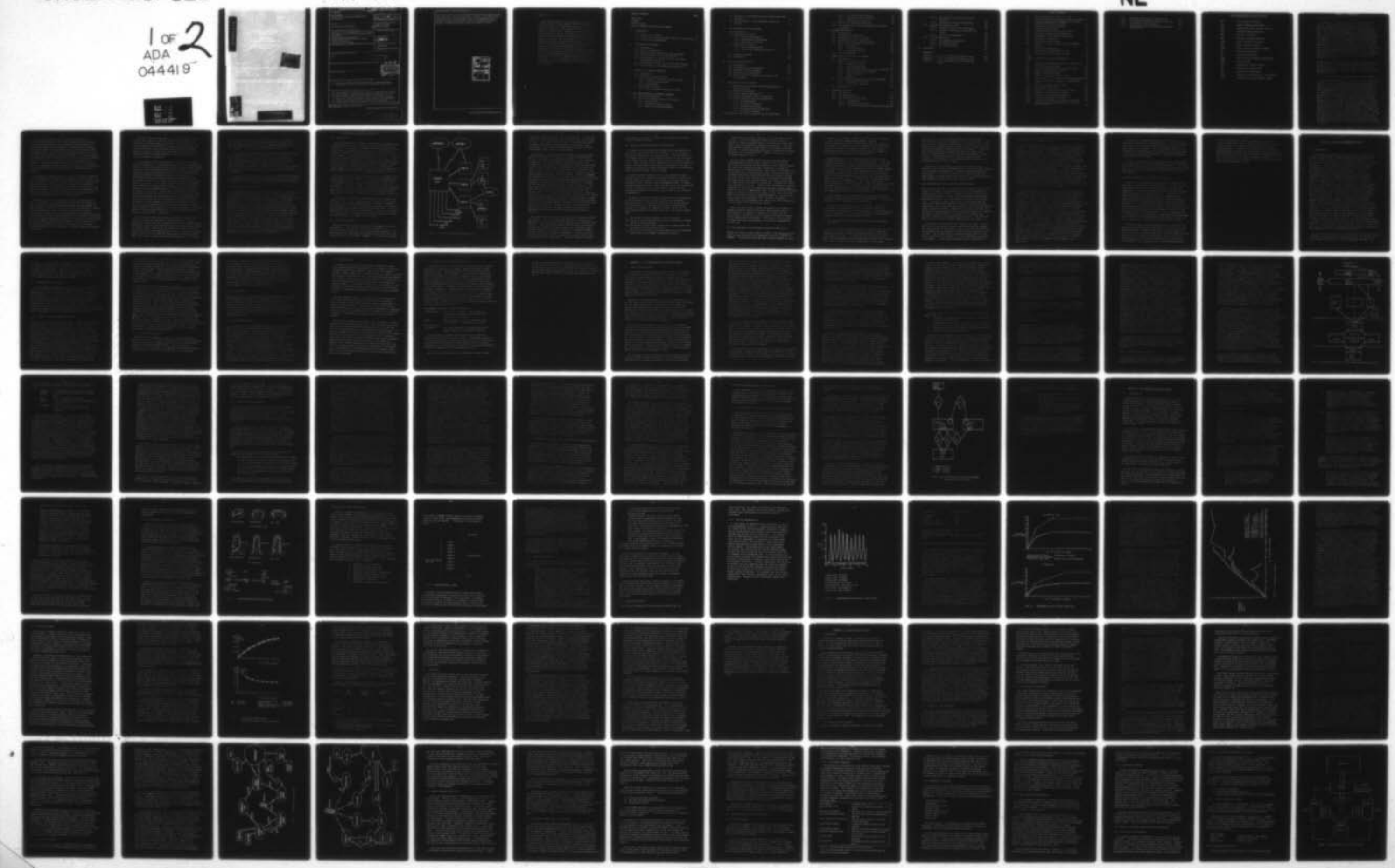
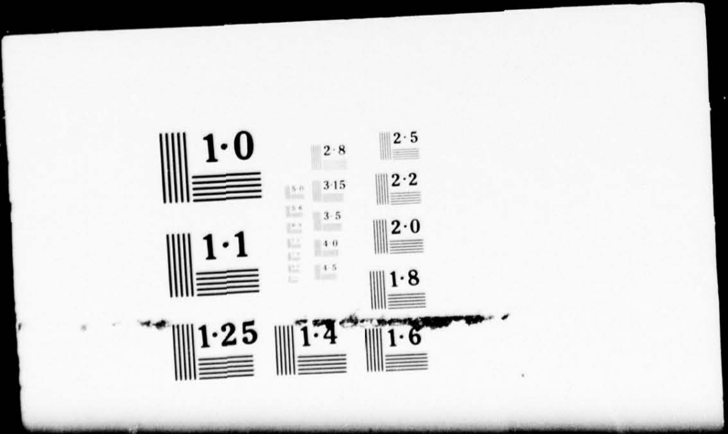
AD-A044 419 UNIVERSITY COLL LONDON (ENGLAND) DEPT OF STATISTICS --ETC F/G 17/2
UNIVERSITY COLLEGE LONDON ARPANET PROJECT.(U)
FEB 77 P T KIRSTEIN N00014-74-C-0280
UNCLASSIFIED TR-34 NL

1 OF 2
ADA
044419



1 OF 2

ADA 044419

Although an important new project, the Atlantic Packet Satellite Project (PSP) started in 1975, our practical involvement in the experimental programs began only in 1976. During this period a

REPORT DOCUMENTATION PAGE

(12)

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
(14) TR – 34

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
(6) University College London ARPANET Project. Annual Report (9)

5. TYPE OF REPORT & PERIOD COVERED
ANNUAL, 1 JAN 1976 - 31 DEC 76

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Professor Peter T. Kirstein (10)

8. CONTRACT OR GRANT NUMBER(s)
(15) N 00014-74-C-0280,
N 00014-77-G-0005

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Prof. Peter T. Kirstein, Department of Statistics and Computer Science, University College London, WC1E 6BT, UK

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
(12) 188 p.

11. CONTROLLING OFFICE NAME AND ADDRESS
Defence Advanced Research Projects Agency,
1400 Wilson Boulevard,
Arlington, Virginia 22209

12. REPORT DATE
(11) FEBRUARY 1977

13. NUMBER OF PAGES
187

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
U

16. DISTRIBUTION STATEMENT (of this Report)
No.1

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

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

DDC
RECEIVED
SEP 22 1977
A

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Internetworking, SWITCH, File Transfer, Facsimile, Network Measurements and Monitoring, TCP, EPSS, Packet Satellite Experiments, Network Usage

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

This report serves as a final report on a contract from the Office of Naval Research (N00014-74-C-0280) which terminated on 30 September 1976, and as the first quarterly report on Contract N00014-77-G-0005 which began on 1 October 1976. Research activities covered in this document include the interconnection of hosts and networks, the development of a system SWITCH for virtual network connection, and our connection to the UK Post Office Experimental Packet Switched Service. Two topics covered in particular

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

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

409 714

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

detail are the packet satellite project and the Transmission Control Program (TCP) experiments. These have particular relevance to other ARPA supported projects and so are treated in some depth. Finally, our measurement and facsimile activities are described, and UK use of ARPANET through the UCL TIP is evaluated. A list of conclusions that may be drawn from our research activities during 1976 is given at the end of the report.

ABSTRACT

This report serves as a final report on a contract from the Office of Naval Research (N00014-74-C-0280) which terminated on 30 September 1976, and as the first quarterly report on Contract N00014-77-G-0005 which began on 1 October 1976. Research activities covered in this document include the interconnection of hosts and networks, the development of a system SWITCH for virtual network connection, and our connection to the UK Post Office Experimental Packet Switched Service. Two topics covered in particular detail are the Packet Satellite Project and the Transmission Control Program (TCP) experiments. These have particular relevance to other ARPA supported projects and so are treated in some depth. Finally, our measurement and facsimile activities are described, and UK use of ARPANET through the UCL TIP is evaluated. A list of conclusions that may be drawn from our research activities during 1976 is given at the end of the report.

TABLE OF CONTENTS PAGE

5. THE TRANSMISSION CONTROL PROGRAM

6. UCL ACTIVITIES WITH EPSS

7. THE PACKET SATELLITE PROJECT

8. NETWORK MEASUREMENTS

9. EXPERIENCES WITH UK USAGE OF ARPANET

10. FACSIMILE ACTIVITIES

LIST OF FIGURES

SOME ABBREVIATIONS USED IN THE REPORT

BBN	Bolt, Beranek and Newman
DCA	Defence Communications Agency
EPSS	Experimental Packet Switched Service
IMP	Interface Message Processor
MOD	Ministry of Defence
NLM	National Library of Medicine
PSDN	Public Switched Data Network
PSP	Packet Satellite Project
PSTN	Packet Switched Telephone Network
PSU	Packet Switching Unit
RL	Rutherford Laboratory
RSRE	Royal Signals and Radar Establishment
SIMP	Satellite Imp
TCP	Transmission Control Program
TPU	Transmission Protocol Unit
UCL	University College London
UCLA	University of California at Los Angeles
ULCC	University of London Computer Centre

CHAPTER 1: INTRODUCTION

1.1 Purpose of the Report

The University College London (UCL) ARPA Project started in 1973. The original contract from the Office of Naval Research (N00014-74-C-0280), was renewed up to September 30 1976; a new contract (N00014-77-G-0005), started on October 1 1976. This report serves therefore as a final report on the first contract, and as the first quarterly report on the second. The project is also partially financed by the British Library (BL, GRANT SI-G-093), the UK Ministry of Defence (MOD, Contract AT/2047/064), and the UK Science Research Council (SRC, GRANT B/RG/59811 and B/RG/67022). The report is therefore also an annual report for these grants and contracts as were the previous annual reports (Kirstein, 1974 and Kirstein 1976A). Support has been provided by the UK Post Office to the Packet Satellite Project, and this report is also aimed at informing them of the progress of work.

Two subjects, the Packet Satellite Experiment (Chapter 7) and the TCP (Chapter 5) are treated in particular detail. This is because those subjects are of particular relevance to other ARPA supported projects, and this report is a final report for ONR N00014-74-C-288.

1.2 Aims of the Project and their Relation to this Report

During the previous years of the project, we have investigated some new principles for interconnecting hosts and computer networks. We had investigated the problems of connecting the Rutherford Laboratory (RL) IBM 360/195, the University of London Computer Centre (ULCC) CDC 6000/7000, the Royal Signals and Radar Establishment (RSRE) GEC 4080 and the Culham ICL Systems 4/72, onto ARPANET using UCL computers as the front-ends. While all the individual computers had been attached separately, during the past year we have developed a system for connecting all of them simultaneously onto ARPANET attached to the same front-end computer. This development was essentially completed by the end of this reporting period. In Chapter 2 we discuss how the UCL system has developed during 1976 as regards hardware, system software and the attachment of hosts. The work has required the definition of a new form of virtual network connection as the number of hosts to be attached grew to four. The implementation of the new system called SWITCH, is discussed in Chapter 3.

The logical next step was to investigate the problems of interconnecting networks. Here we had analysed during the previous period the facilities required in gateways between networks and had started implementing a connection to another specific network, the UK Post Office (PO) Experimental Packet Switched Service (EPSS) (Broomfield,1975). There have been significant changes in the last year due to the acceptance of X25 (CCITT,1976) as an agreed standard interface for access to PTT networks. This is a virtual call network, which can be used with some modification for interconnection of networks. Our general approach is in fact a virtual call mapping, and so is compatible with this standard. Our approach to the provision of services through concatenated networks is developed further in Chapter 4.

One approach to communication between hosts on different networks is the implementation of software to a common interface standard. We have investigated a specific protocol (TCP) designed for communicating between such hosts. These experiments, which have involved both Stanford University and Bolt, Beranek and Newman are discussed in some detail in Chapter 5. We have completed quantitative experiments, and have also simulated some modifications which have been suggested to improve the performance of systems using the protocol.

During the past year we have become increasingly involved with EPSS. An experimental interconnection between EPSS and ARPANET has been devised, with one PDP9 performing mappings between ARPANET and EPSS protocols. The present UCL system is not too advanced; it performs the EPSS mappings at a very low level, which does not use EPSS efficiently. However, this was the connection first adopted by RL, and it has allowed us to get some experience with the problems of using EPSS. We are now running experimental sessions for UCL users of the RL 360/195 using the ARPANET, EPSS and the UCL gateway. This development will be extended considerably during 1977, when mapping of the higher level EPSS protocols will be achieved. This work on EPSS is discussed in Chapter 6.

Although an important new project, the Atlantic Packet Satellite Project (PSP) started in 1975, our practical involvement in the experimental programs began only in 1976. During this period a gateway computer was installed in University College London. We began implementing tools to generate traffic across the packet satellite network, to control the experiments and to collect data flowing between the gateway computers. Our activities in this area are discussed in Chapter 7.

A key component of the UCL activities has been very extensive network measurements. In the previous annual report we discussed our tools for access control and measuring terminal access control through the public switched telephone network (PSTN). During the past year, these tools have been developed further and fairly extensive and varied measurements have been made and analysed, although PSTN monitoring has been restricted by shortage of computer time. We have complete records of all traffic using the UCL/RL link in either direction whenever the RL machine has been connected to ARPANET. The information is output on paper tape and has been kept since the beginning of the UCL ARPANET project, although it was not possible to analyse it until 1976. Another form of measurement that was taken during the year was that of line traffic passing between IMPs; this has been carried out in conjunction with the experiments on end-to-end protocols and has clarified our understanding of which performance problems depended on host-host interactions and which on the interface between host and data network operations. Finally, some measurements have been made on the actual performance of different network control programs themselves. All the above projects are described in some detail in Chapter 8.

The UCL node has been almost unique on ARPANET in that it has a very wide spectrum of users both in the UK and the US. It is possible to identify the user community fairly clearly because of the operational rules set up at the beginning of the project (see Chapter 2 of Kirstein,1976A), under which any UK user must obtain explicit authorisation from the Governing Committee of the project. Similarly, every US user of the UCL node must obtain explicit permission from ARPA. As a result of these authorisation procedures, it is possible to know fairly accurately, who has been using the network and for what purpose. We have carefully evaluated the usage made by the different projects, most of which are collaborative between the US and UK groups. This usage has been analysed in Chapter 9.

Early we became interested in the use of facsimile techniques over packet switched networks and a specific grant in this area (BL Grant SI-G-121) started in January 1976. Our activities in this area are discussed in Chapter 10. They include the connection of a microprocessor to a facsimile terminal, its connection to ARPANET, and the integration of facsimile traffic with message services. The conclusions which can be drawn from the UCL work are discussed in Chapter 11.

In Appendices A and B, we list all the active and non-active research groups who have been authorised to use ARPANET from the UK during 1976. In Appendix C, we list the papers and presentations made by the group.

1.3 Institutional Arrangements

The institutional arrangements described in Chapter 2 of (Kirstein,1976A) continued almost unchanged during the year. The PO continued its representation on the Governing Committee with one representative from the External Communication Executive (ETE,Letford), and one from Network Planning (NP, Broomfield). The direct representation of ETE on the Governing Committee has made it possible to discuss more directly the policy and the practical problems raised by the project; thus it is possible to resolve quickly which uses of ARPANET from the UK are permissible. Several applications were refused which involved use of ARPANET by people not resident in the UK, or in which the principal usage would have been for message traffic. It has also been possible to discuss frankly the criteria which will be adopted during 1977 when another commercial service is expected to become operational between the US and the UK. As a matter of policy, any users who conceivably could use a commercial service will not be permitted to continue their use of the ARPANET link.

CHAPTER 2: UCL SYSTEMS DEVELOPMENT

2.1 Introduction

The UCL development has always been significantly different from other ARPANET host sites. We have never had a significant host ourselves, but have served as a remote front-end computer to other hosts, and as a gateway to other networks. This role has made increasing demands on the UCL system software with the increase of the number of remote systems which we have attached. In Section 2.2 we show how the system has developed in the last year. Two remote hosts are now permanently available, the Rutherford Laboratory IBM 360/195 and the Royal Signals and Radar Establishment GEC 4080; two more hosts are attached experimentally, the University of London Computer Centre CDC 6000/7000 and the Culham ICL System 4/72. The attachment of these hosts is described in Sections 2.4 - 2.7.

The initial host attachment used somewhat ad hoc software in the UCL PDP9s, mapping from ARPANET protocols to ones desired by the target computer (Stokes, 1975) . The amount of software required is significant, and we have been very concerned to improve its performance as more hosts become attached through the same front-end. As a means to improve performance, we have made consistent measurements of the PDP9 software, and changes to the operating system and communication tasks. This activity is discussed in Section 2.3.

There has been a further systems activity which is treated in a separate Chapter. This is the complete redesign of the front-end philosophy; now a system-independent set of interface procedures are defined for both interactive and batch traffic, and all remote systems protocols are mapped onto these universal ones. While the development of this system, called SWITCH, is discussed in Chapter 3, the progress made in intergrating the host software to run under the SWITCH system is discussed in this chapter.

2.2 The UCL Configuration

During 1976 there has been considerable development in the UCL configuration. One important change was the move of our group from Gordon Square into the new Gower Street premises. This move required the construction of a duplicate post office facility and the move of all the group's computers. This move was achieved

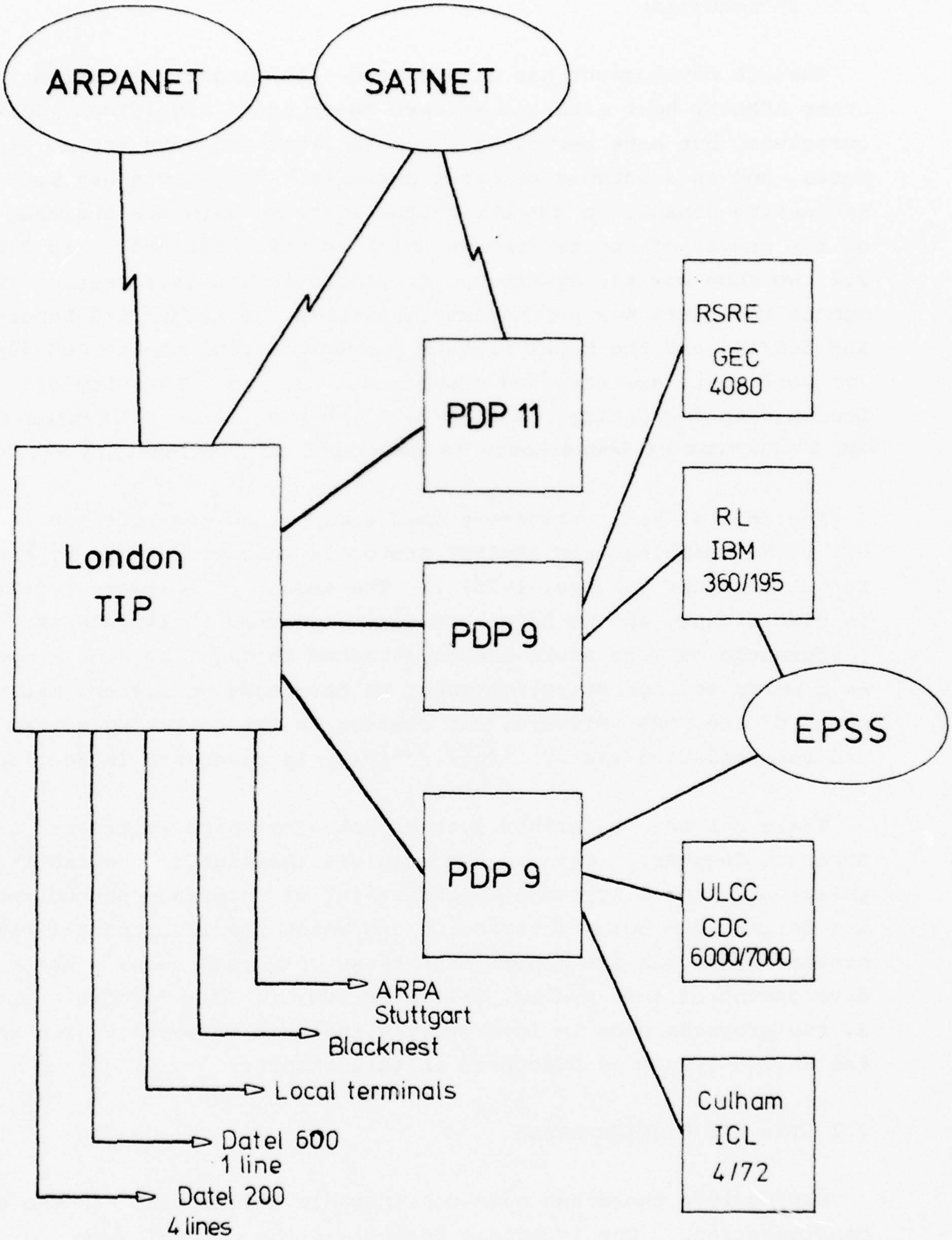


Fig 2.1 UCL Configuration at the end of 1976

relatively painlessly during July and August 1976. The TIP was moved with minimum disturbance and was only off the air for about 48 hours. The movement of one of the UCL PDP9s was also fairly straightforward, with an unavailability of only about 1 week. The second UCL PDP9 became somewhat damaged in transit and gave considerable trouble over a five week period.

The difficulty of reconciling our research and service commitments with only one PDP9, forced us to reduce the availability of the RL and RSRE machines during this period. This resulted in a substantial reduction of usage of these two machines - far sharper than would result from a proportional reduction with hours of prime time availability. This brought home to us again the need to provide as near 24-hour availability as possible for any hosts offering services. A third computer (a PDP11) to act as a gateway between the packet satellite network SATNET and ARPANET was installed also in early July and was moved with little difficulty. During the construction of the duplicate communication facility, a number of changes were made in the UCL communication equipment. The configuration at the end of 1976 is shown in Fig. 2.1. The main changes over the previous year are that the SATNET gateway PDP11 was installed so that the TIP has now no room for any further hosts. The Computer Aided Design Centre Atlas connection was removed, mainly because the Atlas itself was going out of service. The RL and RSRE computers were attached to PDP9A and were in almost continual service throughout the year. The ULCC, Culham and EPSS activities remained experimental throughout the year and were therefore attached to PDP9B. Although it is not obvious from the detail given in Fig. 2.1, we also extended our communication equipment so that PDP9A and PDP9B have essentially identical facilities.

The number of terminals on the public switched telephone network was reduced. One of our major user projects, an experiment of MEDLINE access under British Library auspices, was terminated during the year. Our network measurements indicated that the reduction of ports on the PSTN would reduce costs with a negligible impact on port availability to users. Moreover, the usage of ARPANET through the RL hosts increased considerably. Finally, we wished to start coming into a position under which terminal access would be supported mainly through EPSS. For these reasons we reduced significantly the number of terminal ports offered on the public switched telephone network.

2.3 Operating System Measurement and Improvement

To support our communications software and its development, we have our own operating system for the PDP9. During the year we have devoted some effort to a consideration of the efficiency of the resident sections of this operating system and its core allocation, segment loading, segment deletion and task scheduling functions. There has been little change to the input/output functions: drivers have been added for new devices as necessary, and extra queueing was provided to permit synchronous running of asynchronous format character devices at medium speeds.

A study by an MSc student in 1975 of the ARPANET-RL system running at that time suggested that marked degradation would occur if core space were reduced. As a result of this work and some theoretical notes (Hinchley,1976A and Higginson,1976F) on segment loading overheads and their effects, an MSc project was commissioned in 1976 to investigate the current segment loading and core allocation and deletion algorithms, and to develop new algorithms as appropriate.

The report (Johnson, 1976) describes the extensive measurements of various types of user systems (including the ARPANET-RL real time system) in normal situations and with varying degrees of core restriction. The object of running core-restricted systems is to simulate the effectsof a larger system (ie. the multi-machine system) in the full core size. The results showed that:

(i) The segment deletion algorithm could be improved.
(ii) Design of user system structures could be improved to lessen operating system overheads
(iii) The segment loading times measured were a third higher than predicted in the theoretical notes
(iv) Two specific features of internal operation of the operating system were adding largely unnecessary overheads

Correction of the last of these is trivial now they have been isolated. As part of improving segment deletion (which is the way the operating system regains core space for current activities), the core allocation routines were rewritten. This was also necessary because of known flaws in the current allocation routines. A system incorporating the new routines will be tested during the first quarter of 1977. Results (ii) and (iii) will be studied further in 1977.

It also became apparent during 1976 that the operational task-scheduling routine was not as efficient as it might be, and a long term solution to this problem is being studied. However, two 'stop gap' measures have been introduced to increase the efficiency of the current methods. Most tasks in the system loop on the 100ms clock while waiting for work to do, and this meant that an average 50ms delay was encountered when work became available. The clock interval was redefined to be the minimum of 100 ms real time and 1 ms idle time, thereby reducing the normal average delay to 0.5 ms but not overloading the system with unproductive looping if it had other work to do. Another long term problem is the non-reentrancy of PDP9 code; in particular this has limited access to global variables and intersegment calls in such a way that the normal method of building a system is to define a 'main task' which does most of the work and only use independent tasks in a special situations within totally resident code. 'Serially re-useable' global access and intersegment call code has now been developed, and is to be added to the multi-machine system.

We do not regard further systems development of the PDP9 as a research activity, and will devote as little further effort in this direction as possible. However, it is clear that the multi-machine configuration of this chapter will make new demands on the capabilities of the PDP9s. Our efforts will be limited to those items which are necessary to the system succeeding in its new role.

2.4 The Attachment of the Rutherford Laboratory IBM 360/195

During 1976 there has been almost no change in the operational software for the connection of the Rutherford Laboratory computer to ARPANET. The Help facilities and PDP9 response messages to users on ARPANET have improved, and both the facilities for and extent of usage of the RL system have increased. Various specific problems in the file transfer area were also remedied. For example, the response time was improved by using system level priority in the 360, and premature timeout of long connection periods was eradicated; this had arisen during the transfer of very large files and output from submitted jobs.

A major effort has gone into developing a new version of the ARPANET/360 software to run under SWITCH (Section 3). This has had three components. First, it has been necessary to split the ARPANET Network Control Program (NCP)/360 package in such a way that there are very distinct components interfacing between ARPANET and SWITCH. Second, a 360 HASP component has had to be interfaced between SWITCH and the HASP interactive terminal drivers. Third, it has been necessary to break up the file transfer protocol module into similar parts. One connection is between the SWITCH FTP package and the ARPANET NCP. The second, goes between SWITCH and FTP and a HASP RJE package. Since very similar exercises have been required for all the other computers attached to the UCL PDP9, block diagrams of the changes which have been required in the ARPANET/360 software are given in Section 3.2.

All this software, with the exception of that required for file transfer, has been written and largely commissioned. Most of that required for file transfer has also been written.

We have taken the opportunity to provide much better status information to remote users on the file transfer operations. Although this software is not as advanced as the interactive terminal parts, we anticipate it will also be installed on the operational system by the end of the first quarter of 1977.

2.5 The Attachment of the Royal Signals and Radar Establishment GEC 4080

The Royal Signals and Radar Establishment (RSRE) GEC 4080 was connected in early 1976 to ARPANET through a PDP9; the connection simulates a cluster of several interactive terminals, and a very simple protocol is used. A course was run successfully by the RSRE staff in Washington, with US Department of Defense (DoD) participants, within a week of the line between RSRE and UCL coming into service. At that time the connection was made through PDP-9B and nine ports for simultaneous usage were available. After the course was finished the software and the connection were intergrated into the system running on PDP9A, so that although the number of ports was reduced to five the availability went up. Since that time the system has been available for more than twenty hours a day to US participants in the US DoD High Level Language Evaluation Project. Her Majesty the Queen officially opened the Link on March 26 1976 by sending a welcoming message to some of the US participants during a visit to RSRE.

The operation of the RSRE link during 1976 has been as an addition to the standard RL/ARPANET system; however, a further version of our software to run RSRE as a connection under SWITCH has been commissioned. We forsee no problem in continuing to have the RSRE computer available when we go over to the multi-machine operation under SWITCH at the end of the first quarter of 1977.

2.6 The Attachment of the ULCC CDC 6000/7000 Complex

The connection from UCL to the University of London Computer Centre (ULCC) is a 2.4Kbps line terminating on their CDC 6400 which can access the CDC 6600 and 7600. Currently the PDP9 simulates four CDC 200 User Terminals (a 200UT is the CDC hardwired RJE terminal and normally consists of a card reader, line printer, and display unit). However, the PDP9 software has the card reader and the line printer permanently disabled on three of the four simulated 200UTs; this still leaves the full interactive facilities of INTERCOM (CDCA) available to users. INTERCOM is the portion of SCOPE, the standard CDC operating system, which deals with interactive terminals. The fourth simulated 200UT is reservedfor file transfer; this utilizes a simulated card reader and line printer. The PDP9 software is part of the multi-network SWITCH system (see Chapter 3).

The protocol used by the CDC is a simple polling one. The CDC 6400 acts as a master and cyclically polls the four 200UT slaves. The simple nature of the protocol is due to the fact that CDC 200UTs are hardwired devices and hence no more intelligent protocol could be used. The CDC 6400 sends an addressed message down the line which is decoded by the software. If not recognised, the message is discarded. Otherwise,the software responds appropriately:

'nothing to send' or 'here is some data'. Further details on the CDC 200UT protocol is given in (CDCB).

One of the drawbacks with this protocol is that the error recovery is poor. There is no explicit acknowledgement for the receipt of good data at the CDC 6400. A situation can arise in which data for transmission from the PDP9 may be corrupted by software or store failure. The protocol is such that the 6400 will continually request transmission of the data; there is no timeout and no way the CDC 6400 can tell the 200UT about the erroneous data. The CDC 6400 determines which terminals to poll from internal tables. These tables are dynamic in that the frequency of polling will depend on what has been happenning in the immediate past. If there has been some exchange of information within the terminal recently, it will be polled quite frequently; once the terminal has sent a message that has no information this frequency drops sharply. There is a further state where the 6400 gets no response from the terminal, in which case the frequency of polling will drop even further. In this situation, which is the one prevailing when the PDP9 system is started up, it can take up to 4 minutes to get everything started.

The 200UT protocol is half-duplex and is used primarily for the submission of card-image data to CDC mainframes and the receipt of line printer output. Messages can be up to 1000 characters long, not including control information, and there is space and zero compression in lineprinter data. The method we have adopted, out of necessity, is not very efficient for short interactive messages, as one of the network measurements projects, taking measurements between the PDP9 and the CDC 6400, has shown (Section 8.5). For short messages, on a 2.4Kbps line with a single connection, it was possible to send only 13.5K messages in 10.6 hours - giving a throughput of 135bps. For a 3-connection run, 22K messages were sent in 9.8 hours, giving a thoughput of about 240bps. These low figures are caused by the overhead in the CDC protocol (70 msModem turnaround time because the protocol is half-duplex, three out of every four messages are control). A maximum throughput of 375bps should be achievable on this connection, given that they are no retransmissions and that the 6400 is always ready to accept data. The protocol was really designed for multi-dropped interactive terminals (for use with a controller which blocked the data), which can operate at only about 2400bps, so that a maximum of 375bps for messages of this length is not unreasonable.

By the end of the reporting period the CDC machine had been interfaced to SWITCH and run experimentally. It has been passed over recently to the staff at ULCC for testing by them before a public service is attempted. The file transfer section of the CDC SWITCH and ARPANET programs are coded but not yet integrated. A full service including file transfer is scheduled for the end of the first quarter of 1977.

It should be noted that all the CDC software has been designed to run under SWITCH in the multi-machine environment. The public service will commence only when the complete multi-machine including RSRE, RL and ULCC are available for very substantial periods on the same PDP9.

2.7 The Attachment of the CULHAM ICL System 4/72

Software for the connection of the Culham ICL System 4/72 was described in Section 3.6 of (Kirstein,1976A). At that time it was stated that for terminal access a slight modification was implemented. During the past year the system has been run almost without a change in a stand-alone version. Progression to the multi-machine system is straight-forward. The lower level line and block programs were converted into the mode required by SWITCH when they were incorporated into the current system used for testing. The interface level programs being used are still the RL/ARPANET ones, and we are waiting for the RL 360 SWITCH interface to become operational before we generate a version of those programs for use by the Culham system. Since the latter system is a modification of the RL system, we anticipate very little delay before Culham is also available on the multi-machine system once this is in use (i.e. at the end of the first quarter of 1977).

In (Kirstein 1976A), we mentioned that the file transfer would probably be implemented using modification of the EPSS basic file transfer protocol. However, during 1976, the standardisation activities by the PTTs have reached such a point that EPSS is clearly only going to be a transitionary standard. While we ourselves are continuing actively with EPSS participation (see Chapter 6), it is improbable that Culham will also become EPSS participants.

For this reason we have agreed to implement file transfer to Culham by giving them access to SWITCH - FTP commands; this will be done by putting the appropriate commands as HASP records (Higginson,1976). The design for SWITCH FTP was deliberately made sufficiently flexible that the commands to that program could come from another machine. Partly because of Culham's progress and partly because of our own system developmental schedule, it is unlikely that the Culham file transfer will be operational before the end of the second quarter of 1977.

CHAPTER 3: SWITCH- AS A NETWORK INTERFACE

3.1 Introduction

The UCL SWITCH implementation serves both to map virtual calls between networks, and as a network access interface by which hosts can be attached to networks via an intermediary computer. The application to the interconnection of networks is discussed in detail in Chapter 4. In this section we discuss connection of hosts by considering both interactive terminal level of access and bulk data transfer by use of file transfer protocols. The host attachment may be by established work-station protocols as the 360 HASP station, or by a specifically-designed data exchange mechanism. Reasons for interposing an intermediary computer may vary. The additional load to the main CPU may be too great for comfort, or alternatively the core occupied by full network protocols may be too great. More frequently the problem is the reluctance, with good reason, of the systems staff to provide, and later maintain, new complex software modules buried deep in the operating system. There is a paradox here. One reason for the size and complexity of the host protocol is the intention of providing a very flexible network interface able to support a variety of applications, offering secure message delivery with sophisticated end-to-end flow control. By removing the protocol support into a front-end machine, invariably some of the flexibility may be lost, unless the front end host protocol is itself made of equal complexity. It is a matter of some interest then as to whether this technique of host attachment should be considered the norm, or whether it should only be used in specific cases as mentioned above. In some ways of course, there is a trend to distribute some processing capability into microprocessors attached to a main processor, and sharing store with it. Even in this case, we are interested in the amount to which networking protocols can be contained within such a microprocessor function.

At UCL we have experience now of a manufacturer's full-duplex protocol (IBM HASP), a manufacturer's half-duplex protocol (200 UT), and two ad hoc asynchronous protocols, one to a mainframe (GEC 4080), and the other to a microprocessor (INTEL8080). Only the IBM 360 is an originator of terminal users, the other machines are providing a service to network users. The individual details of these connections is described in Chapter 2, and much of the initial experiences in attaching the IBM 360 to ARPANET were described in the previous annual report (Kirstein,1976A).

3.2 Interactive Terminal Service

All the connections completed at UCL have been at the terminal level or above. Because multiple server host destinations are supported at the same time, and no mechanism exists for selective onward addressing, it is necessary to ask network users to identify the destination they desire, on reaching the PDP9; they are not able to address the required service machine directly from their source machine. The only possible direct method would be to designate a particular host as a particular 'host' service. Additionally, an explicit request for connection allows password control on connection which is necessary in some of the connections which we support.

3.2.1 Message or Character Terminal Service

In general, we are catering for use through ARPANET, where character terminal access is the norm and where there is remote echoing of each character from the host to the terminal. While many Tenex services are orientated to this approach, the IBM 360, for example, operates on a line-at-a-time basis. For users of the 360, the PDP9 provides echoing so the terminal user sees the kind of echo he would expect, although only complete lines are being forwarded to the 360. In the reverse direction things are more complicated, a user attached to the 360 would like control characters forwarded to the net, without carriage returns - the addition of a carriage return would be treated as a following control character itself. The convention has been adopted that a 'line' from the 360 terminating in a space is in fact a partial line of control characters, and is sent out into the net without carriage returns. This is a reasonably pragmatic way of solving the problem. Some more elegant way of providing this ability would be useful. In future networks, remote echoing is likely to be replaced by echoing at the local terminal concentrator/network access point (Higginson,1976D). This PAD (packet assembler/disassembler) approach carries with it the implications of a page-mode terminal, rather than a duplex character-by-character terminal access such as Tenex. . Use of hosts such as the 360 is made easier by this approach - the burden of other than line-at-a-time handling is too much for most networks of hosts to bear! It remains to be seen whether the loss of character access will represent a large burden to bear.

In the current set-up, 360 users into ARPANET see both local and remote echoing. It is rather difficult in front-end situations to get exactly the right balance of facilities. Character-set mapping is another thorny area. The ULCC 6400, for example, only has a 64 character set access through 200 UT protocols. Genuine U200 terminals support two local hardwired control characters for deletion of characters or lines. Use of the CDC 6400 itself does not require control characters. There is no way we can support the U200 control characters easily without changing TELNET conventions for each ARPANET access depending on stated destination! This would mean changing these conventions actually during the exchanges with the user. This is complex due to our multiplexed connections of computers, but would not be too unreasonable in a normal situation where only one host is being front-ended at a time. We have recently changed our default TELNET convention to support two or three editing characters and to allow all other characters to be passed through. For the ULCC machine, of course, this just means that control characters need supression before reaching the ULCC.

