of Multics and to provide a service to the ERDA community.

VI. RESEARCHER'S ALLOCATION OF TIME

The principal investigator, Elizabeth J. Campbell, has spent 20% of her time on this project. She has been assisted by Muriel Kannel who has worked half time on the project. Kalman Saffran who has joined the project recently will contribute about 10% of his time. For this contract period, his time will amount to the equivalent of five weeks. The investigators are members

of the sponsored research staff at the Laboratory for Nuclear Science.

We would like to thank members of the Programming Development Office for their consulting assistance in the use of Multics. In particular, we would like to acknowledge Lyman Hazelton of MIT for his assistance and patience in teaching us the intricacies of the Multics system, for writing routines in Multics for us and for his constructive suggestions on our experiments on the network.

DRAFT OF APPENDIX A

USER MEMO 1. INTRODUCTION TO MAIL FACILITY ON MULTICS

Introduction

As part of the network experiments being done by various ERDA computer installations, an account called "ERDA" has been set up on the Multics computer system at the Massachusetts Institute of Technology. The main purpose of this account is to allow users on it to use Multics as a communications system. Administration of the account is handled by the Laboratory for Nuclear Science through Muriel Kannel, Room 24-021, M.I.T., Cambridge, Massachusetts 02139 (617 - 253-6065). If other ERDA accounts on Multics are set up through her, the users will automatically be added to the account called "ERDA".

This is the first in a series of memos on the use of Multics that will be especially designed for the ERDA community. It describes the Multics mail system at a fairly basic level. Also described are the sm (send message), otc (on line consultant), and help commands for those people who want to communicate with others who are on-line concurrently or to seek on-line help.

Sample Sessions

In the sample terminal sessions that follow, an exclamation point (!) is printed at the beginning of every line typed by the user to distinguish user entries from system entries. The user should not type the exclamation point. In all cases he must use upper and lower case carefully just as indicated.

I. LOGIN PROCEDURE - READ MAIL - LOGOUT

After the user has dialed in and a connection to Multics has been established, an information message about the current system is printed on the terminal. The user is now ready to login. The sample session is as follows.

```
! login MJones ERDA
Password:
! user-password (the typing is suppressed or hidden)
MJones ERDA logged in 7/03/75 .....etc.
r 1506 .751 12.578 400 (This is a sample of a ready
message)
! mail
```

The contents, if any, of MJones' mailbox is printed here. If there is no mail, this will be indicated. The user will also be asked if he wants to delete his mail after it is printed.

```
r 1509 26.08 59.053 511
```

```
! logout
```

If the login is not correct, rather than getting a ready message, the system will prompt the user to try again. In the above sample, MJones is the user's name, also known as the personid. ERDA is the projectid.

II. SEND MAIL

To send mail to another user using the console for input, the sample session is as follows. Each line typed must be followed by a carriage return; one or more spaces should be inserted where indicated on the sample.

```

! mail * DJones PROJA
Input
! Type the message here on as many lines as is necessary.
! Follow it by a line consisting of only a period and a
! carriage return.
! .

```

If a user wants to send information contained in a Multics file rather than sending console input, the command is:

```

! mail filename personid projectid

```

The file in this case could be one created by an editor, read in from an off-line device, etc. Mail consisting of either console input or a Multics file, can be sent to more than one user on the same or different projects. The commands are:

```

! mail * DBROWN PROJA TWHITE PROJ8 MROSE ERDA
! mail filename BCampbell ERDALNS Kannel ERDALNS

```

'erдамail' is a way of sending messages to all ERDA users. The personid's and projectid's are created by the command 'erдамail'.

```

! erдамail *
Input
! This is a message for all ERDA users
! .

```

A message may be sent as well as console input. For example:

```

! erдамail PROBLEMA

```

To obtain a list of the people currently on the erдамail list, type the following line:

```

! print >udd>ERDALNS>public>erдамailist

```

III THE WHO COMMAND

Before using the message facility described in section IV, the user can invoke the who command to check if the other users with whom he wishes to communicate are logged in .

