

B O L T B E R A N E K A N D N E W M A N I N C
C O N S U L T I N G • D E V E L O P M E N T • R E S E A R C H

AD722367

Report No. 2123

April 1971

INTERFACE MESSAGE PROCESSORS FOR
THE ARPA COMPUTER NETWORK

QUARTERLY TECHNICAL REPORT NO. 9
1 January 1971 to 31 March 1971

Principal Investigator: Mr. Frank E. Heart
Telephone (617) 491-1850, Ext. 470

Sponsored by
Advanced Research Projects Agency
ARPA Order No. 1260

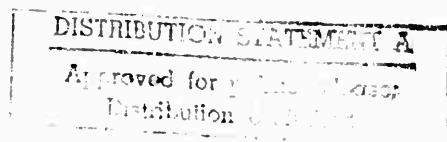
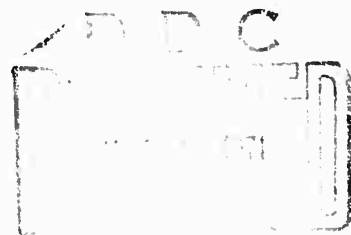
Contract No. DAHC15-69-C-0179
Effective Date: 2 January 1969
Expiration Date: 31 December 1971
Contract Amount: \$2,545,736.00

Title of Work: IMP

Submitted to:

Director
Advanced Research Projects Agency
Washington, D.C. 20301

Attention: Dr. L.G. Roberts
Telephone (202) 0X4-5922



Reproduced by
NATIONAL TECHNICAL
INFORMATION SERVICE
Springfield, Va. 22151

Report No. 2123

Bolt Beranek and Newman Inc.

INTERFACE MESSAGE PROCESSORS FOR
THE ARPA COMPUTER NETWORK

QUARTERLY TECHNICAL REPORT NO. 9
1 January 1971 to 31 March 1971

SUBMITTED TO:

Advanced Research Projects Agency
Washington, D.C. 20301

Attention: Dr. L.G. Roberts

This research was supported by the Advanced Research Projects Agency of the Department of Defense under Contract No. DAHC15-69-C-0179.

Table of Contents

	Page No.
1. INTRODUCTION	1
2. TERMINAL IMP	3
2.1 Multi-Line Controller	3
2.2 Terminal IMP Software	6
3. THROUGHPUT AND PROTOCOL STUDY	9

1. INTRODUCTION

This Quarterly Technical Report No. 9 describes several aspects of our technical progress on the ARPA computer network during the first quarter of 1971. During this quarter, we installed IMP No. 12 at the University of Illinois and IMP No. 15 at Burroughs Corporation in Paoli, Pennsylvania. We also received delivery from Honeywell of a prototype 316 IMP, a prototype Terminal IMP, and the first deliverable Terminal IMP (both to be fitted with Multi-Line Controllers during the next quarter). The 316 IMP was debugged and connected in the BBN test cell to the ARPA Network, and is operating in the normal fashion. [The IMPs actually delivered to Carnegie-Mellon University and to Case-Western Reserve University during the last quarter of 1970 were numbered 14 and 13, respectively, instead of 13 and 14 as previously reported.]

The telephone company installed six additional circuits during this quarter, connecting Utah/Illinois, Illinois/MIT, Lincoln/Case, Case/Carnegie, Carnegie/Paoli, and Paoli/Harvard. The temporary MIT/Utah circuit, which was installed in June 1970, has been discontinued. The total number of wideband circuits in the net is now 19.

The logic design of the Multi-Line Controller (MLC) was completed during this quarter and we commenced implementation of a prototype MLC unit. The design of the Terminal IMP program was advanced almost to completion and efforts on coding of the Terminal IMP program were intensified. In addition to handling characters from terminals, this program will initially incorporate those aspects of Host protocol required for flow control, handling connections, logging, and teletypes. This work is described in Section 2.

A study of the relationship between buffer allocation, the routing algorithm, Host protocol, and subnet performance was completed during this quarter. During this study, several new algorithms were developed to improve system performance. These algorithms are described in Section 3.

2. TERMINAL IMP

2.1 Multi-Line Controller

During the last quarter, we completed the logic design of the Multi-Line Controller (MLC) and fabricated a prototype of the common logic and the Line Interface Unit (LIU). The MLC control panel and additional LIUs will be completed early in the next quarter, at which point debugging will begin.

The MLC is divided into an input section and an output section to handle traffic flow in both directions between terminal devices and the Honeywell 316. The input section of the MLC accepts characters bit serially from each of up to 64 terminal lines and assembles and delivers each whole character, along with a line number indication, into the 316's memory via a single DMC channel. The MLC accepts 5-, 6-, 7-, or 8-bit characters having one start and either one or two stop bits. The start and stop bits are stripped off each character by the MLC and the assembled characters are presented right justified in an 8-bit field of the DMC word.

Characters are delivered up to 64 terminal lines via a second DMC channel and the output section of the MLC. The MLC transmits 5-, 6-, 7-, or 8-bit characters, as delivered by the program, with either one or two stop bits. The start bit is automatically inserted by the MLC and need not be specified by the program.

The MLC contains a 74-bit by 64-word memory and each of the 64 terminal lines is assigned one of the memory locations, which is called the STATEWORD for that line. The STATEWORD records

the instantaneous state of data transmission and control status on a line. A 56-bit portion of the memory, illustrated schematically in Figure 1, is realized as a serial, pseudo-drum memory built from circulating shift registers. The remaining 18 bits of each STATEWORD are stored on a per-line basis on an individual Lin Interface Unit (LIU). The pseudo-drum rotates once per 51.2 μ sec, permitting a complete revolution per bit at 19.2 kilobits/sec. As each STATEWORD is presented to the MLC common logic, the next processing step for that particular line is performed, e.g., a partially assembled character is shifted one bit and a new input bit is inserted. Up to 64 lines are serviced in sequence by the common logic processing the STATEWORDS.

In the STATEWORD for each line, the Terminal IMP program must set the input rate and character size. In addition, it may set a bit to indicate a high-speed input line that is to be given priority access to the DMC channel. A character from a terminal is shifted, one bit at a time, into the input data section of its STATEWORD, until a complete character is assembled. The complete character is then moved to the input data buffer section of the input STATEWORD to await service by the DMC channel, thus allowing the assembly of the next input character to begin.

For output to a terminal, each character is loaded from the DMC into the output data section of its STATEWORD, for double buffering, while the previous character is shifted out. When the previous character has been shifted out, the contents of the output data section are moved from the double buffer. The bits are then shifted out at the rate required by the line being serviced, and a request for the next character is sent via the DMC to the Terminal IMP program.

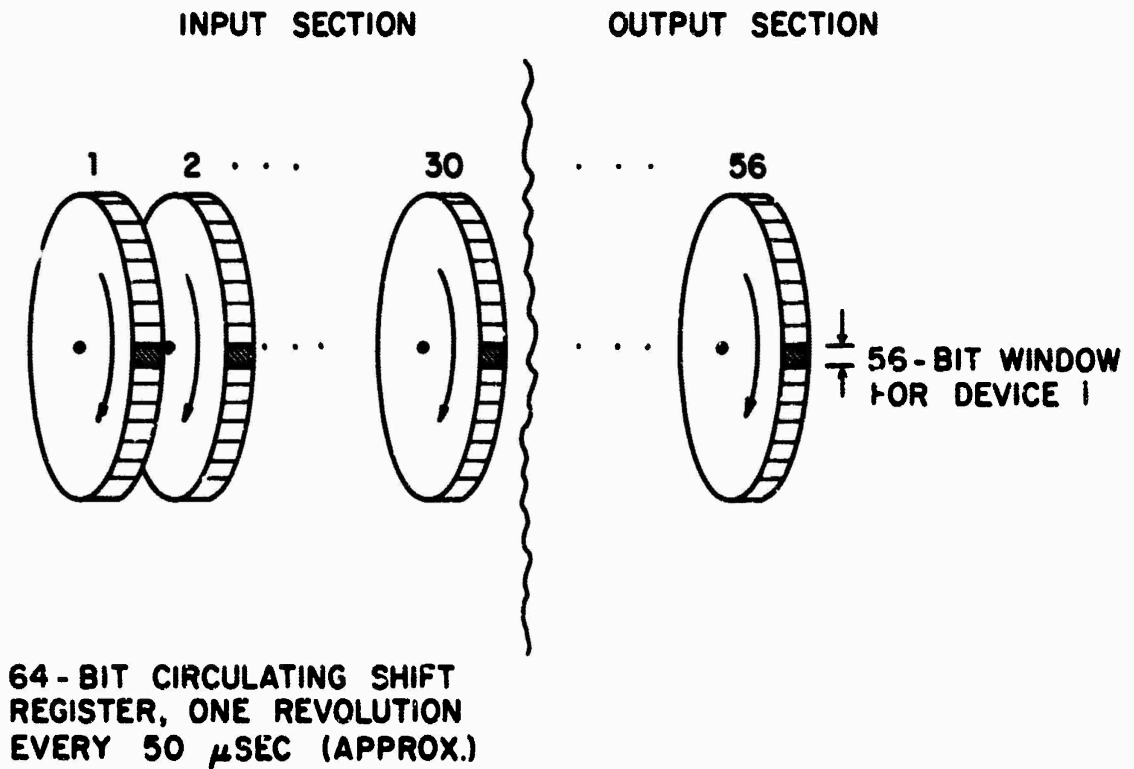


FIGURE 1 SERIAL MEMORY IN THE MLC

A portion of the control circuitry for each line is contained in the Line Interface Unit. The LIU handles signals which cannot be time multiplexed by the pseudo-drum. Such signals are the output data line and modem control lines. A prototype of the LIU has been fabricated on a printed circuit card using 30 ICs.

2.2 Terminal IMP Software

The Terminal IMP program will be composed of the regular IMP program plus additional code and buffering to handle up to 64 terminal devices. The design of the terminal-handling portion of the software is at an advanced stage of progress and several major parts have already been coded. This portion of the program will communicate with the regular IMP program via a software Host/IMP interface. Thus, all devices at a Terminal IMP will be addressable as belonging to a single Host.

Five principal functions of the software are:

- 1) Terminal input — buffering characters from the terminal into appropriate format and shipping them to the net;
- 2) Terminal output — accepting messages from the net, unpacking and reformatting to a form suitable for the terminal, and delivering characters to the terminal as requested by the Multi-Line Controller;
- 3) Protocol — communicating with Hosts, IMPs, NCPs, and loggers in the language they understand;

- 4) Command Language — conducting a dialog with a user via his terminal to provide him with the capability to effect mode settings and otherwise set Terminal IMP parameters;
- 5) All else — initialization, monitoring modem status, etc.

An initial effort has been made to define a command language for the Terminal IMP. At this point, the command language is still under development and we are specifying both the parameters and options the user will have and how he communicates them to the Terminal IMP. It is clear that the Terminal IMP cannot support a very elaborate command language, and that it must be able to do both the job called "Telnet" (in the Host protocol jargon) and a subset of the file transfer protocol. The Telnet protocol is currently under development by a committee of Host protocol representatives and the file transfer protocol will be developed at a later date.

One useful measure of the Terminal IMP performance is the total supportable terminal bandwidth. This measure is obtained from a complex function of hardware organization, network performance, available buffer space, amount of usable phone line bandwidth, and several Host and Terminal IMP program running times. A key parameter is the per character processing time that sets an upper bound on bandwidth. Our current best estimate is that this upper bound is about 150 kilobits/sec for the most critical case of 19.2 bits/sec terminals and that we will be able to achieve our goal of 100 kilobits/sec.

3. THROUGHPUT AND PROTOCOL STUDY

During this quarter, we completed a study on the design and performance of the ARPA Network. Two important objectives of this study were to seek improved algorithms that provide efficient performance under heavy traffic load and to obtain an increased understanding of the relation between the number of packet buffers, the system design, and system performance. Several improved algorithms were developed for handling heavy traffic and are described below. A separate report on the results of this study is in preparation.

To improve congestion control, we have designed a scheme to alleviate congestion at a destination IMP by reserving reassembly space for each message before allowing it to enter the net and by delaying the return of RFNMs from the destination when reassembly space is unavailable. This scheme prevents the insertion of more traffic into the net than the destination can handle.

We have also developed a routing algorithm to achieve high throughput under heavy traffic load and we are currently studying its performance properties. The algorithm combines the use of periodic as well as aperiodic updating of routing tables. The periodic routing update occurs every half second to provide a reliable and efficient means of maintaining routing information when network conditions are not changing markedly. The aperiodic updating results from important status changes such as the failure of an IMP or circuit, or a circuit becoming fully occupied.

Each IMP is allowed to route traffic to a given destination on one or more lines to the destination. If two or more paths share a common circuit, normally only one of the paths will be selected. However, if extra capacity is needed and more than one path is required to obtain it, the IMP allows additional paths to be set up. In this scheme, the selection of routes does not use information about the instantaneous movement of packets in the network. (The routing information cannot be propagated about the net in sufficient time to reflect accurately this instantaneous traffic flow and it is, therefore, decoupled from these events.) Instead, average traffic flow is used in the selection of routes and additional routes are established, if possible, only when the capacity is reached on the established paths. By using an averaging procedure in the measurement, the flow of traffic must build up smoothly. This allows the routing algorithm ample time to adjust its tables in each IMP in advance of the buildup of traffic.

This routing algorithm contains a metering procedure that limits and adjusts the maximum flow of traffic on each circuit. The flow is changed in such a way as to allow the overall network flow to be increased, but to change only slowly for stability.

Several other algorithms were developed during the course of this study. These are:

- 1) An algorithm to provide an "overflow network" in which a small amount of traffic is guaranteed to be delivered to the destination if a path exists. This algorithm guarantees a small throughput at the cost of two buffers per IMP and a few percent of line capacity;

- 2) An algorithm to modify the acknowledgment procedure in a way that substantially saves processing time and makes more efficient use of the phone lines;
- 3) An algorithm to allocate buffers efficiently for use in store and forward and for sharing store and forward and reassembly buffers.

We have been considering the implementation costs and performance benefits of these new algorithms and we expect to initiate implementation of portions of the new algorithms in the next quarter.

UNCLASSIFIED

Security Classification

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. 9 1 January 1971 to 31 March 1971			
4. DESCRIPTIVE NOTES (Type of report and, inclusive dates)			
5. AUTHOR(S) (First name, middle initial, last name) Bolt Beranek and Newman Inc.			
6. REPORT DATE April 1971		7a. TOTAL NO. OF PAGES 11	7b. NO. OF REFS
8a. CONTRACT OR GRANT NO. DAHC 15-69-C-0179		9a. ORIGINATOR'S REPORT NUMBER(S) BBN Report No. 2123	
b. PROJECT NO. 1260		9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	
c.			
d.			
10. DISTRIBUTION STATEMENT			
11. SUPPLEMENTARY NOTES		12. SPONSORING MILITARY ACTIVITY Advanced Research Projects Agency Washington, D.C. 20301	
13. ABSTRACT The basic function of the IMP computer network is to allow large existing time-shared (Host) computers with different system configurations to communicate with each other. Each IMP (Interface Message Processor) computer accepts messages for its Host from other Host computers and transmits messages from its Host to other Hosts. Since there will not always be a direct link between two Hosts that wish to communicate, individual IMPs will, from time to time, perform the function of transferring a message between Hosts that are not directly connected. This then leads to the two basic IMP configurations -- interfacing between Host computers and acting as a message switcher in the IMP network. The message switching is performed as a store and forward operation. Each IMP adapts its message routine to the condition of those portions of the IMP network to which it is connected. IMPs regularly measure network performance and report in special messages to the network measurement center. Provision of a tracing capability permits the net operation to be studied comprehensively. An automatic trouble reporting capability detects a variety of network difficulties and reports them to an interested Host. An IMP can throw away packets that it has received but not yet acknowledged, transmitting packets to other IMPs at its own discretion. Self-contained network operation is designed to protect and deliver messages from the source Host to the destination IMP.			

UNCLASSIFIED

Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Computers and Communication Store and forward communication ARPA Computer Network Honeywell DDP-516 IMP Interface Message Processor						