3.2.2 A Single Service Level

The terminal-level mapping is close to a transparent mapping - especially, as in our case, most control characters are passed straight through, and we are not supporting some of the more sophisticated options of the ARPANET TELNET protocol. The difference is sufficiently small to enable us to implement file transfer protocols (described in 3.3), which use the interactive terminal mapping software as a binary data path. The INTEL 8080 connection is also intended at least in part for binary data transmission, although in this case not by a formalised file transfer protocol (see Chapter 10). It appears that a common service level at the call-level would be more appropriate to all current applications, with some form of convention for terminal protocol mapping where appropriate. Whether this option could be automatic, or whether it needs option commands to the PDP9, is open at the moment. Our internetworking experiences (Chapter 4) also point to a call-level of mapping as the best service base to use.

3.2.3 Flow Control

Adoption of an ad hoc protocol for transmission between the PDP9 and the INTEL8080, for instance, has forced the imposition of a simple protocol for this connection. Even the RL 360 does not have proper flow control in both directions. Across SWITCH, no flow control exists currently, use being made of backing store resources to prevent congestion. Flow control, based on the volume of data currently held on the PDP9 in transit for any particular connection would be useful here.

3.3 File Transfer Service

The implementation of an ARPANET interactive file transfer service to the RL IBM 360 has been detailed in previous annual reports, and in published papers. During 1976, we wished to draw this file transfer service inside the framework of a general file transfer approach. Decisions on the nature of this general SWITCH file transfer protocol were based mainly on internetworking considerations and are discussed in some detail in Section 4.3. However, to date it is the ULCC 6400 and RL 360 which are furthest advanced in interfacing to the general standard adopted, so it is appropriate to discuss these in more detail here. This file transfer protocol is also to be used for the connection to the CULHAM system, except in this case the protocol will be implemented at CULHAM, whereas in the other cases, the protocol is being mapped respectively into the HASP bulk transfer options (+batch job), and the CDC rudimentary file transfer option.

Our model of use in the past has been to implement a server file transfer only where the request for the transfer is initiated outside the PDP9, say in ARPANET. The internal file transfer interface makes requests for a transfer across the internal interface. Thus requests can be made in the previous way by mapping the ARPANET file transfer request onto the internal format, or by introducing a local user interactive piece of software able to formulate file transfer requests. The ARPANET FTP defines a user interface to its protocol; in the first instance it turns out to be more convenient and useful to implement this protocol, and map it onto the internal format.

The file transfer service to the 360 has for a long time been one of the least reliable software services we have provided . It is still not totally clear why we have been so unsuccessful in giving a reliable service. Undoubtedly, the fact that the 360, although looking like an interactive machine (ELECTRIC) is fundamentally batch, and the file transfer approach for ARPANET (and the new PDP9 interface) is fundamentally interactive lies, at the root of the problem.

Moreover file transfer does require a level of service of higher reliability than interactive use, and in this particular service we are seeing the limits of the front-end approach, or at least our ability to write code which is satisfactory for this situation. It will be interesting to see if we fare better on the ULCC file transfer scheduled to be in use by the second quarter of 1977. Certainly, this time we are devoting as much software effort as can be usefully employed on achieving a reliable connection. The CULHAM plan, where the file transfer instructions are obeyed at their site, will also be a useful pointer as to where the weaknesses of the approach lie. The prospects of job transfer protocols for networks front-ended in this way are not high unless we can achieve better results than so far indicated.

3.3.1 The File Transfer Interface in Detail

As mentioned earlier, the SWITCH FTP uses the basic data transfer facilities already present in SWITCH. SWITCH addressing can also therefore be used. An initial request is made to the destination FTP service using a SWITCH CONNECT. This facility was implemented with a service field as an optional parameter, and it is this field which is used to indicate FTP service. Having established contact, the data connection implicitly set up is reserved for later, and a buffer passing mechanism is now employed as a command pathway between the two FTP modules. Commands use an 8 bit byte as the basic working quantity. Commands are variable length option strings, and replies to a command request are formulated as a new command. The control thus oscillates between the two FTP modules, but new command requests do not have to be solicited. In most cases, this protocol funtions within the PDP9 computer. In the case of Culham, it is planned to operate across a communications link; then collision deadlock situations may occur, and must be resolved.

The commands available include the following:

INITIATE/ACCEPT	starts a *file transfer*. After completion of exchanges of options, the transfer starts without further commands.
ABORT	may be required. It can be signalled from the originator or destination of the file.
END OF DATA AND FILE STORED	are the messages from the sender and receiver respectively after a successful transfer.

A substructure of commands also exists mainly to deal with initial options in setting up the transfer. The possible options include whether previous versions of this filename in the destination directory should be deleted if found, whether the file should be appended to an existing file, or whether the file should be treated as a job to be submitted to the host system to be run.

Specific file attributes need to be communicated such as name, account, password of directory, maximum file size, and maximum record length within the file. The flexibility of options allows the destination to refuse the transfer if it is unable to support the required feature. The ability to use this structure for ARPANET, IBM 360, ULCC 6400 and EPSS will be under investigation in early 1977.

CHAPTER 4: THE INTERCONNECTION OF COMPUTER NETWORKS

4.1 Survey and Background

In the previous annual report (Kirstein ,1976A) we contrasted the alternative approaches to connecting networks. One approach is to have any interconnecting "gateway" connected to each individual network so they can continue to act in their individual frameworks; an alternative approach is to ignore any helpful features of the subnet control on an end-to-end basis. The first approach requires complete protocol mapping at a gateway but little extra protocol in each host to permit internetworking - the second approach makes little demand on the gateway, but requires a strong end-to-end protocol in each host.

Accordingly we pursued research objectives on both fronts. On the one hand we participated in the joint Stanford, BBN, UCL program for the design, development, and measurement of an end-to-end protocol (TCP-described in Chapter 5), and on the other have constructed a mapping gateway-SWITCH described later in this chapter. This mapping of virtual calls proved a useful technique in providing both host front-ending and terminal support of network access; these areas are described separately in Chapter 3, with the practical details of the current UCL system in Chapter 2.

During 1976 the definition of the X25 virtual call network interface (CCITT,1976) and its adoption by the major PTTs has made a substantial contribution to the development of packet switched networks. It has aided the eventual solution to the problem of the interconnection of networks, but has not, by itself, provided the total solution. In 1977 we expect to make use of the lessons learned from both TCP and the mapping of virtual call networks, in achieving an internetworking environment, of which X25 is a fundamental constituent. In view of the emergence of X25 as a standard for network access, we will review briefly our current attitudes, before discussing our achievements in 1976.

Any discussion of how to interconnect networks must start with how the individual networks are built. Here the PTTs have gone unanimously for a virtual call solution. There will be further arguments put forward by different groups why they may be wrong (e.g. Pouzin,1976), but that is not our concern here. It is important to stress, however, that X25 is essentially a Network Access protocol. The sender is provided with flow control and delivery confirmation, and messages are delivered in sequence - provided the network is completely reliable. It is not difficult to see how messages could be lost or duplicated on network malfunction, so that users will still probably need to adopt an end-to-end protocol (cf. the Bridging Protocol of EPSS users). This may, however, be comparatively simple; it need only guard against the specific failures which could occur in X25 network failure. Provided some of the modifications proposed (E.g.Roberts, 1976) are adopted, X25 is also suitable for the interconnection of networks. There are still several options which need agreement; some details, such as choice by the data terminal equipment of logical link number, still occur in the areas of error recovery and status reporting. We have still to see how well these subjects are treated in practical implementation of single networks, to know how much end-to-end activity will be required. In the case of connected gateways, there will be new problems of alternative routing between networks, and how far to re-establish call between networks without call interruption, in case of failures of individual nodes or gateways.

There is one other area of concern. The PTTs regard the Public Switched Data Networks (PSDN) as 'theirs', and will reserve specific fields of the packets, for example the address field. All attachments to the PSDNs are regarded as customers with lower priviledges. Often over the next decade at least, very substantial networks will require to connect to the PSDNs. Their operators should, ideally, be allowed similar access to reserved fields; otherwise it will be necessary to set up yet another set of protocols to meet these needs. Examples are facilities for addressing, and alternative routing to several gateways.

Some networks will not be appropriate for virtual call techniques in the subnet. Examples are those where the predominant traffic is transaction oriented, or where there are large rapid fluctuations of average traffic in regions of the network; also where there are requirements to route more data between two terminals than can reasonably be carried on one route, or where parts of the networks may be liable to rapidly varying distrubance such as Packet Radio. Because of such specific requirements, and of compatibility with existing systems, we must expect that a variety of types of networks will exist for a substantial period. Moreover there will continue to be a requirement for such private networks to interconnect with PSDNs and with each other.

Few networks operate on pure datagrams. Usually a datagram at one level becomes a virtual call at the next. EPSS is a network providing a virtual call interface to what is in many ways a datagram subnet. Conversely, ARPANET provides datagrams, by a complicated virtual call mechanism which must be set up for each and every datagram (unless the interval between messages on the same route is less than two seconds). Thus the main discussion seems to hinge on where to place the virtual calls. TRANSPAC is planned as a pure virtual call network.

The pure "Datagram Network" school believes the datagrams should be retained through the whole subnetwork, including the gateways in concatenated networks. The pure "Virtual Call" school believe the calls should be a fundamental property of the subnetwork and hence certainly of the interconnecting gateways. An intermediate group believes that datagrams are alright for subnetworks, but that virtual call are required for the interconnection of networks.

As we have already discussed our attitude to concatenated virtual call subnetworks, we will now discuss those operating on concatenated datagrams. More exactly, we will discuss concatenated networks where it is inappropriate to make any assumptions about delivery order, reliability or duplication. In such networks dynamic re-routing, multiple gateways, multiple paths, reconnection etc., are not conceptually difficult - but strong end-to-end protocols are required. Here our efforts so far have centred on the TCP. Our experiences to date are reported in Chapter 5, and have not been too promising. Both the TCP of Chapter 5 and the EIN/CYCLADES station protocol (Zimmerman,1975) need considerable resources both for implementation, and to achieve high throughput. These end-to-end protocols function best over relatively simple networks. The need to superimpose such a protocol on top of virtual call procedure, leads to excessive redundancy in checking and flow control, both at the level of the software needing to reside in the host, and in the additional header information needed to be carried in each packet. In practice UCL work shows a more subtle and unanticipated problem; the end-to-end controls can be sufficiently "out-of-synchronization" with the lower-level protocols that "protocol interference" results (see Chapter 5). The usual network symptoms of congestion and consequent performance loss may be expected to occur, unless some controls are added at and between gateways. If they wish to minimise gross changes in traffic rate between one pair of terminals due to interfering traffic by another, some flow control is required on a "per network" or even "per connection" basis. Thus one is easily faced again with virtual call-type flow control in the gateways. Moreover, in concatenated networks, it may be essential to achieve reliable transmission between gateways, and thus to have virtual call facilities for flow control, error reporting, accounting and status information.

In view of the above, we believe that all the following areas need further work:

1. Procedures for connections of X25 networks to each other, via modifications of X25
2. Connections of X25 networks to other virtual call networks
3. Connection of virtual call non-X25 networks to each other
4. Connection of virtual call networks with datagram networks
5. Interconnection of datagram networks

We expect to do reasonably little work on (1) in the near future; we will have little access to several such networks, interconnecting them may be difficult politically, and the PTTs and other public bodies are putting considerable effort into this area. We will watch the activities of others in this area, and try to anticipate the problems in interconnecting "Public" and "Private" X25 networks. We expect to concentrate on (2), (3) and (4) and on variations of (5) where the connection is done by virtual call association techniques. We will continually bear in mind that virtual call procedures are likely to move towards X25, unless good reasons appear to the contrary. In all our work with virtual call networks, we will concentrate on the provisions of services through the networks. There has been little standardisation of this field, and the work is needed as much for X25 networks as others. The whole tenor of the work we report in Section 4.2 and Chapter 6 is on this theme.

In Section 4.2 we describe our work in 1976 in the mapping of virtual call services. In our environment, networks with different facilities terminate at UCL; all have a virtual call level. We have developed a general system called SWITCH, which allows the provision of interactive terminal facilities through concatenated networks terminating at UCL by performing protocol mapping in the UCL Gateway. We have developed, but not yet commissioned, a further extension by which bulk transfer services can be provided in this environment.

In Section 4.3 we survey our work during 1976 on strong end-to-end protocols, and comment on its extensions during 1977. Finally, in Section 4.4, we outline our plans for further work in general in this area.

4.2 UCL work in mapping virtual call protocols

4.2.1 Introduction

During 1976 much of the work on the UCL SWITCH, a general mechanism for mapping virtual calls, has gone into providing the front-ending and satellite processor functions of Chapters 2 and 5. Use of SWITCH in this year for packet-switched network interconnection has been for ARPANET-ARPANET mappings and in preparation for connection of EPSS TO ARPANET. The implementation of the EPSS host protocols achieved in readiness for interconnection in the first quarter of 1977 is described in Chapter 6.

There are four main levels at which a mapping of virtual calls might take place:- at the basic level of translating each control or data packet of one network into the appropriate control or data packet of the other net; at the level of the call itself - the main primitives of the call are mapped, but no direct one-for-one mapping takes place on each control signal; at the level of interactive terminals; and in association with the previous method, the mapping of file transfer protocols. We shall describe in Sections 4.2.2 and 4.2.3 the UCL SWITCH interactive terminal mapping and SWITCH file transfer mapping respectively. Mapping at the packet level has been discussed in the literature; a mapping of X25 and EPSS has been reviewed at UCL (Higginson,1976B), but not specifically in an internetworking context. Another recent example is the mapping of EIN/CYCLADES end-to-end protocol into X25. In general, mapping at this level is extremely complex and usually asymmetric. Moreover there is danger, as in the CYCLADES/X25 case proposed by IIASA (Sexton,1976), of confusing the roles of end-to-end protocols as opposed to the network access interfaces like X25. We believe that mapping at the virtual call level is the most promising approach for interconnection; there is an implication that users wishing to communicate must implement the same terminal or other higher-level protocols. It does not follow that there must be universal higher-level protocols; even in ARPANET there are closed communities of hosts which obey different higher-level protocols. User hosts must have a common reliable virtual call facility, or alternatively end-to-end association of calls. Users of network services will essentially be closed groups in many instances. It would be unfortunate and uneconomic to have a multitude of incompatible higher-level protocols; however, complete standardisation at that level is not as necessary as at lower level. In the UCL situation we will provide varying service levels at the gateway; as a beginning we will experiment with a full call mapping in the EPSS-ARPANET situation (Chapter 6) by mapping virtual terminal protocols.

The mechanism of SWITCH described in this section permits the introduction of different service levels. We expect to experiment during the next couple of years with different types of mapping; for this a flexible software structure in the Gateway is a prerequisite.

4.2.2 Interactive SWITCH Mapping

The principles and practice of the SWITCH virtual call mapping software have been outlined in an earlier paper (Higginson,1975B), revised and summarised in (Kirstein,1976A), and described in implementation-level detail in two internal notes (Hinchley,1976B and C). The SWITCH conventions distinguish between the originator and the recipient of a virtual call request (all calls are assumed duplex). A call made to the gateway will result in an ALLOCATE request to a central SWITCH utility, with the identity of the initiator passed to SWITCH also.The address of the destination is now conveyed in a CONNECT request to SWITCH. SWITCH will identify the destination among network(s) currently attached, and itself initiate a call out to that destination by requesting the host software, resident also on the gateway, to make a call. If the call is successful, it is allocated a port number to distinguish it from other future calls; a similar process was carried out for the originator on contacting SWITCH. SWITCH thus holds the identity of two virtual calls, such that data originating on either call can be forwarded to the other, by means of Data Pickup/ Data Forward requests to SWITCH. Appropriate mechanisms exist when either call is unsuccessful to inform the originator.

In order to realise these facilities, not only the basic Host but also the Virtual Terminal Protocols are implemented in the Gateway for each network. By the time messages arrive at SWITCH they are ASCII strings, with no protocol control characters contained. The control information for indicating call destination and error information are also ASCII strings. Since the strings can be originated and interpreted by a human user, interconnection can be achieved without any end-to-end conventions at the level of software; the appropriate commands can be supplied by the user. This choice is not fundamental to the implementation but greatly eases the commissioning. A basic set of connection, addressing, and error codes could be devised to pass on this information from initiating software to the gateway, and if necessary beyond. Such a set of codes would of course be essential in a pure call mapping gateway; such a set, together with suitable algorithms to obey when connections fail, will also be a fundamental necessity in the interconnection of X25 networks.

Figure 4.1 shows the UCL software modules used to provide a mapping between EPSS and ARPANET. The central SWITCH section is essentially a set of mapping tables, but is swelled by the need to keep a rigorous check on orderly establishment, and maintenance

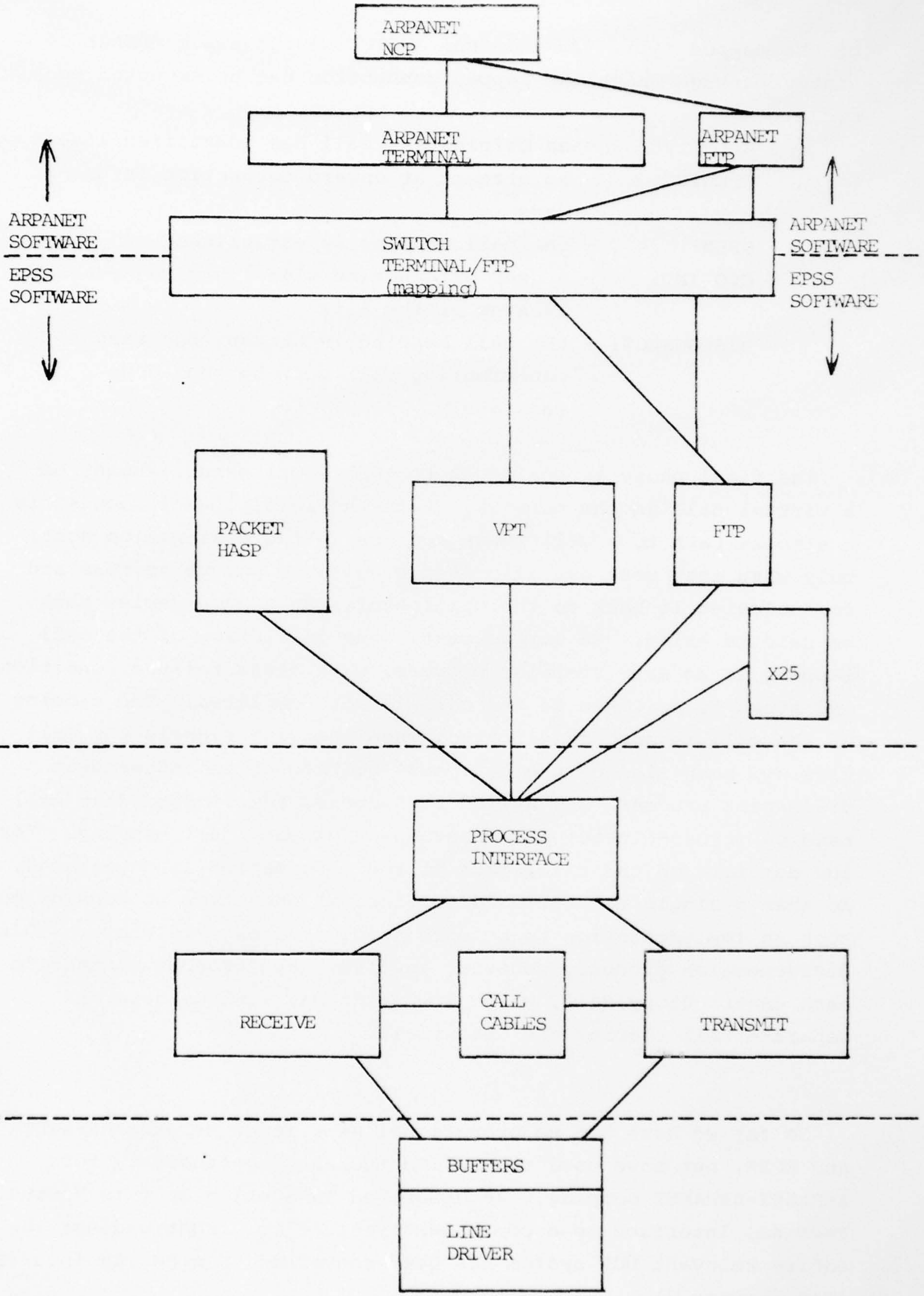


Fig 4.1 UCL Software Modules used to Provide a Mapping Between EPSS and ARPANET

of the mapped call. To this end SWITCH designates a number of states through which the mapped connection can be expected to cycle:

ALLOCATE:	- an originating call has identified itself
OPENING:	- an attempt at onward connection is being made
OPEN:	- the call mapping is established
CLOSING:	- a user originating signal has requested the closing of the call
DISCONNECT:	- the call mapping is broken, but each contributing call must be shut down separately.

The final phase is analogous to the normal establishment of a virtual call in the network, where the initial call request is a simplex path to a well-known address in the destination host; only when that host has allocated a suitable unique address and communicated it back to the originator, can a pure duplex path be said to exist. In our context, the initiation of the call mapping is an easy stepwise process, with clear failure conditions and steps to be taken if the call is not completed. The closing of the call is more complicated; when one side signals the call closing, each side of the call must go through an independent disconnect process. In SWITCH it happened that independent call records (cross-linked) are maintained for each call mapping. For the duration of the call, much of the information is duplicated, so that a single record would suffice; at one stage we considered such an implementation to be more economical on core store. This disconnection process, however, involves asynchronous changes in each contributory call, which are in no way linked; here the separate call records are essential.

So far we have had no operational experience between ARPANET and EPSS, but have used the SWITCH mechanism extensively for ARPANET-ARPANET mapping. As described in Section 3, this Virtual Terminal Interface is a convenient vehicle for front-ending; the entire relevant UCL system has been converted to a SWITCH interface (see Chapter 2).

One of the major problems experienced has been that the terminal protocol of ARPANET is oriented to single characters. This protocol, designed to suit DEC machines, allows intimate character-by-character exchanges between user and host; in some subsystems one can type a character, and the host will fill out the remainder of the word. This is a very pleasant facility to use, but when in addition, the basic mode of use is to let the host also perform all character echoing, then the load of the network in very short packets becomes excessively high and the echo delay long (in our case this is accentuated due to use of a communications satellite). With the per-packet tariffs being introduced by the PTTs, this mode of use would be prohibitively expensive. In the internetwork environment short character packets consume particularly large gateway resources; it is necessary to forward each packet separately, generating the appropriate acknowledgements in each direction. In view of the possible need to repackage each packet, not necessary in a homogeneous network, the processing load may be exhorbitant. In fact ARPANET did design a new terminal protocol with local echoing, but this has never been fully implemented. The trend to terminal controllers handling whole lines of text (PADs in Euronet, Higginson,1976D) means that this problem will not recur in quite the same way with other networks.

Flow control across the gateway has not been a problem so far, because each connection across the gateway is mapped into internal message queues which are automatically buffered on drum. We hope shortly to experiment with different amounts of use buffering on packets in the normal store-and-forward capacity; The processor overhead to transfer messages to and from backing store will become excessive as our requirements in the gateway increase - but removing it may require discarding packets which could cause yet additional delay. On experimenting with buffering, we will encounter congestion problems; a suitable flow control signalling across the gateway is an essential constituent of the design of the gateway SWITCH interface.

The need to handle networks with different maximum message lengths has not yet been encountered at UCL and can be avoided in an interactive terminal mapping. In general, however, fragmentation will be needed at the gateway - for instance for an ARPANET multi-packet message going to EPSS; the consequences of such fragmentation have still to be examined. In the Bridging protocols of EPSS packet boundaries are not rigid; the TCP of Chapter 5, being a general internetworking end-to-end protocol, has very flexible built-in fragmentation facilities.

4.2.3 SWITCH File Transfer Mapping

The more we move away from the communications level towards computer applications, the less likely we are to encounter conventions that are different but yet equivalent. File transfer certainly comes into this latter category. It involves a communications function, but is specified in terms of the characteristics of the file residing on the originating source of the file.

There is a strong need to attempt to map file transfers. A file transfer service is an essential adjunct to an interactive service, and is the basis for all text-processing use of networks. File transfer between EPSS and ARPANET is a necessary condition to support any real internetworking between those two networks. We have been considering various approaches to providing an internetwork file service. None has been implemented yet, but our present thinking on the relevant merits of each is discussed below.

We have considered three alternatives (Higginson,1976E) for supporting file transfer at the UCL gateway namely:

The implementation of EPSS file transfer protocols on Tenex

A direct attempt at providing a SWITCH-type mapping of the initiation,establishment, and progression of the file transfer

A centralised FTP controller capable of initiating transfer of the file onto the backing store of the gateway, and at some later stage forwarding the file to the host on the destination network.

The EPSS file transfer protocol seeks to avoid some of the pitfalls of ARPANET protocols, although no user experience has yet been gained with it. It is essentially network-independent and provides a flexible options list and checkpointing/restart capability. It is, therefore, a good candidate for wider application (EIN are adopting a variant). Furthermore, as general ARPANET use is concentrated on Tenex machines, an implementation of the EPSS protocol on Tenex could be propagated through ARPANET Tenex hosts in such a way as to provide a reasonable service. It is worth noting of course that users wishing to transfer files to hosts other than Tenex would need to perform two successive transfers, something akin to what we could do automatically with a centralised FTP controller, and shortly discussed in more detail. No final decision has been taken on whether to try to implement this approach, nor has the effort needed to achieve a solution been estimated. It obviously involves a much more intimate use of Tenex machines than the UCL group has attempted so far. This subject is discussed further in Section 4.4. In general, if we aim to define the level of gateway activity such that residual end-to-end functions reside in hosts, then either we or collaborators must be prepared to do those implementations.

The second approach, a direct mapping of protocols, requires a relatively simple central mapping facility, much of which is required already in SWITCH. More complex file transfer implementations are also required to support transfer of the file in any particular direction. We have chosen to follow this path, and are well into the definition and implementation of the necessary protocols. At first we will be testing these protocols to support file transfers front-ended to hosts such as the RL 360 and the ULCC CDC 6400; this is analogous to our initial work on SWITCH itself. We expect to have these facilities operational towards the end of the first quarter of 1977, with EPSS/ARPANET file transfer later in the year.

Before describing some of the details of the planned implementation of the mapped transfer, we will comment on a centralised controller. Here a user facility would probably be implemented such that a file transfer request would always be made direct to the gateway or another internetwork file controller. The central facility would allow simultaneous transfers across the gateway suitable for interactive use, but also would allow a staged transfer, where the complete file resides on a centralised repository until a convenient delivery time. This method does not require both source and destination to be up at the same time, and is similar to the MAIL facilities on Tenex. There messages were queued, and delivered when hosts come back on the network. The classic ARPANET pattern of network usage (i.e. file transfer with on-line users) relies on high bandwidth channels with comparatively low utilisation; the queued approach is used mainly with scarce resources, such as ILLIAC IV, archived files and MAIL. Further benefits of the centralised approach is the ability to keep records of all transfers for a specific period, allow retries etc. Disadvantages of the approach include the need for high amounts of backing store, the need to keep journals of transactions, and the inability to correct erroneous commands (e.g. mis-spelling of passwords). All approaches will need accounting and passwords to regulate gateway usage. At the present time this approach is under study as a research student study topic at UCL, and it is the direct mapped solution which will be used for real traffic loads in 1977 at UCL. The staged approach to file transfer is similar to classic Message Switching; it should be considerably cheaper, because we do not need to retain journals of the whole files, but just records of the history of the files.

We will now present a little more detail on the first direct mapping approach currently being implemented; a full description and analysis of the method must await implementation experience and use during 1977. We have defined an internal file transfer description format rather similar to the EPSS definition. The initiating file transfer request defines a control connection at the SWITCH level for that network only (in other words a SWITCH ALLOCATE function). The request is translated into the internal form for the request, and a SWITCH CONNECT call is made to establish a connection to the appropriate file transfer option in the host software in the destination network. In practice this is not a conventional SWITCH connection. It makes use of SWITCH addressing capabilities to route the request to the right place, it also uses the data path set up for the data exchange sequence of the file transfer (but not obeying the terminal protocols).

Once implemented and tried, we should probably add the correct additional primitives separately from the current ones used for virtual terminal mapping. In addition to the data path set up in readiness for the data transfer phase, a control path across SWITCH is now established such that the reply to the initiating message may be conveyed, together with subsequent control exchanges. In fact the EPSS style of control allows flexible variable-length control messages, each specifying options requested or information on the transfer. There is provision for confirmation or rejection messages to be returned to each of the options requested, such that a negotiated common understanding is achieved on how the requested transfer can be mapped onto the facilities available on each network.

One interesting additional point on this interface, is that it has been defined in such a way that it does not have to be interpreted locally. Attachment of the CULHAM processor system will be accomplished such that all relevant file transfer interface interpretation and response will take place in their front-end processor, through a separate control path multiplexed with their data path on the same physical channel.

4.3 UCL work in Internetworking Using Strong End-to-End Protocols

The TCP protocol, described in Section 5, was conceived as meeting both the objections to earlier host-host protocols, and meeting the needs of an internetworking framework, where strong end-to-end control allowed the need to make minimal assumptions about intermediate networks. It is no accident that the EIN/CYCLADES protocol bears many similarities; both were conceived to fill precisely the same role. Many of the conclusions, from experimentation with TCP apply to the EIN/CYCLADES work.

The internetworking environment in which TCP is most appropriate is described in most detail in (Beeler,1976). Here are propounded the ideas of a 'supernet' of gateways, centrally regulated by a gateway control centre, as would be the nodes of a single net. Each gateway obeys only a simple datagram protocol to its connecting network, but obeys a much more complex gateway-gateway protocol to handle dynamic routing and maintenance functions appropriate to a supernode. This environment is anathema to the commercial carriers, who would like to constrain carefully the Internetwork paths for reasons of control of traffic revenue, ease of accounting and national control of traffic. It is an environment which may well apply, however, in situations where the integrity of the data networks is of paramount importance.

The SATNET of Chapter 7, is an appropriate testbed for this form of internetworking, and the PDP11 machines programmed to carry out only the most basic of the 'super-node' functions are being installed in the four terrestial access points to SATNET (Section 7.2). Apart from the general implications of this approach, many detailed points arise from the introduction of Gateways; as the SATNET ones will not become fully functional until the second quarter of 1977, there has been some time to consider some of the relevant problems in detail. One important problem will be the maintenance of SATNET through the gateways; another is access in transitional stages where participatory measurement or other hosts are still obeying ARPA protocols. A note (Hinchley,1976D) deals with some of these problems. While some problems have a general significance (the maintenance of one network through another, will be important in some situations) others are more parochial issues of a transitional software nature.

At a more general level, we at UCL do not think that the idea of a centrally-controlled homogeneous super-net of gateways is acceptable outside DoD circles. Further work is needed to identify precisely what information must be passed between gateways. Adoption of full dynamic routing would put a large additional load on the connecting networks and would also be resisted by the Carriers. However, some alternate routing is required in many configurations. In the light of wide acceptance of X25 procedure, the approach of using end-to-end protocols across datagram nets may be of limited applicability, although directly relevant where participatory nets are closely 'bound' i.e. packet radio nets attached to a terrestrial net. The need to consider different approaches was propounded by (Lloyd,1976). From his work and that of (Kirstein,1976B), we have proposed the concept of transit and terminal gateways in the internetworking framework. This idea has led to some of the experiments we plan for 1977/78, which are discussed further in Section 4.4.

From experiments with the TCP, discussed in Chapter 5, we conclude that it is difficult to achieve both end-to-end flow control and high throughput. We believe that there are many important unresolved issues in the trade-offs on end-to-end flow control, Gateway-Gateway controls, high throughput gateways and protocol interference.

4.4. UCL Approaches to Internetworking Planned for 1977/78

From the preceding sections it may be seen that we at UCL clearly feel that the previous models for internetworking are of limited applicability. Overall guiding factors must be the types of services required, and the topology and behaviour of network interconnection which would meet these demands. Particularly attention must be paid to PTT developments in standards and planning.

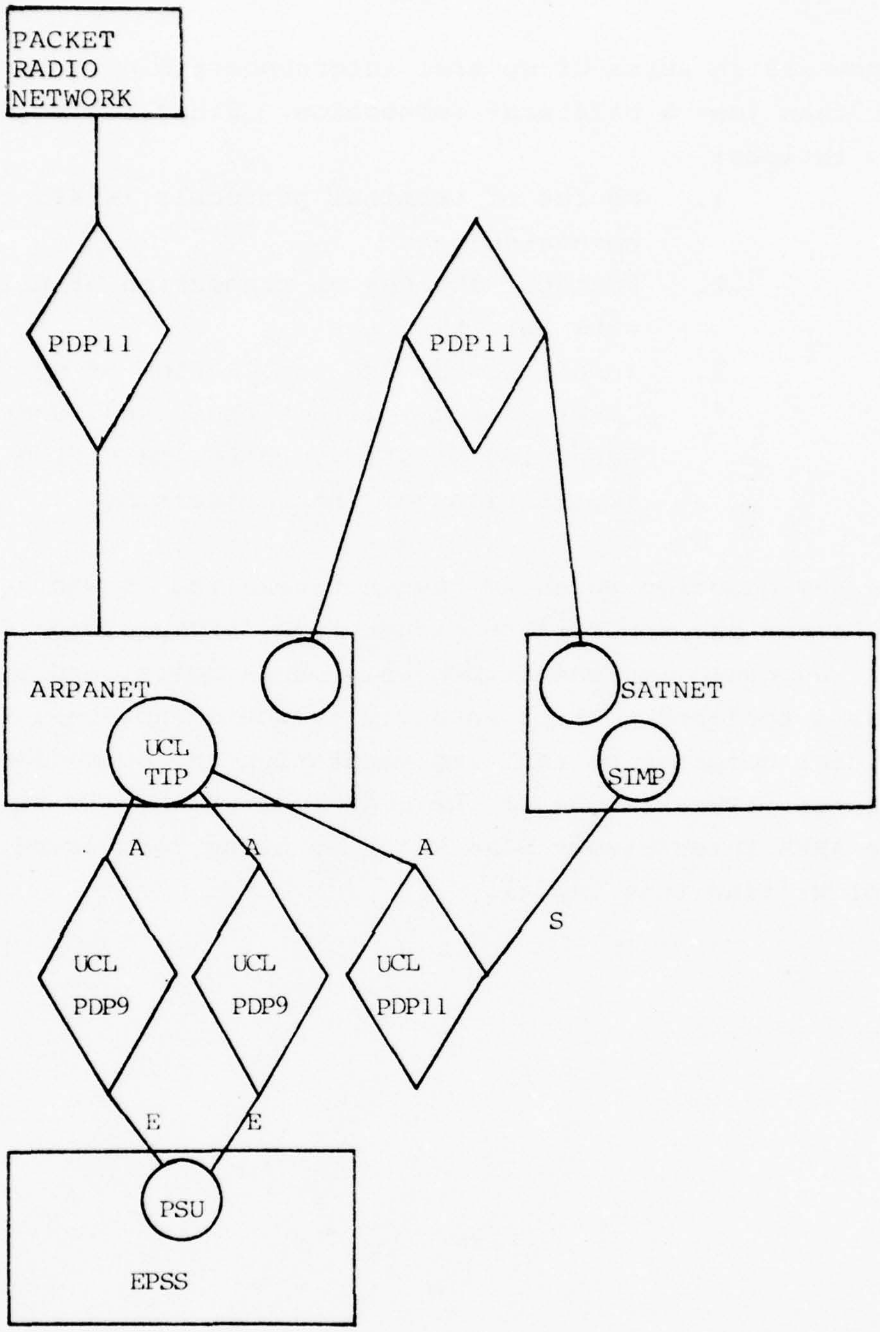
Solutions must resolve issues of addressing, routing, sequencing, flow control, congestion control, retransmission/secure delivery, fragmentation/reassembly, access control, status information and accounting. It would be presumptious to suppose that UCL activities in 1977/78 can hope to resolve all these problems, but we have a reasonable framework in which to validate sub-areas of the complete model. So far we have contended that end-to-end call/association is the desirable goal, not totally common higher-level protocols; we also think that substantial inter-gateway protocols and algorithms will be required, but not of the types yet proposed. We assume that this framework produces a 'strongly-connected' 'super-net' characterised by transit gateways acting as store and forward nodes, and subject to necessary congestion control only. At the periphery of the 'strongly-connected' set, will be terminal gateways talking to the source and destination of traffic. These may be hosts themselves, a single network path, or even multiple nets. The latter we define as 'weakly-connected' nets; although they may have gateways, they do not obey specific inter-gateway protocols. Performance, and throughput on connections to these 'weakly-connected' nets may be lower than in the 'strongly-connected' case. This may still be the optimum configuration when the traffic load does not justify full interconnection when the attachment is a limited transitional stage.

In the overall framework we must allow constructive co-operation between necessary use in some cases of the end-to-end protocols, and the use of X25 type interfaces. To achieve this co-operation the gateway will need to provide a TCP-like protocol in order both to provide a secure path across the 'strongly-connected' part of the internet connection, and also to provide an inter-gateway flow control mechanism between terminal gateways.

We presuppose that dynamic routing will not be justifiable, and alternate routing strategies may be needed. In the case of mapped virtual call gateways for example, this will require common algorithms at PDP11 gateways; when an onward call is broken, attempts must be made to re-establish it by an alternative route, and provide necessary forward and backward signalling to clear the broken sections of the route, and return to an orderly establishment. Moreover, access control status reporting and accounting will be a necessary part of the inter-gateway protocol.