By issuing the command

```

! who

```

the user can learn the number, identification and status of all

users currently logged in. This is usually more information than he needs and should be used sparingly. More often the user will want to check if a particular user is logged in before issuing a send_message command. He would then type:

```
! who personid (! who MJones)
```

He might want to check which users are logged into a particular project. This is done by

```
! who .projectid
    or
! who .ERDA
```

IV. MESSAGE FACILITY

The message facility allows for on-line communication between users on different terminals who are logged in concurrently. It consists of the following commands and their abbreviations:

a.	accept_messages	am
b.	send_messages	sm
c.	defer_messages	dm
d.	print_messages	pm
e.	immediate_messages	im

- a. If the user wishes to receive on-line messages instantaneously, he must type:

```
! accept_messages
    or ! am
```

- b. If the user wants to send a message, he must invoke the send message command with the proper arguments. For example:

```
! send_message personid projectid message
    or
! sm MJones ERDA This is a one line message.
```

If the message is more than one line, the system will prompt for the input as shown below.

```
! sm MJones ERDA
Input
! This is line 1
! This is line 2
```

! .
In the above sample the period indicates the message is complete although each line is sent as soon as the user types the carriage return for that line.

- c. If a user does not want to be interrupted by messages printing on his terminal, he uses the command:

```
! defer_messages
      or
! dm
```

- d. If the user wants to print the contents of his message segment (e.g. his deferred messages), he uses the command

```
! print_messages
      or
! pm
```

- e. If the user wants to cancel the defer_messages command, thereby restoring the immediate printing of messages sent by the send_messages command, he types:

```
! immediate_messages
      or
! lm
```

V. olc - The on_line_consultant

To help people with problems they may encounter while using Multics, an on-line consultant is on duty weekdays from 9-12 and 2-5 (EST or EDT). Assuming the user is set up to accept messages, an example of olc use is:

```
! olc
Input:
! Where can I get more information on using Fortran?
from Bow.IPCTS: see center memo MS-9. Thank you
!
r 1719 1.275 12718 229
```

Note the use of the period to terminate the session with the

consultant.

If for some reason the on-line consultant doesn't respond during the user's current session, a message will usually be sent, and then printed at the beginning of the next session.

V. The help command

The help command helps the users to obtain information about Multics commands, subsystems, system status, etc.

The help files have been organized so that a small but useful amount of information is printed and then the user is asked if he wants more help to which he types Yes or No and a carriage return. Rather than describing any more about the help command, it is suggested the new user types:

```
! help or ! help help
```

This will give him more information about the help command itself.

APPENDIX B

20 April 1976

INITIAL DRAFT OF
LABORATORY FOR NUCLEAR SCIENCE USERS' GUIDE TO
REMOTE BATCH SUBMITTAL TO LAWRENCE BERKELEY LABORATORY
VIA MULTICS AND THE ARPANET

OUTLINE

INTRODUCTION

I. OVERVIEW OF THE JOB SUBMITTAL PROCEDURE

II. JOB SUBMITTAL SCENARIO AND EXPLANATION

- A. Logging into LBL through Multics via the ARPA net
- B. Transferring your Job from Multics to LBL
- C. Submitting your Job to the CDC 7600
- D. Checking Job status and claiming out
- E. Editing the print output
- F. Transferring the output from LBL to Multics
- G. Housekeeping and Logout of LBL
- H. Printing of Output, Housekeeping and Logout of Multics

III. JOB RESTRICTIONS

IV. SAMPLE AND EXPLANATION

- A. Job Card and Control Cards
- B. Program and Data

V. HOW TO READ A CARD DECK INTO MULTICS

VI. MISCELLANEOUS INFORMATION

VII. USER ASSISTANCE

VIII. REFERENCE MATERIAL

INTRODUCTION

This preliminary users' guide presents one method for submitting batch jobs to the Lawrence Berkeley Laboratory's (LBL) CDC 7600 computer from the Massachusetts Institute of Technology. The communications medium is the U.S. Advanced Research Projects Agency's Network (ARPA net).

