BOLT BERANEK AND NEWMAN INC

CONSULTING • DEVELOPMENT • RESEARCH

Report No. 3276

(incomplete OCR, 2nd iteration)

INTERFACE MESSAGE PROCESSORS FOR THE ARPA COMPUTER NETWORK

QUARTERLY TECHNICAL REPORT No. 5

1 January 1976 to 31 March 1976

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

Sponsored by:
Advanced Research Projects Agency
ARPA Order No. 2351, Amendment 15
Program Element Codes 62301E, 62706E, 62708E

Contract No. F08606-75-C-0032
Effective Date: 1 January 1975
Expiration Date: 19 July 1976
Contract Amount: $2,533,832

Title of Work: Operation and Maintenance of the ARPANET

Submitted to:

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

TABLE OF CONTENTS

1. INTRODUCTION

This Quarterly Technical Report, Number 5, describes aspects of our work performed under Contract No. F08606-75-C-0032 during the first quarter of 1976. The previous reports in this series dealt largely with work quite closely related to the development, maintenance, and operation of the ARPANET, e.g., the IMPs and TIPs of the ARPANET and the Satellite IMPs and PLIs connected to the ARPANET. However, beginning with this quarter, our work with the ARPANET has been largely funded under a contract from the Defense Communications Agency and our work with Satellite IMPs, PLIs, etc. has been supported under new contracts from ARPA which will be reported elsewhere. The only significant body of work still funded under this contract is a study into the feasibility of using the Pluribus computer as the basis of a large, secure message-switching system. The remainder of this document describes our Pluribus message switch work.

2. PLURIBUS MESSAGE SWITCH STUDY

The message handling systems which have been developed and used within the ARPANET have spurred widespread interest. We are studying extensions of this technology, contemplating a system which will provide a message handling service of large capacity and high reliability while meeting stringent security requirements. Our task is to explore the suitability of the Pluribus computer for this application. This interim report on our work will be followed by a comprehensive final report to appear in our next Quarterly Technical Report.

Message switching, not inherently a difficult task, becomes difficult when a large number of messages must be processed in an environment requiring high reliability and complex and stringent rules regarding who should and should not have access to each message. A secure system requires means for guaranteeing not only that messages are not misdirected to improper recipients but further that there is no method by which an individual can obtain access to messages without authorization. To guarantee the integrity of both the operation and the security of a message switching system requires a highly reliable system with good fault tolerance. Thus for a large message switch of the kind contemplated, the hardware and operating system together must:

- Be able to support a high volume of traffic, with provision for expansion.
- Be highly reliable and highly available.
- Assure against release of information to unauthorized recipients.

2.1 Application of the Pluribus to Message Switching

The Pluribus multiprocessor, developed at BBN under ARPA support for the ARPANET, has many characteristics which make it attractive as a machine for the message switch. In this study, we are considering how it handles, or can be made to handle, the various problems presented by this application. This will, of course, involve special software, although a substantial part of the reliability/availability software which already exists may be used. Also, some special hardware for helping with the security problems will likely be required. However, it seems possible to capitalize on the multi-resource nature of the Pluribus in coping with these problems and thereby to minimize the amount of new hardware work required.

The structure of the Pluribus has been described in previous Quarterly Technical Reports, and is further described in References 1 to 5. With the structure of the Pluribus in mind, we turn to consideration of how the Pluribus meets the needs of a large message switch.

The ability to handle a high volume of messages is a very important requirement of the message switch. A prototype installation which is being used as the basis for estimates requires on the order of 15,000 messages to be processed daily for about 2000 users. We estimate that the present TENEX system operated in a dedicated fashion could handle perhaps 60 such users. This is not surprising since the TENEX system was designed to provide efficient service of a very different type. The Pluribus, on the other hand, seems quite well suited to providing the required processing bandwidth. It was designed to be able to process, in an economical fashion, large numbers

of small tasks which could be executed in parallel. Just as in packet switching, the task of message processing has a large element of inherent and obvious parallelism stemming from the independence of the individual messages. The Pluribus was designed to gain speed by taking advantage of such parallelism.

Another characteristic of many jobs, and message processing is no exception, is that the capacities required inevitably change with time -- usually by increasing. The modularity of the Pluribus hardware structure, plus the approach we have taken in using the processors as general purpose workers, plus a highly adaptive approach in locating and utilizing available hardware resources, combine to create a system which can be adapted easily to changing requirements. Handling increased traffic should require only the addition of the needed extra hardware resources. This would typically mean adding more communication line controllers in the I/O, perhaps more memory for buffering, and perhaps more processor busses to increase processing bandwidth. Such changes do not require changes in the software, because the adaptive mechanisms necessary to incorporate shifting hardware resources are built into the software at the outset for reasons of reliability. The system thus can grow with a minimum of effort. We have set the upper bounds on growth (as determined by address fields, etc.) very high, so that large amounts of I/O and processors can be accommodated.

The need for reliability hardly needs emphasis. If all messages flow through a central message switch, when it breaks, the flow of messages stops. The approach we have taken with the Pluribus attempts to guarantee that service will rarely be interrupted and that when it is, the system will recover automatically and quickly. Service should be resumed within seconds

with any failed component excised from the system by the program. This will result in a system in which single errors have extremely low probability of stopping operations for more than a few seconds.

The third major requirement is multi-level security, the need to provide protection for several levels of access. Many different levels of messages will be handled by the same system and yet unauthorized access must be prevented. This is a difficult job. A "security kernel" in the software is one traditional approach to be considered. However, just as in the case of reliability, the multi-resource aspect of the Pluribus offers some conceptually simple approaches to the security problem.

The focus of attention in security studies has traditionally been on software and on providing assurance that no amount of ingenuity could circumvent the protective features provided in the program. However, to our knowledge, there has been little work on attempting to cope with the ways in which hardware failures can jeopardize security. The assumption of unfailing hardware is unrealistic in the practical world; any prudent system design must allow for the effects of a failure. In the Pluribus, powerful techniques have been developed for detecting and dealing with the effects of failures, both software and hardware, on the integrity of system operation. A good deal of the program (although only a small fraction of processing bandwidth) is given over to mechanisms for isolating and recovering from failures. Methods have been developed whereby the processors work together to certify functioning and to eliminate bad parts of the system so that they do not damage overall operation. Many of these same approaches can be used to assure the integrity of