We will clearly have a fertile test bed of many networks and gateways. Figure 4.2 shows a partial configuration of those that will be accessible from UCL by the middle of 1977. While none of these networks support X25 interfaces at this time, we have other activities planned for 1977 to implement this access protocol. We will certainly examine it experimentally at UCL for network interconnection, and hope to be in a position to experiment outside UCL at an early date.

In view of the current interest in X25, mention of some of our planned ac tivities is appropriate. An internal note(Kirstein, 1976C) has explored the mapping of terminal and call protocols such that X25 interfaces may be constructed and mapped onto existing network facilities. An alternative approach is to experiment solely with existing protocols, but in such a way that the results may be applied to X25 nets; the topics quoted earlier in this section on call mapping setup, disconnection, and flow control are good examples - especially when their significance



A = ARPANET interface
S = SATNET interface
E = EPSS interface

Fig 4.2 Partial Configuration of Networks and Gateways Accessible from UCL by Mid-1977 is expressed in terms of several interconnected networks rather than just a bilateral connection. Other X25 related topics include:

1. Choice of terminal protocols in X25 interconnected nets
2. Protocol choices on connection of private nets into X25 nets
3. In all cases, the application of any higher-level protocol within these environments is deserving of study, quite apart from wider implications of internetworking.

One key question which we have not resolved is whether to attempt to implement any appropriate higher level protocols on ARPANET hosts. We could implement them only on UK hosts, and use ARPANET just as a test-bed. Such an activity would, however, be very artificial compared to real implementation and operational experience. This is one of the questions which will be addressed in the ARPA Internetwork Plan which is being formulated at the time of writing this report.

CHAPTER 5: THE TRANSMISSION CONTROL PROGRAM

5.1 Introduction

In (Kirstein,1976A), we described our work on one of the possible procedures for end-to-end communication through concatenated networks. In this procedure, special software is put into each host, and a special Internet Packet (IP) is defined to allow a universal host-host protocol to be adopted across the different networks. At the gateway a minimum function is required in this approach; however the host-host protocol must be comprehensive. Cerf and Kahn (Cerf,1974) defined a possible protocol which has been much extended (Cerf,1976). Its embodiment is here called the Transmission Control Program (TCP).

The TCP has been implemented by three groups at BBN (Tomlinson), Stanford U (Cerf), and UCL. A final report of the findings of the experiments is in preparation (Cerf,1977). The UCL interest in the TCP has several origins. We are interested in evaluating the suitability of the protocol for different internetwork operating environments; we wish to be able to propose simplifications and improvements. Finally, the PDP11 SATNET Gateways of Fig. 2.2 use the Internet Packet Header; therefore we have a practical need to have an efficient implementation of the TCP to pass traffic over SATNET.

In Section 5.2 we will discuss the experimental tools. These include methods for Time Stamping packets at critical points, ability to change parameters remotely, and a careful design of the experimental controller and traffic generator.

In some ways, the experiments raised more questions than they resolved. For this reason, we also resorted to other experiments and simulation methods. In one different set of experiments, we examined the behaviour of an individual packet passing between the IMPs. In this way we were able to disentangle end-to-end behaviour from delays encountered inside the Data Transmission Network.

In Section 5.3 some actual experimental results are presented.

We have mentioned that the experiments described are difficult to carry out. For this reason we have also embarked on a simulation exercise. The simulation is particularly useful in evaluating proposals to improve the protocol. Some of our activities in this field are described in Section 5.4. Finally, conclusions are presented in Section 5.5.

5.2 Experiments and Tools

5.2.1 Aims and Environment

A great deal can be learnt about the TCP through theoretical discussion of protocol features and by stepping the implementations through the packet transmission process. However, many questions regarding practical implementations cannot be answered this way. What throughput can we actually achieve? If it is poor, why? What retransmission timeout should be chosen? What are the effects of different buffering strategies? In order to answer questions such as these, a quantitative programme was developed and a number of experiments were carried out between the various groups.

The emphasis of the experiments was on learning the effectiveness and constraints of TCP performance. Thus we were principally interested in examining the throughput achieved and the delays experienced by packets under various constraints. Very early in the programme it was decided to separate the TCP from the rest of the universe; thus data was to be collected as seen by the TCP process, and the experimental parameters varied were to be ones which were specifically features of the TCP protocol. The parameters used are:

i) Window size: This is the basic flow control of the protocol. Window size is defined as being the amount of information the TCP is prepared to have outstanding at any moment. Measurements were carried out on constant windows fixed by the experimenter.

ii) Maximum packet size: This parameter is determined more by the properties of the traffic and the transmission characteristics of the traffic medium than by an internal TCP constraint. At UCL it was fixed to be the size of the maximum ARPANET packet of 126 bytes (which allowed 90 bytes of TCP data).

iii) Thrash size: If data is being acknowledged at a very slow rate, the available window at the sender site may be very small, leading to packets being fragmented into many small segments. Thrash size is the parameter which defines the minimum size fragment the TCP will generate for transmission purposes. In the experiments undertaken, it was felt desirable to avoid fragmentation altogether- thus thrash size was set to the size of the packets being sent.

iv) Retransmission timeout: This is the length of time which passes before an unacknowledged packet is retransmitted. Ideally it should be set to just above the round trip time for the majority of packets. In practice, in experiments where one could assume arrival was guaranteed, an 'infinite' retransmission timeout was set to be 8 seconds.

A number of parameters are implementation dependent, or data dependent. These included items such as choice of buffering strategy which are not easily quantifiable or variable, but which may have to be taken into consideration when understanding final results. The items considered were:

i) Packet size: As the TCP window is clearly related to the amount of information outstanding it is important to study the effects on performance of the amount of data being transferred in any given packet. The limitations are those imposed by the need for a full timestamp field (18 bytes) at one end, and the maximum

ARPA packet size (90 bytes) at the other at UCL.

ii) Choice of acknowledgement strategy: It is possible either to force out separate acknowledgements (ACKs) for each TCP packet, or to piggyback ACKs on existing traffic in the other direction where possible. The choice of strategy may enhance or degrade performance, depending on the nature of the traffic flow; an ACK is needed to stimulate further transmissions, but may take up considerable bandwidth in a congested channel. UCL chose a strategy, whereby one checked that there was no packetised information pending anywhere, before forcing out an ACK. However, one may still compare the two strategies by setting up appropriate traffic situations.

iii) TCP buffering strategy: As noted above, the protocol places great importance on the relationship between window size and buffer constraints. As the measurements opted for a fixed window size, they also opted for a fixed buffer strategy.

The buffer strategy chosen, which proved flexible enough for most requirements, was to maintain a large fixed buffer pool from which fixed sized blocks could be chosen as needed. Space was grabbed according to demand for transmission, and for reception was chosen so as to maintain at least one outstanding receive buffer. If serious space limitation problems occurred, the experiment was abandoned; this tended to occur in the UCL experiments with larger window sizes as there was no necessary connection between the arbitrary window size chosen and the amount of space available. Problems caused by choice of buffering strategy were extensively studied by BBN, and led to a re-examination of many areas of the protocol, which were considered by the simulation study of Section 5.4.

Finally, in order to observe the interaction between TCP and the ARPA subnet, one could vary the ARPA packet type. This is a ready-made control for examining the effects of subnet flow control on TCP performance and vice versa. Type 0 ARPA packets are subject to the full ARPA end-to-end flow control;

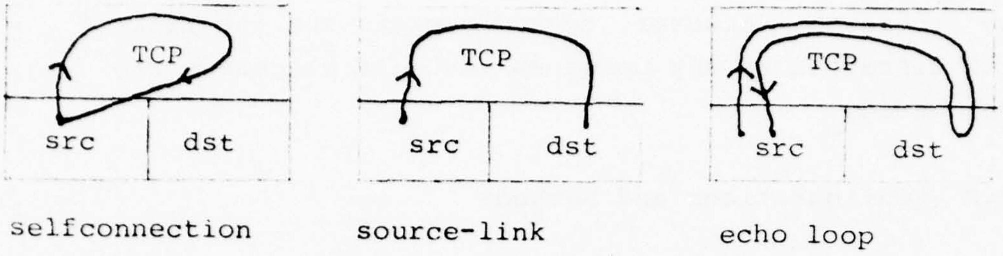
type 3 to none. However, type 3 packets run the risk of being discarded at any point en route if blockages are experienced.

5.2.2 Configurations and Methods

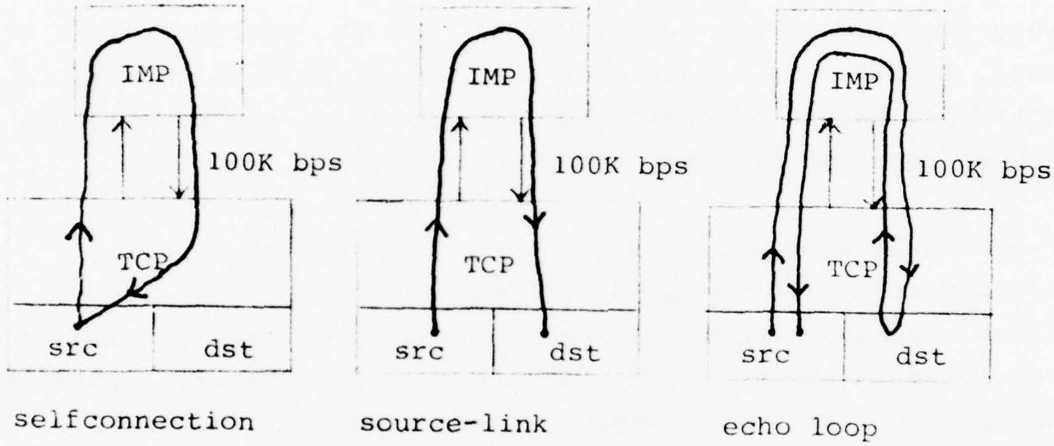
Three basic traffic situations were identified which it was felt could give information on different aspects of TCP performance. These were: the self-loop, the source-sink and the echo-loop. In the self loop, transmitted packets were received on the same TCP port. In the source-sink case, traffic was received on a different port which did not reply beyond acknowledging it. Finally, the echo-loop echoed received traffic to the sender.

The self-loop was used to give an indication of the optimum TCP performance in various situations. The TCP is doing a minimal amount of work in that it is transmitting and receiving on the same TCP; all acknowledgements are automatically piggybacked, and no extensive manipulations are required. This loop is not possible, however, if the TCP is talking to a TCP in a remote machine. The source-sink and the echo-loop provided more normal situations, reflecting full and half duplex connections. The source-sink case by itself, however, could not give information on end-to-end delays to a remote TCP, and this could only be obtained by using an echo connection. In order to obtain data usable at UCL, echo connections were almost always used for experiments to remote sites.

Behaviour was studied in several different physical configurations, illustrated in Fig 5.1. Two local loops were used. In the "internal loop", TCP packets were placed on a transmission queue, and when they reached the head of it were immediately transferred to the receive queue. In the "IMP loop", packets were transmitted to the London IMP across a local host interface. A similar "VDH-IMP loop" was tested but was abandoned due to hardware problems and an unsolved lock-up condition in the VDH RTP. Traffic was also sent to a remote site, usually Stanford, although some



a) Internal loops



b) IMP loops

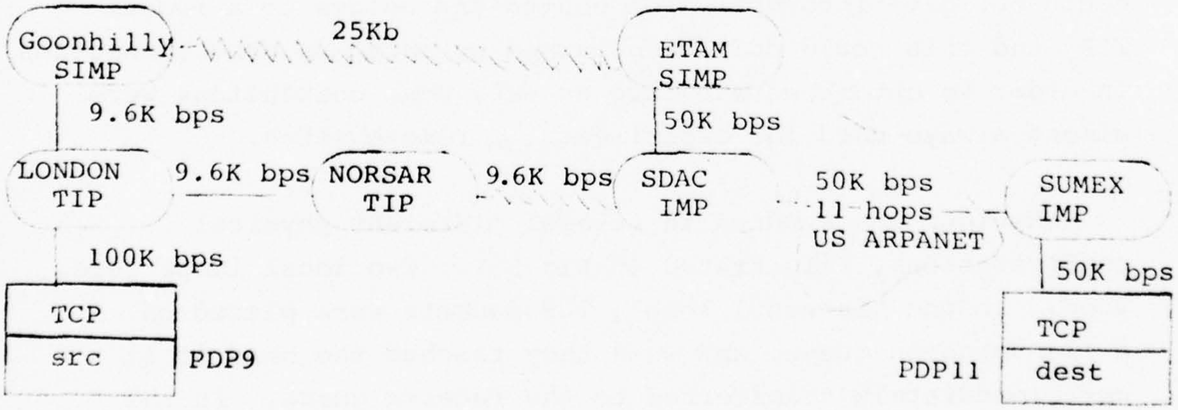


Fig 5.1 The Experimental TCP Configurations

connections were also made with BBN.

In order to examine the effects of UCL's relative isolation from the main ARPANET, two configurations were studied. In one, traffic was sent across the standard NORSAR route, a channel whose nominal capacity is 9.6K bps, but with an effective data capacity of between 5.1 and 7.6K bps for ARPA host-to-host packets depending on whether seismic data is being transmitted from NORSAR. This channel is clearly a bottle-neck, and one can expect delays to build up in Norway and at SDAC on the return journey. The other configuration was to use the 50K bps channel provided for the broadcast satellite experiments using FTDMA to five a one-way capacity of 25K bps. In this case the potential bottleneck is the 9.6K bps London-Goonhilly link.

To obtain figures on throughput and delay, one has to know the times at which various events occur, the critical interfaces being the point at which a packet enters and leaves the TCP. Thus, the basic tool for measuring TCP behaviour was a Timestamp packet, which picked up timestamp values at these points. As packets might be sent to either a sink process or an echo process, the following events had to be allowed for:

1. Generation at source process
2. Transmission from source TCP
3. Reception at destination TCP
4. Reception at destination sink/checker
5. Departure from destination echoer
6. Transmission from destination TCP
7. Reception at source TCP
8. Reception at source logging process

By including an **offset** field to point to the first unstamped location in the field, the timestamping procedure was made simple to code and operate. The format is illustrated in Fig 5.2.

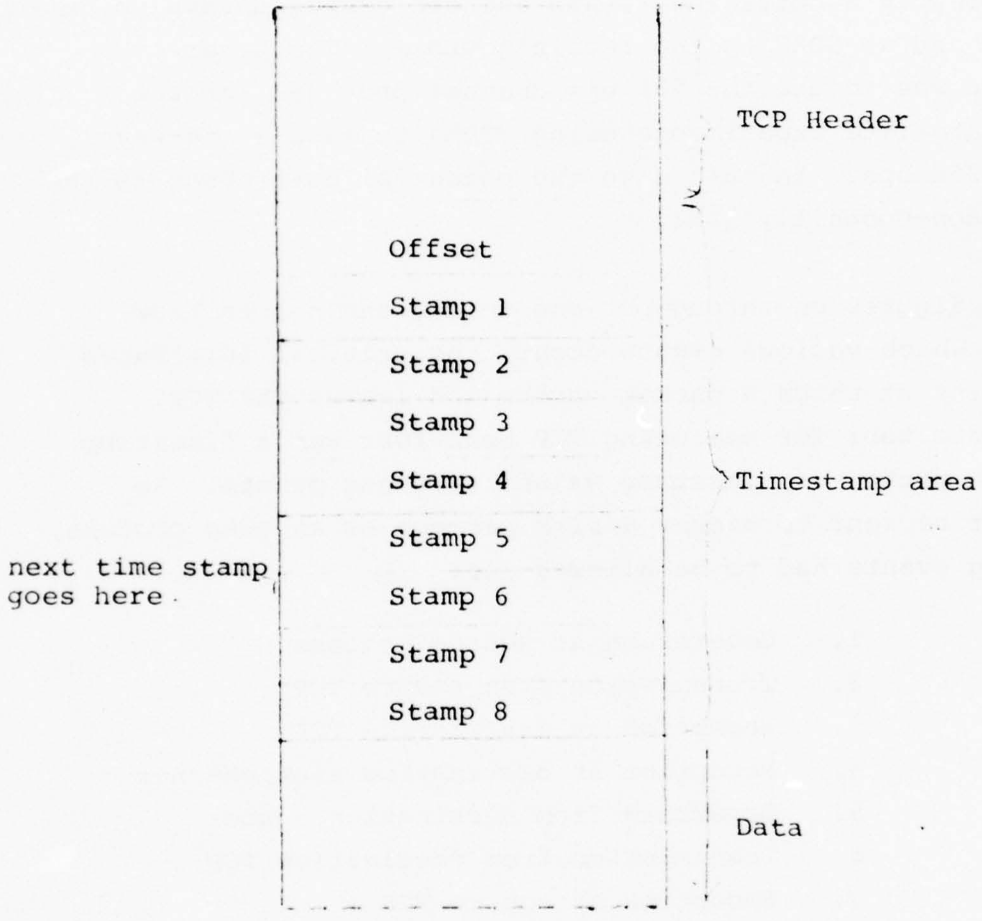


Fig 5.2 Timestamp Packet Format

Throughput can be calculated from the differences between corresponding timestamps in successive packets, delays from the difference between timestamps in the same packet. Throughput was measured directly in terms of the number of letters transmitted per second by the source process, where a "letter" in TCP terms is simply a unit of data which has logical significance to the user processes. It was quickly found that in order to understand what was happening in the TCP in tests between UCL and remote sites, one would need more detail than could be provided by the overall status times obtained from timestamp data. To obtain this detail, two levels of packet tracing were used. One, adopted at both UCL and Stanford, was to dump each TCP packet on transmission and reception. This was used principally in debugging and in various robustness demonstrations, such as demonstrating the correct behaviour of various protocol features. A more sophisticated version of the same trace was implemented at BBN.

In addition, packets passing through the London IMP could be traced using the techniques described in Section 8.6 of this report. Owing to the resources needed and a number of hardware problems, this procedure was only used once successfully. It proved extremely useful, however, in pointing out important and unsuspected TCP interactions.

5.2.3 Experimental Software

In order to support the facilities described above,UCL developed an integrated set of software to maintain the following TCP processes:

i) Manual exerciser: This opened and closed connections and provided basic message transmission facilities. It was intended primarily to demonstrate the correct functioning of aspects of the TCP protocol.

ii) Echoer: A passive process was maintained which always had open a listening echo socket, and which echoed any data sent to it. It was intended to be used with the remote parameter change facility and as such was rendered invisible both to the UCL experimenter and the data logging software. As the remote parameter change was only implemented at UCL, this proved to be an unwise decision in experiments with Stanford.

iii) Traffic Generator: This maintained a constantly blocked TCP transmission interface. Traffic of a fixed letter size was placed on the transmission queue whenever the TCP informed the process that the packet queue was becoming low.

This mechanism ensured that traffic generation would occur at the optimum rate, which could thus be measured directly.

iv) Parameter Change: This was a special process which could request or implement parameter changes for both the local and remote sockets of a connection, and thus in theory enabled a TCP experiment to be run by one side only. In practice it was only ever implemented at UCL, due to space restrictions at Stanford.

v) Data Logging: All incoming data was routed through this process which selected timestamped packets at regular intervals, and monitored parameter change responses as well as maintaining a watch over background information such as the opening and closing of a connection.

All these processes could be driven either from a command console or automatically from commands stored in a command file, written in a simple control language.

Post-processing facilities remained rudimentary. It was originally intended that the TCP data captured should be logged onto magnetic tape for subsequent analysis; but owing to numerous teething problems with both TCP and the tape hardware and software, this was never implemented, and proposals for a certain amount of run-time data reduction were not studied until late in the project. Both timestamp data and the TCP packet traces were logged onto the line printer, with the user specifying the interval at which timestamp packets were traced. Subsequent reduction was carried out by hand.

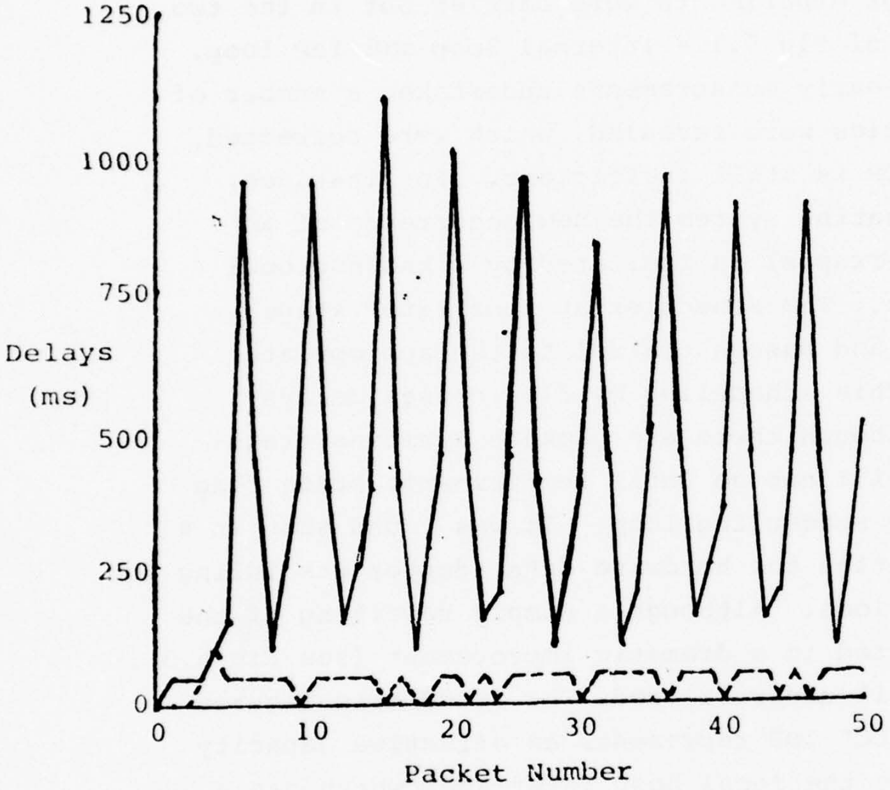
These limitations were never a serious problem, as the experimental program never reached the stage of generating a volume of data too large to handle. All throughput and delay measurements were made by sampling every tenth incoming packet from a group of 100; the sampling indicating the uniformity of the sample. Thus a very few calculations actually had to be performed for each point.

5.3 Experimental Results

For a fuller treatment of the experiments conducted and the results obtained, the reader is referred to the TCP report (Cerf,1977). This section outlines the results obtained and discusses their significance for the TCP in the UCL environment.

5.3.1 The UCL Implementation

A large number of experiments were carried out in the two UCL configurations of Fig 5.1 - internal loop and IMP loop. As a result of the early measurements undertaken a number of serious inefficiencies were revealed, which were corrected, although the UCL TCP is still inefficient. For instance, under our PDP9 operating system the new occurrence of an event (eg. I/0 interrupts) is indicated by a known global flag changing state. The scheduler at some later stage detects the change and passes control to the appropriate service segment. This scheduling by flag causes delays; for example, even though there are packets awaiting transmission, the data will not go until the transmit state flag is inspected in the scheduling loop. It was found that in a heavy traffic situation the hardware data adaptor was idling for relatively long periods. Although a simple rewriting of the TCP scheduler resulted in a dramatic improvement (see Fig 5.3), the adaptor was still underutilised. The round trip figures of Fig 5.4 for the UCL IMP represents an effective capacity of 26.35K bps across the local host interface, which has a nominal rating of 100K bps. This suggests that even in the comparatively idle situation represented here, the UCL TCP is compute-bound. Results in later experiments support this conclusion.



Window size: 96 bytes
Packet size: 18 bytes
Thrash size: 18 bytes
Retransmission: 8 seconds
Configuration: self loop to IMP
Solid line: old scheduler
Broken line: new scheduler

Fig 5.3 Transmission Queue Delays in the UCL TCP

Configuration	Round Trip Time (sec)
UCL Internal	0.073
UCL IMP	0.0107
Stanford via NORSAR	2.06
Stanford via Goonhilly	1.58

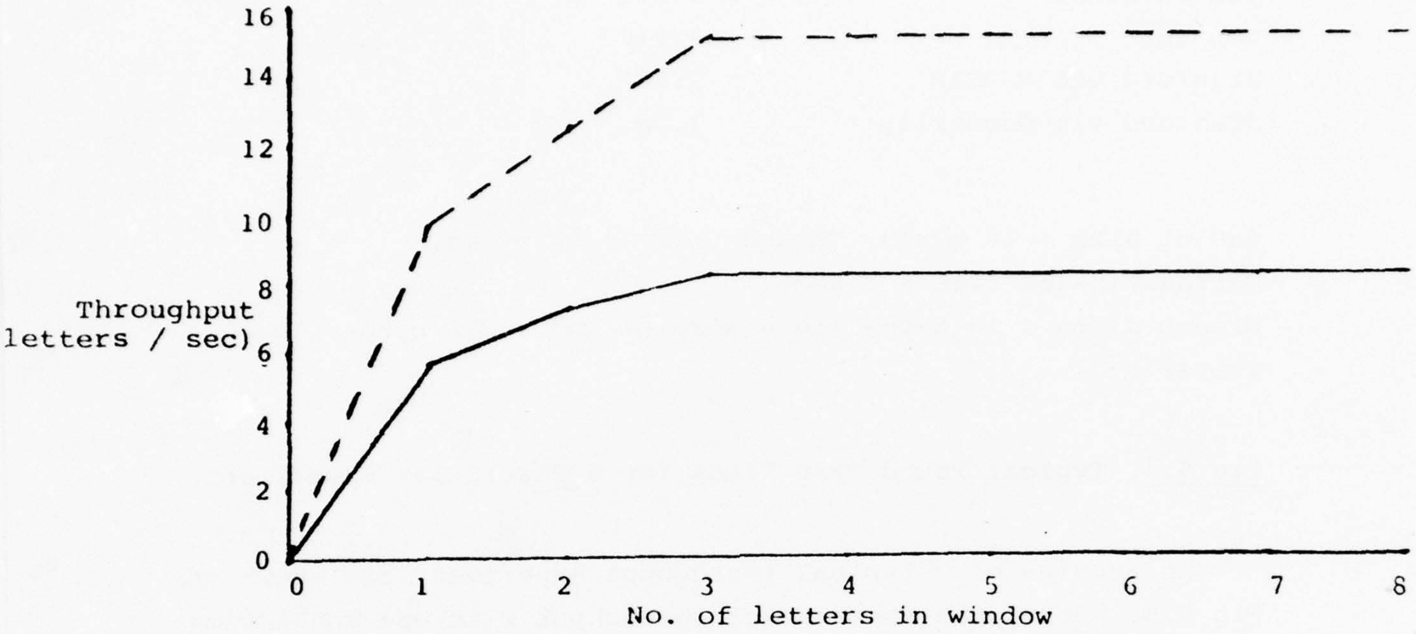
Packet size = 18 bytes, Thrash size = 18 bytes,
Retransmission time = 8 secs.
Window sizes = 18 bytes (to Stanford) and = 256 bytes (local loops)

Fig 5.4 Typical Round Trip Times for a Particular Experiment

The results of a typical throughput experiment are shown in Fig 5.5. Here, the increase of throughput with one connection vs with two connections is 1.86 (for the internal loop) and 1.85 (for the IMP loop). The remarkable agreement of these two figures again supports a simple process-bound model of the UCL TCP. In this model the throughput per connection for a number of connections each handling a similar load is inversely proportional to the number of connections. Such a model would predict the throughput ratio to be 2:1. The difference would be due to the minimum load of idle processing not being taken into account.

Throughput experiments involving variation of letter size, and choice of ACK strategy also suggest that in situations where the UCL TCP was being used for communications in a high capacity network, its use would incur considerable processing overheads. Although other groups had the same experience, this does not mean that the TCP is necessarily wasteful of processing cycles. The problem is largely due to inefficient design; the basic reasons for this will be discussed later. New TCPs currently being implemented by other groups appear to be much more efficient than their earlier implementations, and doubtlessly we would see the same thing with the UCL TCP if it were rewritten suitably.

a) Internal loop



Letter size: 18 bytes   Thrashsize: 18 bytes
Retransmission: 8 seconds
Solid line: echo loop   Broken line: self connection

b) IMP loop

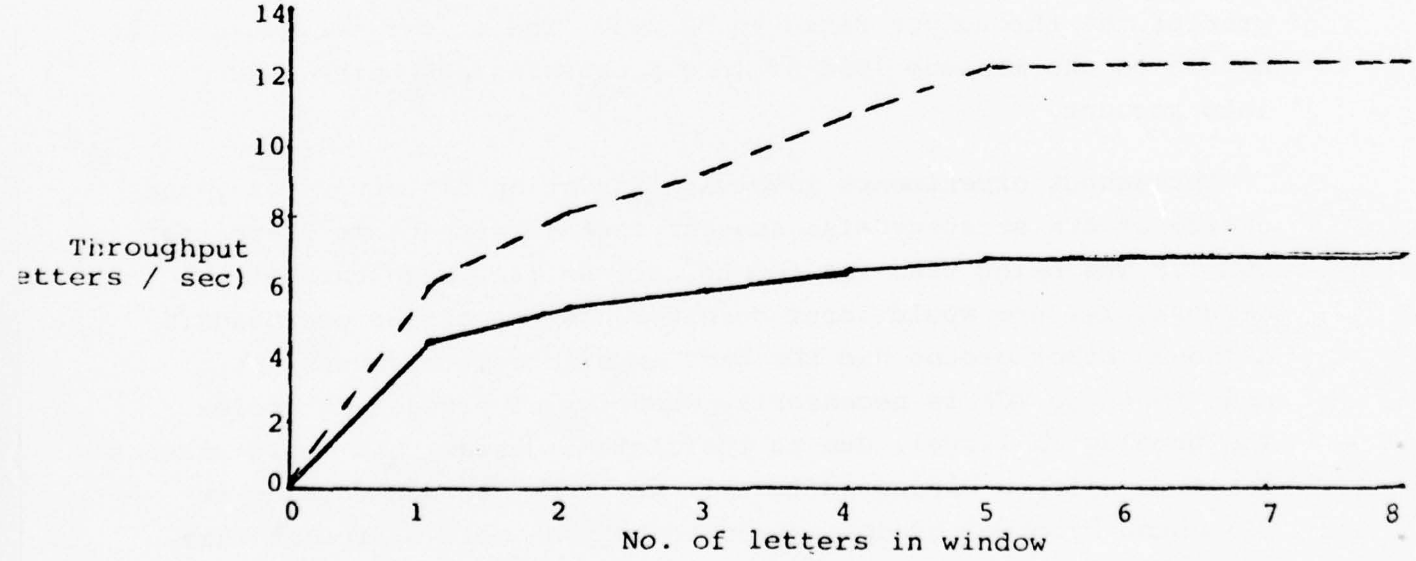


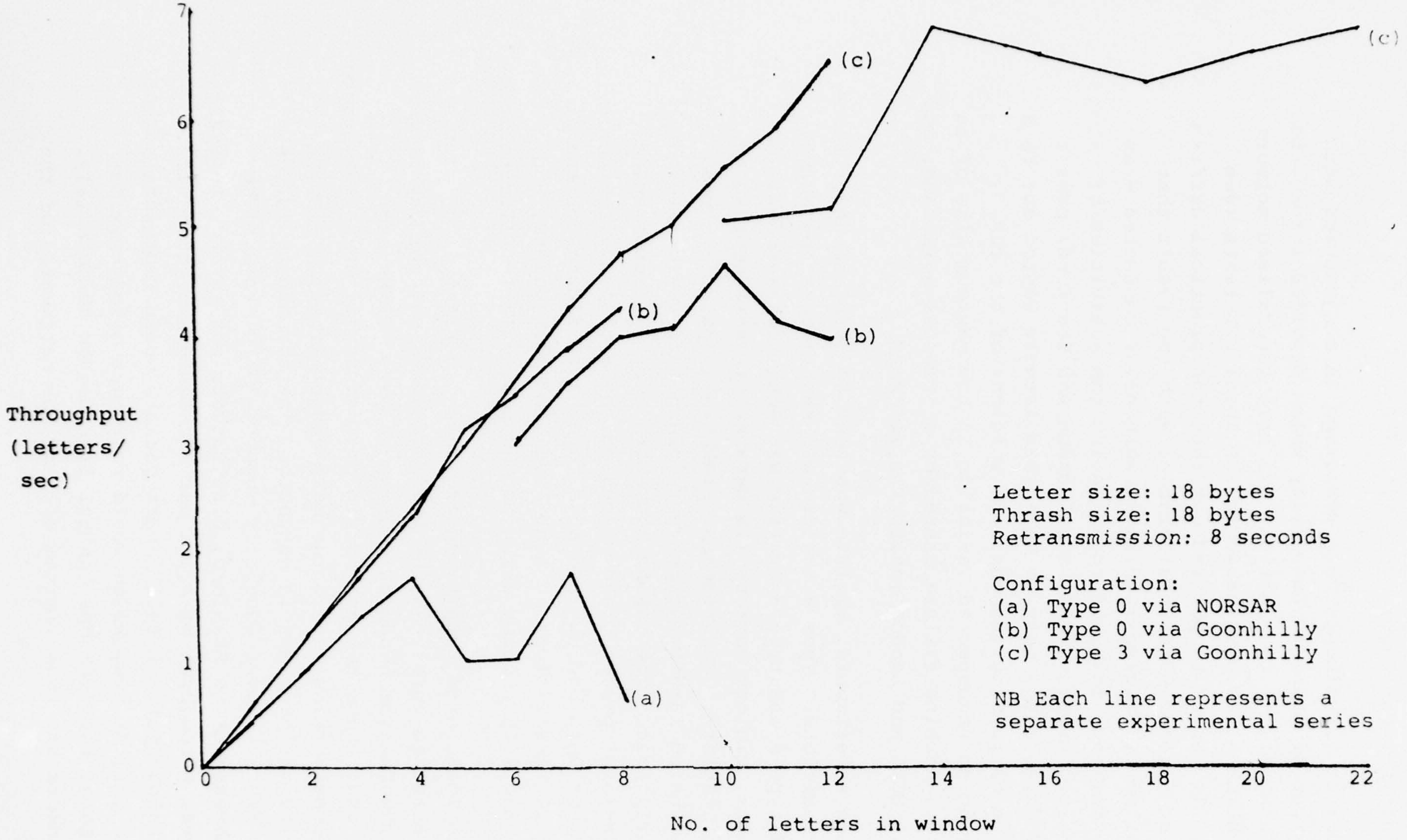
Fig 5.5   Throughput for one and two connections

5.3.2 Internetwork Experiments with TCP

A number of experiments were performed in conjunction with the TCP group at Stanford University which provided information on the TCP's ability to function in a more generalised network environment than was obtainable in the local UCL tests (see Fig 5.1). It should be pointed out that the practical difficulties encountered were considerable, with the result that many experiments originally planned were never conducted - an extreme experience involved coordinating the activities of four dispersed groups (UCLA, NCC, London and Stanford) over a period of three days, only to have them largely wasted due to a combination of running incompatible versions of the TCP, encountering an unsuspected variation in the maximum size of an ARPA packet. machine failure caused by a tropical hurricane, and failure in UCLA and London monitoring software.

In these experiments, traffic was sent to Stanford under three sets of conditions. Type 0 traffic was sent across the NORSAR link, and type 0 and type 3 traffic was sent across Goonhilly. For all runs, minimum packets (18 data bytes) were sent to an echoer at Stanford. Experiments using type 3 packets across NORSAR and using larger packets were planned but were never successfully held. The NORSAR run shown here was accompanied by the low-level packet trace discussed in Section 8.6. From the data thus obtained we were able to extract figures on the number of retransmissions at line level for most points, and also some packet histories which proved particularly illuminating.

Minimum load round trip times are shown in Fig 5.4. A simple model constructed purely on the basis of known channel capacities (less overheads from headers, routing packets, RFNMs etc), indicates that these delays contain no features of any significance at an internetworking level, the main constraint being simply the capacity of the channels en route. The throughput curves (shown in Fig 5.6) over Goonhilly support the picture. The maximum throughput to Stanford, 6.85 letters per second, represents a line-level throughput of 4.11K bps for letters of this size. The same throughput for full length packets would represent a line level rate of 7.9K bps, which would represent a near-maximum utilisation of the 9.6K bps channel from London to Goonhilly. Type 3 packets are thus limited only by the narrowness of the



Letter size: 18 bytes
Thrash size: 18 bytes
Retransmission: 8 seconds

Configuration:
(a) Type 0 via NORSAR
(b) Type 0 via Goonhilly
(c) Type 3 via Goonhilly

NB Each line represents a separate experimental series

Fig 5.6 Throughput to Stanford

channel. The effect of having two layers of host-to-host protocol start appearing when we consider the throughput figures for type 0 over Goonhilly. Disregarding the rather anomalous peak observed at a window size of 10 letters, the maximum of 4.25 letters per second corresponds to a line level rate of 3.26K bps and a predicted maximum of 5.7K bps, rather below the maximum channel rate. This would appear to be due to the maximum limit of 8 IMP buffers for a connection imposed by the ARPA as throughput starts to level off at this point.