Additions to this preliminary guide will be forthcoming as members of the LNS Computer Facility Systems Group continue their experimentation.

CDC FORTRAN IV programming is not discussed in this guide. A working knowledge of CDC FORTRAN IV is assumed.

I. OVERVIEW OF THE JOB SUBMITTAL PROCEDURE

1. In order to proceed with job submittal to LBL, you will need a Multics personid, projectid, and password, as well as an LBL account number and password. Please obtain these from Betty Campbell, Room 24-017, ext.3-6067.
2. First prepare your job for submittal to the LBL CDC 7600 computer. To accomplish this, consult sections III and IV of this guide.
3. After preparing and J29 keypunching your job, it must be put into a Multics segment (file). Section V explains how to do this.
4. When your job has been put into a Multics segment, proceed by logging into the LBL Sesame time sharing system through Multics via the ARPA net. This procedure is accomplished by lines 1-15 of section II.
5. After logging into Sesame, you will want to transfer your job from Multics to Sesame via the ARPA net. This is accomplished by lines 16-45 of section II.
6. Once your job has been transferred to Sesame, you will want to submit it to the CDC 7600. This is accomplished by lines 46-51 of section II.
7. You should then check on the status of your submitted job. To do this, refer to lines 52-54, section II. When your job is queued for print (see line 55, section II), enter line 56, section II.
8. You are now ready to edit your print output into a form acceptable to the file transfer protocol (FTP). This is accomplished by lines 58-74 of section II.
9. After editing, the next step is to transfer the print output to Multics. Lines 75-102, section II accomplish this.
10. After the edited print output is transferred to Multics, delete the print output queued for local printing at LBL, logout of Sesame, and exit from Multics user telnet. Lines 102-115 section II accomplish this.
11. You are now back at Multics command level and should queue your print output file for printing on a Multics line printer, request the Multics session cost, and logout of Multics. This is accomplished by lines 116-123, section II.

II. JOB SUBMITTAL SCENARIO AND EXPLANATION

An exclamation point (!) is printed at the beginning of every line typed by the user to distinguish user responses from system responses. The user should not type the exclamation point at the beginning of lines, only those in the line itself.

A. Logging into LBL through Multics via the ARPA net

```

1)  !  l Saffran -bf
2)      Password:
3)  !  #####
4)      .c..
5)  !  user_telnet lbl
6)      !
7)      WELCOME TO LBL, ENTER >LOG
8)  !  >log
9)      ?
10) !  >log, saffran.981005,saffran
11)      LOGIN CP-20 ITY-226 22.35.01.**BKY63C*B*04/18/76
12)      ENTER PASSWORD
13) !  #####
14)      SAFFR26 LOGGED IN.  SESAME 2.4
15)      OK - SESAME

```

Explanation:

- 1) Multics login command
- 2) Multics prompting message
- 3) Your Multics password
- 4) Multics ready message
- 5) Invokes the ARPA net protocol user_telnet to connect you to LBL
- 6) user_telnet acknowledgement
- 7) LBL prompting message
- 8) LBL login command (must be typed again on line 10)
- 9) LBL acknowledgement
- 10) LBL login command and jobcard
- 11) LBL login acknowledgement
- 12) LBL prompting message
- 13) Your LBL password
- 14) LBL login complete message
- 15) LBL Sesame time sharing system response message

B. Transferring your Job from Multics to LBL

```

16) ! common,uftr.
17) OK - SESAME
18) ! uftr.
19) HELLO, THIS IS USER FTP, ENTER COMMAND
20) ! ^new
21) OK
22) ! ^log,44
23) TRYING TO OPEN CONNECTION
24) CONNECTION OPEN
25) 300- Multics 28.2a: MIT, Cambridge, Mass.
26) Load = 12.0 out of 85.0 units: users = 12
27) 300 Enter USER command:
28) ! user Saffran
29) 330 Password:
30) ! pass 88888888
31) 230 Saffran ERDALNS logged in for file transfer:
    04/19/76 0122.0 est Mon
32) ! ^file,input
33) OK
34) ! retr>udd>ERDALNS>Saffran>LBL_forf_test5.kal
35) 255 SOCK 261
36) 250- Beginning transfer of >udd>ERDALNS>
    Saffran>LBL_forf_test5.kal
37) 250 to LBL in Stream 8-bit ASCII.
38) 252 Transfer complete.
39) LBL RETR COMPLETE
40) ! bye
41) 231-Saffran ERDALNS logged out 04/19/76 0134.3 est Mon
42) CPU Usage 1 sec, memory usage 40.0 units.
43) 231 430-hangup
44) CONNECTION CLOSED
45) OK - SESAME

```

Explanation:

```

16) LBL command attaches common file containing
    their User File Transfer Protocol (UFTP) to your
    session
17) LBL Sesame time sharing system response message
18) Execute the CDC UFTP protocol
19) LBL UFTP acknowledgement and prompting message
20) Use the new file transfer protocol
21) LBL UFTP new protocol acknowledgment
22) Connect you to MIT-Multics (No.44)
23) LBL informative message
24) LBL informative message
25) Multics informative message
26) Multics informative message
27) Multics prompting message
28) Your Multics personid
29) Multics prompting message
30) Your Multics password

```

- 31) Multics informative message
- 32) Tells LBL UFTP to name file to be transferred from Multics input
- 33) LBL UFTP acknowledgment
- 34) Tells Multics serverftp (SFTP) to send the Multics file to LBL
- 35) File to be transferred via Interface Message Processor (IMP) socket 261
- 36) Multics SFTP informative message
- 37) Multics SFTP informative message
- 38) Multics SFTP informative message
- 39) LBL UFTP informative message
- 40) Tell Multics SFTP to log you out
- 41) Multics SFTP informative message
- 42) Multics SFTP informative message
- 43) Multics SFTP informative message (disregard)
- 44) LBL UFTP informative message
- 45) LBL-Sesame acknowledgment message

C. Submitting your job to the CDC 7600

```
46) ! sesame,add,input.
47) LOAD COMPLETE, ENTERING ^EDIT
48) OK - ^EDIT
49) ! ^submit
50) SUBMITTED SAFFR27
51) OK - ^EDIT
```

Explanation:

```
46) Attach transferred file to your terminal session
47) LBL acknowledgment and informative message
48) LBL editor acknowledgment
49) Submit the transferred file (job & data)
    to batch
50) LBL Acknowledgement message
51) LBL-editor acknowledgment
```

D. Checking job status and claiming output

```
52) ! ^cc
53) OK - SESAME
54) ! >>saffr27
55) IN OUTPUT QUEUES - PR - SAFFR27
56) ! claim,saffr27,pr.
57) OK - SESAME
```

Explanation:

```
52) Change mode from Editor to Sesame (command level)
53) LBL-Sesame acknowledgment
54) Display status of job just submitted
55) LBL informative message - job in printer queue
56) Attach printer queue file to your terminal session
57) LBL-Sesame acknowledgment
```

E. Editing the print output

```
58) ! neted,saffr27.  
59) EDIT  
60) ! l eor  
61) EOR  
62) ! d  
63) ! l eor  
64) EOR  
65) ! d  
66) ! l eor  
67) EOR  
68) ! d  
69) ! l eor  
70) EOF  
71) ! d  
72) ! save  
73) FILE WRITTEN  
74) OK - SESAME
```

Explanation:

```
58) Invoke an editor (neted) to edit the print out-  
put from job SAFFR27  
59) Neted acknowledgment  
60) Locate end of record card image  
61) Neted acknowledgment  
62) Delete End of Record card image  
63) Same as Line 60)  
64) Same as Line 61)  
65) Same as Line 62)  
66) Same as Line 60)  
67) Same as Line 61)  
68) Same as Line 62)  
69) Locate End of File card image  
70) Same as Line 61)  
71) Same as Line 62)  
72) Replace original with edited version and exit  
from editor  
73) Neted acknowledgment  
74) LBL-Sesame acknowledgment
```

F. Transferring the output from LBL to Multics

```

75)  !  uftp.
76)  HELLO, THIS IS USER FTP, ENTER COMMAND
77)  !  ^new
78)  OK
79)  !  ^log,44
80)  TRYING TO OPEN CONNECTION
81)  CONNECTION OPEN
82)  300- Multics 28.2a: MIT, Cambridge, Mass.
83)  Load= 11.0 out of 85.0 units: users = 11
84)  300 Enter USER command:
85)  !  user Saffran
86)  330 Password:
87)  !  pass XXXXXXXXXX
88)  230 Saffran ERDALNS logged in for file transfer:
89)  !  ^file,saffr27
90)  OK
91)  !  stor test5_results
92)  255 SOCK 260
93)  250- Beginning transfer of >user_dir_dir>
    ERDALNS>Saffran>test5_results
94)  250 from LBL in Stream 8-bit ASCII.
95)  LBL STOR COMPLETE
96)  252 Transfer complete.
97)  !  bye
98)  231- Saffran ERDALNS logged out 04/19/76 0140.1
    est Mon
99)  CPU usage 2 sec, memory usage 40.0 units.
100)  231 433- hangup
101)  CONNECTION CLOSED
102)  OK - SESAME

```

Explanation:

```

75) Same as Line 18)
76) Same as Line 19)
77) Same as Line 20)
78) Same as Line 21)
79) Same as Line 22)
80) Same as Line 23)
81) Same as Line 24)
82) Same as Line 25)
83) Same as Line 26)
84) Same as Line 27)
85) Same as Line 28)
86) Same as Line 29)
87) Same as Line 30)
88) Same as Line 31)
89) Same as Line 32)
90) LBL UFTP acknowledgment

```

- 91) Command to Multics Server FTP to get CDC File
Saffr27 and to store it on Multics with
with relative pathname test5_results
- 92) Connect to Interface Message Processor socket 260
- 93) Multics SFTP informative message
- 94) Continuation of previous message
- 95) LBL UFTP informative message
- 96) Multics Server informative message
- 97) Logout of Multics
- 98) Multics informative message
- 99) Continuation of previous message
- 100) Multics disconnected (only SFTP level, not at
User TELNET level)
- 101) Multics informative message (disregard)
- 102) LBL-Sesame informative message - You are now
back at command level of LBL-Sesame

G. Housekeeping and Logout of LBL

```

103) ! delete,saffr27.
104) OK - SESAME
105) ! >wipe
106) TT WIPE
107) *J DR2235F 981005 76/04/18 B N 00023
.....
108) CP SECONDS 5.663
109) MONITOR REQUESTS 291
110) TTY MINUTES 12
.....
111) TOTAL JOB CUS 23
112) ESTIMATED COST $ 2.31
.....
113) JOB ENDED - DISCONNECTED
114) ! !exit
115) .r...

```

Explanation:

```

103) Deletes job print output queued for local print
      at LBL
104) LBL-Sesame acknowledgment
105) Logout from LBL-Sesame and delete all "work" files
106) LBL-Sesame acknowledgment
107) LBL-logout message
108)
109)
110) . LBL terminal session accounting information
111)
112)
113) LBL logout message
114) Command to Multics user_telnet, telling it to
      disconnect
115) Multics ready message

```

H. Printing of Output, Housekeeping, and Logout of Multics

```
116) ! dprint test5_results
117) 1 request signalled, 22 already in printer queue
118) .r.....
119) ! estimate_bill
120) shift 3 cost:      $.87
121) .r...
122) ! logout -bf
123) hangup
```

Explanation:

```
116) Multics command to queue to a line printer the
data set test5_results
117) Multics informative message
118) Multics ready message
119) Multics command to print estimated cost of
terminal session
120) Multics terminal session cost (not including
Multics cost of file transfers)
121) Multics ready message
122) Logout of Multics
123) Multics disconnect message
```

III. JOB RESTRICTIONS

1. Your job cannot explicitly request auxiliary storage devices such as disk or tape.
2. Your job must be punched on an 029 keypunch.
3. Your job must conform to CDC FORTRAN IV, VERSION 4.

IV.~ SAMPLE JOB AND EXPLANATION

A. Job Card and Control Cards

- 1) SAFFRAN,,300.804185,SAFFRAN
- 2) *HOLDOUT
- 3) *NOSTAGE
- 4) *7600
- 5) FTN4.
- 6) LG0.
- 7) EXIT.
- 8) CXIT.
- 9) FIN.
- 10) EOR

Explanation:

- 1) Jobcard format:
Jobname, default priority, 300 computing unit limit,
account number, programmer's name
Notes:
 - a. The default priority is "standard".
Do not use any higher priority.
 - b. 300 computing units is sufficient for
short to medium jobs.
- 2) Hold print output in queue for 12-24 hours
(hold time depends on time of day output was held)
- 3) Job does not require staging (e.g. no disk or tapes)
- 4) Job to be run on CDC 7600 computer
- 5) Invoke the FORTRAN IV FTN4 compiler
- 6) Load and execute the object code
- 7) Exit if there are any execution time errors
- 8) Exit if there are any compile time errors
- 9) Exit and Cxlt terminator card
- 10) End of Record card.

8. Program and Data

```

11)    PROGRAM MAIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
        DIMENSION X(10),Y(10)
1      FORMAT(10X,5HINPUT/10X,10F10.6//10X,6HOUTPUT)
2      FORMAT(5E16.6)
3      FORMAT(5I5)
        X(1)=0.259830
        X(2)=-0.134815
        X(3)=-0.351128
        X(4)=0.769003
        X(5)=0.926647
        X(6)=-0.529984
        X(7)=0.148433
        X(8)=0.651835
        X(9)=0.865956
        X(10)=0.351365
        READ (5,3) NMAX,IMAX,JMAX,KMAX,LMAX
        WRITE (6,1)(X(I),I=1,10)
        DO 25 N=1,NMAX
        DO 20 I=1,IMAX
            Y(I)=0.0
            A=X(I)
            DO 15 J=1,JMAX
                B=X(J)
                DO 10 K=1,KMAX
                    C=X(K)
                    DO 5 L=1,LMAX
                        D=X(L)
                        Y(I)=Y(I)+((A*B-C)/D)
5          CONTINUE
10         CONTINUE
15        CONTINUE
20        CONTINUE
25        CONTINUE
        WRITE (6,2)(Y(I),I=1,10)
        CALL EXIT
        END
12)EOR
13) 100  10  10  10  10

```

Explanation:

- 11) Program name and I/O definition card format:
Program program-name (I/O definitions)
- 12) End of Record card
- 13) Data

V. HOW TO READ A CARD DECK INTO MULTICS

To read a card deck into Multics, the deck must be preceded by the following two cards.

```
personid.projectid;  
pathname MCC
```

To designate capital letters on the above two cards, precede each capital letter by a cent sign (¢). Once the card deck has been read into Multics, type the following command.

```
copy_cards pathname
```

This will create a permanent segment (file), containing your card deck.

VI. MISCELLANEOUS INFORMATION

1. Useful character equivalencies:

ASC. II		EBCDIC
<hr/>		
back arrow (←)		underscore (_)
up arrow (↑)		not (↵)
or		
(^)		

- 2) Multics system status recorded information MIT X8-7700.
- 3) LBL filenames can be a maximum of seven characters.
- 4) The LBL command for the time and date is >t.
- 5) The LBL command for aborting the terminal initiated process currently in progress is:
>drop

VII. USER ASSISTANCE

For assistance please contact:

Kal Saffran
MIT Room 24-019
X 3-6066

Muriel Kannel
MIT Room 24-021
X 3-6065

LBL Consulting Office
Tel. No. 415-843-1009
FTS No. 451-5981

VIII. REFERENCE MATERIAL

Reference material available in MIT Room 24-019 includes:

1. LBL Users Handbook
2. Selected LBL Writeups
3. LBL FORTRAN IV FTN4 Reference Manual
4. Complete set of Multics Programmers Manuals