II

BOLT BERANEK AND NEWMAN INC

CONSULTING • DEVELOPMENT • RESEARCH

Report No. 2852

July 1974

INTERFACE MESSAGE PROCESSORS FOR THE ARPA COMPUTER NETWORK

QUARTERLY TECHNICAL REPORT NO. 6

1 April 1974 to 30 June 1974

Submitted to:

IMP Program Manager
Range Measurements Lab.
Building 981
Patrick Air Force Base
Cocoa Beach, Florida 32925

This research was supported by the Advanced Research Projects Agency of the Department of Defense and monitored by the Range Measurements Laboratory under Contract No. F08606-73-C-0027.

INTERFACE MESSAGE PROCESSORS FOR THE ARPA COMPUTER NETWORK

QUARTERLY TECHNICAL REPORT NO. 5
1 April 1974 to 30 June 1974

Submitted to:

IMP Program Manager
Range Measurements Lab.
Building 981
Patrick Air Force Base
Cocoa Beach, Florida 32925

This research was supported by the Advanced Research Projects Agency of the Department of Defense and monitored by the Range Measurements Laboratory under Contract No. F08606-73-C-0027.

TABLE OF CONTENTS

1. OVERVIEW

This Quarterly Technical Report, Number 6, describes aspects of our work on the ARPA Computer Network under Contract No. F0806-73-C-0027 during the second quarter of 1974. (Work performed from 1969 through 1972 under Contract No. DAHC-69-C-0179 has been reported in an earlier series of Quarterly Technical Reports, numbered 1-16).

During this quarter, no new network nodes were delivered. At the end of the quarter, the Real Time Clock retrofit program described in Quarterly Technical Report Number 4 had been completed at all 316 IMP and TIP sites with the exception of Hawaii. We expect very soon to begin similar retrofits for the 516 IMP sites.

The Very Distant Host program has been extended this quarter to support up to four Very Distant Hosts on an IMP. The code has been introduced into the network, and is now operational.

Among our activities in the Satellite IMP project this quarter were further interactions with COMSAT, including a proposal for BBN to provide Satellite IMPs directly to COMSAT under a "Use Charge" agreement. We hope that this may permit installation of Satellite IMPs in COMSAT ground stations.

Development of the 316 Satellite IMP program has continued with the introduction of statistics and tracing similar in form to the IMP statistics and programming of other channel protocols. The program now has a program settable switch which selects either Slotted ALOHA, TDMA, or a rudimentary Reservation ALOHA protocol. This switch may be changed during operation without adverse effects. In addition to our work on the 316 Satellite IMP program, we also began development of the program for the Pluribus Satellite IMP.

During this quarter we experienced considerable difficulty in accomplishing the release of new IMP software on one occasion (May 14). This trouble resulted in several hours of interrupted service to many Hosts and users, due to problems both with the IMP to which the users were attached, and also more global problems. These problems did not manifest themselves in our pre-release test procedures, or in the first part of the release. It was not until more than 30 IMPs were reloaded with the new version that any software crashes occurred. The release was therefore completed (somewhat slower than usual) and the software staff worked to solve the network problems for about 3 hours. During this time about 20 IMPs had to be reloaded, and the staff still had not found the underlying cause. At that point, the decision was made to withdraw the release, which took about 1 1/2 hours more.

Subsequent debugging uncovered a bug that had been dormant in the IMP for more than a year (an interrupt bug in which a resource was released, in two stages, with interrupts enabled.) The bug had been changed from a harmless oversight to a relatively high probability event (on the order of once per 10 machine-hours of normal operation) by a change in some conventions necessitated by the new message number resynchronization code. The pre-release testing apparently did not involve the right circumstances to provoke this particular bug: steady traffic, at a high rate, for long periods from an IMP with considerable line traffic. Of course, finding such bugs is quite a difficult process, and devising appropriate checkout procedures is also difficult. We will continue our efforts to increase the reliability of the network, including the task of producing reliable software without disrupting normal network operations.