The full implications of the effects of having two host-to-host protocols in operation (a normal internetworking environment) became clear when we considered the information gained from the line-level measurement of Section 8.6 obtained with the NORSAR link. The picture here is more complex. The maximum throughput of 1.75 letters per second (1.34K bps at line level) occurs at a window of 4 letters; this is followed by severe degradation accompanied by a high level of IMP retransmissions from London to NORSAR. The effective capacity of the NORSAR-SDAC link is reduced considerably by the introduction of seismic data at NORSAR. A sufficiently large window in the source TCP will accordingly lead to the formation of a queue at NORSAR of more than 8 packets - the maximum allowed. This will cause an IMP-level retransmission after 2 seconds and a consequent degradation of throughput. IMP-level retransmission rates of up to 32.3% were observed. These figures confirm something that European users of the ARPANET have long suspected. The ARPANET IMP-IMP protocols, which are perfectly satisfactory on the high-capacity low-delay 50K bps lines for which they were designed, are not really adequate for the low-capacity high-delay satellite link which connects us to the US side. The figures show more than this, however. TCP throughput is reduced further by the fact that the TCP duplicates features of the ARPA net protocols - in particular, retransmission and positive acknowledgement. Since the IMP cannot distinguish a TCP retransmission as such, we end up in the situation where the London IMP is still retransmitting a TCP retransmission of a packet that the UCL TCP already knows has arrived successfully. Thus in this sort of bad situation, the features built in to give end-to-end security lead to a duplication of IMP-level effort, increasing performance degradation as the duplication of packets will only be detected by the receiving TCP.

5.4 Simulation Studies

As a result of studies undertaken at BBN (Cerf,1977) into buffer strategy, a number of proposals were made regarding the relationship of TCP window size and buffering strategy. These were investigated at UCL by a simulation study of methods of credit return. In TCP credit is returned relative to the last acknowledged packet and defines a contiguous area (the window), extending in the data stream from the end of the data in the last packet acknowledged, in and only in which the sender is allowed to transmit/retransmit data.

Generally, as packets are acknowledged by a TCP on behalf of a receiving process, credit is also returned dependent on the amount of unfilled storage in the Receive Buffer Queue and in TCP storage allocated for the Receive Packet Queue. Two general approaches may be distinguished. In conservative schemes, the allocated credit must be guaranteed by available storage which may come from internal TCP buffer exclusively allocated to the process, and from any user buffers which are provided. In the variant proposed by BBN there are no buffers provided by the TCP and the window size is governed purely by the buffers made available for the connection with the received data being placed directly in the user buffers. In optimistic schemes, of which the fixed control window used in the experiments discussed in Section 5.3 is an example, sometimes more credit may be granted than is actually available, on the assumption that the extra storage will become available later. It is clear that, provided no subnet loss occurs, a conservative scheme is 100% efficient, since space is always guaranteed on the receive end. However, with optimistic schemes, it is possible that packets may have to be discarded as the promised space may not, in fact, be available when they arrive.

In considering these schemes, we should bear in mind the general criteria by which they may be judged. These fall into two groups- resource protection (achieving maximum efficiency, minimum storage utilisation, minimum CPU utilisation by the TCP) and service utilisation (maximising throughput, minimising delay). Throughput will be governed by the minimum of the channel bandwidth, the sender production rate and the receive consumption rate.

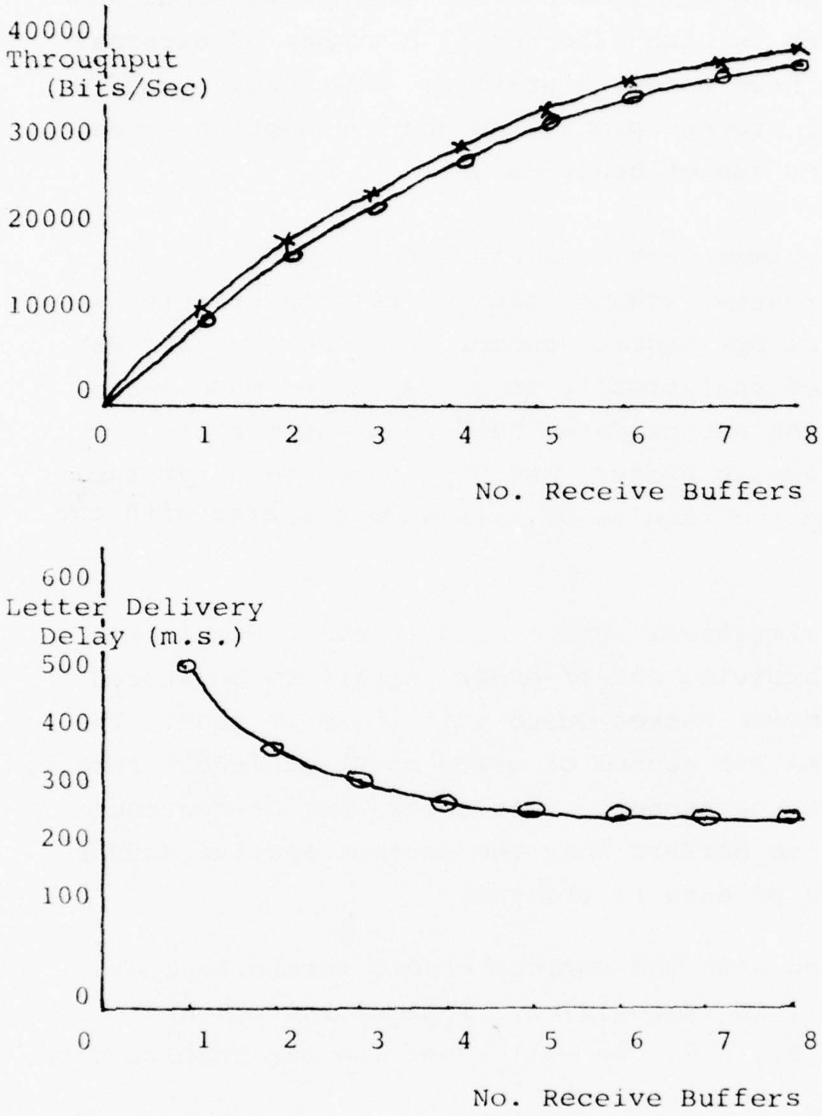
A further objective may be that the performance should depend as little as possible on the processes connected, though this has been deliberately rejected in BBN scheme. The criteria are somewhat incompatible; it is unlikely that any one protocol will succeed in optimising all flow control objectives simultaneously. The simulation is also affected by a number of external events of the kind we have noted in previous sections: traffic distribution patterns, process production and consumption rates, local host activity and subnet behaviour.

Three particular schemes were simulated (Edge,1976) - the BBN version of a conservative scheme, the general conservative scheme, and the general optimistic scheme. The conservative BBN scheme was also studied analytically on a simplified model with an exponentially varying subnet delay held constant for all packets in the same receive buffer, and receive buffers for the optimistic scheme, and the results of this were compared with the simulation results.

Two variants were simulated. One - used in every simulation study - requires all arriving out-of-order packets to be stored temporarily in the Receive Packet Queue - if there is room. The other is based upon the BBN scheme of using constant length Receive Buffers, made known to the sender. This allows out-of-sequence entry of data to Receive Buffers when the correct Receive Buffer for a particular piece of data is present.

The results obtained with the various credit return schemes are discussed in detail in (Edge,1976). Typical results are shown with figures in Fig 5.7. We will summarise the results here.

In general, throughput with the BBN scheme is very sensitive to particular process behaviour. It will improve considerably if one has a large number of small buffers rather than a small number of large buffers. However, throughput and delay vary in a similar way to buffer utilisation and are shown to be highly dependent on it. Consequently this scheme is very sensitive to letter/buffer size mismatch. General sensitivity of the BBN scheme is less in an environment of low subnet delay, though as with all schemes, throughput and efficiency are reduced if the subnet is unreliable. Ideally, a large number of small Receive



× = Analysis
O = Simulation

Receive Buffer size = 400 bytes
Letter size = 400 bytes
Mean propagation delay = 100 m.s.
Subnet loss factor = 0

Fig 5.7 Conservative (BBN) Scheme
Variation of Number of Receive Buffers

Buffers should be used - given a certain amount of Receive Buffer storage available, it should be split into many small buffers rather than a few large ones. The letter size should be matched to the buffer size. When subnet propagation delay is low, these factors are less crucial.

For the general conservative scheme, high throughput is obtained at the cost of setting aside large quantities of TCP storage. Sensitivity to process behaviour is also apparent, but less so than in the BBN scheme. Roughly equal amounts of total storage (from the TCP & Process) are needed in these and the BBN scheme to achieve the same throughput. When both schemes are idealised (with respect to process behaviour), storage (Receive Buffers in BBN and Receive Packet queue storage in the General Conservative scheme) spends most of the time 'in the TCP' performing the functions of a Receive Packet Queue.

With the optimistic scheme, high throughput is obtainable at a cost to the efficiency, and to the amount of TCP storage set aside or average TCP storage utilised from a common pool. Much less storage overall is needed to obtain a given throughput than in either of the other schemes.

Figure 5.8 briefly summarises the extent to which the various schemes satisfy the flow control criteria listed above.

Criteria	BBN Scheme	General Conservative Scheme	Optimistic Scheme
Efficiency	2	2	1
TCP storage	2	0	1 (Trade off with efficiency)
TCP CPU time	2	2	0
Throughput	1	2	2
Delay	1	2	2
Contention	0	2	1

Fig 5.8 Ability of Various Schemes to Meet Different Criteria

2 : Criteria easily achieved.

1 : Criteria achievable, but only at cost to another objective, or through special process behaviour.

0 : Criterion cannot be achieved.

It is seen that no single scheme manages to achieve all flow control objectives. Some are achievable only at the expense of others or through special process behaviour. In practice, choice of a particular scheme, for a communications system, will depend on the priority attached to each objective. This in turn will depend on the availability of resources within TCP using hosts, the kinds of communication envisaged, and whether or not processes served by TCP's can be expected to participate in achieving flow control objectives.

The use of a particular scheme by a receiver TCP is completely transparent to the send TCP with respect to protocol functions and interferes very little with protocol functions at the receiver. Thus communication between machines using different schemes is not disbarred, and each TCP in a network may use one, several, or all the above schemes, depending on resource limitations and its requirements.

5.5 Conclusions

The lessons that can be drawn from the UCL experience with the TCP fall into two broad categories: those that related to our specific implementation, and those of more general interest. Many of the implementation lessons we learnt were not drawn directly from our experiments but from the background work in setting them up. The UCL TCP, in common with most of the other early TCPs, was large and complex; typically the size of the TCP was of the order 8-10 K. It took over 6 months to implement, and was accompanied by long and painful debugging sessions. In these circumstances it is not surprising that our local measurements show low throughput and high processor time. The only first generation TCP that did not show these defects was the TCP0, a single-connection TCP implemented on an LSI 11 at SRI, which was rendered much simpler by discarding the multiplexing function. Obviously, this is not in general an acceptable choice, but it does raise the question of what protocol simplifications can reasonably be made, which we shall return to later. Later versions of TCP0 have included the multiplexing function with relatively little increase in program size.

Undoubtedly, part of the reason for TCP inefficiency is connected with various operating system features which have high overhead costs.. In the BBN TCP, TENEX JSYS traps were identified as a major source of overhead. In the UCL TCP intersegment calls and the run-to-completion nature of the operating system were identified as sources of waste. However, a much more fundamental reason for the problem is simply lack of experience of implementing protocols of this order of complexity. For instance, we saw that rewriting a mere 50 lines of code - the TCP scheduler - resulted in a 10-fold improvement in the transmission queue delays. Other TCPs now being written seem to be much smaller and more efficient than the early ones, based on the design lesson learnt from the first implementations. One problem is that the TCP is really only completely specifiable in a suitable high level language; although we did have an incomplete BCPL listing of Stanford's TCP, the original specification was written in English and we did not attempt to write our own in a suitable high-level form. There is a clear need for such a specification for future implementations.

Given that the implementation left much to be desired, the question still remains as to what changes, if any, should be made to the protocol, and how will they impact the efficiency of the implementation. The BBN proposals studied in our simulation work were clearly intended to make the implementation less complex. From the point of view of protocol design, the changes proposed are fairly small - the sequence-numbering scheme is partially decoupled into separate control and data sequences, and the window becomes specifically a description of the available buffering at the receivers end. From the implementation point of view, the changes are dramatic; they mean that packets can be placed directly into the correct position in user buffers, thus avoiding the whole process of letter reassembly. Moreover, this means that the TCP no longer needs to manage its own internal storage for "cushioning" a user process currently unable to accept received data. The effect on the TCP is a reduction in its intelligence, as it is reduced from being an active participant in making flow control decisions to being a carrier of decisions made effectively by the receiving process.

As the simulation studies noted, this reduction in intelligence makes the throughput/delay characteristics much more sensitive to the behaviour of individual processes. In particular, efficient throughput requires that send-letter size be matched closely to receive-buffer size, and the end-of-letter flag thus becomes an indication to the sending process that it should pad out the rest of the current transmission buffer. In other words, the concept of "letter" takes on the specific meaning to the TCP of "receive buffer size". Since the earlier function of "end-of-letter" as initiating reassembly automatically disappeared in the BBN scheme the idea would appear to have no meaning to the protocol in the proposed revision. To a user process, a letter was intended to be a unit of data with logical significance. Not all user processes will have need for such structuring, and thoses that do, will contain indications of it in the data field in any case, so the user-orientated definition of a letter would also appear to be redundant. At least under the BBN scheme, therefore, there is a strong argument for eliminating the "letter" altogether.

If the simulation studies suggest that the BBN proposals potentially simplify the relation between the TCP and the user processes, the crossnet experiments indicate the failure of the TCP to relate to existing ARPANET protocol structures. This would appear to be a general problem for internetworking - the potential for redundant activity between protocol layers has already been pointed out for the X25 protocol agreed for PTT network access (Pouzin 1976). We certainly intend to study the implications of this situation as part of our internetworking activity in 1977.

The importance of these interactions depends very much on the future direction of internetworking. If all internetwork communication is to be handled by a network of gateways acting as a transit net, then we have a very clean design environment in which the problem disappears. On the other hand, the situation of concatenated networks across which we must maintain an end-to-end connection of an indefinite number of network hops presents serious difficulties of this kind for any end-to-end protocol. What appears to be needed is a defined "network interface" which would enable the protocol to decide when certain functions should be suspended. This was done on an ad-hoc basis to the ARPANET protocols governing the establishment of a virtual connection, since this obviously duplicates the same function in the TCP. The normal procedure of defining a link number of link 0 was suspending in favour of using a "well-known" link number agreed in advance. However, it is clearly not possible, in general, to suspend a function by user diktat and a more formal procedure should be found.

In short, the conclusions we have drawn from our experience with the TCP suggest that not only would it benefit from more careful design, but also from design decisions and protocol changes which make it less rather than more intelligent. As an end-to-end protocol, the TCP should act as a carrier for flow control decisions rather than as a participant in them. As an internetworking protocol, it needs to find a method of dividing responsibility for protocol functions between itself and the various network protocols it encounters in transit. By following these leads, we should be able to produce a TCP which is better designed, smaller, more efficient and less complex than the existing one.

CHAPTER 6: UCL ACTIVITIES WITH EPSS

6.1 Introduction

In Section 4.5 (Kirstein, 1976A), we mentioned our plans to connect one PDP9 to the UK PO Experimental Packet Switched Service (EPSS). The UCL configuration has been shown in Fig 2.1. Our progress has been slower than anticipated, mainly because EPSS has had some unexpected delays.

At the beginning of 1976 EPSS was providing an echo service for hardware testing by which all packets sent were subsequently returned. The service was gradually advanced to provide calls to oneself, calls to test number and calls to other users on the same Packet Switching Unit (PSU). This last state lasted a considerable time and was made more difficult by the availability of only three 48K bps ports on the London PSU, to be shared between the five users requiring access, of which we and the Rutherford Laboratory are two. Service was (and still is) Monday to Friday mornings only (8.45 to 13.00), and this restricts access to what is normally regarded as prime time. A marked improvement in the level of service came in the late Summer when trunk switching, a second London PSU and access from character terminals became available. With minor improvements this is the current situation. A PSU became available in Manchester and in Glasgow during the course of 1976.

We have had to modify our communication hardware to cope with EPSS; The modifications are discussed in Section 6.2. We decided to develop the lower level software in a structured way. Our progress here is discussed in Section 6.3. It was felt very important to get early real experience with EPSS; therefore both the Rutherford Laboratory and ourselves decided to mount an experiment. In this we at UCL simply adapted the software module driving the communication device on the leased line to be able to make a call through EPSS. RL had to make many more modifications, since they drive EPSS via a front-end computer. This activity is discussed in Section 6.4. We still plan to support services better via EPSS; our immediate plans are discussed in Section 6.5.

6.2 UCL Hardware and Line Handlers

In order to connect to EPSS we purchased a Transmission Protocol

Unit (TPU). This unit was designed by a group of users and the PO; its development cost was supported by the PO and it was manufactured by Computer Electronics Ltd. The TPU checks incoming packets for line sequence number and CRC; it passes all packets received to the processor and appends a status byte to the end. On transmission the processor passes a packet to the TPU, which adds a line sequence number and CRC before sending it on; the TPU keeps a copy of the packet so that any retransmissions necessary can be generated internally. The TPU has a V24 modem interface to the processor, so that standard interface boards can be used on the processor. The usual control signals of the V24 interface are used to control the TPU and to signal status from the TPU to the processor. The TPU performs the line initialisation and loop delay measurement procedures required by EPSS without processor interventions.

The standard PDP9 synchronous adaptor, which is used for all other remote links except the CDC, had to be modified in order to cope with the higher speed of the EPSS line. The adaptor had to be modified to switch automatically from its "Ignore SYN" state to its "Data Channel Receive" state so that characters would not be lost if an interrupt for another device were being serviced. The software handler for this adaptor has been written to support both the TPU and the "Simplified Protocol" of EPSS. This second protocol is available as an alternative to using a TPU; however, it has a much higher processing overhead and it is not clear that it can be adequately supported by a PDP9 at 48K bps. The code was written as a "backup" in case the TPU was late or unreliable, but neither of these eventualities occurred. The simplified code will be used to test out the second 2.4K bps line (see Section 6.5) when it arrives.

6.3 Software for EPSS Attachment

The line handler, the adaptor and the TPU were tested using a packet generator in the PDP9 and the loop facility either in the PSU or internal to the TPU. Several points of ambiguity with regard to TPU operation were uncovered, particularly with regard to the status signals and SYN sequences. These came to light only when the TPU was connected to a processor through a standard modem interface and these points were passed on to other TPU users.

Much planning and some coding of the call-level software had been done during 1975; however greater clarity from later documentation and a better understanding of the SWITCH interface necessitated some revision. In addition it was realised that the number of protocols that had to be supported, and the desire to be able to add monitoring tools and possibly datagram access, required that a more rigid and well-defined interface between programmed processes and the EPSS call-control software be defined.

Implementing this new interface involved reorganisation of the call-control software. The delay that this caused was well justified by the ease with which the standard RL 360 driver from the RL/ARPANET system was interfaced to EPSS.

This unconventional use of EPSS is discussed in the next section. The standard interface is also being used for the File Transfer and Virtual Call Protocols, which are planned for implementation, and for the Virtual Packet Terminal Protocol (i.e. the protocol for character terminals dialling up the PSE), which was coded and being debugged at the end of the year. All the EPSS software runs under SWITCH, and could be run in a multi-machine, multi-network environment.

6.4 The Packet Hasp Experiment

The Packet Hasp Protocol was developed jointly by the Rutherford and Daresbury Laboratories as the simplest modification that could be made to connect a standard multileaving workstation to the Hasp program on the 360 over EPSS or using EPSS protocols over leased line. The workstation uses a single EPSS call onto which all the data for the card readers, line printers and interactive terminals is multiplexed in the standard way. So far, use of this protocol between UCL and RL has proved its viability and usefulness. During the first quarter of 1977 we will also monitor the performance of EPSS; the Packet Hasp system is particularly suitable for this because it keeps up a one second exchange of messages even in the absence of user traffic.

The Packet Hasp system became operational during the fourth quarter of 1976. We now run test sessions for an hour or so per day, in which UCL staff do routine work via the UCL TIP and the PDP9/ARPANET/EPSS gateway. At present the system requires a dedicated PDP9; both for this reason and because of the limited availability of EPSS, we do not plan to offer a service outside the UCL group before the second quarter of 1977.

Packet Hasp requires a significantly greater number of packets to be passed over EPSS than is needed for that amount of useful data transmission. This extra traffic during idle periods is useful at the moment. However, tariffs have been announced for EPSS, which are scheduled to commence early in 1978. These include a very significant charge based on traffic volume (£0.95/kilo packet). Once charging is introduced, some way will have to be found to remove unnecessary packets. This would involve use of a two or three buffer call instead of the present single buffer call, and possibly will require extra buffering at each end. The lower than maximum throughput is not such a serious problem for workstations interfaced at 2.4 or 4.8K bps because the RL 360 interfaces at 48K bps. It is more marked on our connection, but we would expect traffic to other hosts to use up the line capacity and would not place great emphasis on increasing our Packet Hasp throughput. (A throughput increase may come as a by-product of a two buffer call).

Another aspect of EPSS charging is that some level of empty buffer return must always accompany a one-directional traffic flow*. In fact this is a trade-off between user buffer space and network charges for buffer management traffic. The one buffer case and its potential massive overhead has been mentioned already. A two buffer call can result in a 100% overhead and a three buffer call in a 50% overhead because of buffer management traffic. A two way data flow of course reduces the overhead. These aspects will be examined during the coming year.

6.5 Future Support of Asynchronous Terminals and File Transfer

It is planned to support the VPT protocol (PO,1976) which is a simple terminal protocol for both incoming and outgoing calls. Since this is a "real terminal" protocol it requires extra adaptation to and from the SWITCH virtual terminal. The future problems which we are likely to encounter with this protocol are that we will have to provide different adaptations for different

---

*Since this was written the P.O has announced that no charge will be made for buffer management traffic.

terminals and that we may have problems with mapping other hosts adaptations back to a virtual terminal.

However we anticipate no problems in making initial connections between EPSS character terminals and ARPANET hosts, and ARPANET terminals and EPSS hosts. We also intend to support any virtual terminal protocols which gain wide acceptance within EPSS or which seem to have merit from an experimental poin of view. The EURONET virtual terminal protocols (Higginson,1976D) fall into this latter category.

The SWITCH-FTP primitives of Section 3.3 are a subset of the standard EPSS FTP primitives; hence we anticipate no theoretical problems in building an interfacing module between the standard process interface of the call-control software and the SWITCH-FTP. Planning of this module is currently taking place and we expect it to be operational during the second quarter of 1977.

For several reasons we have ordered a second line, at 2.4K bps, to EPSS. This second line will allow us to test our gateway software completely. It will allow also experimentation with the software needed through concatenated networks, which was discussed in Chapter 4. We will experiment also with our facsimile terminal attached directly to EPSS via this second line (see Chapter 10).

When the high level protocols have been implemented on several other UK hosts, and we have mapped the terminal and FTP protocols via SWITCH on to TELNET and the ARPANET FTP, the number of UK hosts accessible to ARPANET will increase considerably. It is quite possible at some time that the PO and we may agree to restrict host access to computers which are attached to EPSS, and not continue direct links like the present Culham, RSRE and ULCC ones. However, before this can be contemplated, the availability of EPSS must increase very substantially - probably to at least 18 hours/day from 9am to 3am. Even without this high availability we expect to have a considerable portion of the UK traffic arriving via EPSS during the second half of 1977. We cannot yet guage the throughput of the UCL PDP9s running the multi-machine system under SWITCH, so we cannot yet predict on which UCL machine EPSS gateway software will usually be run.

CHAPTER 7: THE PACKET SATELLITE PROJECT

7.1 Objectives of Broadcast Satellite Techniques for Packet Switching.

Early work on broadcast radio techniques in the ALOHA project has been widely published,and is early planning for and simulation of a broadcast satellite channel for packet switching (Hinchley, 1976D).

The fundamental objective is to optimise bandwidth by having all stations transmit, and receive on the same frequency. The degree of sophistication in allocation of the channel between users will depend on the application. The simplest scheme allows random transmission into the channel, with arrangements for retransmission when two stations broadcast at the same moment; more complex schemes make use of the broadcast nature of the medium to allow all stations to monitor closely the allocation of the channel, and impose pre-reservation constraints on virtually all use of the channel (Lin-nan Lee, 1976).

The broadcast nature of the channel may be used to genuinely allow multi-destination applications, such as speech conferencing to be supported. The main use of the broadcast nature of the channel is still, however, to optimise communications bandwidth,in a more flexible way than is possible with current techniques. Its main application is where many source-destination paths may be required concurrently, and where the short term fluctuation of traffic density on any one path is variable.

The packet satellite project uses INTELSAT IV as a communications medium, and has used ARPANET as its constituent packet switched network. Neither of these is fundamental to the project's objectives. In fact, a separation from ARPANET is scheduled during 1977 into a separate SATNET. This decision was taken because of the widely differing requirements of the protocols of a broadcast satellite channel compared with terrestrial networks. This separation raises interesting questions on network interconnection in which UCL expects to be active in 1977 (See also Chapter 4). These issues are not addressed in detail in this chapter as they are a future activity, and this chapter is confined specifically to the design and measurement of SATNET. INTELSAT IV also is not a necessary vehicle for the project - it would have been possible to use other Satellites. It is indeed possible that other satellites including the European Orbiting Test Satellite, may be used in the future. However, the present project is concerned only with the use of INTELSAT IV, and with ARPANET as an access network.

The detailed objectives of the project are discussed elsewhere (Linkabit 1976). Because the project involves many parties - ARPA, BBN, COMSAT, DCA, LINKABIT, the UK Post Office, and UCLA, it is clearly not appropriate to discuss these here. For the purpose of this report, it is enough to describe the actual configuration (Section 7.2) and the general progress during 1976 (Section 7.3.).

The principal UCL activities in the Packet Satellite Project have been to develop tools for traffic generation and data acquisition for the configuration of Section 7.2. We have concentrated on high level traffic generation measurement outside SATNET itself - though measurements data will also be provided as the traffic traverses SATNET. While we have been able to develop the tools, in the absence of Gateway computers and other sites, no real UCL measurements will be made before April 1977. The general description of these tools is the main subject of this chapter, and they are discussed in Sections 7.4 to 7.6. Our progress during 1976 is summarised in Section 7.7; how the tools will be utilised is the subject of Section 7.8.

Because the UCL measurement tools are directed at high level measurements, they are almost independent of the network being measured. Clearly the packet formats and the exact timestamp mechanisms are governed by the data network being investigated. Also the exact parameters which must be set are also dependent on the data network facilities. For example there is no point in setting priority, multi-destination or multipacket traffic in networks which cannot cater for them. The application of these tools to SATNET will involve provision by SATNET switches of additional information on their behaviour. For other networks, similar information would have to be provided by the packet switches in the network itself.

7.2. The PSP Configuration

In the interests of brevity, it is assumed here that the reader is familiar with the basic ARPANET technology, and the functioning of the Satellite IMPs (Butterfield, 1974). The present PSP configuartion is shown in Fig. 7.1. In this configuration the UCL Gateway PDP11 has no real function- the Satellite IMPs (SIMPs) in the earth stations are still connected to the IMPs or TIPs of ARPANET in the normal IMP-IMP way. It is only between each other that they have specific ways of communication via the single satellite channel. The UCL Gateway Computer has two addresses in the ARPANET routing tables; it is host 234, the fourth local host on the UCL TIP (counting the terminal handler of the TIP as the third) and host 124, the first Very Distant host on the Goonhilly SIMP (which ARPANET treats as if it were an IMP for routing purposes). The path between Goonhilly and ETAM has been made to appear artificially long, so that normal US - London ARPANET traffic passes through Norway and not the ETAM - Goonhilly path. This avoids the disturbance of measurement experiments by normal traffic, but permits this satellite channel to be used for normal traffic if the circuits through Norway are down.

In Fig. 7.1. two lines are shown between UCL and the Goonhilly SIMP. The 50 kbps one terminates on the PDP11, the 9.6 Kbps one on the TIP. The second line has been put in only temporarily. It is to permit diagnostics and control of the Goonhilly SIMP to continue via Norway from the Network Control Centre even if there is difficulty with the satellite channel within some contention strategy, or with some hardware modifications.

The configuration will change in several important aspects during the second quarter of 1977. The new configuration is shown in Fig. 7.2. There will be two additional SIMPs - at the Nordic earth station in Tanum, and the COMSAT earth station at the COMSAT laboratories. The first of these is another standard INTELSAT earth station, the second has a smaller dish - but still with approval to access the INTELSAT IV satellite. There will again be an additional line to control the Tanum SIMP; the COMSAT one may be controlled via the switched telephone network if necessary. The access line from ETAM is also being shifted to terminate at BBN, in Cambridge, Massachusetts, on a PDP11 Gateway computer. A similar configuration will exist in Kjeller and COMSAT. Now there will be a real Packet Satellite Network, SATNET, attached to ARPANET for data transmission purposes only through the PDP11s at London, Cambridge and Kjeller. The COMSAT PDP11 will be attached also to their own IBM 360 for measurement, data acquisition and analysis.

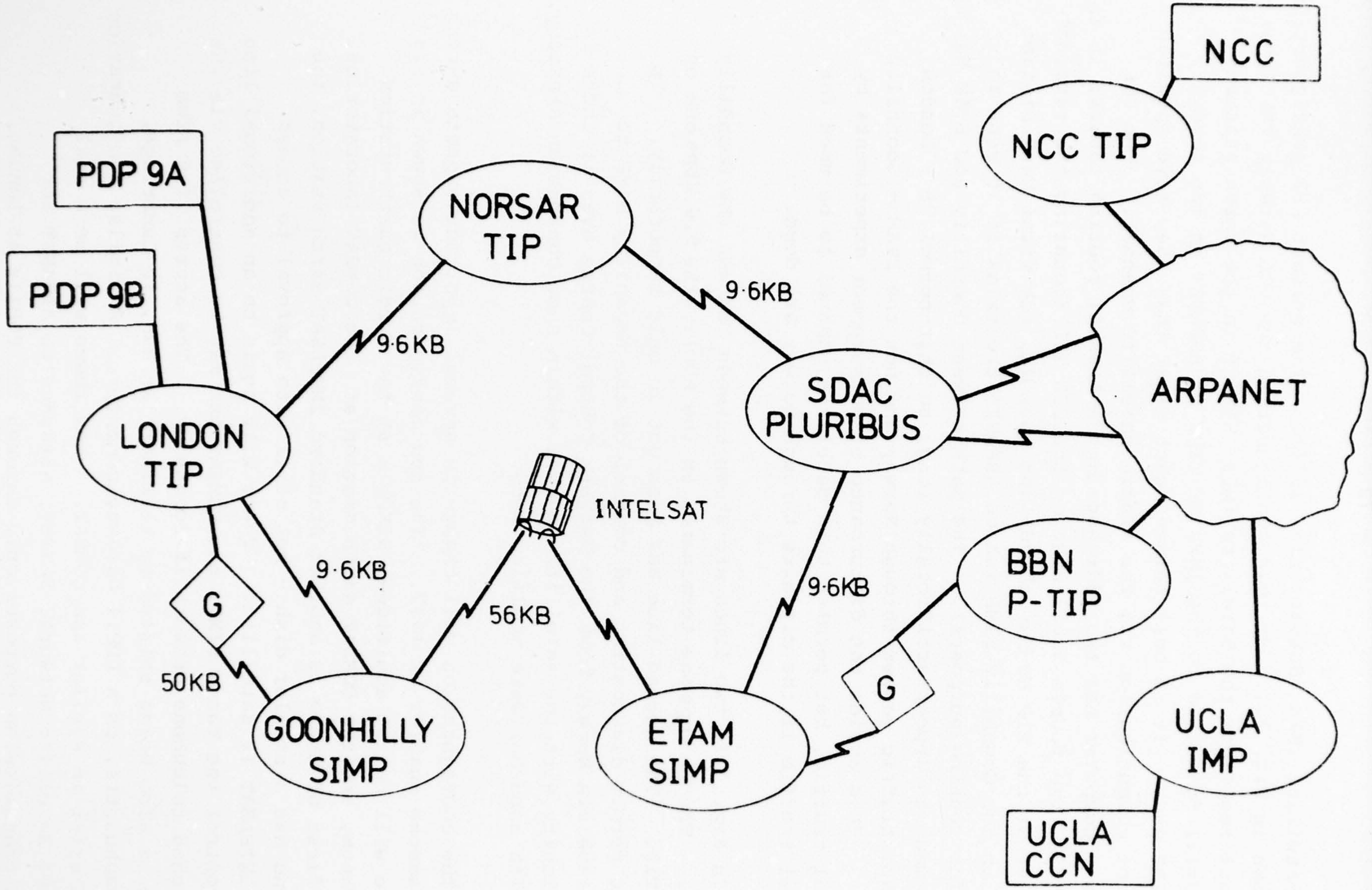


Fig 7.1 Current PSP Configuration

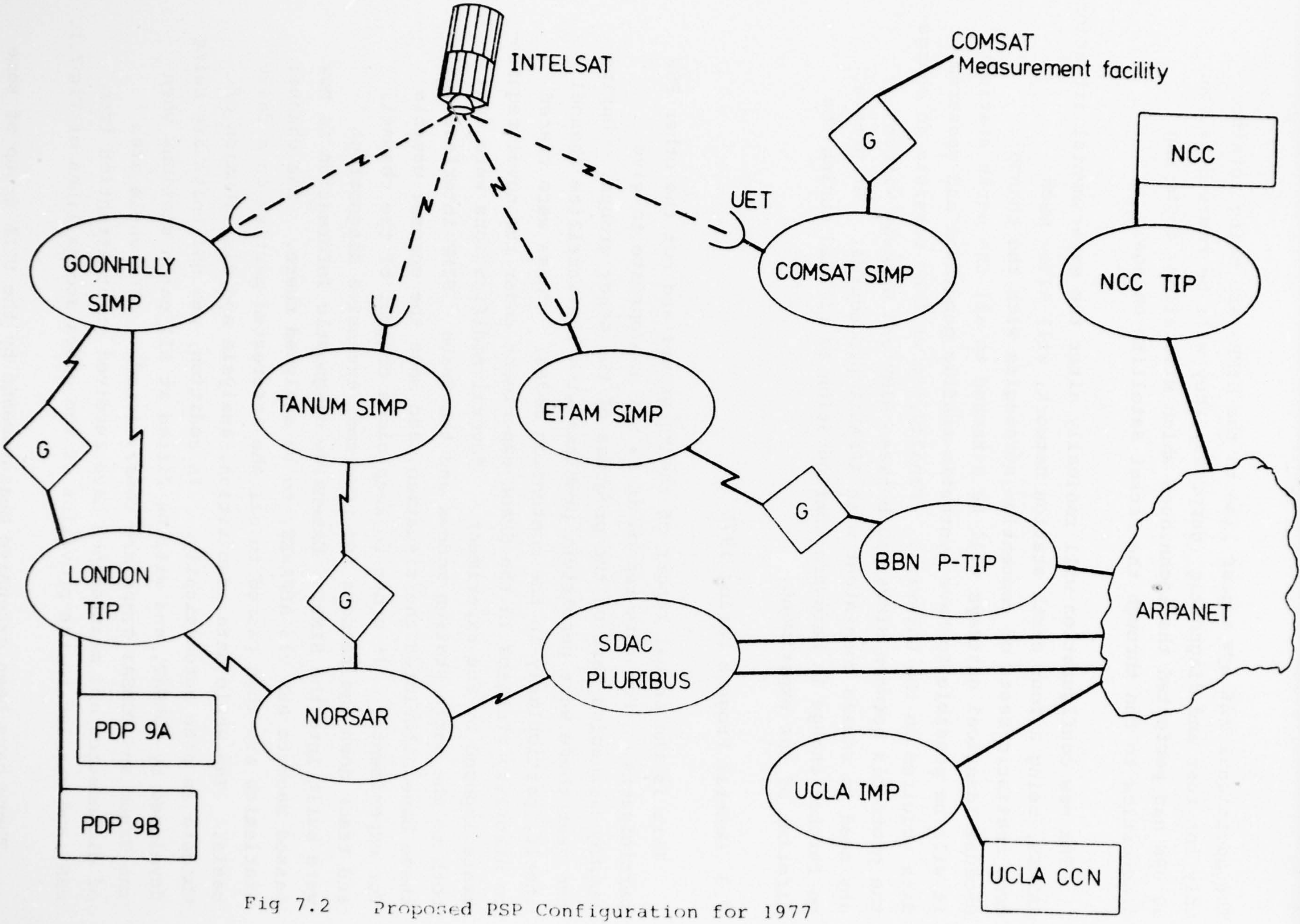


Fig 7.2 Proposed PSP Configuration for 1977

The additional 9.6K bps access line to the SIMPs are being retained only for test and diagnostic purposes; they will be removed as soon as BBN has perfected the techniques which will allow tests and diagnostics to run through the Packet Satellite channel.

The new configuration will radically alter the experimental situation. SATNET, being a four earth station network, will allow much more realistic tests of contention strategies with the channel. Because identical gateways will be attached to all the earth stations, it will be possible to have synthetic traffic generated and measurement data acquired in the Gateways. Finally, it will be possible to change the protocols between SIMPs and between SIMP and Gateway without any need to remain consistent with ARPANET procedures. At present no further change in hardware configuration is planned during the lifetime of the experiment.

7.3 General Progess During 1976

This is the annual report of the UCL group, and not the other PSP participants. For this reason it is not appropriate to give lengthy discussion here of the progress of the other groups. During the year there were persistent problems with the satellite channel itself, particularly in the contention modes. These were traced to stability problems in the SPADE experiment under the novel requirements imposed by the experiment. Several modifications were made both to the earth station modems and the Spade - SIMP interface (SSI); these have alleviated the situation, and made the channel useable for experiments. In order to keep close control of the channel, and track down the reasons for problems, extensive diagnostics were built into the SIMP. Extensive diagnostic information is now passed back to BBN via ARPANET, to be analysed there. The channel statistics are then passed to all the interested parties on a daily basis. The whole data acquisition, analysis and distribution of the data is done automatically. In addition, new SSI units are being developed by COMSAT, and will be fitted at all earth stations when the Tanum and COMSAT SIMPS are commissioned. The whole area of diagnostics and maintenance have received great attention from BBN, and has led to the provision of the extra access lines of Fig 7.1.

There have been extensive measurements by the UCLA group of some of the simpler strategies for using the satellite channel. These have been described in reports from Kleinrock's group. Extensive design and simulations of new contention strategies for using the satellite channel have been carried out at BBN, LINKABIT and UCLA. These strategies aim at using the channel efficiently, and with good recovery from errors, overloading, and with different traffic mixes. They also allow, for the first time, the attachment of priority and maximum permissable delay on the data packets. Both of these will be required in the commercial or military environment. Implementation of these strategies in the SIMP has been started by BBN, with the aims of deployment during 1977.

Details of the above are contained in a series of internal working notes, the Packet Satellite Project Working Group notes. These may be available to interested parties from the authors of the notes themselves.

One gateway PDP11 has been installed at UCL as an ARPANET host on the UCL IMP and also on the Goonhilly SIMP. Gateway code has been provided by BBN so that incoming packets with a particular Internet header (as used in Chapter 5) can be timestamped and passed to a program being developed at UCL. There are also facilities for accepting and forwarding packets generated by the UCL program, to either the IMP or SIMP attached. The functions of these UCL programs will be described in Section 7.4. A similar PDP11 has been attached as a host to ARPANET at BBN, and can be used for testing the UCL code.

7.4 The UCL Measurement Tools - An Overview

A high level traffic generator/measurement tool, (i.e. one from outside the network itself) requires that the network under measurement be seen and interpreted from a host level point of view. This has two immediate consequences. First, the experimental traffic must be a realistic model for traffic generated at host level; second,host level measurements must be oriented towards a user measurement of network performance. While it may be possible to extrapolate from lower level figures for variables such as throughput, some variables such as network response to user demand can only be measured at host level. Neither of these requirements depend significantly on the features of a particular network, and accordingly much of the following discussion is couched in network independent terms; in any particular implementation, the network interfaces need clear specification. On its own, such a tool provides a useful means of assessing the overall performance of a network, but would not tell us much about the importance of any single feature. This type of information can be derived by feeding 'realistic' multi-user traffic through lower level measurement tools in an integrated measurement scheme.

Bearing these considerations in mind, there are three important aspects of an integrated high level tool, viz: the generation and acquisition of the traffic to be measured; the control and variation of traffic generation in the experimental environment; and the actual measurements to be made. These are discussed in Sections 7.5 and 7.6.

Three main traffic patterns are recognised in this experiment; each of these traffic patterns may be varied so that a wide spectrum of data types is actually covered. The three main patterns are:

(1) High data rate bulk transfers
(2) Low data rate steadystream transfers
(3) Interactive traffic

In the SATNET protocols to be implemented in 1977, categories 2 and 3 are treated as similar traffic types, for purposes of satellite channel reservation. Priority and delay can be specified however, for all traffic.

An example of a high data rate transfer might be that of a file across the network. The message generator program would attempt to push messages as fast as possible into the network. The generator may try to transfer a prescribed number of bytes through the net and measure the total elapsed time, or may just transfer as much data as possible in a given amount of time. In either case the system will measure the maximum throughput that the user could expect. Combining this traffic with other controller traffic on the net (see below) will indicate how the performance deteriorates with interfering traffic.

The low data rate steady stream transfer covers such data types as voice traffic; where the total amount of data may be small per unit of time compared to a file transfer, but the messages enter at very precise intervals. Once voice messages enter the network, they must be delivered not only at a constant rate but also within a specified time period. If for instance, a message enters the net every half second, it must be delivered every half second at the other site; 'bunching' of data is not permitted here (unlike bulk transfer). Much real time data such as voice, facsimile, telemetry, or control data falls under this category. The measurements will consist of examining how consistently the messages are delivered to the destination, and the number of messages that arrive at the destination within a given time of entering the system. Here too, the effects if other traffic in the net will be measured,

Interactive traffic is the most difficult to simulate and the most difficult to measure. Typically, the user sends short messages of variable size at variable intervals, and receives replies consisting of one or more lines a short time later. The important aspect of this type of traffic is whether the messages can be delivered within some threshold time. The traffic generated by the program will be based on statistics gathered on usage of the ARPANET, the type of messages sent and the type of replies received.

The ability to handle all these traffic types is the major design requirement of the traffic generator. This requires greater flexibility than can be provided easily by a SIMP level traffic generator.

7.5 Traffic Generation and Data Acquisition

7.5.1 The Environment

This is a description of the program that simulates users' interaction with the network. Each gateway taking part in these experiments has an identical copy of this module, which consists of three main segments; the control segment, the data generation segment and the data reduction segment.

The control segment accepts commands from an external host, or another gateway, and controls the other segments of the program. The generator simulates the user's interaction. In the following discussion, the word 'user' will denote a simulated user. Messages are generated independently for each user and the messages are transmitted after passing through a simple flow control mechanism.

The data reduction segment will accept messages from any gateway taking part in the experiment. Pertinent statistics, selected by the ecperimenter, are extracted from the received user messages and gathered in histograms. At present intervals, the histograms are sent back to a data collection site.

7.5.2 The Traffic Generator

The generator is controlled by a User Traffic Table which contains the message generation details for each user being simulated. Each entry in the table describes a separate user and at present the table has 32 entries allowing up to 32 users to be simulated from each gateway at a time. The user traffic table is loaded by the control module. The table can be loaded or modified only between experiments, when the message generator is not using the table. Each entry in the table contains all the information required to simulate a user, although the number of entries and their actual value can be given. For instance, the experimenter can specify that for a particular user messages are to be transmitted every 10 seconds ± 2 seconds. The generator uses a random number generator to select the actual value within the spread constraints. The most important parameters that can be set are shown in the following table:

User Number*	A number from 1 to 32
Mean Message Size*	Mean Size of generated messages. In bytes.
Message Size Spread*	The Variation allowed in message sizes
Mean Inter Message Delay*	The mean delay between sending each message.
Inter Message Delay Spread*	The variation allowed in the delay between messages.
Next Message Number	A sequence number inserted in the messages.
Next Message Length	Length of the next message to be sent
Scheduled Departure Time	The time at which the next message will be sent.
Traffic Type*	Interactive, steady stream or file transfer
Destination*	The gateway these messages will be sent to.

Table 7.1 The Important Parameters of the User Table

All the entries marked with an '*' are the entries that must be loaded by the control program.

Whenever a user's message is transmitted (and at the beginning of the experiment) the data generator goes through a calculation to determine at what time the next message for that user should be sent, and what length that message should be. As soon as the Scheduled Departure Time for a message is equal to the current clock time, the generator attempts to transmit the message. However, this attempt may fail; if the generator determines that the gateway is unable to accept the message, the user entry is put in a frozen state until the gateway is ready.

7.5.3 Expanding the Message

Up to this point the message has consisted of only a few entries in the User Traffic Table. The message generator actually manipulates pointers to the table rather than moving potentially massive messages around. The message is expanded to its full length just before transmission. Each message always contains the following information:

Source Gateway Number
User Number
Message Sequence Number
Message Length
Scheduled Departure Time
Actual Departure Time *
Traffic Type
Destination

The entry marked * is inserted at the instant of the message departure. After these fields have been inserted the message is expanded to its full length with dummy characters.

Special consideration must be given to the steady stream traffic type. For these messages it is important that they are sent within a threshold time. If the delay prior to transmission exceeds some threshold value the message must be deleted: there would be no point in transmitting it as it is already out of date. In this case the message is deleted before transmission and the generator proceeds to calculate the next transmit time for that user. For stream traffic, the delay to the next message is based on the scheduled departure time of the previous message.

Using the Scheduled Departure Time of the previous message as the base time (whether the previous message was cancelled or not) means that messages are generated at regular intervals regardless of the acceptance rate of the network. With other traffic types, where the current clock time is used as the base time, the input stream can 'hiccup' when the network is unable to take any more traffic. This is consistent with actual network use. If the network refuses to accept a message because the destination host is not rapid enough, or if the IMP cannot accept more traffic, the user gets any entries rejected. However, this simply causes the user to wait and try again, not to discard the messages he would have sent. This is also true of file transfers which always try to transmit as fast as the network will accept.

7.5.4 Data Acquisiton and Reduction

In this segment, messages are accepted from other gateways in the experiment and the data is reduced before being sent on to the collecting host. The data reduction segment never denies entry to traffic from the network. It will accept messages from any gateway at any time and in any sequence.

As user messages pass through the satellite network to the destination gateway they are timestamped at up to eight different places. By inspecting these timestamps at the destination, the time spent by the message in various parts of the network can be determined These times can then be entered in histograms. To permit the greatest flexibility, a general histogram method is employed. The experimenter can inform any gateway (via its control module) which users, from which source gateways, are to be histogrammed. The experimenter can also select the timestamps in which he is interested and which of the modes are desired. In one, the 'successive' mode, the difference in a particular timestamp between successive messages is used, in the other the 'difference' mode, the difference between two timestamps in each message is used.

Each histogram can also be given a 'lifetime'. A lifetime of one minute, for instance, would mean that every minute the histogram must be sent back to the collecting site and then re-initialised to zeroes for subsequent statistics collection. Thus a series of histograms for a user can give a 'motion picture' view of the users performance.

7.5.5. Building-in Response

In many circumstances it would be more realistic if the Data Acquisition and Reduction segment responded to some of the user interactive messages sent it. In this way the segment would behave more like an actual remote machine. This can be accomplished by the following method: As well as the information currently in the User Traffic Table, values are kept for the generation of expected responses. These values are also loaded by the control program. When the messages are sent from the Data Generator, information is included in the messages to dictate the response required. Whenever the Data Acquisition and Reduction segment receives a message it inspects this response field. If a response is required the segment creates a message in the normal fashion and transmits it. It then treats the received messages in the normal way. The message that is sent as a reply will itself get reduced at the remote gateway. Such messages contain a special flag to indicate that they are response messages.

Note that the reply is controlled by the Data Generator. This is consistent with making the Data Acquisition Reduction segment as passive as possible, so that it acts in a slave mode. The response information would have to include the delay time before the response message length. Only one message is ever sent as a response.

7.6 Control of the Traffic Generator and the Experiment

7.6.1 The Function and Structure

The traffic generator is designed to be controlled from any host in ARPANET. Accordingly, any host may communicate with the Traffic Generator through a well defined interface, either through a resident controller or through one resident elsewhere in the net which accepts external control. In order to maintain the integrity of this interface, and to maintain a clear separation of the control functions from the measurement functions they might otherwise affect, it is better that there be no controller resident in a gateway.

The control pathways are shown in Fig. 7.3. Overall direction is invested in an experimental host, sitting outside SATNET, in which a controlling process (the 'controller') resides. This process accepts commands interactively or from a previously created command file, validates them, and passes them to the net. In the reverse direction, it accepts messages from the net and acts on them where necessary.

Communication with the net is via one of the participating gateways (the 'service gateway') which for control purposes acts as a relay. All control messages will be sent to the gateways involved. The basic control functions at gateway level are:

- to initiate an experiment
- to start and stop traffic generation
- to change traffic distribution or simulate protocol parameters

7.6.2 Controller-specific Features

The traffic generator is controlled by a set of simple commands which will be common to all controllers. The controller can be made as sophisticated as the experimenter requires, and the following discussion describes some features of the UCL implementation.

The controller will accept commands interactively. However, it will also execute command files which have been previously prepared for it and are resident in the same host. The detailed syntax of the various commands will await implementation, but it should be possible to nest files, and the following local file commmands at least should be available:

READ<filename>	
WAIT<interval>	(before executing next command)
RETURN	(from this file level)
LOOP<counter>	(repeat n times)

7.6.3 Gateway-level Functions

The broad categories listed above are all functions requiring

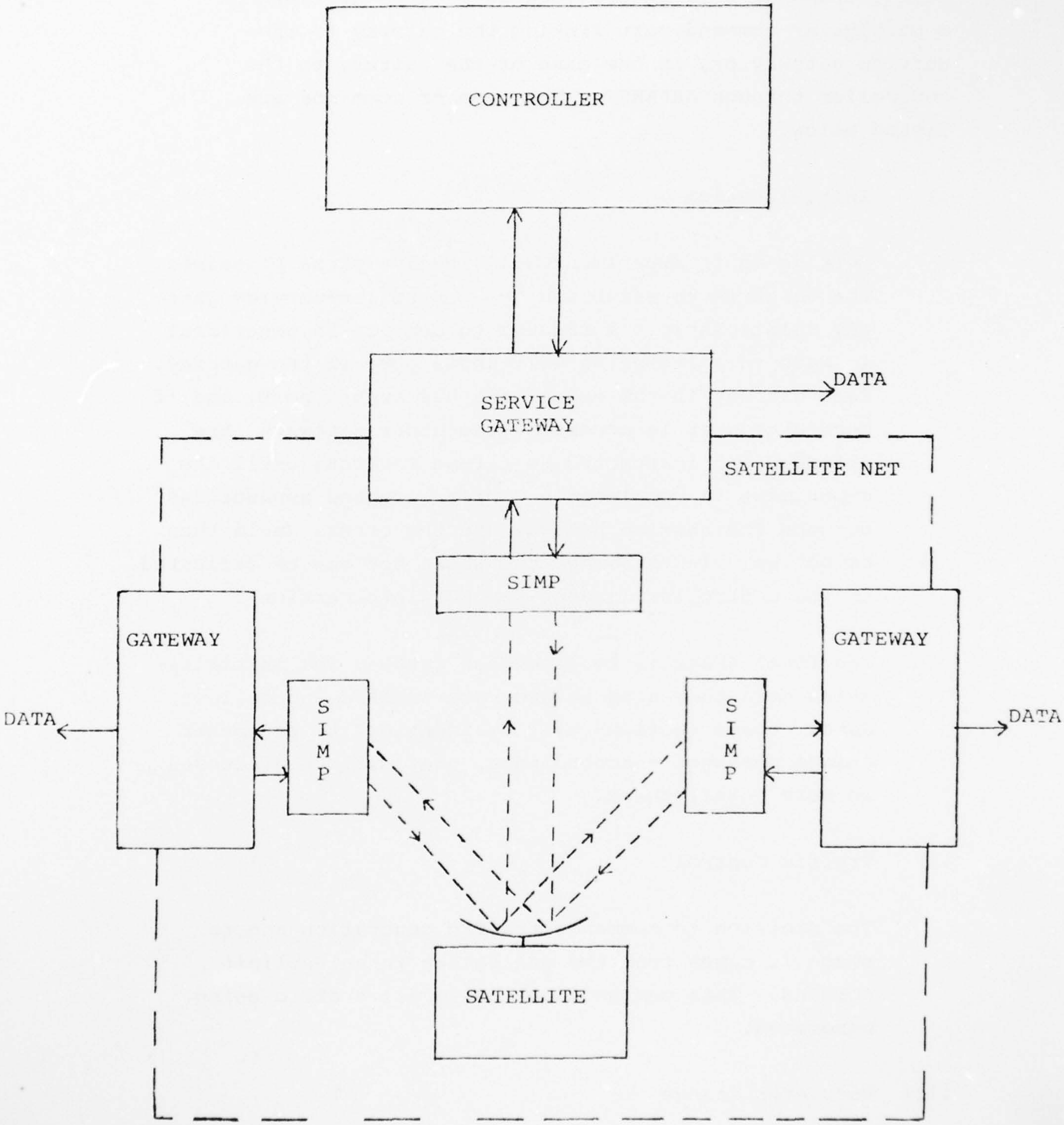
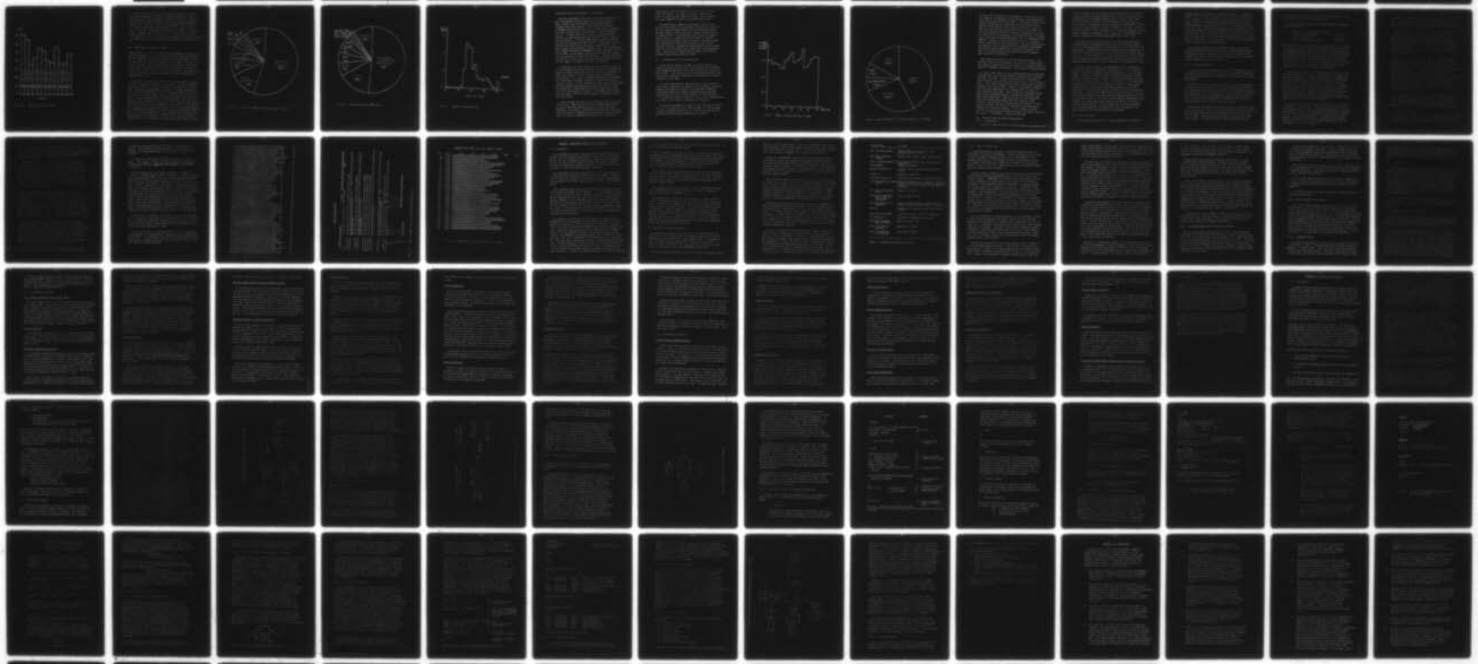


FIGURE 7.3 CONTROL PATHS IN THE SATELLITE NET

AD-A044 419 UNIVERSITY COLL LONDON (ENGLAND) DEPT OF STATISTICS --ETC F/G 17/2
UNIVERSITY COLLEGE LONDON ARPANET PROJECT.(U)
FEB 77 P T KIRSTEIN N00014-74-C-0280
UNCLASSIFIED TR-34 NL

2 OF 2
ADA
044419

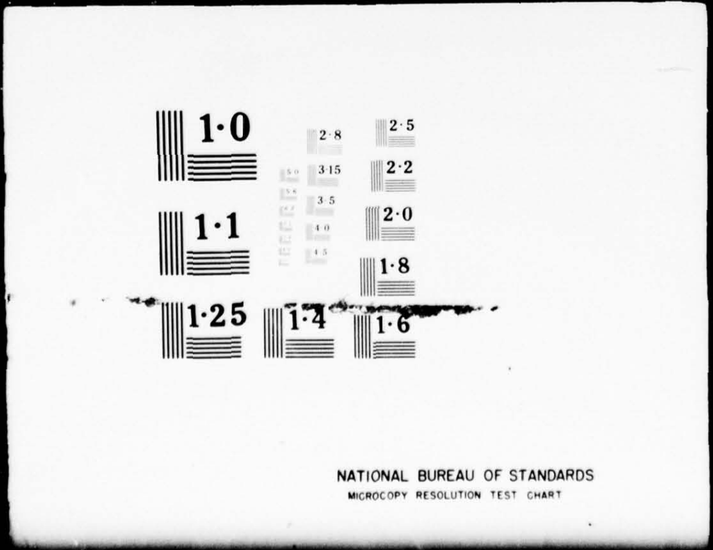


END
DATE
FILMED
10-77
DDC

2 OF 2

ADA

044419



On the other hand most university usage occurs in the late morning or afternoon - because of university lectures. Many of performance by the gateway traffic generator. At this level, commands will be provided as packets arriving on a particular command port linking the gateway to the service gateway or, in the case of the latter, to the controller through ARPANET. The type of commands are listed below.

i) Initialisation

Setting up an experiment will involve three processes. The first is to establish the controller-service gateway relationship. A request to control an experiment is made to a listening well-known port in the gateway. Each gateway in the experiment has such a port, and if such a request is accepted, the other gateways are informed and instructed to refuse requests until the experiment is completed. Similar command connections between the service gateway and the others would then be set up. In this way, the whole net can be dedicated to the controller (except for NCC intervention).

The final stage is to give each gateway the initialisation data they need to generate traffic and collect data. These sections will be identical to parameter change messages - accordingly, they will be discussed in more detail there.

ii) Traffic Control

The decision to commence traffic generation and to cease it comes from the controller in an explicit command. This command applies to all traffic being generated.

iii) Parameter Change

The parameter change mechanism can only be described in general terms. The significance of the particular values will depend entirely on the traffic type being simulated. The Change Packet must contain:

- the simulated connection number
- the number of parameters to be changed
  and for each parameter
- the indentifying code
- the parameter value

The information in a Traffic Description packet can be specified more completely. The packet contains:

- the number of simulated connections, and
  for each connection:
- the connection number
- the source gateway
- the traffic type (e.g. steady stream, file transfer)
- the destination gateway
- the destination role (e.g. sink, echo, simulated response)
- the mean packet length at source
- the variance
- the mean generation interval at source
- the variance
- the corresponding figures for the destination
- the initial packet length
- the initial transmission interval

iv) Messages to control

Messages from the net to the controller will usually be success/failure messages in response to a controller request. Such messages might include:

- traffic being sent (on connection)
- no traffic being sent
- protocol parameter change accepted
- initialisation successful
- access prohibited (experiment already in progress)

7.7 The State of the Measurement Tools

A first version of the Controller and Traffic Generator have both been written and commissioned. Both are still undergoing modifications as our experience with their use increases. Some programs have already been written to collect the data from the experiments although there is still much to be done in this area. Since the gateway software arrived from BBN only during December 1976, our experience with our tools is very limited.

The first task has been to calibrate the performance of our gateway and test the generation and data collection aspects of the system. For this we arranged to have the generator transmit messages to its own gateway. This means that the messages are not actually transmitted into the network; they are turned around at a low level in the gateway, and are handed to the Data Reduction Module. Figures 7.4, 7.5 and 7.6 demonstrate some typical results. In each case over 1000 messages were used to build the histograms shown.

These histograms are important for future experiments, as they demonstrate the performance we could expect from the software at present in our gateway. For instance, the round trip delay of an internally looped interactive user can be seen to be about 48ms. This is higher than we expected, and the reasons for this are being investigated at the moment. BBN expects to modify the software they have supplied to reduce the internal processing time of the gateway.

7.8 The Application of the Measurement Tools Planned for SATNET

It can be seen from Fig 7.3 that the gateways are at the extremities of SATNET. They also may be viewed as SATNET hosts, or even as SATNET terminal multiplexors. They are therefore an excellent place to be doing measurements on all of the possible modes of use of SATNET. Normally, traffic will originate at one point at the edge of the network designated, or a single destination at another. The usual measurement mode will be to generate traffic at one gateway, and collect it at another. As experimentation proceeds, this pattern will be extended to generating multiple streams of traffic bound for different destinations, and eventually to be generating concurrently this traffic from several gateways. The need for one gateway to control all the traffic generators is obviously useful in this more complex picture. However, data collection onto backing store can be done more easily outside the direct SATNET; thus the necessary software and the processor time to deal with measurements can be isolated from the gateway machines, which can be left to perform the basic traffic emulation task. Eventually, however, it may be necessary to provide local data collection facilities, because of the absence of sufficient bandwidth to drain data away from SATNET.

SATNET protocols will allow priority and delay information on traffic to be encoded on each message, and will act on this information in ordering requests for channel transmission. No problem exists in making the gateway traffic generators encode this information according to pre-set parameters on traffic characteristics.

The small station participation and conferencing application will involve measurement issues which are not yet properly resolved and will be relevant only in late 1977.

The SATNET SIMPS provide the ability to collect status information on queues and channel behaviour for every few seconds. On an even more detailed level, data is generated by any SIMP 'exception condition', and must be collected. Assimilating this information into data analysis programs is a problem still to be tackled.

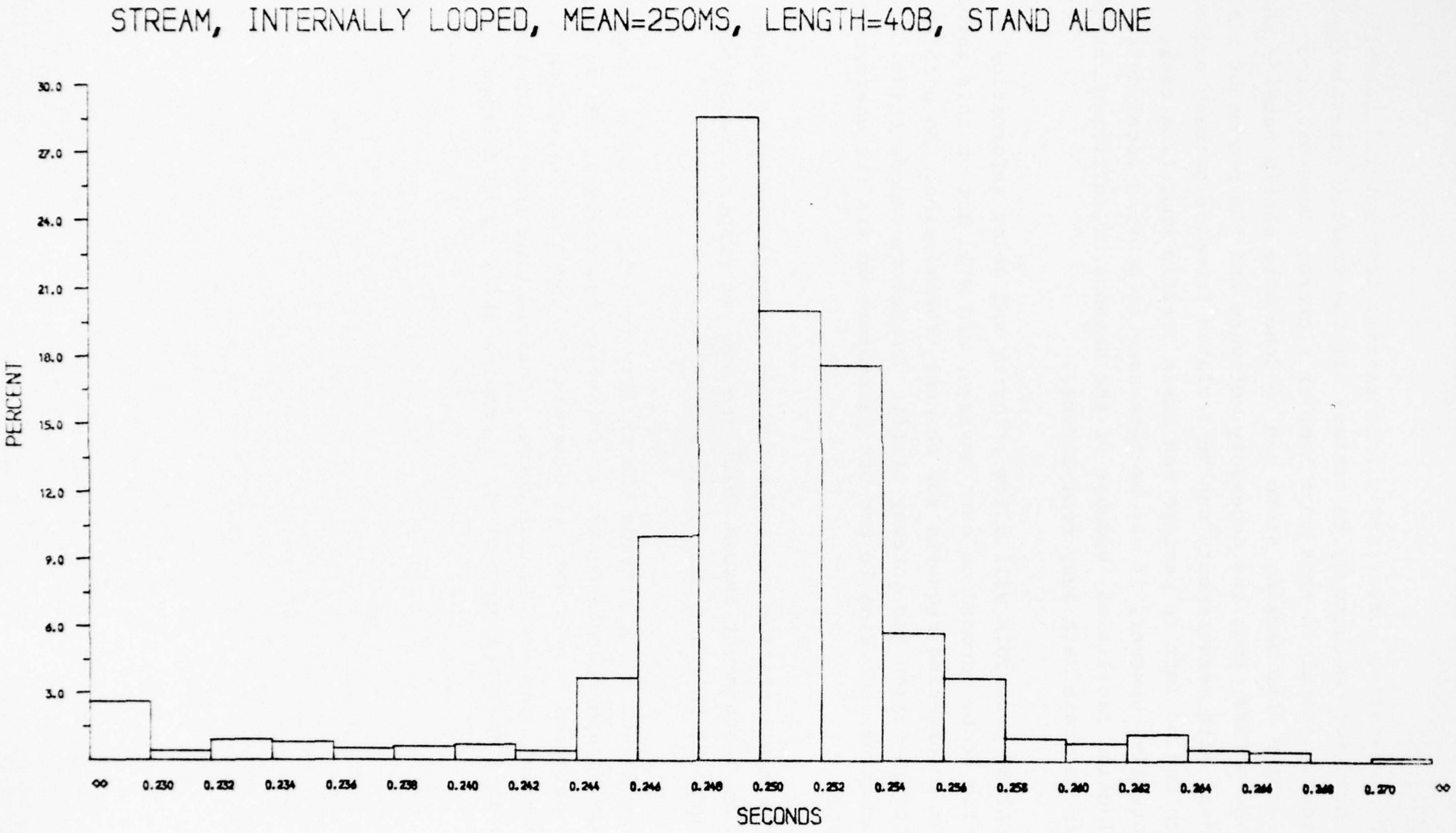


Fig 7.4 Histogram of Message Turnround Times

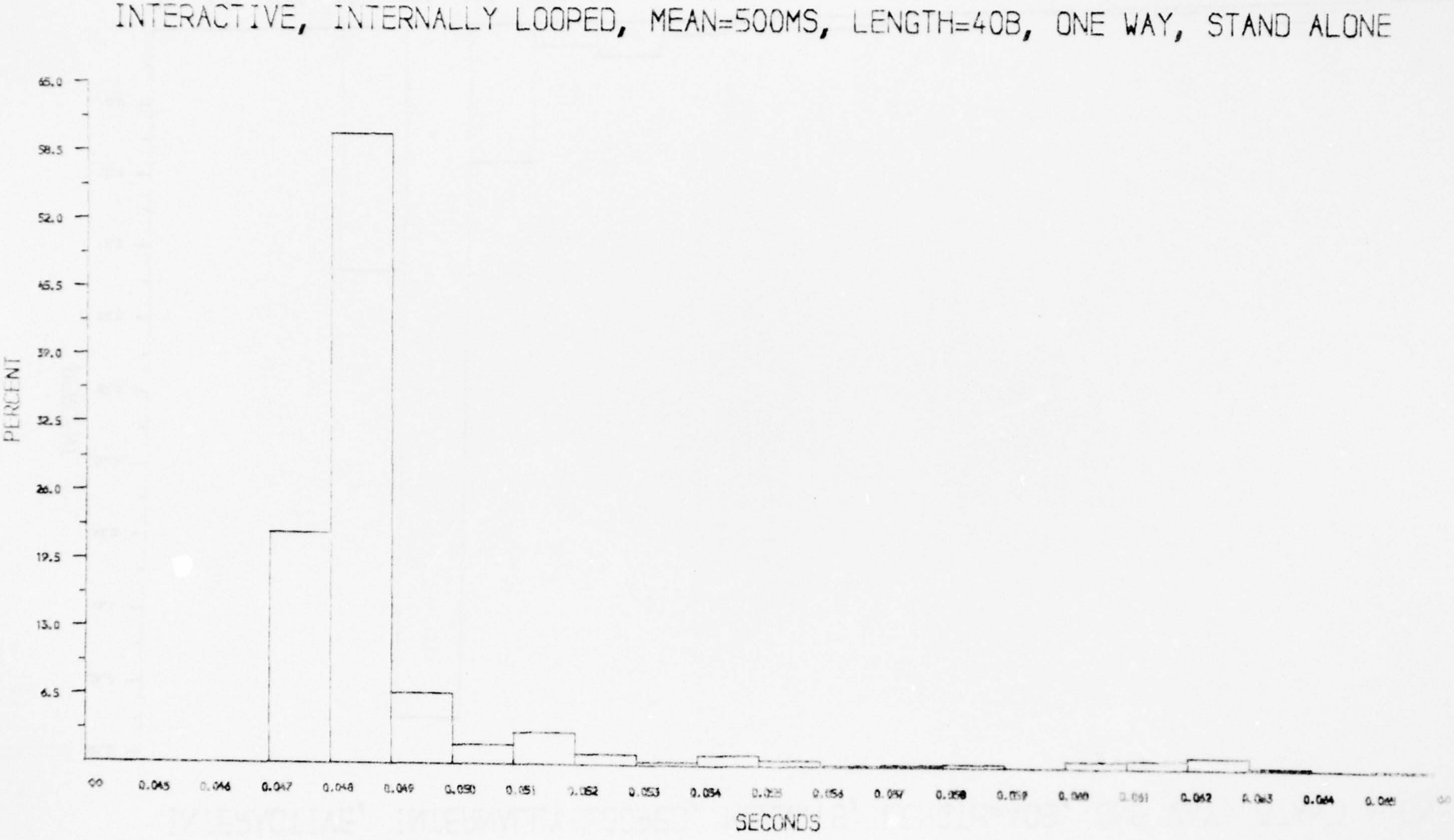


Fig 7.5 Histogram of Message Turnround Times

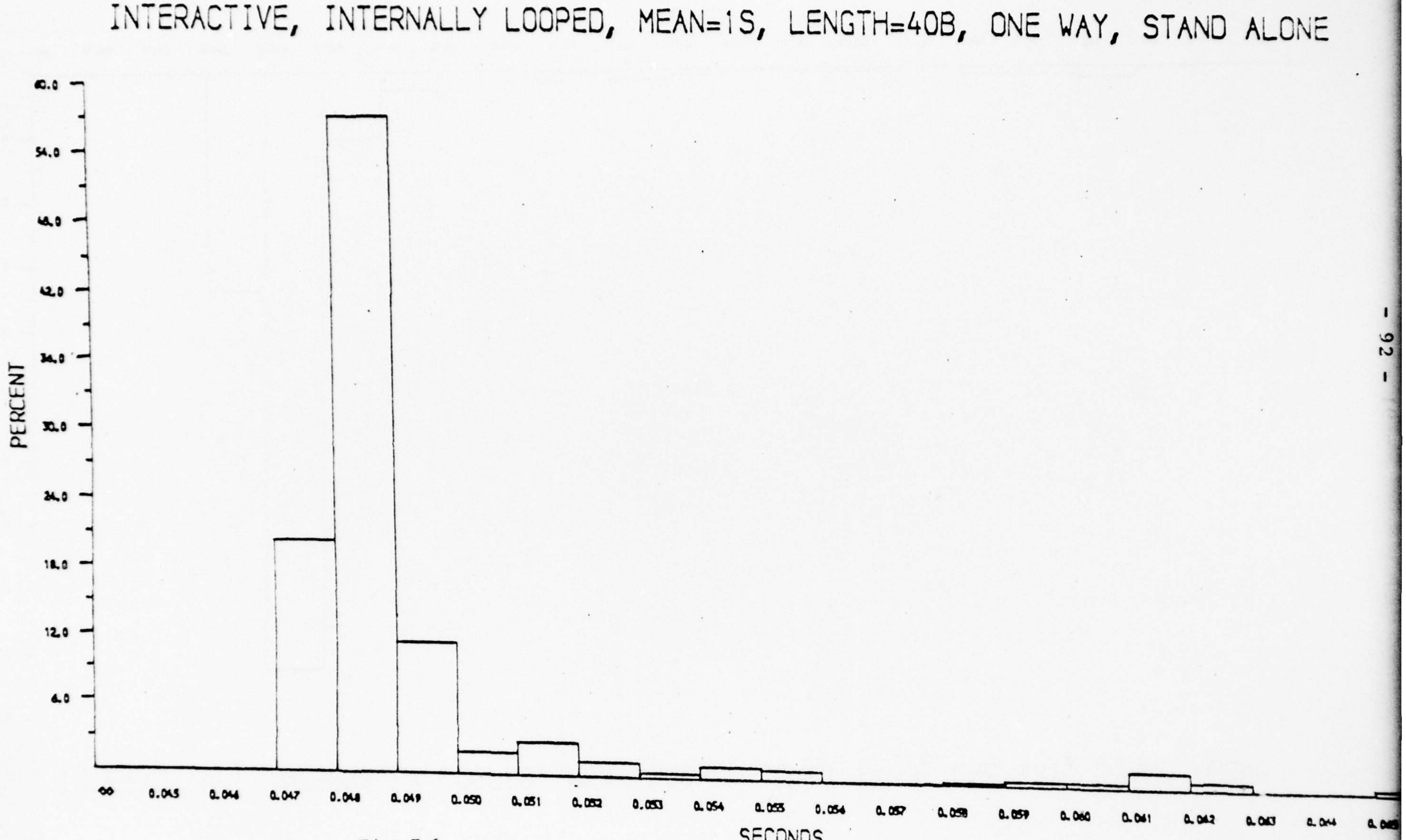


Fig 7.6 Histogram of Message Turnround Times

CHAPTER 8: NETWORK MEASUREMENTS

8.1 Introduction

In the previous annual report (Kirstein,1976A), we discussed a number of network measurement tools which we had developed. These, together with the programs needed for the analysis of the data gathered, have been developed further during the past year. This development work is now at an end and data up to the end of the reporting period is being analysed. This work and the results are described fully elsewhere (Stokes,1977A and Stokes,1976A). In this chapter we give a review of the work and some selected results. Of course, the data acquisition is continuing and further results will be presented in due course; however, it is expected that these will merely complement the significant amount of data already gathered and presented here.

The data-gathering tools we have developed give us information on four types of usage. First, we are able to monitor the status of the dial-up ports on the London-TIP and, at the same time, provide an access control mechanism. This program (QUES) was described in some detail in (Kirstein,1976A). It requires a dedicated computer (PDP9B) and, due to other pressures on this machine's time, we have not been able to run it for a sufficient proportion of the time to get accurate statistics on use via the PSTN (especially as our sample is strongly biased by PDP9B being available for this purpose almost exclusively in the mornings and at weekends, if at all; we have very little data for afternoons). Due to pressure on machine time, no such monitoring has been carried out during the second half of 1976. The results we have obtained are discussed in Section 8.2.

The second usage we are able to monitor is that through PDP9A. All usage of the RL 360/195 and the RSRE GEC 4080 is via this machine; since the PDP9A system monitors all interactions through it, we obtain very full data on such usage. The data is in a similar (although significantly different) format to that of PDP9B. We record, for US users accessing British machines, the machine that is accessed, the machine from which the access was made, the ports used, the identity of the users and the account used: we are able to record similar information for users of the RL machine accessing ARPANET.

Much of the usage of RL through the TIP is from UK users, mainly local users with hardwired terminals or a few remote users with leased lines. In general, it is difficult to obtain a consistent picture of leased line usage. This information is valuable, for example, to PTTs who wish to know whether the characteristics of leased line usage differ significantly from those for PSTN usage. Our analyses do indeed show significant differences and the results are given in Section 8.3.

The third measurement tool runs on PDP9B in conjunction with the monitoring and access control program QUES mentioned above. In general, QUES obtains its information from the user and then closes the connection, provided the user is authorised, and allows him to log in as usual. However, it is possible to make the connection for the user, and since now all data passes through the PDP9B, we are able to make a complete transcript of the interaction. Due to the high cpu overheads involved, we have only used this technique with one host, with users of the National Library of Medicine (NLM) information retrieval systems. Analysis of the data thus obtained allows us to determine various parameters of usage of such a system, for example the distribution of system keywords. Although the tools have been developed completely in 1976, the number of runs made have been very few. However, we plan a week long experiment early in 1977 which will provide a significant quantity of data. Our activities in this area are recorded in Section 8.4.

Another measurement tool is totally different from those described above. The previous three were all at the user level; this is at the line level, i.e. measurement of the IMP subnet. While there have been extensive measurements of the subnet by the Network Measurement Center (NMC) at UCLA, these measurements have relied almost entirely on cumulative statistics gathered by the IMPs or messages traced through the subnet. As a result of these measurements, the NMC has identified a number of significant problems in the subnet, particularly logical errors in the IMP-IMP protocol. However, for certain specific purposes (in particular, the TCP of Chapter 5) we have been interested in measuring the IMP-IMP traffic ourselves. We have set up some fairly elaborate tools to make this possible, but they have been used only very rarely. Their usage has enabled us to identify a number of specific problems, both with the TCP itself and with the effect of the point-to-point satellite line on the IMP-IMP protocols. This work is described in Section 8.5.

Finally some measurements have been carried out of the performance of specific Network Control Programs (NCP). These measurements have been performed to give us yardsticks for comparing different implementations. They are discussed in Section 8.6.

8.2 Monitoring of PSTN Usage

The techniques to monitor usage via the PSTN were described in some detail in the previous annual report (Kirstein,1976A) and some simple results were presented. Little change has been made to the data gathering program, QUES, but the analysis programs have been modified and extended. In addition, we now have results for over a year's monitoring and present these here.

For clarity, at the risk of some repetition of the previous report, we now describe the way the final results are obtained. PDP9B, when not being used for other purposes, runs a program called QUES. This program connects itself to a number of TIP ports (currently those available through the PSTN); when a user dials in, this breaks the connection. On noting this, QUES immediately seizes the connection and interrogates the user for his name (not checked), his TIP password and the number of the host he wishes to access. Provided these are acceptable (he is allowed one error on each), QUES closes the connection and allows the user 20 seconds to connect to the required host. It then attempts, at one second intervals, to reconnect to the user and only succeeds when he closes the connection to the remote host. It then asks him the next host number required and so on.

For every interaction, it prints data onto a paper tape log,together with other information (e.g. every half hour, it prints a message stating it is still monitoring). This paper tape is then sent to the RL 360/195 and analysed by a program called XFSTATS (Stokes, 1976B). This is written in Babbage (Stokes,1974) and makes extensive use of the facilities in its system library (Stokes,1977B). The analysis consists of two phases. The first is to reduce the output to a canonical form; this consists of reducing each total interaction to a single line and tidying up in the case of system crashes. Considerable error checking is performed and the output may be assumed to be error free. The second phase is the actual analysis; in this, the data is read into core in a specified structure and this structure is traversed in the requisite fashion. In this manner, we are able to analyse usage of the London TIP via the PSTN in some detail. To be specific the analyses available are:

Times monitored
Surnames used for each Ident
Number of logins, connect time and number of hosts
  used for each Ident
Number of logins, connect time and number of users
  for each host
Matrix of connect times for each host number/Ident pair.

For simplicity, the results are available in graphical or tabular form.

We have a considerable amount of data for the period July 1975 to July 1976 and it is upon this that we concentrate. Before that date, although password validation was effective, we did not prohibit access for illegal users. After July 1976,we have done relatively little such monitoring due to other pressures on the computer.

Over the year in question, we monitored the TIP for about 50% of the time but much of this time was at weekends and in the evenings when usage was extremely low. In addition, the amount of monitoring performed varies significantly from month to month. Figs 8.1 and 8.2 show this variation; the former shows the breakdown by hour of day, the latter by month. The largest usage was by the British Library research groups accessing the National Library of Medicine databases. However, this use is biased because the usage of NLM coincides almost exactly with the times we monitor.

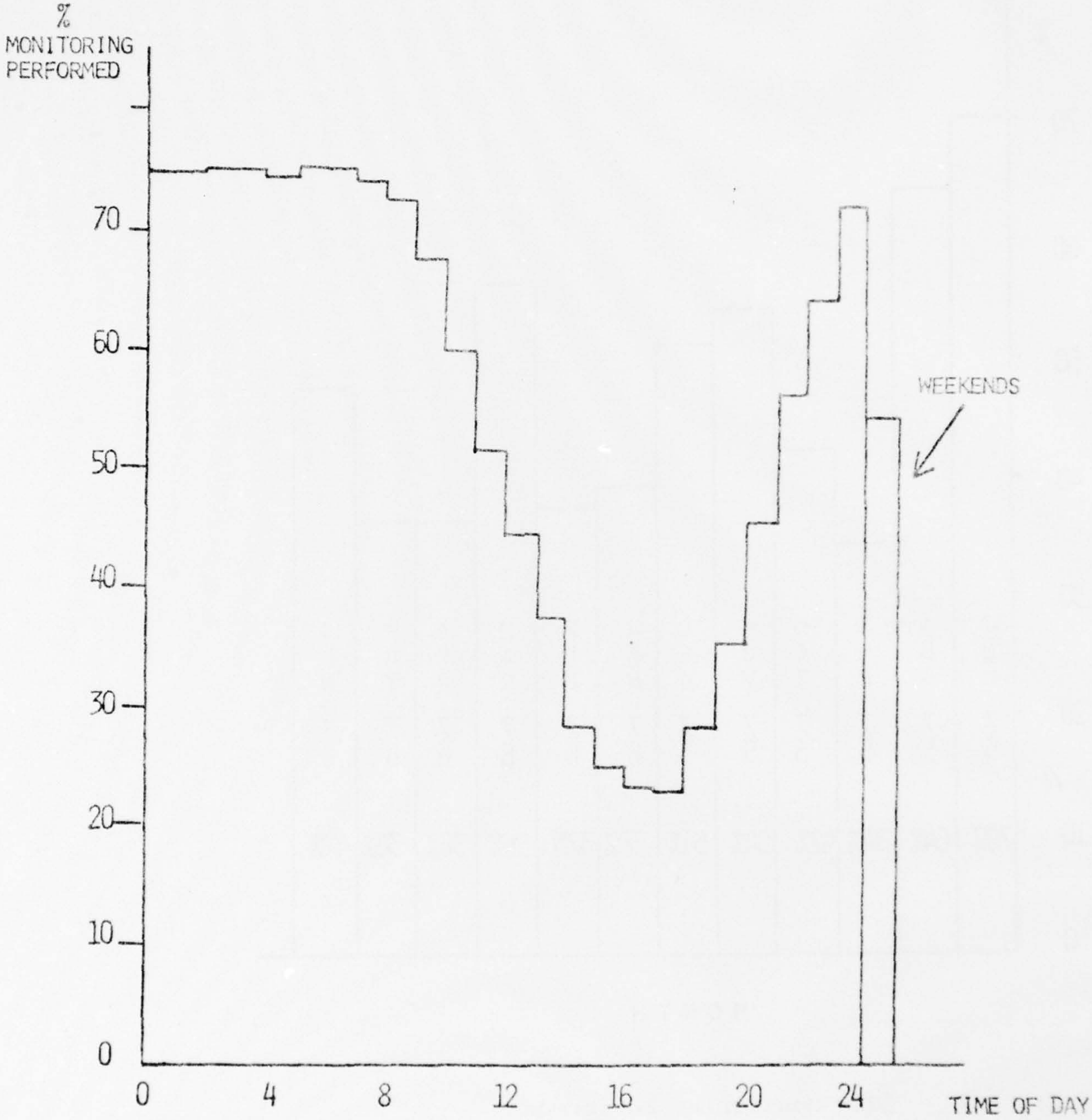


Fig 8.1 Monitoring Performed as a Function of Time of Day

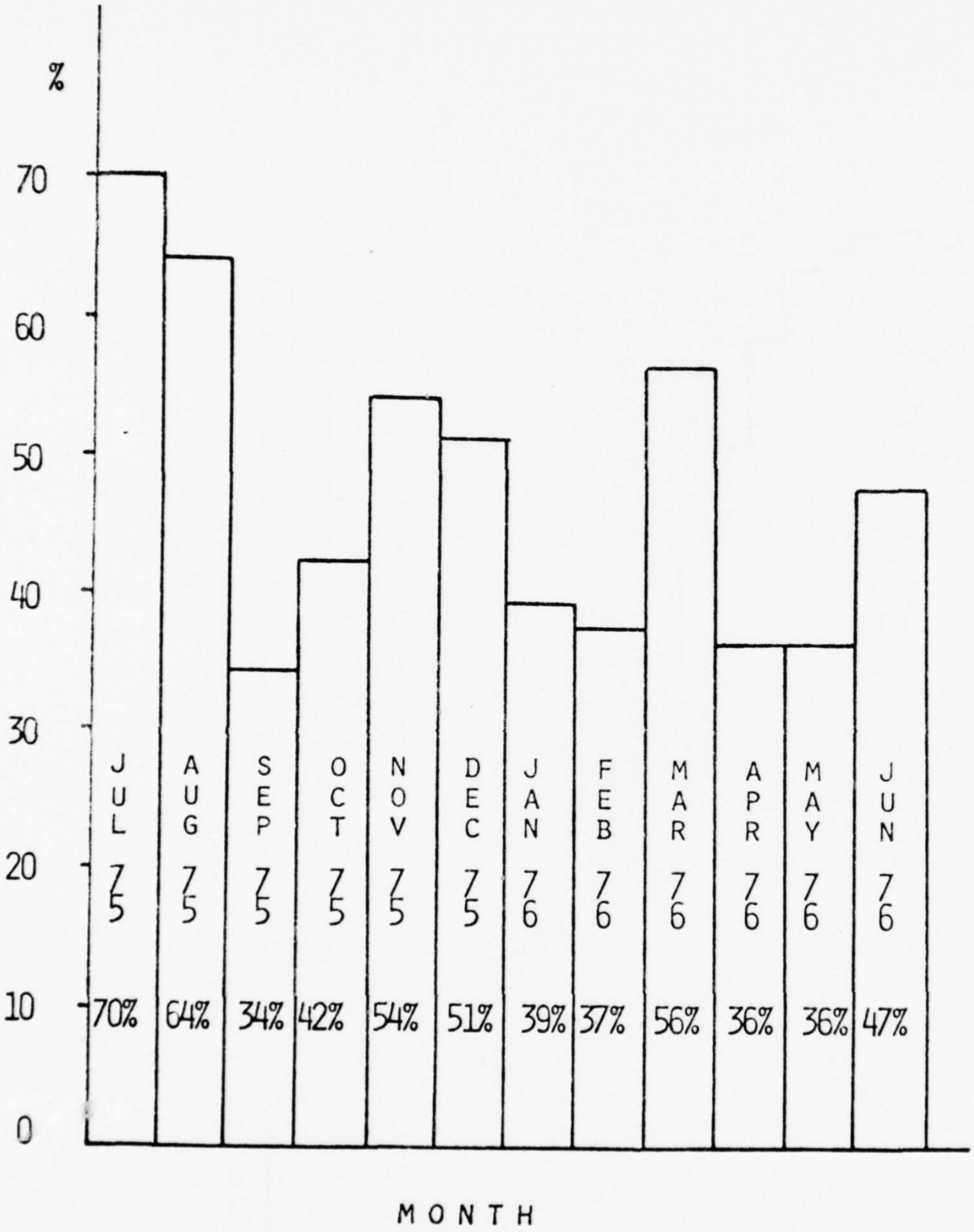


Fig 8.2 QUES Monitoring Performed

On the other hand most university usage occurs in the late morning or afternoon - because of university lectures. Many of the more active users (Blacknest, Salford U and the High Energy Physics Users) access via the RL 360/195 or a direct leased line to the TIP. Thus their usage is not reflected in the figures. The overall pattern of usage monitored over the year is shown in Figs 8.3 and 8.4. The results are examined in detail in (Stokes, 1977A). Another type of statistic of interest to PTTs and the systems manager is that of port usage. Figure 8.5 shows, for example, the number of logins/hour monitored during specific periods of the day.

8.3 Measurement of Usage via PDP9A

PDP9A frontends a number of computers onto ARPANET as has been discussed in Section 2. At the present time these are the RL 360/195 and the RSRE 4080. All traffic through PDP9A is monitored. There are two types of traffic, and hence of measurement. The first is similar to that of the QUES of Section 8.2; instead of monitoring lines to the TIP from the PSTN, we monitor terminals attached to the RL 360 (either directly or via a workstation). The second usage is the converse, the usage from ARPANET accessing the RL 360 or RSRE 4080 as a host.

The analyses performed are similar to those for QUES, although in addition system performance is monitored. For users to ARPANET through the PDP9, the analysis available is identical to that for QUES (since we are considering essentially the same interactions, that is a user connected to a host on the London IMP; in one case the TIP, in the other, PDP9A) with the single exception that the user is identified merely by his terminal address. For usage from ARPANET, again similar data is noted as in Section 8.2; now, however, the source host and 360 user IDENT are noted instead of the destination host and user password of Section 8.2. On a system performance, the information collected is the time at which each machine (PDP9A, IMP, RL 360 and RSRE 4080) went down (scheduled and unscheduled) and came up again. Each occasion of receipt of an incorrect protocol message is noted. In addition the start time, end, source and des-

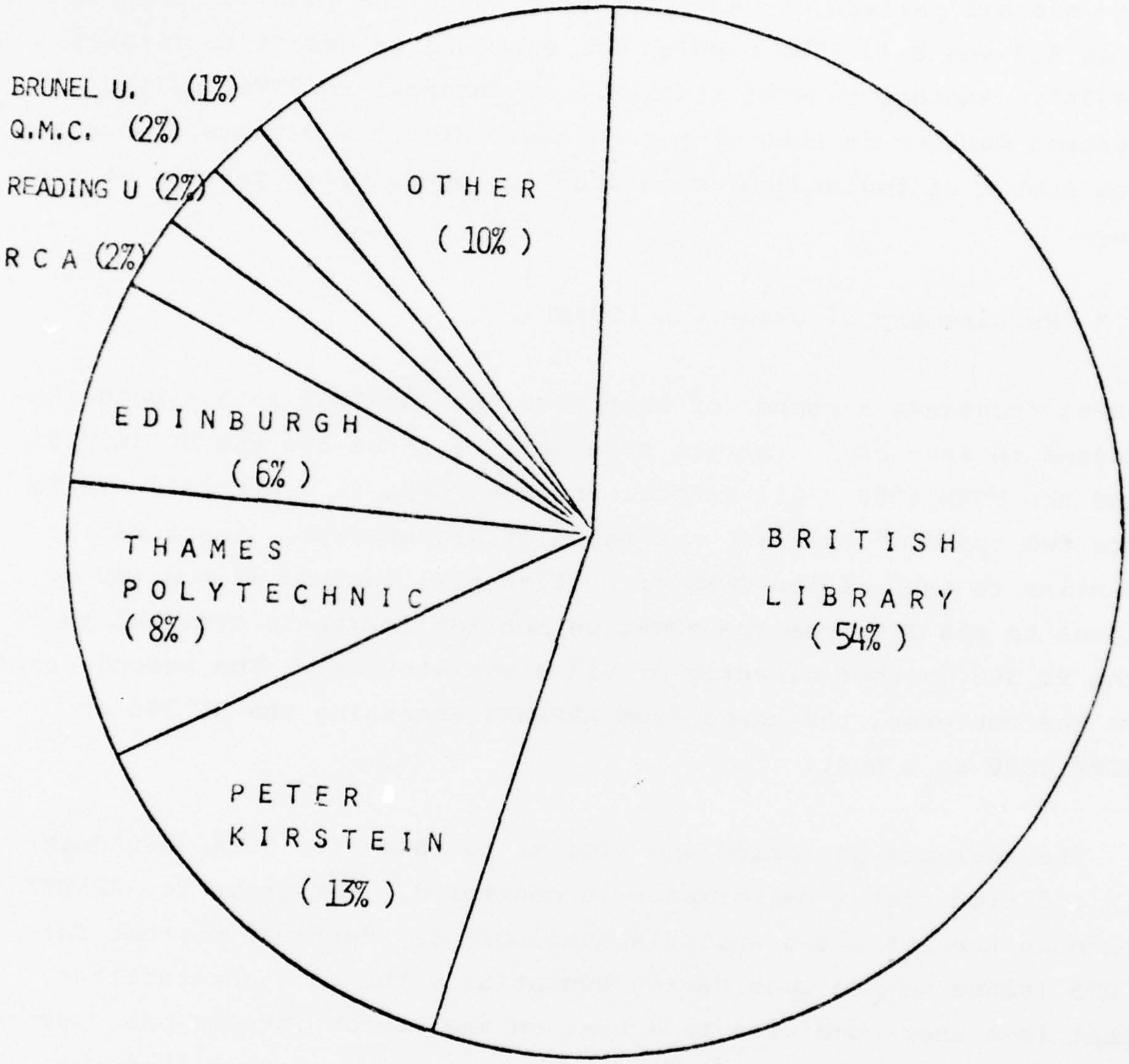


Fig 8.3 Usage of ARPANET by Various Research Groups

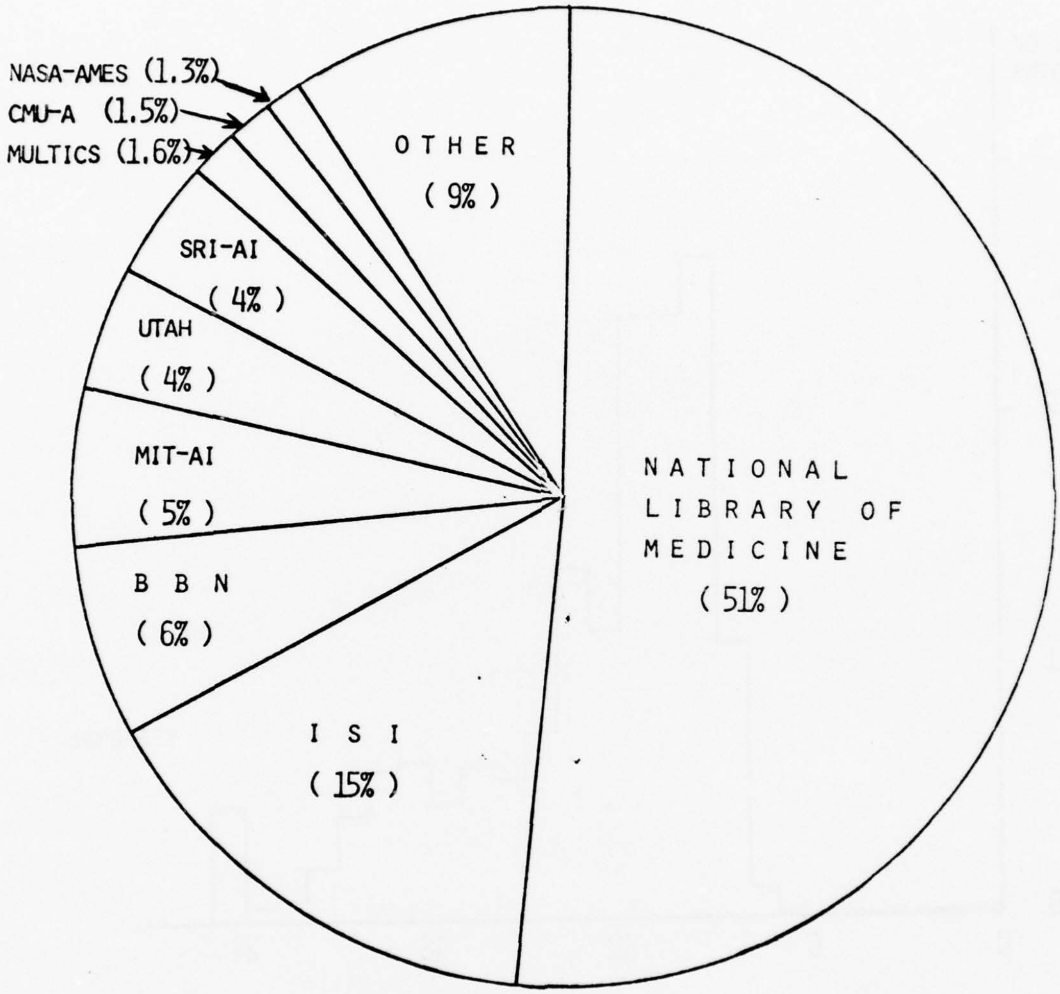


Fig 8.4 Usage of Various ARPANET Hosts

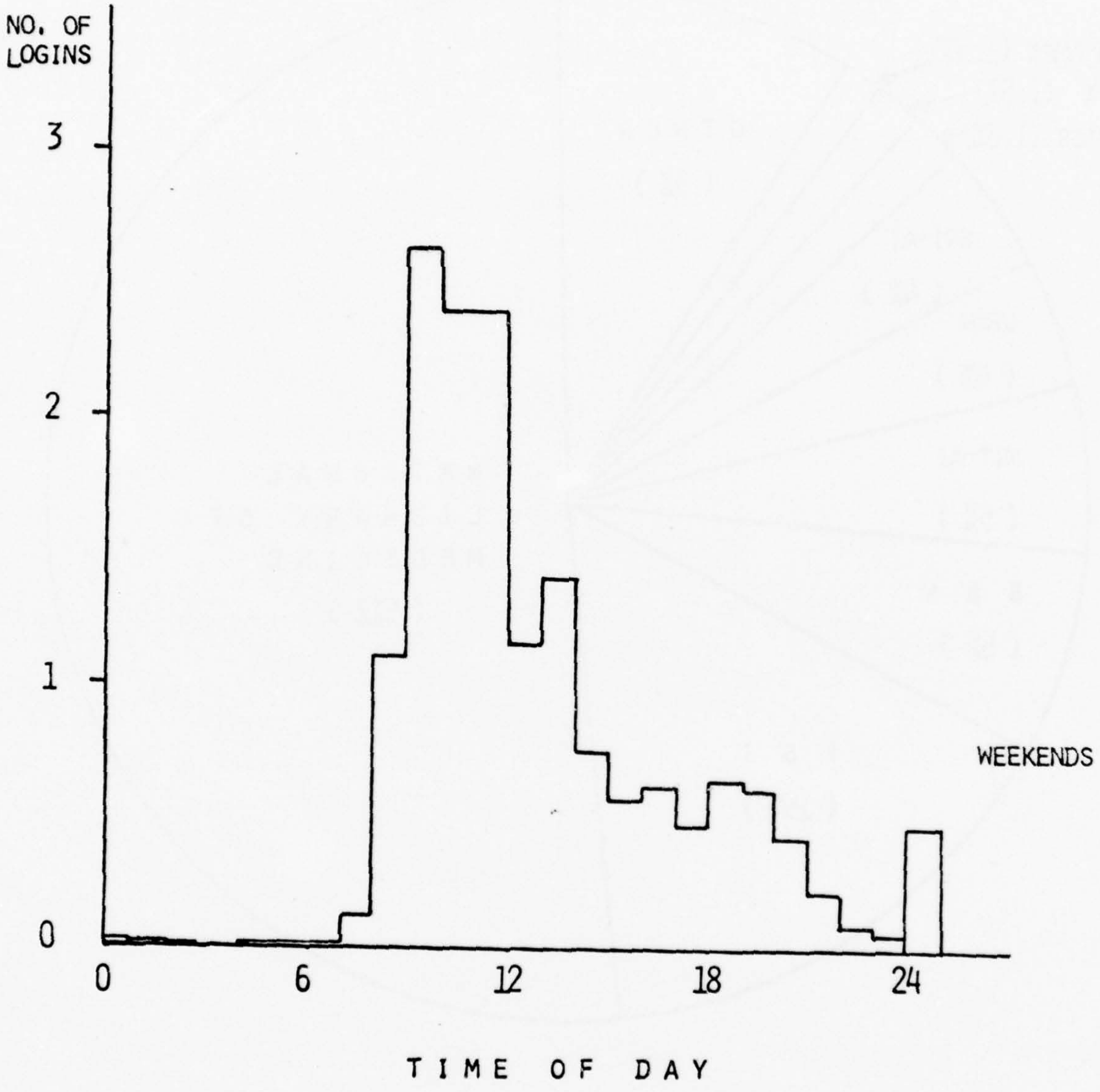


Fig 8.5 Number of Logins per Hour

tination of each file transfer is also noted.

From the performance data noted, we derive figures for the Mean Time Between Failure (MTBF) and the Mean Time to Repair (MTTR), and the frequency of erroneous protocol messages (and their source). From the data collected on ARPANET usage from the RL 360, we derive results identcal to those of Section 8.2, with terminal location replacing user ID. In many cases this is more reliable, because a terminal is used exclusively by a specific group. From the data collected on RL 360 and RSRE usage, again the results derived are identical in form to those of Section 8.2; now the service host and RL 360 ID are used as identifiers. The file transfers are analysed in some detail. For files transfers between the PDP9 and ARPANET, we list the number of files by the Ident of the person initiating them, and the elapsed time and direction of each file transfer. Finally histograms are derived of port usage; for each machine, the distribution of port usage as a function of time is presented.

On examining the global statistics over 1976, we find that the availability of all the machines is high. The number of times the PDP9 crashed (847) is also high but on examining the data more closely it can be seen that on 5 occasions, a single fault (e.g. a host on ARPANET sending illegal messages or a drum fault on PDP9A) manifested itself and caused the system to crash. If this occurred at night, with no-one present, the auto-restart mechanism would enable the crash to recur at intervals of a few minutes until corrected, thus giving rise to a high number of crashes. 307 crashes were due to this and hence, the real number should be 545 giving a MTBF of over 9 hours.

Another global statistic is the number of times all ports were in use. PDP9A has five ports to ARPANET and on many occasions (155 times during the year), these were fully occupied. From PDP9a to the RL machine, there are only three ports available to ARPANET users (two other ports are available; the first is permanently connected to a specific local device, and the second is used for traffic in the other direction). The number of times a fourth user tried to login and was rejected was 358. Fig 8.6 shows the total number of logins to the PDP9A during 1976.

On examining the connection statistics (Fig 8.7) we see that the greater part of usage of the RL 360 is made by the UCL research group (INDRA) and the Blacknest Seismic Research Centre, who have set up a seismic database which is updated daily. The relatively small amount of usage by other groups is a reflection of the nature of host-to-host use of the network; such use tends to be concentrated and concerned with the routine running of jobs or the exchange of files, compared with dialled-up terminal access, which is often concerned with communication and exploratory work. Host-to-host access therefore tends to be of much shorter duration than terminal access.

8.4 Characteristics of a Specific Host

A part of our project has been to provide access to a number of British Library (BL) centres accessing the MEDLINE and CANCERLINE databases on the NLM IBM 370s and to monitor such access. Some monitoring can be performed by QUES; other monitoring is done by a program called MEDLINE which runs in conjunction with QUES.

If a user specifies his required host as 147 (the TIP to which NLM is connected) or NLM he is not released, as usually happens with QUES, but rather is transferred to MEDLINE. This program maintains status information about NLM and informs the user of the current status. If satisfactory, it is assumed he wishes to proceed- if not, he is asked whether he wishes to wait while a connection is attempted.

The program attempts to connect to the five ports in turn, and, if it is able to do so, logs the user in (using either an Ident specified by the user or a free one chosen by the program out of the group allocated to UK users).

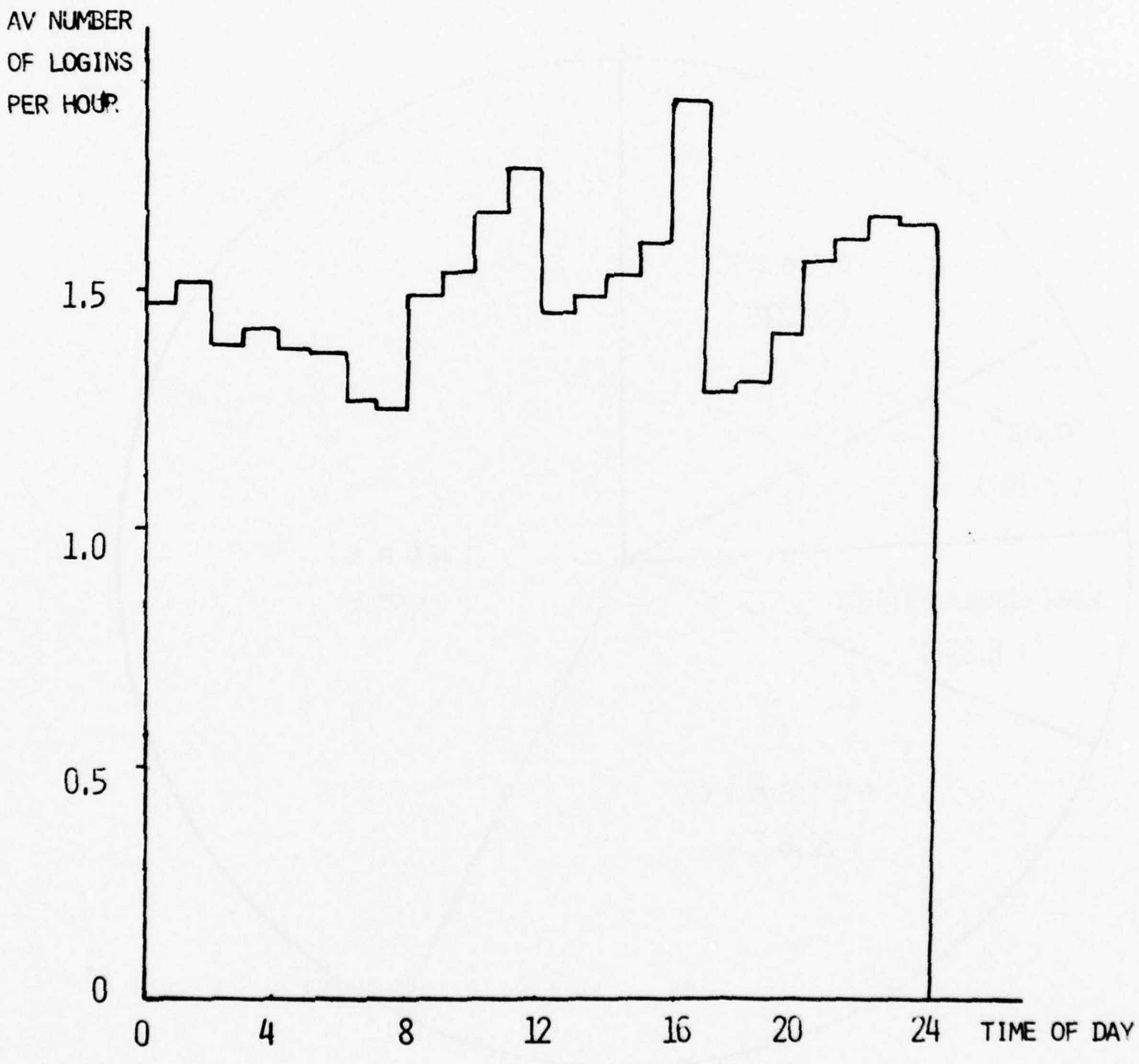


Fig 8.6 Number of Logins Per Hour on PDP9A

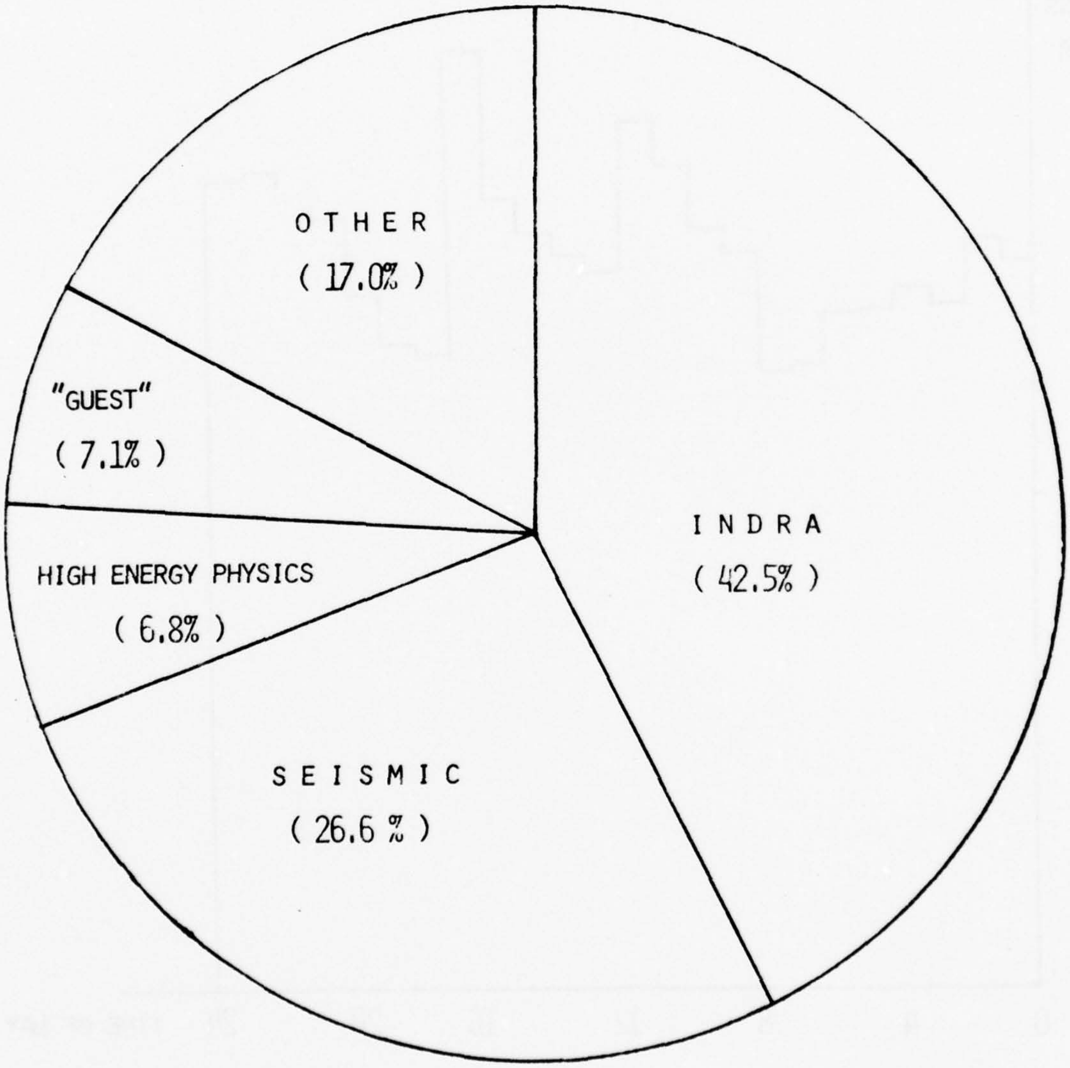


Fig 8.7 Usage of RL360/195 by Various Groups Via UCL PDP9A

It then asks the users for his password, and from this stage on, permits a transparent connection with a few exceptions. These exceptions cover three areas. The first is that simple control characters are permitted, to allow slight local editing of the current input line; the second is that the user is given access to certain data held in the PDP9, e.g. the total connect time of the current interaction; the third is that some of the verbosity of the NLM dialogue is removed. Up to this point, the program merely acts as a Network Access Machine. However, if the monitoring option is selected, a direct transcript of the conversation (or, more specifically, a transcript of all conversations in progress, datestamped and with a channel identifier, multiplexed together) is sent to the current monitoring device, usually magnetic tape, but disk if required.

This tape is then sent to the RL 360 for analysis. The analysis consists of two phases; the first is to demultiplex the separate conversations; the second is to examine characteristics of the usage, for example, occurrences of system keywords.

At the present time, we have considerable data on NLM usage from OUES but little from the MEDLINE program due to various technical problems. A week long experiment is planned in early February 1977 and the results from this will be reported later. The results obtained from OUES provide some interesting insights into usage of NLM. Because of the mode of operation of QUES, some connections are recorded when, in fact, no real connection was made. This is due to a connection being made to the appropriate port on the NBS-TIP correctly, but NLM itself being down. Removing these from the statistics (and also unfortunately some very short genuine interactions), we find that the average time per session is just under 23 minutes (see Fig 8.7) indicating that most users perform one search per session: in one session, however, thirteen searches were performed in nearly two hours, involving the output of over 20,000 characters from NLM.

8.5 Network-Level Performance Measurement

8.5.1 Introduction

In previous years we have constructed simple measurement tools to look at particular throughput/delay characteristics for traffic between the PDP9s and other ARPANET hosts, or to look at particular improvements in the ARPANET Network Control Program (NCP) that we were thinking of making. We found usually that the effort to get reasonable measurements was not negligible, and moreover we were left with a data reduction problem to solve in each case. Unlike the TCP evaluation effort of Chapter 5, we were not prepared to build powerful general measurement tools, as the set of measurements we wanted to make was a changing requirement from time to time.

The introduction of SWITCH (Chapter 3) was a good opportunity for designing a throughput/delay tool that could be used equally to look at throughput/delay to destinations, or at a particular NCP performance. The SWITCH interface allowed the traffic generator to look like a network itself, and to be accessible for running experiments as a similar interface. It was now justifiable to put effort into constructing automatic generation, collection, reduction, and graphing facilities into the tool, because of its generality of use.

Certainly in our context, the usefulness of this measurement facility is akin to a performance measurement tool in an operating system. We wish to evaluate possible improvements in our network support software, to monitor changes in subnet protocols or topology over a period of time in order to detect any effects which may not have been anticipated. Since in our system, we are supporting host connections by RJE-type protocols, we wish to gather performance figures on these hosts, particularly where, in the ULCC attachment, we are using half-duplex protocols with low throughput potential. In the case of EPSS, this tool may be useful in providing thorough statistics on the switching exchange availability from a user point of view. A facility of this type for performance measurement is a useful additional tool for the design of a virtual circuit mapping gateway. A full description of this measurement tool is given elsewhere (Williams,1976); here we summarise the tool facilities and some preliminary measurements.

8.5.2 Tool Facilities

A traffic generator capable of varying frequency and length of message generation resides on the SWITCH gateway as a 'network' address. It is capable of generating to a number of destinations, or on a number of connections to the same destination simultaneously. Traffic is timestamped on generation,and on receipt of echoed messages, allowing overall delay to be measured by halving round-trip time. Additional timestamps can be inserted at the NCP level when necessary. The method does require an echo process to be used at the destination; for hosts it is possible to get round this by using a protocol positive acknowledgement signal to measure delay. If necessary, an echo process can be written at the destination to allow additional timestamping.

The returned messages have timestamps removed, and history records are written to magnetic tape in an IBM-compatible format. An offline analysis program takes the records and formats them suitably for input to the BMD graphing package (a UCLA statistics package commonly available on 360s).

8.5.3 Measurement Results

At the time that the measurement tools were completed, only the ULCC CDC 6400 connection was functioning to the SWITCH interface; the other hosts and network connections to SWITCH were still under development.

The CDC 6400 proved a useful and interesting first application of the tool, as the problems of achieving reasonable throughput with half-duplex protocols are well-known, and we wished to confirm this behaviour on our own connection. We also wished to examine the polling frequency, and message processing delay at ULCC, as well as examining the behaviour of our software which had not previously been used in other than a test setup.

Measurement consisted of generating messages at fixed intervals to be sent to ULCC. As we had no echo program available, we used the protocol acknowledgement as an indication of delay to and from the CDC 6400. Measurements were carried out both at peak times and at night; to facilitate control over measurement periods, the software was extended to allow automatic shutdown at a time given in the set-up of an experimental run.

The results showed a fairly consistent pattern of throughput/ delay as follows:

mean time for a message to CDC 6400 and its acknowledgement:	2.82 secs
time in the PDP9:	0.97 secs
time in the CDC/PDP9 software:	0.015 secs

Subtracting the third figure from the second shows a considerable time spent in the PDP9 outside the actual protocol driver for the 200UT emulation. Although in the PDP9 system, messages do get buffered to backing store, this time is obviously excessive. We were able to track down a scheduling error, which accounted for most of this delay (the PDP9 processes are all currently clock-driven). Time in the protocol handler at 0.015 secs would appear reasonable - the accuracy of our measurement clocks is insufficient to analyse this figure more precisely.

Subtracting the second figure from the first gives the time outside the PDP9 as 1.85 secs. The 24 byte test messages used for these measurements are considerably swollen by addressing, header, and padding characters (the CDC puts out such things as screen clear characters) and the calculated transmission time for messages (both directions) is 0.51 secs. In addition there is a modem switching time associated with the use of a half-duplex protocol. Time spent in the 6400 would therefore appear to be slightly in excess of one second per message. Although the measurements were very consistent, occasional delay peaks as high as 7.37 secs were recorded.

Throughput figures were calculated for both single and triple connections to the CDC 6400. Triple connection is achieved in this implementation by emulating three 200UT terminals in multi-dropped mode. The CDC polls infrequently addresses not currently active, so the additional overhead of using the multi-dropped software is marginal. However it is interesting to compare whether the sum of three loaded connections compares with the throughput on a single connection.

Observed data throughput for a single connection was 135 bps while for cumulative throughput for three connections was 241 bps. The improved figure results from use of the 'dead' time waiting for the 6400 to return the message in a single connection.

Taking into account the overhead bytes, actual throughput is 375 bps. This figure is very low for a 2400 bps line, but highlights the problems of half-duplex multi-drop protocols; it is surprising that these are still being used in contexts where a full-duplex protocol is demanded. Of course a protocol of this type will always have a market in cases of distributed low-activity terminals, where no packet-switching capability is available, or where the capital cost of attaching to a packet-switch network is too high.

While this measurement tool has only been used to ULCC,some conclusions can be drawn already. Firstly the BMD package combines extensive statistical processing capability with very limited graphing capabilities. For the kind of data gathered in these measurements, limited data reduction with better graphing would be more appropriate. Probably a specific formatting program calling a graph-plotting package will be developed in the future. Secondly, our experiences in ARPANET last year showed that host echoing facilities as specified in the host protocol were not appropriate to this sort of measurement. TALK/LINK subsystems on Tenex would seem a much more appropriate echo method, and preliminary tests show that this will probably be the most appropriate method to use.

Talk/Link subsystems are designed to allow several terminal users to talk to each other, and operate by copying the user's line to another terminal connection. Copying here operates at a higher priority than echo sockets, and therefore can be adapted more easily to the mode of message reflection required.

8.6 Line Measurements

The measurements described in the preceding sections are entirely at the user level; this is only one of a number of levels at which measurements may be performed. Another level, with which this section is concerned, is that of the IMP-IMP line. We have a program running on PDP-9B which can measure the traffic between our IMP and the adjacent IMP (more specifically it copies data headers from the line between our IMP and the modem). The headers are copied onto magnetic tape for later analysis on the UC 360/65.

One serious problem that arises is that of data throughput. The line is 9.6K bps, which means that data can arrive at one character every 0.8 ms. Since the communication adaptors operate under program interrupt, for each packet - and the minimum packet size is 4 characters - there is a possible crisis time of 3 ms. The only way to achieve this throughput is to use a circular buffer pool (currently three buffers per line) which is used for data in from the line and data out to the tape. However, there would be serious problems in attempting to record data from a line of considerably higher bandwidth.

The analysis program effectively models the network protocol implementation. It permits analysis at any (or all) of the three levels of protocol: IMP-IMP, Host-Host or User. The volume of data which is monitored is extremely high; for example, in an hour, two megabytes of routing data (data which enables the IMPs to alter dynamically their routing tables) are recorded. Most of the data is at the IMP-IMP level. An example of the type of data which can be derived is shown in Fig 8.9. Here we show a histogram of the types of packet as a function of time; R denotes routing packets, I IMP-IMP control, H is Host-Host Protocol, and D is Data.

Since the information of interest is often about a specific connection or about a specific facet of a protoçol, the analysis programs were written accordingly. In a typical analysis session, there are two (or more) runs. The first scans the tape, and produces a table of contents for the tape including such information as the duration of measurement and the amount of data analysed. The second and subsequent runs give specific details. The various analyses that are possible are specified in detail in (Treadwell, 1977). The following are some of particular interest.

The first is the "Main log" which is a verbal description of each packet on the line (see Fig 8.10) although the new version may decide to omit certain types of packet, such as routing packets. These may be analysed separately to determine, for example, the instantaneous state of the network, as seen by the UCL IMP.

In addition to the logs we are able to produce five types of graph. One example, a plot of the delay between the sending of a message and the RFNM (Ready for Next Message) as illustrated in Fig 8.11).

This measurement tool has proved valuable on a number of occasions. Our preliminary results revealed two interesting aspects of the London-Norway line. First the routing messages accounted for over 60% of the total data traffic, even when the latter is high. Secondly, the number of retransmissions was found to be very high. This is caused by the large amounts of seismic data being injected at Norway, about 2.4K bps average, and the limited number of packets that can be outstanding at any one time. The combination of these facts, with the long transit time over the satellite circuit and the low bandwidth of the UCL-Norway and Norway-Us lines, leads to the Norway IMP being swamped and rejecting data until it has freed some buffers. This explained some anomalous results encountered with our TCP experiments (see Section 5.3.2), when it was noted that contrary to expectations, increasing the "window" size past a certain threshold actually reduced throughput.

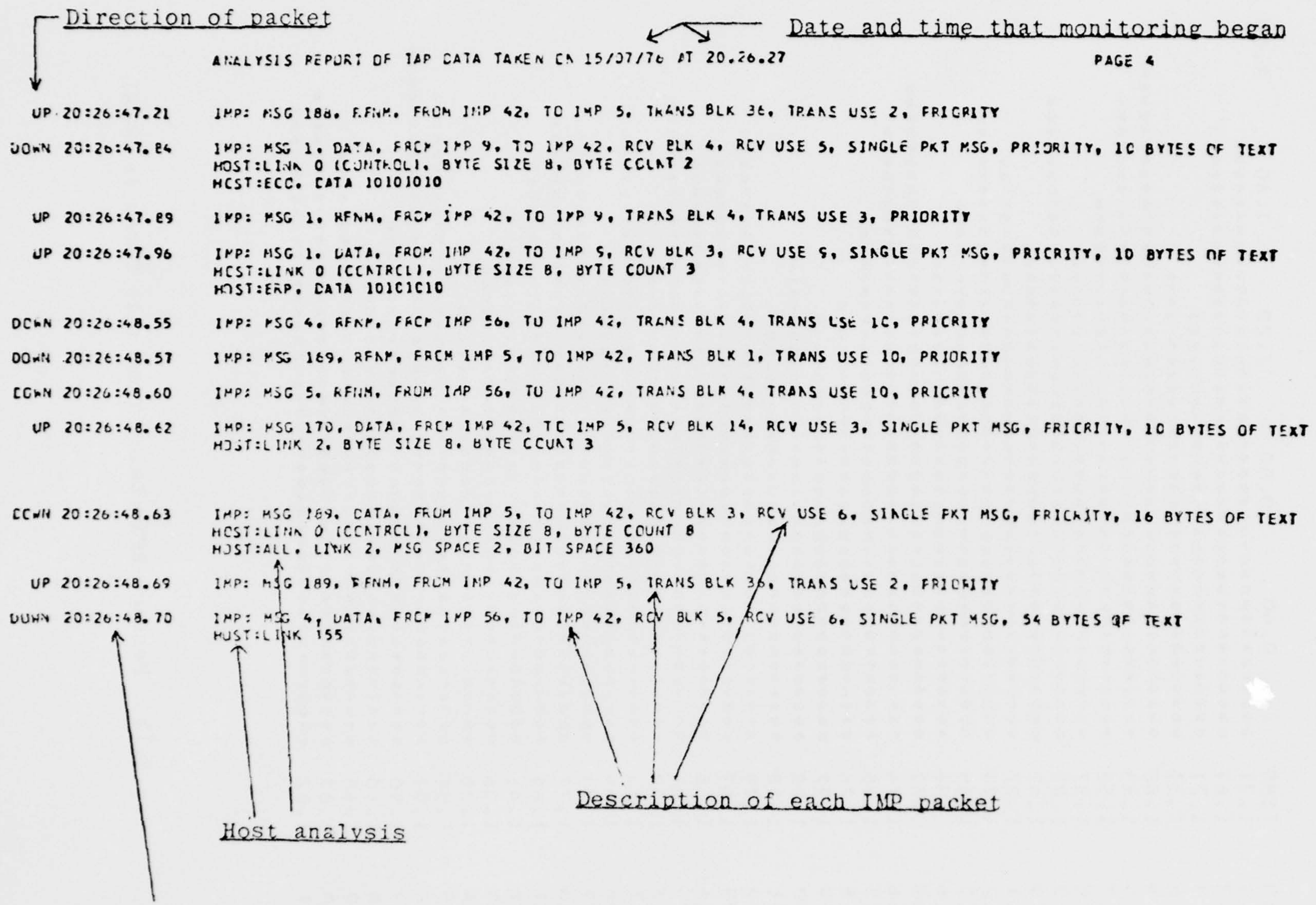
Another point of interest is that when a host on our IMP comes up, it attempts to send "resets" to every other host on the network. Due to the mode of operation of the protocols this results in the sending of three times as many messages into the network and this again causes the Norway IMP to flood.

Thus, this tool is a very valuable addition to our repertoire of measurement tool and provides a flexible means of examining the lowest level. It is, however, most useful in areas where we have a requirement to examine a specific type of behaviour, owing to the vast quantity of data generated.

```
10.04|RRRRRRRRRRRRRRRRIIIIIIIIIIIIIIHDDDDDDDD |          |
10.14|RRRRRRRRRRRRRRRIIIIIIIIIIIIIIIIHDDDDDDDDDDDD       |
10.24|RRRRRRRRRRRRRRRIIIIIIIIIIIIIIIHDDDDDDDDD|          |
11.34|RRRRRRRRRRRRRRRRIIIIIIIIIIHDDDDDDDD     |          |
11.44|RRRRRRRRRRRRRIIIIIIIIIIIIIHHDDDDDDDDDDDDDD         |
11.54|RRRRRRRRRRRRRRRIIIIIIIIIIIHDDDDDDD      |          |
11.04|RRRRRRRRRRRRRRIIIIIIIIIIIIHHDDDDDDDDDDDDDDDDD      |
11.14|RRRRRRRRRRRRRRRRIIIIIIIIIHDDDDDDDDDDDDD |          |
11.24|RRRRRRRRRRRRRIIIIIIIIIIIIHDDDDDDDDDD    |          |
12.34|RRRRRRRRRRRRRRRRRRIIIIIIIIIIIIIHDDDDDDDDDD         |
12.44|RRRRRRRRRRRRRRRIIIIIIIIIIIIIHDDDDDD     |          |
12.54|RRRRRRRRRRRRRRRIIIIIIIIIIHDDDDDDDDD     |          |
12.04|RRRRRRRRRRRRRRRRIIIIIIIIIIIIHHHDDDDDDD  |          |
12.14|RRRRRRRRRRRRRRRIIIIIIIIIIIIIHHDDDDDDDDDD|          |
12.24|RRRRRRRRRRRRRRRIIIIIIIIIIIIIIIIHHDDDDDDDD          |
13.34|RRRRRRRRRRRRRRRIIIIIIIIIIIIIIIIIHHDDDDDD|          |
13.44|RRRRRRRRRRRRRRRIIIIIIIIIIIHDDDDDDDDD    |          |
13.54|RRRRRRRRRRRRRRRIIIIIIIIIHHDDDDDD        |          |
13.04|RRRRRRRRRRRRRRRIIIIIIIIIIIIIHDDDDDD     |          |
13.14|RRRRRRRRRRRRRRIIIIIIIIIIIIIHDDDDDD      |          |
13.24|RRRRRRRRRRRRRRRRIIIIIIIIIIIHDDD         |          |
14.34|RRRRRRRRRRRRRRIIIIIIIIIIHDDDDD|         |          |
14.44|RRRRRRRRRRRRRRRRIIIIIIIIIIIIHHDDDDDD    |          |
14.54|RRRRRRRRRRRRRRRRIIIIIIIIIIHDDD|         |          |
14.04|RRRRRRRRRRRRRRIIIIIIIIIIIIIHHDDDDDD     |          |
14.14|RRRRRRRRRRRRRRIIIIIIIIIIIIIHHDDDDDD     |          |
14.24|RRRRRRRRRRRRRRRIIIIIIIIIIIIIIIIIHHDDDDDDDD         |
15.34|RRRRRRRRRRRRRRRRIIIIIIIIIIIHHDDDDDD     |          |
15.44|RRRRRRRRRRRRRRRRRIIIIIIIIIHDDDDD        |          |
15.54|RRRRRRRRRRRRRRRIIIIIIIIIIHDDD |         |          |
15.04|RRRRRRRRRRRRRRRRRIIIIIIIIIIHDDD         |          |
15.14|RRRRRRRRRRRRRRRRRRIIIIIIIIIIIHDDD       |          |
15.24|RRRRRRRRRRRRRRIIIIIIIIIIIIIIHHDDDDDD    |          |
16.34|RRRRRRRRRRRRRRRRIIIIIIIIIIIIIIIHHDDDDDDDD          |
16.44|RRRRRRRRRRRRRRIIIIIIIIIIIIIIIHHDDDDDD   |          |
16.54|RRRRRRRRRRRRRRRIIIIIIIIIIIIIIIIIIHHDDDD |          |
16.04|RRRRRRRRRRRRRRRRIIIIIIIIIIIIIIIHHDDDDDDDD          |
16.14|RRRRRRRRRRRRRRRRIIIIIIIIIIIIHDDDDDDDD   |          |
16.24|RRRRRRRRRRRRRRIIIIIIIIIIIIIIHDDDDDDD    |          |
17.34|RRRRRRRRRRRRRRRRIIIIIIIIIIHDDDDDD       |          |
17.44|RRRRRRRRRRRRRRRIIIIIIIIIIIIHDDDDDD      |          |
17.54|RRRRRRRRRRRRRIIIIIIIIIIIHHDDDDDDDD      |          |
17.04|RRRRRRRRRRRRRRRIIIIIIIIIIIIIHHDDDDDDD   |          |
17.14|RRRRRRRRRRRRRRRRIIIIIIIIIIIIIIIIIHHDDDDD|          |
17.24|RRRRRRRRRRRRRRRIIIIIIIIIIIIIIIIIHHDDDDD |          |
18.34|RRRRRRRRRRRRRRRRRIIIIIIIIIIIIIIIIHHDDDDDD          |
18.44|RRRRRRRRRRRRRRIIIIIIIIIIIIIIIIIIHHDDDDDDDDD        |
18.54|RRRRRRRRRRRRRRRRIIIIIIIIIIIIIHDDDDDDDDD |          |
18.04|RRRRRRRRRRRRRRRIIIIIIIIIIIIIIHHHDDDDDDD |          |
18.14|RRRRRRRRRRRRRRIIIIIIIIIIIIIHDDDDDDDDD   |          |
18.24|RRRRRRRRRRRRIIIIIIIIIIIIIIHDDDDDDDDDD   |          |
19.34|RRRRRRRRRRRRRIIIIIIIIIIIIIIIHHDDDDDDDDD |          |
19.44|RRRRRRRRRRRRRRRIIIIIIIIIIIIHHDDDDDDDDDD |          |
19.54|RRRRRRRRRRRRRRRRRIIIIIIIIIIIIIIIHHDDDDDDDDDD       |
19.04|RRRRRRRRRRRRRRIIIIIIIIIIIHDDDDDDDDD     |          |
19.14|RRRRRRRRRRRRRIIIIIIIIIIIIIIHDDDDDDDDD   |          |
19.24|RRRRRRRRRRRRRRRIIIIIIIIIIIIIIIIHHDDDDDDDDDDD       |
20.34|RRRRRRRRRRRRRRIIIIIIIIIIIIIHDDDDDDDD    |          |
20.44|RRRRRRRRRRRRRRIIIIIIIIIIIIIIIIHHDDDDDDDDDD         |
20.54|RRRRRRRRRRRRRIIIIIIIIIIIIHDDDDDDD      |          |
```

Fig 8.9 Packet Types as a Function of Time

SAMPLE REPORT OUTPUT



```
                ANALYSIS REPORT OF TAP DATA TAKEN ON 15/07/76 AT 20.26.27                          PAGE 4

  UP 20:26:47.21   IMP: MSG 188, RFNM, FROM IMP 42, TO IMP 5, TRANS BLK 36, TRANS USE 2, PRIORITY

DOWN 20:26:47.84   IMP: MSG 1, DATA, FROM IMP 9, TO IMP 42, RCV BLK 4, RCV USE 5, SINGLE PKT MSG, PRIORITY, 10 BYTES OF TEXT
                   HOST:LINK 0 (CONTROL), BYTE SIZE 8, BYTE COUNT 2
                   HOST:ECO, DATA 10101010

  UP 20:26:47.89   IMP: MSG 1, RFNM, FROM IMP 42, TO IMP 9, TRANS BLK 4, TRANS USE 3, PRIORITY

  UP 20:26:47.96   IMP: MSG 1, DATA, FROM IMP 42, TO IMP 5, RCV BLK 3, RCV USE 5, SINGLE PKT MSG, PRIORITY, 10 BYTES OF TEXT
                   HOST:LINK 0 (CONTROL), BYTE SIZE 8, BYTE COUNT 3
                   HOST:ERP, DATA 10101010

DOWN 20:26:48.55   IMP: MSG 4, RFNM, FROM IMP 56, TO IMP 42, TRANS BLK 4, TRANS USE 10, PRIORITY

DOWN 20:26:48.57   IMP: MSG 169, RFNM, FROM IMP 5, TO IMP 42, TRANS BLK 1, TRANS USE 10, PRIORITY

DOWN 20:26:48.60   IMP: MSG 5, RFNM, FROM IMP 56, TO IMP 42, TRANS BLK 4, TRANS USE 10, PRIORITY

  UP 20:26:48.62   IMP: MSG 170, DATA, FROM IMP 42, TO IMP 5, RCV BLK 14, RCV USE 3, SINGLE PKT MSG, PRIORITY, 10 BYTES OF TEXT
                   HOST:LINK 2, BYTE SIZE 8, BYTE COUNT 3


DOWN 20:26:48.63   IMP: MSG 189, DATA, FROM IMP 5, TO IMP 42, RCV BLK 3, RCV USE 6, SINGLE PKT MSG, PRIORITY, 16 BYTES OF TEXT
                   HOST:LINK 0 (CONTROL), BYTE SIZE 8, BYTE COUNT 8
                   HOST:ALL, LINK 2, MSG SPACE 2, BIT SPACE 360

  UP 20:26:48.69   IMP: MSG 189, RFNM, FROM IMP 42, TO IMP 5, TRANS BLK 36, TRANS USE 2, PRIORITY

DOWN 20:26:48.70   IMP: MSG 4, DATA, FROM IMP 56, TO IMP 42, RCV BLK 5, RCV USE 6, SINGLE PKT MSG, 54 BYTES OF TEXT
                   HOST:LINK 155
```

Fig 8.10 . Time of packet in hours:minutes:seconds.hundredths

```
                MESSAGE/RFNM DELAY TIME (ALL TIMES IN SECS)

MSG    TIME        0.40      0.80      1.20      1.60      2.00      2.40
  1    1.71  ******************************************
  2    1.67  *****************************************
  3    1.21  ******************************
  4    1.33  *********************************
  5    1.88  ***********************************************
  6    1.83  *********************************************
  7    1.52  **************************************
  8    1.37  **********************************
  9    1.82  *********************************************
 10    1.42  ***********************************
 11    1.67  *****************************************
 12    1.77  ********************************************
 13    1.55  **************************************
 14    1.36  **********************************
 15    1.87  **********************************************
 16    1.74  *******************************************
 17    1.46  ************************************
 18    1.26  *******************************
 19    1.22  ******************************
 20    1.38  **********************************
 21    1.18  *****************************
 22    1.70  ******************************************
 23    1.67  *****************************************
 24    1.66  *****************************************
 25    1.81  *********************************************
 26    1.39  **********************************
 27    1.74  *******************************************
 28    1.34  *********************************
 29    1.73  *******************************************
 30    1.77  ********************************************
 31    1.64  *****************************************
 32    1.41  ***********************************
 33    1.36  **********************************
 34    1.25  *******************************
 35    1.85  **********************************************
 36    1.97  *************************************************
 37    1.50  *************************************
 38    1.40  ***********************************
 39    1.68  ******************************************
 40    1.87  **********************************************
 41    1.82  *********************************************
```

Fig 8.11   Message/RFNM Delay Time (All Times in Secs)

CHAPTER 9 EXPERIENCE WITH UK USE OF ARPANET

9.1 Introduction

Use of ARPANET from the UK during 1976 has largely followed the pattern that was becoming established in 1975; there is more host-to-host use for collaborative projects between UK and US research groups, and less terminal access for more casual use. Thus we were able to reduce the number of TIP ports in the PSTN to 4, but channels to the RL 360 were frequently congested.

The requirement for host connections to the UCL TIP results from a problem which has inhibited network use for a number of groups, that of input and output with only terminal access. This problem was discussed in the users report for 1975 (Chapter 7 of Kirstein, 1976A).

Other problems have arisen this year which are peculiar to our node and the attachment of the RL 360 to the network. Because of our slow link to the rest of ARPANET (we have a 9.6K bps line to Norway compared with 50K bps lines over most of the network), and the growing congestion of the transmission links, there were timeout problems with the numerous Tenex systems we use. This meant that it was very difficult for us for a time to actually establish a Login to a Tenex. This problem was solved by the extension of the relevant Tenex timeout period, to allow enough time for a response to be received from the UCL node.

Our method of attaching the RL IBM 360/195 as a host to the ARPANET has been to implement all the ARPANET protocols in our local PDP9, which simulates a remote workstation to the 360. This has worked very well, but some disadvantages have become noticeable this year. One major snag is that we have to implement our software in terms of the 360 operating system, over which we have no direct control; our ARPANET File Transfer Program has to submit two jobs to the 360 to get a file transferred from Rutherford to a site in the US. This can take a few minutes or more, a frustrating wait for the user. Another difficulty is that having to go through the UCL PDP9 to get to the RL 360 means that there is twice the probability of a failed connection; this makes remote interactive use from the US less attractive. Finally, the operating system at RL was not designed for remote interactive use and it has been difficult for a new US user to find his way round the system.

These problems are all being tackled. An improved file transfer program will soon be available which should at least make the process less frustrating to the user, by relaying messages on the progress of the transfer. Online documentation is being put on the 360 which will help users get more information about the system, and it is hoped to improve Help facilities when time permits.

The effort is well worth while, for, in spite of the above difficulties, collaborative research between groups using the RL 360 and sites in the US is proving very valuable to both parties. The network provides a means of truly working together which was not in any way possible before.

The pattern of use as a whole from the UK is analysed in Section 9.2, and comparisons are made with 1975. Details of individual groups use are given in Section 9.3.

The information is defective in two respects. First, the UCL usage is not mentioned; most of the rest of this report covers that topic. Second, the information is based on the utilisation of the network by the UK research groups - not on direct feedback from US collaborators. The UK groups are obligated to furnish us with research and progress reports - thus making the writing of this chapter possible. It is impractical to obtain the same quality of feedback from the US collaborators directly. Finally, in Section 9.4 conclusions are made from experience so far about the usefulness of the UCL node to UK and US research.

9.2 Analysis of Usage during 1976

This report considers the activities of 34 research groups for whom approval was received from the Governing Committee of the UCL ARPANET project for their use of the ARPANET node. It is based on their individual reports submitted to UCL at the end of 1976. A number of other groups were also approved, but so late in the year that it is too soon to make any assessment of their use.

Of the 34 groups considered, 10 may be classified as being 'highly-active', 12 have 'low-activity' status, and the remaining 12 have made virtually no use of the network during the year. This compares with 1975, when only 7 groups out of 30 were highly-active in their use of the network, although only 5 groups were non-active.

There is no predominant reason why so many groups have been inactive this year. The reasons vary widely, and relate mostly to the individual situation; some have been held back by lack of time or money, some by illness. A few have been unable to get the accounts they needed on network machines - usually where they were unable to establish mutually beneficial contacts with US fellow research workers with ARPA contracts.

The distinction between high and low activity on the network is not always easy to make. It is not a reflection simply of the amount of time spent accessing hosts; it is more an indication of the importance of network access to the research group concerned. A number of users find network access useful but could progress with their research without it; these we classify as 'low-activity'. 'High-activity' groups are doing work which would either not be possible at all or would progress much more slowly without the network.

The resources available on the ARPA network vary tremendously. There are therefore many different ways of using the network, and it is interesting to compare these; this will be done later in this section. A number of points can be made, however, about network use in general. It does seem, for instance that a new user should expect that it will take 3-6 months before he can begin useful work. The only exception to this has been database access, where the instructions for information retrieval are clearly laid-out and the system has been designed for easy use.

There are a number of reasons why it takes longer to establish a means of using a computer on a network than on a local machine. It may take a while to arrange an account on a remote machine; having established an account, there may be delays in getting documentation and information on using the system. Again, some machines connected to ARPANET have operating systems which were not designed for interactive terminal access and are therefore difficult to use. Most important of all, a pattern of use has to be established, particularly if the project is a collaborative one; it is very easy to see that

ARPANET SITE	UK USERS
ANL (Argonne Nat.Lab.)	Oxford Numerical Algorithms Group, Oxford Nuclear Physics.
BBN (Bolt, Beranek & Newman)	Medical Res. Council, Queen Mary College.
CMU-A (Carnegie-Mellon U)	Queen Mary College, Royal College of Art, Liverpool U.
Datacomputer	Thames Polytechnic
HARV-10 (Harvard U)	Oxford U Nuclear Physics, Royal College of Art.
ILLIAC IV	Salford U, Reading U.
ILL-NTS (Illinois U)	Oxford Nuclear Physics, Queen Mary College.
ISI (Information Sciences Inst.)	Blacknest Research Est., Brunel U. Cambridge Computer Lab., Ministry of Defence, N.London Polytechnic, Royal College of Art, Reading U, Salford U.
LBL (Lawrence Berkeley Lab.)	Bristol U, Durham U.
MIT-Multics (Massachusetts Inst. of Technology).	Royal College of Art.
NLM (National Lib. of Medecine).	British Library Research Division, British Library Lending Division, Dept. of Industry.
Office-1	Communication Studies Group, Royal College of Art.
RAND-UNIX (Rand Comp)	Queen Mary College, Thames Polytechnic.
SU (Stanford U.)	Edinburgh U Computer Science Dept.
SUMEX-AIM (Stanford Medical Centre)	Royal College of Art.
UCB (U California, Berkeley)	Queen Mary College.
UCLA (U California Los Angeles)	Queen Mary College.

TABLE 9.1: ARPANET sites used by UK users.

9.2.2 Mode of Network Use

As mentioned earlier, the ways in which UK research groups use the network vary quite considerably. There are basically four types of use at present: Database access, use of teleconferencing and message facilities for the exchange of ideas and opinions, access to software and/or data at a remote site, transfer of files from a remote site to a local one. Some groups use only one method, others may use a combination of methods.

Database access is made by the group of users who are participating in the British Library CANCERLINE Project and by the Thames Polytechnic Project. CANCERLINE is available via the NLM machine, which is not connected as a proper host to the network and is difficult to access. This difficulty has been overcome by a program written at UCL which makes the connection for the user. CANCERLINE itself is a good system and has been used successfully without any further problems. The Datacomputer, a mass storage device, is still largely an experimental system, and requires considerable time to become familiar with. Access to it is therefore not typical of a database system; the difficulties involved are similar to those of other newly-developed special resources, such as the ILLIAC IV parallel processor. Resources of this kind are so unique and so new that the intrepid user must tread slowly and carefully around the system until he identifies the hazards.

Use of the network for teleconferencing and the exchange of ideas is now so well established that there are few problems involved. Members of the UK Ministry of Defence have collaborated with the US Department of Defense in the evaluation of the MoD real-time language CORAL over the network. Wilkinson of the Medical Research Council, Cambridge, is collaborating with Donchin of Urbana on the analysis of neurophysiological data, using teleconferencing methods. There have been no problems, except that teleconferencing can be somewhat tedious using a slow terminal.

The most satisfactory way of using the network from a terminal is to access remote software and data. This usually requires only a small amount of input, and if the program being run involves lengthy computations, any delay in response from the net will not matter too much. Brunel U and Cambridge U Computer Laboratory have used this method successfully. If the output from the remote program produces large files, there is the problem of getting it into hard copy form without tying up the terminal for hours; this problem has been largely overcome for UK users by the attachment of a line-printer to the UCL TIP and using this to print user files.

The fourth way of working is to transfer software or data from a remote host machine to a local one, and use it there for development and testing. This method is used for the high-energy physics collaboration between Oxford, Harvard, Illinois and Chicago, as they have encountered some frustration in using the RL 360/195 from the US. The problem stems from having to rely on three computers all being operational at the same time (Illinois, UCL, RL). With widely differing local times at each end, different maintenance periods, together with broken or aborted connections, it can be difficult at times to achieve a useful work session. This situation has been reported by only one group, however, and should not be taken as a typical network situation. If the only viable way to use the network was to transfer files from one site to another, then it would be difficult to justify its existence in terms of resource-sharing.

Flexibility would appear to be the keyword for successful exploitation of the network's facilities. Most projects develop by stages, and each stage may need a different approach or a different type of resource. It is interesting to see how two of our user groups, Bristol U and Durham U, have used the network this year. Bristol began by transferring programs written originally on the CDC 7600 at LBL onto the RL 360/195, where further development took place. They are now entering a second stage of work. For technical reasons the next stage in the program is best run at LBL, where the facilities are better for their purpose. They are currently developing the programs to output the analysis from the 360 into a form for transferring to LBL for further analysis. At the same time, Kelly of LBL scans the results of the 360 runs from LBL and keeps a log of the status of the runs.

Working in a similar way, Durham U have implemented a version of the LBL Data Management Program, BKYBDMS, on the RL machine. During implementation it was invaluable to be able to test the machine independence of their IBM subroutines by running them on the LBL CDC machine. A second stage of the work has involved the design of a coding language. This has been set up at LBL to enable others in the collaborative group to add data and carry out experiments. Future use of the net for Durham U will involve the querying of the databases at LBL for book-keeping purposes.

Another group whose use has developed interestingly this year is the Blacknest Research Centre. They began in quite a modest way by accessing seismic data files at ISI. Now they have set up their own seismic database on the RL 360/195 and regularly exchange data with seismic centres in the US. For them, file transfer is the ideal way of working, since the files are small and the data is needed locally.

A final point needs to be made about the use of special resources such as the ILLIAC IV parallel processor. This is such an expensive resource that compilation and debugging of software to be run on it must be carried out on another host machine and then transferred to the ILLIAC for execution. This type of work can be carried out from a terminal on a small scale, but any large-scale development of programs really needs to be done via a local host, so that listings and dumps can be obtained for examination. This, of course, will tend to apply to software development on any remote machine, but is especially true of a specialised resource where so many problems are likely to occur before the user comes to grips with it.

In summary, full use of ARPANET is best achieved through a local host rather than by terminal access. A flexible approach to the way in which one works will enable the resources and facilities the network has to offer to be exploited most fully.

9.2.3 Problems Encountered in the Use of ARPANET

Most of the problems encountered by UK users have already been mentioned. Four groups had some trouble in accessing and making use of two hosts, the NLM machine and the ILLIAC IV. US users have also found the operating system of the RL 360 difficult to use. The lesson to be learned from these experiences must surely be that when the decision is taken to connect a machine to a network, adequate consideration must be given to providing Help facilities and suitable user interfaces to the system.

Several groups found their work severely hampered by the lack of an account on a suitable US machine. Use of a guest account can usually be arranged for short term experimental work: for serious use, a UK user needs to establish collaborative links with fellow research workers in the US - preferably before he contemplates using the network.

The inability to cope with large I/O files from a terminal impeded full collaboration for two groups, Queen Mary College and Cambridge U Computer Laboratory. This problem should be solved when EPSS is in full operation (see Chapter 6).

The only other problem of note was the slowness of response from the net. This is not so apparent early in the UK day, but becomes a nuisance by early afternoon, when US workers start to use the machines.

On the whole, most users found the network reliable and most of the hosts easy to use.

9.2.4 Mode of Access to the Network

Out of the 20 users actively using the net this year, 15 did so from terminals, 4 via the Rutherford Laboratory and 1 group has a leased line connection to the UCL TIP. Of the 15 terminal users, 6 groups admitted to finding this type of access unsatisfactory, because it was slow and inhibited input and output. Even for fast terminals, I/O is restricted by very limited buffer space in the TIP. The only type of ARPANET user for which terminals seem perfectly adequate is for database access or for running programs when I/O is minimal. There is also a problem of poor quality transference of data over long distances via the public switched telephone network. This may distort I/O to a degree where work becomes very difficult.

9.2.5 Connection Times

The Medline group used most connection time through the UCL TIP, since there are about 15 participating groups and they carried out searches on a daily basis. Apart from this, most users were able to carry out their work quickly. The group reporting the longest average weekly connection time was Salford U, with 6 hours a week.

On the other hand, Bristol U used only a few minutes a week in the early stages of their project and expect to be spending no more than 30 minutes a week in the next stage of their collaboration. The average connection time for highly active users is around 2 hours a week.

9.2.6 Summary

Use of the network this year has been much more interesting to analyse, because the patterns of use have become much clearer. With the connection of important laboratories such as the Lawrence Berkeley and Argonne, important collaborative work has developed for which the existence of the network is absolutely vital. UK users working collaboratively with US groups have increased their use, whilst others have tended to decline. The main exception to this are the two groups using the ILLIAC IV, who have spent a long time getting over the experimental stage of using this advanced machine and are just beginning serious work.

Use has been made of 17 (out of 43) ARPANET Server hosts, most of which proved easy to use, with help facilities and message systems. No UK user has stopped using the network because he found it too difficult.

9.3 UK Research Projects using ARPANET

This section describes the ways in which the numerous UK research groups who have had approval during 1976 have used the network to aid their work. They are subdivided into high and low activity groupings in terms of the extent of their use of the network.

On the whole, the high-activity section consists of groups who are well-established ARPANET users; they have got over the experimental stage, have the software and accounts that they need and the confidence to use them over the network. The Blacknest Research Establishment, Durham, Bristol, Salford and Reading Universities, all come into this category. Oxford University Nuclear Physics group have also continued their collaboration with Chicago, Harvard and Illinois, begun in 1975. The British Library Research and Development Division have continued their experimental information retrieval project, using the CANCERLINE system this year instead of MEDLINE.

Two other groups, Culham Laboratory and the Royal Signals and Radar Establishment, have been active in another sense. They have been working hard to establish their own machines as hosts on the network, for the purpose of collaborating with specific groups in the US. These will be described first.

9.3.1 High-Activity Projects

Royal Signals and Radar Establishment (RSRE)

The RSRE at Malvern has a GEC 4080 computer which runs the real-time language CORAL. The latter is being evaluated by the US Department of Defense as part of their project to examine various real-time languages in terms of military requirements. In order to facilitate this evaluation, the RSRE machine has been connected as a host to the ARPA network via one of the UCL PDP9s (see Section 2.5). This connection was particularly useful during March, when a group from RSRE visited the US DoD to give an instruction course on CORAL; the use has continued, however, during the rest of the year.

Culham Laboratory

The Culham fusion laboratory has joint interests with US laboratories in the same field and work has been going on this year to develop software to allow the Culham ICL system 4/72 to be connected as a host to ARPANET in order to facilitate UK/US collaboration (see Section 2.7).

Blacknest Research Establishment

The Blacknest Research Establishment's project to exchange seismic data with centres in the US via ARPANET began in quite a modest way in 1973, when the ARPANET link was first established at UCL. They began by accessing data from the Seismic Data Array Center (SDAC), Washington and then began to set up their own database on the RL 360/195 for access for US counterparts (Blamey,1976). This data is now being accessed by the Vela Seismological Center, SDAC, and soon the National Earthquake Information Service (NEIS).

The latter is the foremost collector of earthquake data from diverse sources. Computers and computer networks are opening many opportunities for improving the quality of data which NEIS collects, and by participating in the transmission of data by computer networks, Blacknest is assisting in a project that will permit near-real time location of important earthquakes.

NEIS have always received data from Blacknest, but rather indirectly. By transmitting data via ARPANET, it may be obtained far more quickly than before. The current data files on the 360 are in a different format from that used by NEIS, so Blacknest are implementing software which will reformat the data in the standard NEIS format, before sending it to the PDP-10 at the Information Sciences Institute, California (ISI).

For their part, Blacknest have been receiving NORSAR bulletins from the Norwegian Seismic array. Unfortunately these stopped being produced at the end of September due to cuts in the Norsar research budget. Blacknest are also now receiving daily bulletins from a station in Ottawa. These are received via the Vela Center, who make the data available in files on the ISI machine. In due course, when NEIS is fully integrated with ARPANET, they hope to get much more quickly the seismic epicentre location data which they currently receive via air mail. Blacknest claim that their attachment to ARPANET greatly aids their research.

Bristol University

Alcock of the Physics Department, Bristol, has been using ARPANET via the RL 360/195 to collaborate with Kelly at the Lawrence Berkeley Laboratory, California (LBL). They are engaged on a joint project to analyse pion-pion to nucleon-antinucleon scattering data. Use of the ARPANET has enabled them to establish programs written originally on the CDC 7600 at LBL onto the 360, where further development has taken place. Total connect time for this was only a few minutes per week on average.

They are now entering a second stage of work. For technical reasons the next stage in the program is best run at LBL where the facilities are better. They are currently developing the program to output the analysis from the 360 into a form for transferring to LBL for further analysis. This part of the work could not be contemplated without the network and the ability to transfer files. It is also immensely useful for Kelly to scan the results of the 360 runs from

LBL and to keep up-to-date log sheets on the status of the runs.

British Library Research and Development Division

The experimental project to access databases and the information retrieval system at the US National Library of Medicine has continued this year,with the modification that the database used has been CANCERLINE instead of MEDLINE. Comparisons of the two databases have been made by the user centres participating in the experiment. These user centres must record details of their searches for BL records in return for terminal access to the information retrieval system. Their pattern of usage of the database has been measured also by the UCL group to corroborate the data furnished by the user centres. These measurements have been discussed in Chapter 8.

Cambridge University Computer Laboratory

Fitch has been engaged since 1973 in a collaborative project over ARPANET with Hearn of Utah U. This work has been concerned with the comparison of algebra systems, CAMAL and REDUCE, produced respectively at Cambridge U and Utah U. Independent comparisons of each other's systems have led to substantial improvements in algorithms and the development of new solution techniques. This project was essentially completed by the end of 1975.

Work has begun on a new project; Fitch has taken a copy of the second version of the LISP compiler over the net and is currently using it. This cooperation does not require much machine time on the ARPANET, but relies heavily on file transfer facilities. It is hoped that EPSS will make two-way file transfer much easier.

They have continued to use MACSYMA as a help in their algebra research. On a number of occasions they have been able to use this unique resource to considerably shorten the development time of new algorithms. Of particular note here is the help they afforded Jackson (Jackson,1976). A package for Petrov classification was implemented on MACSYMA, and as a result file space was made available to their group on MIT-MC.

Durham University

Gault of the Physics Department, Durham U, heads the UK section of the International Particle Data Group. His work is concerned with the development of high energy elementary particle-scattering databases.

During the past year they have continued their collaboration with the Particle Data Group at the Lawrence Berkeley Laboratory. The purpose of the work has been in part to implement an IBM version of the Berkeley Data Management Program, BKYBDMS, on the RL 360/195. This aspect of the work started in March 1976 and a first version 1.0 was running in July. An improved version was implemented in October.

BKYBDMS is a complex program written on a CDC machine and during the implementation periods it was invaluable to be able to test the machine independence of their IBM subroutines by running them on the Berkeley CDC machine. Also the ability to run BDMS at Berkeley from Durham was invaluable.

Another aspect of the work has involved the design of a data encoding language, PPDL (Particle Physics Data Language). The design is being carried out at Durham, LBL and Cal. Tech. As Cal. Tech. has ARPANET access through its LBL workstation, they have been able to compare coding languages directly by setting them up on the most convenient machine and allowing the others in the group to add data and experiment with them.

They have not yet got to the stage of regular data transfer between the two groups using ARPANET, and it is not obvious that they will; however a major future use of ARPANET will be the querying of the databases at LBL for their own book-keeping purposes rather than the transfer of data. The actual data will most likely be exchanged by tape at regular intervals.

Gault summarises his ARPANET activities by saying that the network has played a significant and useful role in their work, and that without it the nature of their project would be altered considerably.

Oxford University

Quirk of the Department of Nuclear Physics, Oxford, leads a group who are working in collaboration with groups at Harvard, Chicago and Illinois. This work concerns experiment 398 at the Fermi National Accelerator Laboratory, Illinois. Muon-scattering data gathered at Fermi is reduced and analysed using software implemented on the RL 360/195.

The collaboration has been working successfully for several years (Anderson,1976). The main difficulty is that use of the RL 360 from a host site in the US depends on three computers (the US machine, the UCL PDP9, and the RL 360) being operational at widely different local times and maintenance periods. This,together with broken or aborted network connections has caused the groups to use the network for swapping data and programs rather than attempting to remotely control the data processing or debugging of programs. The characteristics of the UCL node (described earlier in Section 9.1) have thus affected the way in which the network has been used by these collaborating groups. Local difficulties at Harvard and Chicago have also caused some problems; there has often not been a suitable printer available at the time of transfer of line printer output files.

Nevertheless, work is proceeding satisfactorily and close contact is kept between members of the UCL research group and Illinois to smooth out any problems that occur with transferring files.

Reading University

Hockney's group in the Computer Science Dept, Reading U, is engaged in a project to evaluate and develop rapid elliptical solvers and particle/mesh **algorithms** for parallel architecture computers. The project involves the study of different parallel architectures, in addition to the numerical algorithms.

They have established an association with Lomax and Stevens at NASA Ames, Moffet Field. A collaborative project has been arranged and they hope to transfer their P^3M algorithm to the ILLIAC IV for use in vortex calculations. In return for this they have an account number and file space on the ILLIAC IV (Jesshope, 1976) and on the Ames IBM 360 TSS system. The latter is used for compilation and checkout of the CFD code (a Fortran-based parallel language) before submission to the ILLIAC IV.

Preliminary results using the ILLIAC IV have been disappointing in terms of the computing power available; its performance on benchmark tests was only twice that of a 360/195 rather than ten times (as expected). The great difference between estimated run time and measured run time is due to the restrictions imposed by non-overlap mode processing. In spite of this, use of ILLIAC IV continues to be invaluable to this project since no other parallel computer is accessible.

Salford University

A project to develop and verify algorithms for predicting steady supersonic flow fields is headed by Walkden of the department of Mathematics at Salford U. Such algorithms involve very complex multi-dimensional matrix calculations that are impractical for ordinary serial processing; for this reason, the ILLIAC IV parallel processor machine is being used to develop programs.

Programs are written in CFD and compiled and debugged on the RL 360/195 machine. Only when they are ready to be run are the program files transferred to the Tenex interfacing the ILLIAC IV (I4-Tenex). Until February 1976, lack of an I4-Tenex file transfer facility meant several days delay when a file had to be transferred between I4-Tenex and RL. The introduction of this facility at that time has enabled file transfers to be initiated by the user, and this has been extremely useful. Some difficulties with file transfers have been experienced; the main difficulty is the length of time taken to transfer files from the heavily-loaded 360 via the slow UCL link to ARPANET.

The work carried out so far at I4-Tenex has consisted of small-scale tests of simple computational procedures. The file space allocation of 100 Tenex pages has been insufficient to allow more than one job to be processed at any one time, since the ILLIAC system produces large output files during processing and halts a job if insufficient disk space is available. It is expected that more file space will be made available soon to ease this situation.

Progress has been made towards achieving their short term aim of obtaining run times for the calculation of solutions of partial differential equations associated with fluid flow field problems. The experience gained through performing rather simple tests has also suggested programming strategies for more complicated problems (Walkden, 1976).

The next stage of the work is to test a full supersonic flow calculation program, followed by full scale tests of algorithms and computer strategies involving the complete I4-Tenex system. This will begin in 1977.

9.3.2 Low-Activity Projects

British Library Lending Division

Harley of the British Library Lending Division (BLLD) has been leading a project to investigate the feasibility of an inter-library loan network between the British Library and the National Library of Medicine (NLM). The purpose of the experiments has been to develop and test a system DOCLINE which minimises input and reporting costs and uses a central computer (NLM) for routing, reformatting and address file maintenance.

DOCLINE is now operating satisfactorily and all the 16 regional medical libraries in the USA are now able to channel their photocopy requests to the NLM computer. Requests which cannot be serviced by NLM are transferred to the BLLD file, from which they are retrieved daily by BLLD staff. 7931 requests were retrieved in the 12 months ending June 1976, and 5668 items were supplied. Reports of items supplied, and reasons for non supply of others, are entered from the BLLD terminal daily.

~~Al~~though the system works well, and use of ARPANET was invaluable in getting it established, access to it through ARPANET has been terminated. By mutual agreement the system is now operated via TYMNET, since it has become a routine operation.

Brunel University

The Decision Analysis Unit at the Institute of Organisation and Social Studies, Brunel, is participating in a decision analysis project with five other groups from the UK and US.

The six groups have accounts at ISI, these accounts having been established in order to facilitate the sharing of computer software and the exchange of ideas; Brunel's part of the project has been to develop CALIBA to determine an individual's accuracy in assessing probabilities (Whalley, 1976).

The project has progressed well, but considerable difficulty has been experienced with local telephone lines; extraneous 'noise' on the lines has caused transmission errors during input. Apart from this, Brunel's use of the network has been very successful and they have been impressed by the reliability and speed of the Tenex system at ISI.

Department of Industry

Pinkerton, of the Technology Report Centre, has been accessing the MEDLINE system at NLM in order to investigate developments in information retrieval software and databases in US as compared with those in Britain and Europe. Comparisons have been made between MEDLINE and a similar system RADAB on an ICL 1900 computer at the Department of the Environment, Hastings. Apparently MEDLINE access has some advantages over RADAB; however some MEDLINE symbols are not easy to understand, while RADAB uses full English terms or abbreviations of these. RADAB has been found to be more satisfactory, and in the second half of 1976 the DoI terminal has been used almost continuously for RADAB searches.

Edinburgh University

A new proof-generation system for LCF has been designed and implemented at Edinburgh in the department of Computer Science. The system is implemented in UCI-LISP. A new version of this is maintained at Stanford, and Gordon has been investigating the Stanford version in conjunction with the Edinburgh implementation.

Medical Research Council

It has been proposed by Donchin of the U of Illinois that a test be made of the feasibility of making his laboratory equipment available for use by remotely located investigators via the facilities of ARPANET. Wilkinson of the Applied Psychology Unit, MRC, Cambridge, is collaborating with him on this basis. Use so far of the network has been restricted by lack of a terminal at Cambridge and has been confined to the exchange of comments on graphs of statistical analysis of data. The overall aim is for Wilkinson to view and analyse data over the network. He envisages a series of studies in the general area of the relationship between event related potentials in the EEG and performance.

Ministry of Defence (MoD)

Curry of the MoD has been using the network for collaboration with the US Department of Defense on the High Order Language project currently in progress. The network facilities have proved useful in coordinating this project, and in providing access from the US to the MoD language CORAL.

North London Polytechnic

Survey analysis packages are the subject of an investigation by Rowe of the London Polytechnic Computer Unit. He assigned a student to the project, allotting him only ten weeks for the task, even though he had no prior knowledge of the network, or US contacts. This was far too short to make any real progress, and the exercise was useful only in assessing the problems and experiences of a naive network user (Drinkwater, 1976).

Numerical Algorithms Group, Oxford

The Numerical Algorithms group, of Oxford Computing Laboratory, are collaborating with a group from the Computational Mathematics Division, Argonne National Laboratory (ANL). Their research investigations are seeking to explore the role of computer networking in the development of numerical software, and to establish the problems of remotely installing and maintaining a numerical software library. Use of the network so far has been largely experimental, while they assess the best way to proceed with the project. Also, the necessary software for file transfer has become available at ANL only recently.

Queen Mary College (QMC)

The Computing Laboratory at QMC, under Coulouris, is engaged in research aimed at developing software and system concepts to enable low-cost information processing systems to be more effectively applied. The most important goals are the development of highly-responsive interactive systems, and the position of effective and generalised communication and cooperation between distributed systems. Use of the ARPANET enables them to obtain information about research related to theirs in the US. This may be in the form of online text unavailable in printed form or in the use of systems developed as research projects.

They have been prevented from full two-way collaboration by not having their software files on a UK host; they can receive files from the UK on their PDP11, but they are unable to send files back via the UCL TIP because of limited TIP buffer space. It is hoped this problem will be alleviated by the use of the EPSS development of Chapter 6.

This year, QMC have supplied software to a number of US users. This has been facilitated by ARPANET, mainly be providing minor updates through normal terminal use.

Royal College of Art (RCA)

During the last year the primary use of ARPANET by the Department of Design Research at RCA has been for maintaining contact with US research and running programs of interest. The areas of interest are in architecture and design research, interactive graphics, and artificial intelligence. The host machines used were CMU-10A, Sumex-Aim, Mit-Multics, HARV-10 and Office-1.

The heaviest use was at CMU-10A, involving the BDS system under development by Eastman. It has not been possible to implement this system in the UK, so this use has been a valuable experience.

Thames Polytechnic

Crowe and Avison of the Systems Analysis Division, Thames Polytechnic, are researching into the implementation of relational databases. They have obtained an account on the Unix system at RAND-ISD and are using the INGRES relational database there to study the implementation of logical data structures. This will enable them to gain some insight into the problems of a practical implementation of relational theory. They also have the use of a guest account on the Datacomputer, on which they are doing similar work for the purpose of comparison with INGRES.

University College London, Communications Studies Group (CSG)

"Technology assessment of the interaction between travel and telecommunications" is the collaborative project being carried out by the CSG, the Stanford Research Institute and Bell Canada. All have access to the Office-1 machine, on which a common bibliography is being developed. Pye of the CSG uses the network for developing and accessing bibliographies on communication and telecommunication.

9.4 Conclusions

There can be no doubt that the existence of the UCL link to ARPANET is now invaluable to many of our UK users. This point has been made clear in Sections 9.2 and 9.3. Some of the projects involved are important not only to the groups concerned but also to the world of research generally. This is not an overstatement; for example, the Oxford/Harvard/Chicago/Illinois high-energy collaboration is concerned with the world-wide search for the basic structure of matter. Again, the exchange on a daily basis of seismic data between Blacknest and the US, which previously may have taken weeks, enables immediate detection of seismic disturbance.

Although slightly fewer groups than in 1975 were actively using the network, the work being done was more constructive. More use was made of the file transfer facility, without which many of our users could not have proceeded. The availability of the Post Office EPSS in 1977 will enable more users to work in a mode where they can exchange files with US sites. The availability of the file transfer facility has been essential to almost all the applications which have made really significant progress.

CHAPTER 10: FACSIMILE ACTIVITIES

10.1 Introduction

The present message and text manipulation services offered over ARPANET are extremely important for processing textual data input directly into one of the hosts on the network. However, we have long felt that these services have one element missing; there are no facilities to deal with material which does not originate in machine readable form, and the facilities for processing non- textual material are poor.

We are investigating the input of documents, their transmission, their storage and subsequent retrieval, their manipulation in conjunction with other files, and their final output on devices different from those on which they were input.

This work is not a purely theoretical exercise. As reported in (Kirstein,1976A), during Stage 1 (1975-1976) specific experiments were undertaken involving ISI and ourselves only. These experiments were mainly concerned with the data input, transformation of facsimile files into a canonical form, and their output on different devices. The work done during this period resulted in a possible, but impractical system. However, it has provided us with the necessary information as to how such systems should be designed to meet the requirements.

Amongst others this system had the following shortcomings:

1/ It was heavily dependent on the UCL PDP-9 and used too much processor time.

2/ It required considerable manned intervention and was awkward to use.

3/ Storage and Retrieval aspects were explored only superficially.

Hence the Stage-2 work was planned to overcome these problems. Unlike the previous work, Stage 2 demanded a Systems Model. Our conceptual thinking, and how we have attempted to realise it, are discussed in

Section 10.2. An essential part was to put considerable intelligence into the facsimile terminal. This was accomplished by putting a microprocessor between the standard analog facsimile device and the data network. Both our short term progress and our long term plans for this terminal are discussed in Section 10.3. A key ingredient of our concept is the use of an Information Storage and Retrieval Node (IR). For this purpose we are using the Datacomputer of the Computer Corporation of America. In Section 10.4 some of the operation of that machine is outlined in the light of our requirements. The status of our actual implementation is described in Section 10.5.

One specific activity, which will require further extension, is that of driving all the remote systems through the intelligent terminal. A microprocessor is difficult to program. In order to be able to try out different alternatives more simply, the concept of a "Network Access Machine" was implemented as part of a separate project described elsewhere (Kent,1977). It is a subsystem which runs interactively in the PDP9, and can be table-driven. It allows multi-user control of the FAX terminal on one hand and of the Datacomputer and Message Processor on the other. For small scale deployment of different types of FAX terminal, the NAM is an interesting concept. It permits the considerable control and synchronisation software to deal with all these systems to be centralised, and allows a considerable simplification of the software required in the FAX terminal. The NAM could be used as a networkwide service, supporting all FAX-type terminals. We are pursuing this approach, and will describe it more fully in the next annual report.

For the purpose of demonstrating one specific FAX system implementation, it does not provide any functionally superior facilities. For this reason the project is not described in detail in this report.

Finally, we have been investigating further the economic implications of the type of system outlined in Section 10.2. Several Carriers have announced tariffs for packet switched networks (e.g. the French, UK, Canadians and Telenet). Based on a comparison of these tariffs with the traffic flow implied here, we have tried to analyse the cost of such a solution but our work here is still at an early stage.

However, it has clearly shown the importance of three factors for facsimile, namely:

1. Bulk Rate Discounts
2. Off-Peak Rates
3. The importance of integrating the Information Storage and Retrieval Node (IR) into the network.

The first two of the above are self evident. Even with a reasonable data compression, a full A4 page of text will typically require 25K bytes of data. Even with optimum packet fill, this implies a transaction cost of 12c/page on TELENET and 17c/page on EPSS (at peak tariffs). In both cases call duration, port access and fixed charges are ignored in the above.

For significant traffic, it clearly is important whether the IR is integral to the network or considered outside it. In one case there will be two impositions the transaction charge - one to the IR and one on retrieval; in the other case only one. Clearly the importance of this factor depends on the amount of traffic, the level of charges for IR, the ratio of transaction cost to call duration of fixed costs, the cost of message processing etc. The analysis of the impact of tariffs on this application are so complex, that we have started developing a computer program to study the effects. This program accepts as input:

Matrix of Transaction Tariffs
Matrix of Traffic Volumes
Vectors of Fixed and Time Charges.

Terminal, IR, Message Processing and communication costs will all be considered. This work is still at an embryonic stage, but will be discussed more fully in the next report.

10.2 The System Overview

In principle the system we are striving to develop is shown in Fig 10. The UCL facsimile terminal FAX1 is connected, via the TIP and a character terminal interface, to ARPANET. Also attached are an information storage and retrieval system (IR), a message processing system (MP) and another facsimile terminal (FAX2). There is no reason why MP and IR should be at different nodes - but they need not necessarily be at the same one. We would like to develop a system where the facsimile terminal is sufficiently intelligent that it can read documents with a text header and address list; the facsimile data is then transmitted and stored in IR, while notification of the storage of the document is passed to all addressees via MP. Later the addressees may access IR and retrieve the file on their own facsimile device (F2). At the recipients' option (or possibly at the sender's to mimic "recorded delivery") each addressee may automatically inform the originator that the document was received. As an added feature, we wish to store the facsimile data in a canonical form, so that it may be retrieved from different facsimile devices. There is an important difference between the storage of the facsimile and the textual information. The textual information is short - typically less than 100 bytes : the facsimile data is much longer - typically 25K bytes for an A4 page with optimum data compression. Hence the textual data can be stored in multiple copies - one for each addressee in his "Mailbox". The facsimile information is stored only once in IR - and the path name to retrieve it is known by the addressees from their text messages.

In an attempt to implement this system, we have used, of course, ARPANET as the data network. For FAX1, we attached a microprocessor to an analog 4/6 monitor facsimile device (Section 10.3); this was then connected via an asynchronous data link to the UCL TIP. We found that it was impossible to pass binary data at any reasonable speed through the TIP characters ports. Under certain circumstances (which we understand) the TIP would actually lose characters. While the microprocessor was informed of the fact (by a Bell character being echoed back), it was not possible to tell locally how many characters had been lost. In fact it was quite arbitrary whether one particular character had been transmitted or not. For this reason attaching the UCL FAX machine to the TIP character port had to be abandoned. For simplicity, it was attached instead to one UCL PDP9, via an interface protocol slightly modified from the RSRE one (Section 2.5).

For the MP of Fig 10.1, we used a Tenex (usually ISI) running

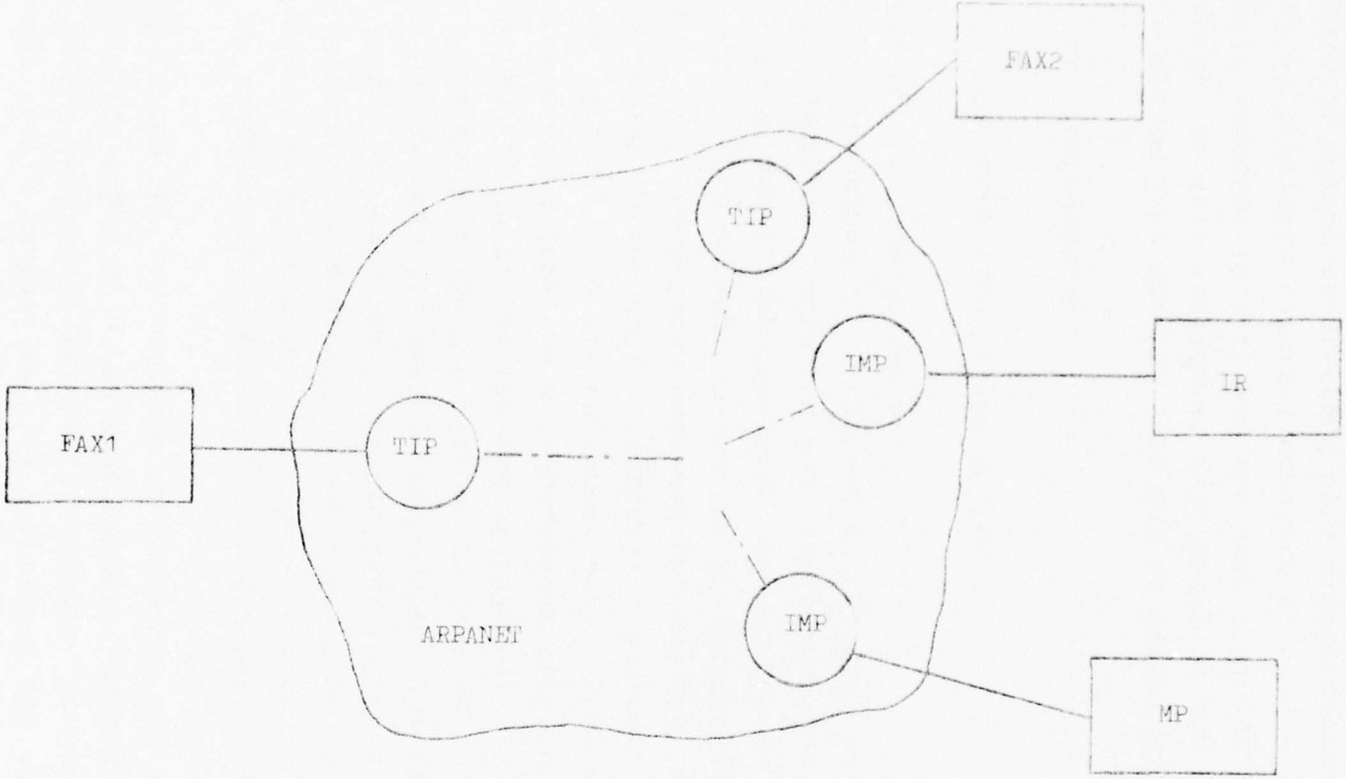


Fig 10.1 Overview of an idealised Facsimile System

the Message system MSG. As far as possible we will, for the time being, use MSG unchanged and try to assess which features in it are really required for our application. For IR the Datacomputer at CCA is a natural candidate, and has been used exclusively. As FAX2 we will eventually use an XGP on ARPANET, as we did previously (Kirstein, 1976A). It will probably be controlled again by a PDP Tenex - PDP11 configuration, because that is the way the software to drive the XGP has been implemented.

An independent research project has been under way for a couple of years to establish a Network Access Machine (NAM). It seemed to be an excellent application of the NAM to use it to automate the functions of controlling the connections to the Data Machine and the Message Processor. The systems overview with and without the NAM with which we have been working is shown in Fig 10.2. At the end of 1976, the text stream was not even going from the microprocessor (µP) to the PDP9, but was going directly to a TIP character port. This was due to software limitations in the PDP9. Eventually, however, there will be a simple dialogue between user and µP, and the latter will automatically open connections to the DC and MP, send the appropriate text to MP and binary facsimile data to µP, and then close down the connections. With the NAM, the paths look the same, but the control functions are exercised by the NAM, which runs in the PDP9, rather than in the µP.

10.3 The User View of the System

10.3.1 Introduction

A system like the one here can be described at many different levels. In this section we will describe the facsimile terminal itself (Section 10.3.2) and the User Dialogue. Although the present facsimile terminal consists of only a slow 4/6 minute facsimile device front-ended by a microprocessor, the system will be unchanged when it is replaced by an all digital system.

The user dialogue has one portion for document transmission (Section 10.3.3) and one for document retrieval (Section 10.3.4). The document transmission dialogue is mainly concerned with the

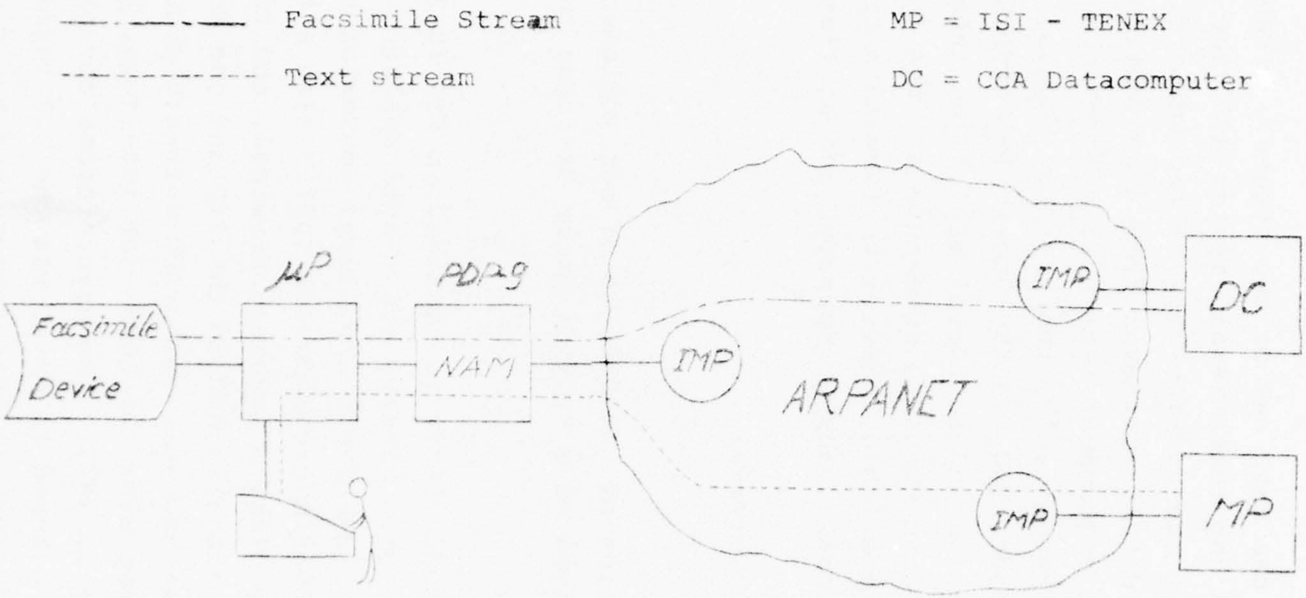


Fig 10.2 Schematic of UCL Facsimile System

composition of a text message to the MSG message system; the binding of the linkages to the Datacomputer and the supervision of message transmission to both is almost transparent to the user.

The retrieval node can have one or two stages. Because the notification comes via the text message system MSG, the user may receive notification of the availability of facsimile files during straight forward text retrieval from a conventional terminal. The facsimile document retrieval must then be a separate exercise. Alternatively, the user may access, via a somewhat simpler dialogue, MSG via the facsimile terminal. In this case the notification and retrieval will be an integrated operation. We do not have space to enumerate all the facilities of MSG - they are all available, however to forward notifications to whole lists of addresses, answer messages and similar functions.

10.3.2 The Facsimile Terminal

Prior to a description of the details of our implementation, an overview will be given of a typical users dialogue in our model.

The facsimile terminal (FAX1) is sketched in Fig 10.3. Physically it consists of a Plessey 4/6 minute analogue facsimile device (FAC) attached via a simple A/D 2 level converter to a 24K 8 bit word INTEL 8080 microprocessor (μP). Also attached at present is a floppy disk, a keyboard terminal, and two asynchronous channels are attached to the TIP and one to the PPDP9 - because we have had some temporary software problems in multiplexing two streams over the PDP9 - μP interface, but this will be remedied early in 1977. Two asynchronous format channels were used, because we planned to enter via the TIP terminal ports, these would only support that format, and would only have one virtual connection over each port. With our abandonment of this approach, the μP now connects straight to the PDP9; it would now be natural to use a single synchronous interface and also to multiplex the control, text used and facsimile data. This we will do eventually. Conceptually, therefore, Fig 10.3 shows both the present and planned configuration

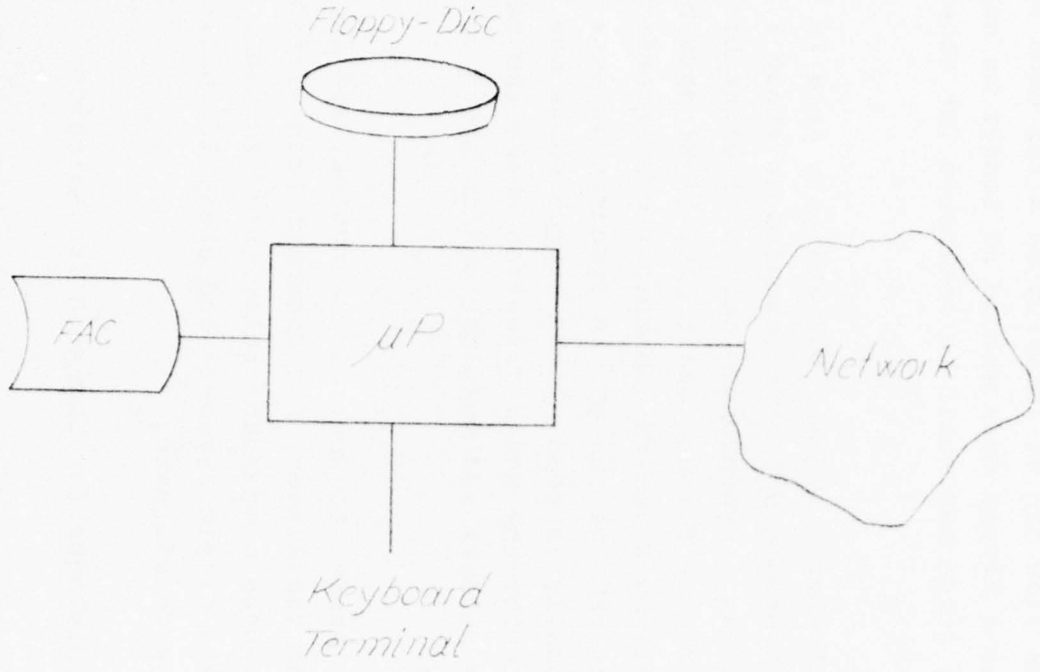


Fig 10.3 Schematic of UCL Facsimile Terminal

The standard 4/6 minute analogue facsimile devices must first synchronise with each other, which takes up to 15 seconds, and then read a full page at a time without interruption. It is impossible to guarantee any specific data throughput via a packet switched network like ARPANET, therefore data is staged always via a floppy disk. At a later stage we can plan to replace the analogue facsimile device by a digital one, in which case the data flow can be interrupted after each scan line. In this case a floppy disk should no longer be necessary. However, a backing store may still be desirable for other reasons (see Section 10.5.2).

As described (Kirstein, 1976A), the analogue data is passed through a simple threshold detector, sampled at about 2.4K bps, and the data output as a synchronous series of eight bit bytes, though with asynchronous format (start bit, eight data bits, stop bit). This allows standard communications interfaces to be used between the FAC and the μP. A synchronous data adaptor will be used eventually instead of the asynchronous one. This is more appropriate for the usual situation where the terminal is remote from the network and must use medium speed data transmission facilities.

The keyboard is needed to control the system, and add addressing and document naming facilities, for document transmission and retrieval. In our case a standard teleprinter is used. In an operational system a simple keyboard and cheap 20 character wide printer would be quite adequate.

10.3.3 The User Dialogue for Document Transmission

A typical user dialogue for transmitting a 5 page document is shown in Fig 10.4. A commentary on this dialogue is given below :

1. Start-up

The facsimile system resides on a floppy-disk, and it is loaded into the memory by typing an 'L' on the user console. It then initialises itself and prints out the title and

```
                 DIALOGUE                                COMMENTS

$ L<CR>

*** UCL FAX-MESSAGE SYSTEM VERSION X.04 ***
YOUR NAME:  KIRSTEIN<CR>                          1. Logon
PASSWORD:  XYZ<CR>
LOGIN O.K.

... TYPE ? CR  FOR HELP                           2.  Optional User
                                                      Guidance

$C <CR>

COMPOSE A FAX MESSAGE FILE
TO:  KIRSTEIN at ISIA <CR>                        3.  Addressing and
CC:  YILMAZ at BBNE <CR>                              naming the document
NAME:  FAX DOC.1 <CR>
NUMBER OF PAGES:  5 <CR>
INSERT THE FAX DOCUMENTS AND TYPE YOUR            4.  Document Feeding
   MESSAGE

THIS IS A TEST MESSAGE SENT BY FACSIMILE
   SYSTEM AT UCL <CR>.                            5.  This message is
                                                      composed

↑Z                                                6.  Sending the message
TEXT IS O.K.         KIRSTEIN AT ISIA             7.  Delivering confirm-
                     YILMAZ AT BBNE                   ation of text
----
----
----
FAX IS O.K.                                       8.  Delivery confirm-
                                                      ation of facsimile
                                                      data
```

Fig 10.4 Dialogue for sending a document
(Characters input by user are underlined, Carriage Returns indicated by <CR>)

the current version number, and asks the user; his name and password to perform the login procedure. If the login is successful it prints out the FAXSYS prompt characters '<-' and waits for a command input as shown below. For subsequent facsimile operations, this portion of the dialogue will not be repeated.

2. