DISTRIBUTED COMPUTATION AND TENEX-RELATED ACTIVITIES

J. Burchfiel, et al

Bolt Beranek and Newman, Incorporated

Prepared for:

Office of Naval Research
Defense Advanced Research Projects Agency

May 1975

BOLT BERANEK AND NEWMAN INC

CONSULTING • DEVELOPMENT • RESEARCH

BBN Report No. 3089 May 1975

DISTRIBUTED COMPUTATION AND TENEX-RELATED ACTIVITIES

Quarterly Progress Report No. 2
1 February 1975 to 30 April 1975

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the United States Government.

This research was supported by the Defense Advanced Research Projects Agency under ARPA Order No. 2901. Contract No. N00014-75-C-0773.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER BBN # 3089	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER AD-AC11 131
4. TITLE (and Subtitle) DISTRIBUTED COMPUTATION AND TENEX-RELATED ACTIVITIES		5. TYPE OF REPORT & PERIOD COVERED Quarterly Progress 2/1/75 to 4/30/75 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) J.Burchfiel, R.Thomas, T.Myer, R.Tomlinson		8. CONTRACT OR GRANT NUMBER(s) N00014-75-C-0773
9. PERFORMING ORGANIZATION NAME AND ADDRESS Bolt Beranek and Newman Inc. 50 Moulton Street Cambridge, Mass. 02138		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE May 1975 13. NUMBER OF PAGES 39
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution of this document is unlimited. It may be released to the Clearinghouse, Department of Commerce for sale to the general public.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

This research was supported by the Defense Advanced Research Projects Agency under ARPA Order No. 2935.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

distributed computation
TIP access control
distributed file system
TENEX security
message processing
CINCPAC interactive test
RSEXEC

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes continuing refinement of our TENEX RSEXEC distributed file system which supports geography-independent computing on a number of ARPANET TENEX sites. It also describes our efforts to upgrade existing ARPANET message service to meet NAVY requirements for an interactive message processing test at CINCPAC.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

TABLE OF CONTENTS

		Page
I.	INTRODUCTION	1
II.	DISTRIBUTED COMPUTATION	3
	A. TIP Access Control and Accounting System	3
	B. Distributed File System	5
	C. Coupled Message Service	6
III.	TENEX Related Activities	8
	A. Protocol Development	8
	B. Security	13
	C. TENEX Monitor	18
	D. BCPL	19
IV.	COTCO ACTIVITIES	20
	A. System Development Effort	22
	B. Documentation	26
	C. Human Factors Analysis	28
	D. Initial Planning for Interactive Tests	29
	E. System Level Changes	30
	F. TENEX Security Model and a Multi-mini Sutdy	33
	G. Future Plans	33

I. Introduction

During this quarter, we demonstrated significant progress in our efforts for a distributed file system, internetwork protocols, improved TENEX security, and message technology.

The distributed file system provides a file environment which spans TENEX systems in a geography - independent manner. This environment is provided both to direct user commands and to executing programs: this is done by encapsulating the program, trapping its monitor calls, and interpreting them in the context of a multi-site file system. The interpreter within the encapsulator is undergoing continuous revision to incorporate all the facilities of a single TENEX environment into the multi-machine environment. This process has now progressed to the point where many TENEX tools (editor, compiler, loader) will execute in this distributed environment. The next step is to make our message processing system, MAILSYS, execut efficiently in this environment. This will provide uniform site-independent message services to network users.

We now have an operational Internetwork TCP (transmission control program) on TENEX. The Internet protocol we have implemented is based on "A Protocol for Packet Network Communication," Cerf and Kahn, IEEE Transactions on Communications May 1974. It will support reliable, high-bandwidth communication between hosts and terminals on the same or different networks, (e.g., ARPANET, Packet Radio Net, Satellite Net). Our initial implementation is written as user code and provided as a program

encapsulator which interprets TCP calls of the encapsulated user programs. In the near future, we will demonstrate the ability of TCP's on different logical networks within the ARPANET to communicate through an Internet gateway.

In order to encourage all TENEX sites to switch over to this improved host-host protocol, we plan to:

1. Create TELNET services which utilize the TCP.
2. Create FTP services which utilize the TCP.
3. Perform comparative measurements which demonstrate the improvements achieved in both bandwidth and reliability.

TENEX security has been improved by encrypting passwords in such a way that "clear" passwords are never stored in the system. Other improvements include a "create job" facility which will remove the FTP server from the TENEX security kernal, and a facility to limit the capabilities of a privileged user to protect him and the system from "Trojan Horse" programs which may maliciously attempt to usurp his capabilities.

Our message technology effort has resulted in release of a unified message processing system, MAILSYS, which incorporates both message reading facilities and message creation facilities in a uniform manner. This combination of facilities has permitted us to provide several new services: forwarding of received messages, and convenient means of replying to received messages.

We now have four significant near-term goals:

1. To simplify the user interface to the message system along lines suggested by human factors studies.
2. To interconnect our message service with that provided by the Navy's LDMX systems on AUTODIN I.
3. To make the message service available in a site independent manner by using the TIPSER/RSEXEC to provide a processing environment and the network users' data base to provide addressing aids.
4. To plan for a secure, message-only TENEX service facility at Moffett Field.

II. DISTRIBUTED COMPUTATION

A. TIP Access Control and Accounting System

When the TIP access control mechanism was installed, one of the major negative user reactions was that a TIP user was required to login twice in order to access a network host: once to the TIP/network and once to the target host. During this quarter, while the future of the TIP access control system was being reconsidered as part of the general effort to improve network performance, we designed a solution to this "double login" problem. We documented and submitted the solution for evaluation to the ARPA office and distributed it for review to personnel of several network installations.

The solution we proposed would facilitate access to network

hosts by authenticated TIP users by: providing automatic transfer of a TIP user from the TIPSER/RSEXEC authenticator to the target host; and, providing automatic but controlled access to the target host by eliminating the second (target host) login. Were this solution to be implemented, an authenticated user would be able to say to the TIPSER/RSEXEC that he wishes to use a particular target host. As a result, shortly thereafter, and with no further interactions with the TIP, TIPSER/RSEXEC or target host, the user would find himself directly connected to a logged in job at the target host. (This assumes, of course, that the user is authorized to use the target host, that the target host accepts such logins and that the user has previously arranged with target host personnel to have such logins accepted on his behalf.) The "double login" solution is described in detail in the document submitted to the ARPA office.

At present, the TIP access control and accounting mechanism has been disabled at the direction of the ARPA office and it appears that it will not be re-installed. However, we regard the effort that resulted in the double login solution to have been more than an academic exercise. The mechanisms, such as reconnection and secure authentication of remote processes, that were designed to support the solution are technically interesting and important ones that are applicable to a variety of other situations that arise in the network environment (for example, refer to the coupled message service discussion below).

B. Distributed File System

Our last QPR (BBN Report 3012) reported on a prototype version of the program execution environment which makes the RSEXEC distributed file system available to executing processes. In order to improve the responsiveness of this environment, we are developing a technique for locally "caching" remote files. This mechanism will insure that a remote file which is frequently referenced during an RSEXEC session (by being repeatedly opened, accessed and closed) need not be repeatedly transferred across the network unless the file is modified between references. (Modifications to the file may be the result of the actions of programs at its own site or of those at other sites.) We believe that the file cache technique will significantly improve the responsiveness of RSEXEC sessions consisting of edit, compile, load and execute cycles when these cycles include remotely stored files.

Adding the file cache capability to RSEXEC requires extensions to the RSEXEC file protocol as well as modifications to the RSEXEC module that encapsulates executing programs. The necessary protocol extensions have been designed and they have been implemented and checked out for the RSEXEC server programs. During the next quarter we plan to modify the RSEXEC encapsulator module to implement the file cache logic.

We have made a number of improvements to the file handling capabilities of RSEXEC at the command language level. The most notable improvement is an extension of the "NEED (file)" command

which enables a user to move a number of files to a number of sites in a single command. For example, the command

```
-NEED (file) FILE.ONE [ISI]<JONES>FILE.TWO [LOCAL]FILE.THREE,
--(in component directory) [SRI-AI]<SMITH> [BBN]<JONES>
```

will move a copy of each of the three files to both of the destinations specified. Our software maintenance personnel have found this capability to be extremely useful for maintaining up-to-date copies of subsystems and data bases at a number of sites.

In addition, we have modified the RSEXEC module which performs the GTJFN function for the distributed file system. (The TENEX GTJFN JSYS converts a file name string into an internal representation and performs file name recognition and completion and field defaulting as requested.) These modifications were made so that the module behaves more like the standard TENEX GTJFN and to support several new features that are planned such as the ability to use (the "wild" designator) in the distributed file environment.

C. Coupled Message Service

We have implemented a mailbox locator service which is accessible through the network at the TIPSER/RSEXEC sites. This service returns the location of a network user's mailbox given (a possibly incomplete specification of) his name. Should the name supplied be ambiguous in the sense that it specifies several users (e.g., Smith) the service will return a list of the users. The service uses the network user ID data base that was originally developed to support TIP login.

Our plan is to have MAILSYS use this service to provide user addressing aids. A user who is unsure of another user's mailbox location will be able to cause MAILSYS to access the service on his behalf. Of course, message systems other than MAILSYS can make use of the mailbox locator service also.

In order to support the mail reading capability within TIPSER/RSEXEC for authenticated users, it may be necessary to access mailbox files at remote sites. For example, a user whose mailbox is at BBNB may find himself connected to a TIPSER/RSEXEC at ISI. One way to provide access to his mailbox is for the TIPSER/RSEXEC to act on his behalf to invoke a process running MAILSYS at the mailbox site and to forward his commands to that process. We have designed and implemented in the RSEXEC server programs the machine-to-machine protocols necessary to support such remote program invocation. During the next quarter we plan to add the command interface to the TIPSER/RSEXEC which will use these protocols to access remote mailbox files.

Another way to provide user access to remote mailbox files is to use a reconnection protocol (see RFCs #426 and #671 and the TIP double login solution paper) to connect the user to a MAILSYS process at his mailbox site. We plan to investigate reconnection as a possible alternative to the approach currently being implemented.

III. TENEX Related Activities

A. Protocol Development

1. TENEX Internet Experiment

The coding and checkout of the initial implementation of the user mode transmission control program (TCP) was completed during this quarter. Packets have been successfully transmitted with the internet protocol using the TCP. The current version has efficiency problems. These are being investigated and eliminated as they are found.

Two versions of the TCP have been produced, one which operates on link 157 and one which operates on link 158. The two versions are identical except for this link and also the network number. This permits experiments to be performed as if more than one network exist by partitioning the ARPA net into two logical subnetworks.

An initial pass at producing a version of the TCP for the PDP-11 was made to determine the complexity of making that transformation. It now appears that that transformation will not be as easy as originally anticipated due to the reversed addressing structure of the PDP-11 and the number of double precision operations required. More specifically, each multi-byte field, when stored in the PDP-11, in the format required for transmission to/from the network, cannot be operated on directly but must first be repacked into the natural PDP-11 format. There are two 32-bit

fields and one 16-bit field to which this applies. Repacking these fields on each packet will contribute a certain inefficiency in the PDP-11 reservoir of the TCP.

2. Network Responsiveness

Problems of network unresponsiveness (long-delayed echoing of characters to network terminals) were investigated and traced to conflicts of access to critical NCP tables. References to these tables are interlocked to avoid timing problems. The lack of responsiveness was due to the process that handles background network processing blocking due to another process having gained access to these tables and then losing priority to run, as a consequence of normal scheduler activity. A previously installed mechanism to avoid this problem had been eliminated by the introduction of the "pie-slice" scheduler.

A general solution to the problem was generated which sets aside a special scheduler queue of runnable tasks which require priority service because they have set critical locks, such as the network interlock. As long as such a process has one or more such locks set, it is run with priority. Any benefit that such a process may gain from being able to run with priority is cancelled by a reduction in priority as soon as the lock is released. This action is a natural consequence of the fact that the pie-slice scheduler attempts to equalize the distribution of CPU service according the size of the pie-slice. A process running with priority naturally gets ahead of schedule and is automatically reduced in priority.

3. Interprocess SIGNAL Facility

Performance of the initial version of the Transmission Control Program (TCP) has been somewhat disappointing in terms of throughput. One source of delays centers around communications between the six asynchronous processes which make up the TCP. Currently a processes is told that something is on its input queue by a cross-process pseudo interrupt. Handling such an interrupt requires a considerable amount of code to be executed in the TENEX monitor and is quite dependent on the policy used by the scheduler for determining when to look for new processes to run.

In an attempt to minimize overhead in the monitor a different mechanism has been implemented on an experimental basis -- the SIGNAL facility. Four system calls (JSYS's) are involved: GTSIG, RLSIG, SIGNL, and WTFOR. The first of these creates a signal identifier (SID) and specifies what other jobs if any have access to it. Once created the SID may be broadcast to those other jobs through a shared file.

The WTFOR call is used by a process to wait for a SIGNL event from some other process which as access to the SID. SIGNL and WTFOR may optionally transmit one word of data, which might be a file address where more data is located. In this case, the data transmitted (a buffer in the case of the TCP) is protected in the normal way by the TENEX file system.

RLSIG is used to release an SID back to the system pool.

More detailled information regarding the SIGNAL facility will be published in the forthcoming edition of the TENEX JSYS Manual.

Since the SIGNAL Facility is operational, the next step is to integrate it into a version of the TCP and perform comparison measurements.

B. Security

1. High Security Login in the TENEX monitor.

An important step has been taken toward securing the TENEX operating system by implementing a "High-Security Log-in Procedure", based on the paper of that title by G.B. Purdy in Communications of the ACM, August 1974, pp 442-445. In that paper, Purdy states that "the most vulnerable part of a password system is usually the list of passwords stored in the computer." TENEX has been an example of the truth of this statement.

The Purdy algorithm was implemented in BCPL in the TIP login system by Johnson and Thomas. During this quarter, the algorithm was transferred into the TENEX monitor sources, and a long list of operational support programs, including the TENEX EXEC, were modified and/or tested to assure their operation with the new system.

The TENEX monitor currently running on all systems at BBN does not have a plain-text copy of any passwords available to it. All passwords are encrypted by a modified version of Purdy's algorithm. This encryption is NOT the same as that used for TIP login; the coefficients of the scrambling polynomial are intentionally different. All of the already existing protection checks and reporting mechanisms are still in effect, such as privileges required to read a password from the system, automatic reporting when passwords are read, forced logout on repeated password failures

by a user, and the like. However, should a user succeed in extracting TENEX's copy of some password, he now has only an encrypted form of the password. Even with the algorithm used for encryption in hand, decrypting the password is believed to be practically impossible.

The practical effect of this change is that no user, even a system operator or wheel, can determine what another user's password is. In fact, if a user forgets his password, the operators cannot tell him what it is. They can only create a new one for him.

2. CRJOB - The Create Job JSYS

Further work has been done on the CRJOB JSYS and on programs which will make use of it in further efforts to secure the TENEX system. The CRJOB JSYS has been modified to allow the File Transfer Protocol server process to operate as follows:
Each instance of the File Transfer service will start a job which has the capability to deliver mail (which is sent on the same sockets as other FTP operations). This instance will be in a separate job, and will not be logged in. The amount of code in this job will be small. Upon receipt of a User and Password command, this job will perform a LOGIN JSYS as that user. At this point, if the login is successful - as determined by TENEX's normal validation code - the system will restrict the user to have only the capabilities which he would have when logged in through the conventional means, via the EXEC. This means that any file activity

which he attempts to perform will be checked by the access control mechanisms of the TENEX monitor, not those of the FTP server program.

Only when a successful login has been done will the small program call in the larger system required to perform general file transfer operations. Thus, if the small program can be verified, there is no possibility of doing any file operations in an environment which has more privileges than the user being served.

An unfortunate side effect of this change will be the prohibition of a useful feature of FTP, the "anonymous login" which allows reading of public files from a TENEX desiring to provide this service. This service will probably have to be sacrificed in the goal of security. We are continuing to look into possible alternate ways to safely provide this service.

The modifications to CRJOB described above have been completed. The changes to FTPSRV (the File Transfer Protocol Server) are much more extensive, and are only partially completed. This work will continue in the next quarter. A dummy server has been written which successfully performs the ICP and CRJOB as the first step of this system.

3. "LIMIT CAPABILITIES" EXEC Command

The LIMIT CAPABILITIES command is available to users with wheel, operator, or confidential information access special

capabilities. This command causes TENEX to temporarily prevent use of these capabilities so that a privileged user may leave his terminal unattended without completely denying use of the terminal. This is the typical situation for operator terminals -- they must be available to average users for finding out what scanner lines are free, and the like, but not for creating new users or other privileged operations.

In the "LIMITED" state a wheel may safely run suspect programs -- either those which he himself has written which might malfunction and destroy the file system, or "Trojan Horses" -- programs which grant special capabilities to ordinary users when run by a privileged user.

In order to exercise his special capabilities, the user must type an "ENABLE" command and supply his own password.

4. Specification of "Login-only" Users

Login-only users are permitted to login to TENEX but have no permanent file directory. Instead they may optionally be assigned a default connected directory; lacking this, a temporary working directory will be assigned from a pool. For instance, Smith, Jones, and Doe might might be in one project and share the XYZ directory. When any of these login, he will be automatically connected to <XYZ> and may send or receive mail there or do normal editting and compilations.

This facility would allow a given TENEX system to support several thousand message users since no disk space would be required to store permanent directories for them.

The first step towards implementing login-only users has been taken: the TENEX monitor has been scanned, and all problem areas catalogued. The various support programs such as the archive system and dump programs have also been examined.

C. TENEX Monitor

1. Swapper Performance

As part or the overall attempt to analyze the poor system performance observed while running NLS programs, the disk swapping activity was monitored and evaluated. The analysis was aimed at determining why the swapper did not appear to perform as well as a model of the swapper indicated it should. The model predicted transfer latencies by summing all the known delays encountered in performing a page swap. These delays include those due to controller/channel contention, positioning delays, rotational delays, transfer time, and queuing delay. The discrepancy was traced to a fairly high peak to average traffic ratio. The model predicts a non-linear relation between traffic level and delay. Delay increases more rapidly than linearly with traffic level. The much greater delay resulting during periods of peak traffic levels biases the average delay to a higher level than would be predicted on the basis of the average traffic level.

The results of this analysis indicate that the disk swapper should be able to sustain swap rates of more than 80 pages per second without substantial delays. Furthermore, the performance of the swapper does not appear to be a significant factor in the poor performance of NLS programs.

D. BCPL

We are continuing to re-write the BCPL compiler, using the parsing section of the PDP-11 BCPL cross-compiler and the new code generators written for the PDP-10 as bases for this effort. Many sections of the compiler are being extensively modified in order to produce a compiler that is highly machine-independent and therefore easily transportable. BCPL structures are used instead of the basic machine word slicing operations in all parts of the compiler we consider transportable.

Much of our re-writing is concerned with making a compiler that is self-consistent, structured, and highly self-documenting. Our "clean-up" work consists of renaming obscure variables, formatting the source files in a consistent manner, reordering program flow wherever it is illogical, and in general re-doing anything that is unclear on a first reading.

The re-writing of the lexical analyzer, parser, and tree-building phase (CAE) is essentially finished, and the tree-walking and code-generating phases (TRN and CG) are well on their way to completion.

IV. COTCO ACTIVITIES

We started this quarterly period with an experimental, limited capability version of Mailsys, the BBN message system, which was first put into use in January 1975. The January version was capable of a restricted range of operations on incoming mail; it had no built-in facilities for creating outbound messages. We now have an initial full capability version of the system with message creation tools, and an expanded range of message processing functions. We view this current version as containing the full range of capabilities required for an initial Navy/DOD test. The major work to be accomplished in this area before test commencement will consist of system refinement, rather than the creation of new features. At this writing we have released the current version of our software for experimental use on our own Tenex systems.

Our most intense work during this reporting period went into the creation of the current Mailsys. However, we also devoted considerable effort to system documentation. We have prepared a new tutorial guide to the system, created on-line documentation, and drafted a single page summary guide.

COTCO is dominated by the need to provide effective tools to real users largely outside the academic research community. We have tried to recognize this strong user orientation in the selection and design of basic features in the message software. During the current quarter, we intensified the effort to achieve a user-effective system by initiating a new project activity in human

factors. The human factors group of the BBN experimental psychology department is now working closely with our software development group in the analysis and design of the system. Our hope in initiating this effort is to achieve the kind of balance between function as perceived by the user and internal design structure that characterizes the best human oriented computer software.

Closely related to human factors in the software is the matter of planning for the actual Navy test. For the test to be successful, its plan must reflect a thoughtful evaluation of numerous psychological and experimental design issues. During this quarterly period we completed a preliminary study of the various considerations that should go into a test structure. The result of this initial study will be released shortly in the form of a memorandum. The BBN Human Factors group are also helping us in this portion of our COTCO work.

A second software development effort during this quarter was in the area of underlying TENEX system changes to enhance security and reliability. Finally, we did some initial ground work on two study tasks in the contract: a security analysis of the entire network wide message system, and a study of the potential use of multiple minicomputers for implementing military message processing facilities.

In the paragraphs that follow we will begin with a discussion of the major software development effort during the quarter, and then cover the above topics in order.

A. System Development Effort

During the current quarter, work on Mailsys has focussed on development of a new message creation mechanism integrated directly into the structure of Mailsys, and intended to supersede the earlier SNDMSG subsystem.

As opposed to SNDMSG, the new message create software permits a command driven mode of message composition. The user is presented with a vocabulary of commands through which he assembles the parts of a message in any order he wishes and at his own pace. This contrasts strongly with the internally or prompt driven operating characteristic of SNDMSG in which the user is led through the various stages of message creation in an order determined by the system, and given no way to return to earlier stages.

The new composition software is capable of creating all fields specified by the ARPA Message Service Committee in the current (RFC 680) protocol. Each message field is created through use of a command of the same name. Thus, to create the subject field of a message, the user types the command "SUBJECT" followed by the text that he wishes to appear in the message. This scheme has what we consider to be the elegant feature that command names, the names of message parts, and the identifying labels that appear on these parts when displayed to the user are all uniform. Thus, should the user display the addressed field of the current message, he would see the word "TO:" just as he typed it in.

Since a key goal of this new command driven approach was to permit as much flexibility as possible, a number of commands have been added to the system to manipulate message fields prior to final transmission. Thus, there are commands to ERASE or DISPLAY one, several, or all of the message parts one has created. One can also edit message parts or apply a simple formatting function. The latter resembles Runoff in the sense that it can provide line filling and, if desired, right as well as left justification. Unlike Runoff, however, it requires no embedded commands, and has the unique property that it can be applied repeatedly to the same body of text without any ill effect (try that with Runoff).

In keeping with the design reflected in RFC 680, there can be multiple instances of any message field in a given message. (RFC 680 treats all message parts except the main text as single line fields, but permits multiple instances of each of these fields. The message text is treated as an intrinsically multi-lined field.) There are two ways which to create such instances. While typing in its contents, one can extend any field into additional lines by preceding each intermediate carriage return with a line continuation character. Multi-line fields can also be created by repeated use of any field creation command (scattered as desired). Each use of such a command appends new lines of information onto any pre-existing field contents.

It is also possible to add the contents of a TENEX file to any message field, either through use of a special control character

during the type-in process, or by a command which appends the contents of a named file to a named message field. The major intent of these features is to permit the insertion of address lists into the addressee field of the message or arbitrary pre-existing text to the message body. However, the new commands also make it possible to add a file to any message field.

The ability to insert file contents into message fields was a desirable feature of SNDMSG. Other SNDMSG features have also been carried over into the new composition subsystem. As with SNDMSG the new system checks local addressees and the names of foreign hosts. It can handle addressees lists (using the same format as SNDMSG) and attention specifications within an addressee. The new subsystem also permits messages to be addressed to TENEX files through the inclusion of a new message field called FCC (file carbon copy). As suggested above, the main intent of this command driven approach was to let the user, rather than the system, guide the message creation process.

The majority of our local BBN users so far have liked this approach, but they represent a sample population heavily biased toward long experience with the ARPA message service, and so it seemed to us that it might be desirable to retain a prompt driven form of message composition for new users, and anyone else preferring this form of input. To make this possible, we created a new command in Mailsys, called SNDMSG, which, however, unlike the old SNDMSG, invokes a prompt sequence composed of message field

commands from the new system. The major difference between this and the old SNDMSG is that the user is not forced to make a choice between sending the message and aborting the whole process. Rather, after message composition is complete, he is free to apply any of the editing or creation commands above before final transmission of his message. As well as commands for field entry and manipulation, the new composition subsystem provide a command for message transmission, and a command that invokes the subsystem MAILER for message pickup and delivery.

With message composition and processing functions integrated into a single system, it became possible to create interlinkages between them. This possibility was taken advantage of in the preparation of three new commands each of which extracts portions of received messages in order to assist the user in the preparation of outgoing messages. The command REPLY maps the FROM field of a received message over into the TO field of an outgoing message, maps the SUBJECT to SUBJECT field, sets up an IN-REPLY-TO field, and then pauses for the user to enter the text of the reply message. The REPLY command can also map all TO and CC recipients of the received message into the outbound CC field. The command FORWARD takes as argument a list of items from the current inbox, assists the user in setting up basic message header fields, and then copies the indicated items into the text of the outgoing message. Both FORWARD and REPLY use a prompt driven technique similar to SNDMSG. By contrast, the command INCLUDE, makes it possible to append copies of one or more received messages to the text of an outgoing message

without resorting to a prompt driven approach.

The new composition software also includes commands that make it possible to direct outgoing messages to the data computer for archiving purposes, and to retrieve previously archived messages from the data computer.

B. Documentation

During this period we have created a revised and updated version of existing user documentation and prepared new user documents. We now have a manual intended as a guide for the user who is naive not only about computer message service, but possibly also about computers in general and computer networks. The new guide reflects a tutorial approach toward organization and style of exposition. It makes heavy use of examples to illustrate the details of message processing composition, and attempts to create simple mental models that will assist the user in comprehending not only Mailsys, but its computer and network environment.

It is anticipated that message creating software will sometimes by used by infrequent users with no knowledge at all of computers, networks, or computer message processing, and little desire to acquire such knowledge. In order to bring the system within reach of this extreme element of our expected population, we are also preparing a user document which we have elected to call a "Cheatsheet". This will be a one page card to be attached to appropriate computer terminals. It will contain the minimal

information needed to allow a totally naive user to log in, read any messages that may be awaiting him, and create simple outgoing messages. The Cheatsheet will contain clear, step by step instructions, but no attempt will be made to provide explanatory material or create suitable mental models of the system. We expect to make use of graphic presentation techniques to make clear the appropriate sequences of operation in message manipulation.

During this period we have also updated the on-line documentation as required to keep in reasonable synchronization with the rapidly developing state of the system. We have also added some new features. In order to help frequent users keep track of the evolving system, we have created a command called NEWS, which tabulates in reverse chronology a capsule summary of all visible system changes accompanying each new release. Typing "NEWS" will produced a listing that summarizes all of the system changes mentioned in this progress report together with any number of smaller changes and refinements. The on-line HELP command now generates a capsule summary of the entire system. HELP is intended to provide a much briefer form of the information spelled out when the DESCRIBE ALL" command is given.

We have also made a change in the on-line documentation for the sake of efficiency and ease of maintenance. The on-line material is now contained in a special file that accompanies the basic MAILSYS.SAV file. This not only reduces the core memory requirement for the system, but also makes it possible to edit and update the

material contained without needing to reload the entire Mailsys system.

C. Human Factors Analysis

The major goal of this effort is to establish guidelines for system design that will assure that the end product meets the needs of users inexperienced with computers and unfamiliar with the various environmental "shells" that surround the message system. In order to be successful in the long run, the system must not only perform the functions needed by these users, but also be attractive. This means in particular, that many "features" and details of operating procedure that appear natural, tolerable, or even desirable to sophisticated users such as the system developers, must be thrown open to question in the light of the eventual user population.

In order to guard against undue bias caused by past experience with computers, the human factors group selected a team of analysts with varying computer experience, including one individual with no past experience at all. As a vehicle for system evaluation, we worked out with them a series of graded exercises in the use of Mailsys, beginning with very basic message manipulations, and ending with tasks intended to exploit the full range of features currently available in the system.

Based on these exercises and subsequent analysis, the human factors group developed a set of conclusions which were presented in

a series of memoranda. Strong emphasis was placed on simplicity and a parallel notion that less can be more when it comes to the design of an effective working tool. The user of a message system must be given as straightforward and simple as possible a conceptual model of what is going on.

The naive user will experience great difficulty in grasping the various hardware and software "levels" that surround the Mailsys software. In particular, the sequence of TIP LOGIN, TENEX LOGIN, and subsystem invocation, should be done away with in favor as simple as possible a one level protocol that leads the user directly into Mailsys. By the same token, the notion of "inferior" forks for running Teco or a subsidiary Exec will most likely be anathema to a naive user. In similar fashion, the full power of Tenex file designators is probably more than the naive user will wish to understand or have use for. Beyond this, the human factors group had a number of suggestions for streamlining the inner workings of Mailsys, and improving the documentation. Many of these are reflected in the design goals we have established for the next round of Mailsys refinement (see below).

D. Initial Planning for Interactive Tests

As suggested previously, since the focus of this development project is concentrated on a single future event, namely the interactive COTCO test, we felt it highly desirable to begin some exploratory planning for that test as early as possible in the

project. Consequently, working with the human factors group, we spent some time exploring a number of issues that we felt would be important in assuring the effectiveness of the test. Initial conclusions about these issues will be released shortly in the form of a memorandum.

The review included such features as goals for the test, the selection of test subjects, experimental design so as to maximize control, minimize bias and maximize information gained; the creation of appropriate statistical measurement tools. An important recommendation is that there be a pilot study prior to the actual test in order to make final adjustment to the system before beginning to gather data on its effectiveness. We feel this is quite important, because there is no way that the system can be precisely tuned to military needs without being exposed to actual military users. An initial shakedown with a very small group of friendly users, should go a long way toward making the system effective in the actual test.

E. System Level Changes

During this reporting period we made certain changes to the TENEX operating system to enhance the security, reliability, and in certain cases human factors of message service operation. These are outlined briefly below.

High Security Login Procedure

We designed and have installed on all BBN Tenex systems a high security login procedure that uses an irreversible transform technique to safeguard the secrecy of user password strings. In particular, only the transform of each password is stored in the system. From this transform there is no effective way to get back the original string, and so even a breach in system security that lead to theft of the transform tables would not compromised the integrity of the actual passwords.

RCTE TELNET Option

This provides for remote character by character echoing on TIP connected terminals so as to avoid the long delays often associated with single character transmission through the Network. At this writing, the TENEX code to effect the RCTE option is substantially complete, and we await completion of the required modification to the TIP code.

Capabilities from File

A critical issue in ensuring the security of the message system is to provide system processes that run as inferior forks to user jobs with higher levels of enabled capability than are permitted the driving user fork. We have completed an initial design specification for software to provide this capability, and expect to complete the final specification in the subsequent period.

Absolute Socket Capability

We have installed changes to the TENEX Operating System that make it possible for privileged subsystems, including the Mailer subsystem, to secure an absolute numbered socket. This capability is one link in the security chain that provides sender authentication of messages.

Secure Message Facility

A military requirement is anticipated for a TENEX site dedicated exclusively to the needs of military message processing. In order to provide the required levels of security in such a site, a number of changes must be made to the TENEX system. During this period, we explored these and have drafted an initial plan for implementing such a system. Fortunately, the achievement of security is aided by the fact that the facility will be a single purpose system. Thus, the implementation plan emphasizes the need to remove access to many programmer oriented features provided in the standard TENEX operating system.

F. TENEX Security Model and a Multi-mini Study

During this reporting period we performed a small amount of preliminary work in these two areas. In particular, we have prepared an outline for the security analysis and model to be undertaken this fall, and have drafted some initial ideas on the topic of multiple mini-computer networks. In the latter, we suggest a modification of the basic multi-mini idea to take advantage of the economies of scale that are possible when major data management and other "back-end" processing tasks are put into the hands of larger machines.

G. Future Plans

Our most important immediate plan is to continue our revision and refinement of the system so as to tailor it to its military users. The following list spells out a number of specific project goals designed to effect this refinement. Along with refining the system software, we will continue our work on refining and updating the various types of user documentation.

The Human factors analysis which commenced during the last period is felt to have been a highly valuable addition to the project. Consequently, this activity will continue at about the same pace during the ensuing period. We also expect to continue our work on more detailed planning for the COTCO test during this forthcoming period.

The test will require provision of TENEX software in order to effect an interface between the ARPA network and the LDMX computers of the military. We expect to develop such software during this next quarter.

In the area of system security, we expect to continue our work on underlying refinements to TENEX, to commence our security study and analysis, and to begin development of the secure, message only TENEX site which we described briefly above.

During this period we also expect to begin a general overhaul of our software in order to achieve a sufficient degree of modularity for integration into other related work, such as the efforts now underway at ISI.

A critical component of military message processing is the coordination phase in which a message passes through possibly multiple levels of review prior to final release. We have discussed with NAVELEX a number of basic coordination features that could be built into our software.

We have developed plans for a more highly efficient and secure message distribution and delivery system then the one currently in effect on the ARPANET. This system will emphasize the use a central cache for received message at each site, the delivery of just one copy of a message to any given site, and the provision for status feedback and an audit trail capability. We expect to begin design and development work on this distribution system toward the end of

the current reporting period.