We presented two professional papers and issued one new report during the quarter. The papers were: "Some Computer Network Interconnection Issues," by A.A. McKenzie, at the National Computer Conference and Exposition, Chicago, Illinois, May 1974; and "Networks and the Life Sciences: The ARPA Network and Telenet," by F.E. Heart, at the FASEB Conference on the Computer as a Research Tool in the Life Sciences, Aspen, Colorado, June 1974. The report, "Adaptive Routing Algorithms for Distributed Computer Networks" (BBN Report No. 2831), by J.M. McQuillan, is discussed in Section 2 below.

Subsequent sections of this Quarterly Technical Report describe a plan for access control in the ARPA Network; progress on the development of the Pluribus IMP; TEMPEST considerations for the Private Line Interface; and the technical characteristics of the Remote Job Entry mini-Host system.

2. ROUTING STUDY

We recently completed a detailed study of routing, described in BBN Report 2831, "Adaptive Routing Algorithms for Distributed Computer Networks".* That report has two primary objectives: first, to provide a broad introduction to the subject of system design for computer networks, including the specification of the communications algorithms, and, second, to present a deeper discussion of routing algorithms for such networks. We hope that it will prove to be a valuable tool for other network designers and implementers.

The first half of the report contains some historical background, and a systematic outline of the basic technical considerations. There is also an extensive annotated bibliography, and several mathematical analyses of key network parameters. The aim of the first part of the report is to provide a balanced and complete framework for the subsequent investigation of routing algorithms.

In the report, we include a terminology for packet-switching networks, and define the basic variables and parameters which are important in subsequent analysis. The analytic material, some of which has been presented in earlier papers and QTRs, includes the following parameters:

*The author, John M. McQuillan, also submitted this report as his Ph.D. thesis at Harvard University.

- Delay: the components of delay, including those critical in minimum round-trip delay, the effect of packet size on delay, and so on.

- Throughput: the factors determining processing bandwidth of a store-and-forward node, the overhead on network circuits, buffering required, both on lines and paths, and tradeoffs.

- Cost: the cost of network connectivity and use, in relation to the other parameters.

- Reliability: the reliability of network connectivity and use, in relation to the other parameters.

Then the process of designing a routing algorithm is examined:

- The specification of an algorithm in terms of its inputs and outputs, basically data on the topology and traffic of the network.

- The specification of the processing goals of the algorithm, such as simplicity, reliability, correct steady state solution and adaptation to change, global optimality and fairness.

- The evaluation of the performance of a routing algorithm in terms of delay, bandwidth, cost and reliability.

- The evaluation of the cost of a routing algorithm in terms of nodal delay, bandwidth and storage, and line delay and bandwidth.

The second half of the report contains the main results of our research into routing algorithms, based largely on our experience with the ARPA Network. It takes up questions which have been addressed previously by other authors, as well as subject matter on which very little published work exists to date. The study begins with a classification of routing processes based on the structure used for each of four functional components. That analysis forms the foundation for a detailed discussion of the synthesis of routing algorithms.

The second half of the report is divided into two main sections: one on the classification or analysis of routing algorithms, and the other on the construction or synthesis of routing algorithms. The classification scheme put forward is based on a functional analysis of the operation of a routing algorithm; it is used in the second part as a guideline for constructing a routing policy to fit a given set of circumstances.

Some of the most important new results are:

- A control scheme in which each network node has independent decision-making power ("distributed" control) is shown to have significant advantages over other alternatives. This approach is the one in the ARPA Network, and it has clear advantages over fixed, isolated and centralized structures.

- New techniques for the rapid and accurate determination of whether a path exists to a node, under changing network conditions, are described and analyzed. The use of "hold down", a new solution to this problem, speeds up the decision process greatly, and also eliminates the need for explicit path length computations, which in turn reduces storage requirements. The problem of reachability determination is fundamental to routing, and therefore this technique is of central importance.

- Problems in traffic assignment, such as the use of multiple paths, and resolution of contention among several traffic sources, are solved through the use of variable frequency updating, and the communication of excess capacity among nodes. Again, this technique represents a very efficient solution to a fundamental network problem, one that does not depend on explicit examination of all possibilities, or knowledge of the network topology.

- The reliability of a distributed routing process is examined carefully, and a set of proposals is advanced to protect this function from any local component failure, no matter how severe. The experience of the ARPA Network routing algorithm, and the many safeguards we have installed, are described in depth here.

After the presentation of these general results, some new problem areas and specialized results are examined in subsequent sections. Much of this research is not complete, but some important conclusions are presented in the areas of heterogeneous networks, and very large networks which would require significant changes to the structure and operation of the routing algorithms examined in the report. Networks with broadcast links and networks with interconnections to other networks are also studied. The main conclusions include:

- Networks with widely differing components require novel techniques for the representation and manipulation of routing variables.

- The simple approach to area routing, in which each node is part of a single area given as part of its address, is shown to have significant drawbacks similar to those for non-adaptive routing.

- Some adaptive area routing strategies are examined, using the concepts of algorithm structure developed previously, and their advantages and disadvantages over fixed area policies are pointed out.

3. THE PLAN FOR ARPA NETWORK ACCESS AND USAGE CONTROL

In response to ARPA's requests, we have developed and are now in the process of implementing a plan which will provide accounting for the usage of the network packet-switching resources by the Hosts and TIP users, control over TIP access to users, fairness in the Host's use of the IMP's resources, and an implementation of the concept of logical subnetworks. In the areas of the RSEXEC and the TIP user data bases, the work will be done primarily by the Computer Sciences Division at BBN rather than the IMP and TIP development groups.

3.1 Accounting for Host Use of the ARPA Communications Subnetwork

The Network Control Center (NCC) continuously accumulates (in addition to other statistics) the following usage data on each network Host: total number of packets sent to Hosts on other IMPs, and total number of packets sent to Hosts on the same IMP as the sending Host. Each month the monthly accumulation of this data for each Host will be sent to the person responsible for each Host in the form of an invoice for packet transmission services used. A copy of all such invoices will also be sent to ARPA.

The Computer Sciences Division will also construct and the NCC will maintain a data base giving the allocation for each network Host. Additions to a Host's allocation will be made at the direction of ARPA. The allocation for each Host will be decremented according to the Host's usage. When a Host's allocation is exhausted, ARPA and the Host will be notified. Arrangements will be made to avoid invoicing a Host for traffic during the first few weeks after initial connection to the network to allow for initial connection checkout.

3.2 TIP Access Control and User Accounting

Our solution to the problem of TIP access control and user accounting will be based on TIP use of the TIPSER/RSEXEC.

When a user dials into a TIP, the TIP will automatically do a broadcast ICP to the (most responsive) TIPSER/RESEXEC. The RSEXEC will ask the user for his name and password. If the user gives a legitimate name and password and if the RSEXEC database indicates that the user is authorized to use that TIP, the RSEXEC will send the TIP a control message stating that the user has been authenticated; in addition it will send the TIP an identifying number for the user which the TIP will use later when it sends the RSEXEC usage accounting information on the user. If the user does not successfully identify himself to the RSEXEC, the RSEXEC will break the connection with the TIP, and the TIP will hang up on the user. If the user has been authenticated, the TIP will count the number of messages sent by the user and the user's connect time so the user may be charged for them. At periods throughout the user's session (and unbeknownst to the user), the TIP will send the user's incremental message count and connect time to an available RSEXEC; we will call this an accounting checkpoint. To end his session, the TIP user hangs up. This action causes the TIP to send the RSEXEC final accounting information for the user's session.

The TIP will only count messages sent, not those received. This is consistent with our Host accounting plan in which messages are counted as they enter the network. Thus the user will be charged for messages sent from the TIP at the TIP end and for messages received at the TIP at the serving Host end. Receive-only terminals such as line printers would not be charged.

Each month, ARPA-designated Principal Investigators will be sent TIP usage data for all their minions, and each TIP "owner" will be sent TIP usage data for his TIP. There will be no account numbers for TIP users; that is, a user has permission to use a given TIP or not to use it — no distinction is made about what job he is using the TIP for. A TIP user will not be cut off when some arbitrary allotment of connect time or number of messages has been used; a TIP user's allotment is for some time period (most probably one year) independent of usage during that period.

No allowance will be made for work lost due to TIP crashes, hung connections, etc.

3.3 Fairness

There has been concern expressed by ARPA about guarantees that one Host on an IMP doesn't somehow unfairly usurp all the IMP's resources to the detriment of the other Hosts on the IMP. In fact, we have heard no complaints in this area, and we believe that the IMP already behaves quite fairly regarding Host usage of the IMP's resources. However, we have reconsidered this issue and now make the following statements.

A. There is little problem at a source IMP as input from the Hosts is interrupt driven in the fairest possible way with necessary background processing done in a fair round-robin (random) order. The interrupt and round-robin background structure of the program naturally distributes the available CPU bandwidth in a fair manner. Since the CPU bandwidth is obtained randomly (fairly), the IMP Host input routines naturally acquire buffer storage also in a random (fair) manner. The CPU bandwidth and source IMP buffer storage represent no problem.

B. There is a slightly greater problem at a source IMP in acquiring message numbers for transmission of messages to a destination IMP. Again, the various Hosts on an IMP randomly make demands for message numbers, so in some sense it is fair. However, message numbers are presently such a scarce resource that moderately heavy use of message numbers to a given destination can interfere with other Hosts on the same IMP trying to communicate to the same destination. We propose two steps to reduce this interference between Hosts on an IMP: first, we will expand the source/destination message number window from four to eight numbers; second and a little later, we will go to a hashed message number table scheme in which all of the Hosts on an IMP share a reasonably large pool of message numbers in a very dynamic way so that statistically the effects of Host interference are greatly reduced (controls will be put on the pool so that all Hosts may always acquire some message numbers while allowing one Host to acquire most of them in the absence of other Host traffic).

C. At a destination IMP the fairness problem is greater as a slow Host can tie up buffer space for abnormally long periods of time thus depriving the other Hosts on the IMP of the use of this space. We will solve this problem by putting controls on destination IMP buffer storage such that when several Hosts are actively competing for the storage, it is divided fairly between them while permitting one Host to use most of the buffer storage in the absence of other Host usage.

3.4 Logical Subnetworks

In an operational network it is not reasonable to assume that every Host can or should be able to communicate with every other Host; some Host/Host access control mechanism must be provided in the communications subnetwork. We suggest the following mechanism:

Every IMP would maintain for each of its Hosts a pair of Host Access Control Words. Each of these words is 16 bits long and the individual bits in these words indicate membership in one of sixteen logical subnetworks or the ability to communicate with Hosts in one of the sixteen logical subnetworks. The first of the pair of words indicates which of the sixteen logical subnetworks the Host belongs to. The second of the pair of words indicates which of the sixteen logical subnetworks the Host may not belong to, but which contain Hosts with which the Host can nevertheless communicate. These words are regularly reported to the NCC in the IMP status messages to assure their correctness.

The Host access control words are used as follows: a pair of Hosts may communicate with each other only if they are members of the same logical subnetwork or if one is allowed to communicate with Hosts in a logical subnetwork of which the other is a member. For example, Hosts A and B are members of the logical subnetwork of ARPA researchers, Host C is a member of the logical subnetwork of Air Force weather workers, and Host D is a member of the logical subnetwork of service Hosts. Further, say Hosts B and C are marked as able to communicate with the service Host logical subnetwork. In this example, then, Hosts A and B can communicate because they are members of the same logical subnetwork. Hosts A and C are not able to communicate with each other because they are neither members of a common logical subnetwork nor is either marked as able to talk to Hosts in a logical subnetwork of which the other is a member; the same rules hold for B and C. Hosts B and C can talk to Host D because they are both marked as being able to talk to Hosts in the logical subnetwork of service Hosts. Note that though Hosts B and C can both talk to Host D this does not imply that B can talk to C.

The design for these changes is now done for some and well under way for others. Implementation has begun in some areas. We expect to finish these changes in the fourth quarter of this year.

6.2.2 User Level Protocols

The RJE mini-Host implements, in some form, three so-called "user" level protocols: these are Telnet, the Initial Connection Protocol (ICP), and the File Transfer Protocol (FTP). In general, the current version of the system provides the default version of each protocol for the "user" side. The system is not a "server" in the ARPA Network sense.

A user ICP is provided according to specification. The system will perform the correct sequence of commands to a designated ICP, or "listen", socket. This connection sequence may be initiated directly by a user command specifying the foreign Host number and ICP socket number. It may also be requested implicitly by the user setting up for a file transfer; in this case, the FTP process will automatically initiate the ICP to the FTP logger socket at the appropriate time.

At this writing, the Telnet protocol itself exists in two versions which differ significantly. The old version is relatively simple and uses various "special" characters to represent Telnet commands. The new version defines a command syntax using a single escape character and provides a structure for negotiated options between Hosts. This new Telnet protocol has not yet been universally implemented. Where necessary the RJE mini-Host will provide the old protocol; wherever possible it will use the new. A minimal user Telnet is implemented according to the new Telnet protocol. The current system does not support any options and will refuse any option requests that may come in. The RJE mini-Host will not send the Telnet go ahead (GA) command. Since the only terminal now supported is non-interactive, it is not clear what action the system should take on receiving a GA command; currently, the system takes no action. The control functions described in the Telnet protocol are not provided locally and hence are not provided to network users. In addition, there is currently no way for a user on the RJE mini-Host to direct the system to send these control functions. It is understood that changes or additions to the current implementation of the Telnet protocol may be necessary or desirable.

A user FTP exists for the default case of stream mode, ASCII Non-print type, and file structure with an 8 bit transfer byte size. No options are implemented in the current version. Various additional possibilities that might be useful for Remote Job Entry users, however, seem relatively easy to implement and could perhaps be added given sufficient interest. These are: EBCDIC code, Telnet format effectors and carriage control representation types, and record structure. A description of the procedure for using FTP on the RJE mini-Host appears in section 6.4.

6.3 The RJE Mini-Host Command Language

The RJE mini-Host implements a local command language to allow the user to set parameters and to initiate actions. An example of the former function is setting up user parameters for the File Transfer Protocol, such as user identification and account number. An example of the latter function is requesting that a connection be opened between the user and some foreign Host. All commands follow the same general format: the command is initiated by an escape character, which is followed by a command number in octal, which is sometimes followed by one or more parameters; the command is ended with a terminator character,

6.2 Implementation of ARPA Network Protocols

As a general comment, the current version of the RJE mini-Host attempts to implement protocols correctly though in a sometimes minimal fashion; a few exceptions to protocol are noted as appropriate. It is understood that the system will likely grow to fix these exceptions and to include some "useful" options as these become clear (and time and core memory permit).

6.2.1 "Network" Level

The RJE mini-Host is a Host computer on the Arpanet. As such, it performs the IMP-Host and Host-Host protocols necessary to connect to the network and to communicate with other Host computers.

The current version of the RJE mini-Host includes a few exceptions to the Host-Host protocol, namely that the system will ignore the incoming commands "give back" (GVB) and "interrupt by receiver" (INR). It is believed that these commands are seldom used, but the current version is nonetheless violating protocol by ignoring them. It may also be noted that the RJE mini-Host, in its current version, will not send the commands "give back" (GVB), Echo (ECO), or Error (ERR).

A problem faced by all Hosts is that of correctly replying to "unsolicited" Host/Host commands; an example is an unexpected request for connection (RFC) to which a Host should reply with a close (CLS) rather than ignoring it. The RJE mini-Host places such commands on a queue for prompt attention. The available space for saving these commands (currently about 50-60 words) should assure replies in all but pathological cases, for instance a "broken" NCP sending out streams of such commands.

command appear on each card; it need not begin with the first column.

6.4.1 Using the FTP Procedure

The anticipated mode of operation is that involving the ARPA Network File Transfer Protocol. What "user engineering" exists in the current system has been concentrated here. The concept of the FTP is that the user maintains two separate connections with the server Host. The first is the Telnet connection which is used for control purposes; FTP commands and replies are exchanged over this connection. The RJE mini-Host automatically performs the required command sequence for the user once the parameters have been set and the user gives a retrieve or store command. The second connection is used for the data file being transferred; the data is considered transparent to both Telnet and local command interpretation.

Ideally, an FTP user has access to more than one device; for instance, a Teletype-like device associated with the control connection, and readers, printers, or other devices associated with the data connection. In such cases, both control information and data can be exchanged independently; a system can always look for commands from the "control device" and never need look for them from the "data device". Since the IBM 2780 has only a single non-interactive device for input, this "ideal" implementation is clearly impossible; it is instead necessary to define an operational means of distinguishing data and control characters. The FTP procedure is thus divided into two distinct phases: in the first (and initial) phase, characters read from cards are considered control information by the system and interpreted for commands; when the command to Retrieve or Store a file is given (number 25 or 26), the system shifts to the second phase and assumes the file transfer is taking place. In the case of the Store command, characters read from cards are assumed now to be transparent data and are sent over the FTP data connection; the system will not intercept the local command escape character. In the case of the Retrieve command, data will arrive from the FTP server Host via the data connection and be sent out to the printer or punch; the system will neither expect nor accept characters from the card reader until the file transfer is complete. As with any "non-intercept" mode, this second phase poses the problem of when and how to end it, and return to a control phase. Local commands cannot be used since the system specifically stops looking for them in this second phase. The RJE mini-Host will use the FTP end-of-file condition to return to control mode. The EOF condition causes the system automatically to terminate its FTP process, release resources, and generally clean up and return itself to an initial state ready to begin another FTP transaction. In performing a Retrieve (data from the server), the EOF is sent or indicated by the server; in a Store (data from the IBM 2780), the EOT character sent by the terminal will be translated into the appropriate EOF character or action by the system. The EOT character is sent by the IBM 2780 terminal when the end-of-file key has been depressed *and* the card reader runs out of cards.

The current system does not allow re-initialization or multiple files to be sent to the same Host; each file must be a separate transaction as described above. The same physical, or dial-in, connection between the RJE mini-Host and the terminal, however, may be retained for as many transactions as the user desires. In addition to the EOF condition, various other error states may cause the system to abort an FTP process and return the terminal to the initial control phase; such conditions are currently any problem with the FTP process. (It is expected that experience with real users may define more appropriate or specific conditions, perhaps including well-defined user action, to cause an abort.) The system should print a message to the user indicating that the process has been aborted and giving a reason, however. The prototype RJE mini-Host may not always provide such messages.

Specific steps for doing file transfer are described below.

1) Connecting to the RJE mini-Host.
Initially, the user should power on the 2780 and dial in to the RJE mini-Host; an indicator light on the 2780 console will light on receipt of data terminal ready.

2) Setting the 2780 to ready.
The card reader must be cleared of any cards and placed in a ready mode. Since the user initiates the procedure by sending a sequence of cards, the mode switch should initially be set to transmit. The printer should be generally ready to go (e.g. paper in place) but need not be set "ready" at this time. If the user expects to use the card punch, it should be known to work, with a supply of blank cards handy.

3) The control deck
An FTP process will be initiated in accordance with a control deck containing commands to the RJE mini-Host (as described in the section on commands). The first card must contain a command #20, which causes the system

to reserve the necessary resources for the file transfer process. This command takes a parameter, the number of the Host (in octal) designated as the FTP server for this transaction. Following this initial command card are a number of cards containing the various parameter commands. These in general may appear in any order (unless noted otherwise); the system saves the parameters and sends them in the proper order to the server. Different FTP servers may require different parameters; for instance one Host may require an account number, another not. It is assumed that the user will know and supply the appropriate set of parameters required by the designated server Host. Should a required parameter be missing, the process will abort. Because the current system only supports the default conditions, there are only four parameter commands implemented: user ID, password, account number, and pathname. As options are provided, further parameter commands will be defined. The last command card in the control deck, which follows the set of parameter commands, is an action command. On receiving this command, the RJE mini-Host will take over control and automatically initiate and perform the file transfer process. The system will not interpret further commands until success or failure of the transfer has occured.

4) Entering the control deck (this may change).
To begin the procedure, the user should place the control deck in the card hopper and press the start key and the end-of-file key. When the deck runs out, the user should change the mode switch to Receive and set the printer to ready. The system will print out a message indicating

success or failure. On receipt of a success message the user may proceed with steps 5 or 6. On receipt of a failure message the user may assume the system is again initialized and proceed to step one and try again; the failure message should contain some indication of why the process was aborted.

5) Retrieving a file (from the server).
If the transfer request was Retrieve (#25), then the punch or printer (as appropriate) should be placed in a ready condition and the mode switch left in Receive. The file should arrive and be punched or printed automatically. At the end of the file (or on abort), a message will be sent to the printer indicating success or failure. In either case, once a success or failure message has been printed the system is again ready to accept a control deck as in step one.

6) Storing a file (send to the server).
If the transfer request was Store (#26), then the system expects to read the data file from cards. The mode switch must be set back to Transmit and the reader set ready (the card reader may again need to be cleared). The user should place the card deck in the reader and press the start and the end-of-file keys. The cards should then be read by the system and sent to the server Host. When the hopper is empty , the terminal will transmit an EOT character to the RJE mini-Host; this character is translated into the appropriate EOF marker by the system. Once the hopper empties, the user should again turn the mode switch to Receive and put the printer in a ready state. The system will print a success or

failure message and the user may then proceed with step one.

6.4.2 Telnet Connection to a Host

The system will support a simple (duplex or simplex) Telnet connection between a user and a Host. Because the IBM 2780 is not interactive, and because the current system does not support any options, it is not expected that this mode will be particularly useful, at least initially. This mode is available, however, if a Host has some well-defined procedures for servicing RJE terminals via Telnet connections. In addition, a user might wish to define a private protocol to a system which does not support FTP. Note that Telnet in this context means essentially conversion to NVT seven-bit ASCII code and interpretation of Telnet commands.

Once the dial-in connection to the RJE mini-Host is made, the user should set the 2780 to Transmit mode, initialize the card reader, and enter a deck. The first card must contain the command #3 as described in the section on commands; the last card must contain a command #15. On receipt of the first card, the RJE mini-Host will perform an ICP to the designated Host (and socket if other than to the Telnet logger). On receipt of the last card, the system will close the connection. Intervening cards and operator actions are the responsibility of the user. At the present time, the connection is assumed to follow the Telnet protocol; non-Telnet or transparent modes of operation may be supported at some later time. It should also be noted that the system will intercept the command escape character and attempt to interpret the following characters as a command; the escape character must be doubled if it appears as data.

It is also possible to set up a connection directly without use of the ICP. In this case, the user's card deck must begin with the sequence of commands as follows: 1, get a connection parameter block; 2, get a Host parameter block; 4 and 5, set the socket parameters; 14, open both connections (or 10 or 11 if only one connection is desired). This sequence effectively replaces the command 3 card, which causes an ICP procedure. It may be desirable if the user and Host wish to experiment, or if a Host provides some special service on a particular socket. The session should end with a card containing the command 15, close the duplex connection (or 12 or 13 if only a simplex connection was used). The intervening cards and procedures are again the responsibility of the user. The same cautions as above prevail with respect to Telnet protocol and local command interpretation.

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)*

Bolt Beranek and Newman Inc.
50 Moulton Street
Cambridge, Mass. 02138

2a. REPORT SECURITY CLASSIFICATION

UNCLASSIFIED

2b. GROUP

3. REPORT TITLE

QUARTERLY TECHNICAL REPORT NO. 6, INTERFACE MESSAGE PROCESSORS

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

1 April 1974 to 30 June 1974

5. AUTHOR(S) *(First name, middle initial, last name)*

Bolt Beranek and Newman Inc.

6. REPORT DATE July 1974	7a. TOTAL NO. OF PAGES 44	7b. NO. OF REFS

8a. CONTRACT OR GRANT NO. F08606-73-C-0027 b. PROJECT NO. 2351	9a. ORIGINATOR'S REPORT NUMBER(S) Report No. 2852
c. d.	9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*

10. DISTRIBUTION STATEMENT

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Advanced Research Projects Agency Arlington, Virginia 22209

13. ABSTRACT

The ARPA computer network provides a communication medium which allows dissimilar computers (Hosts) to interchange information. Each Host is connected to an Interface Message Processor (IMP), and IMPs are interconnected by leased common carrier circuits. There is frequently no direct circuit between two communicating Hosts, and the intermediate IMPs store and forward the information. IMPs regularly exchange information which is used to adapt routing to changing network conditions. IMPs also report a variety of parameters to a Network Control Center, which coordinates diagnosis and repair of malfunctions. The Terminal IMP (TIP) permits the direct attachment of 63 character-oriented terminals. The Satellite IMP (SIMP) will allow multi-station use of a single earth satellite channel. A High Speed Modular IMP (HSMIMP) is under development; one goal of this effort is to increase IMP performance by an order of magnitude. Specialized mini-Hosts under development will provide for: connection of remote batch terminals; simulation of a leased point-to-point circuit; encrypted Host communication.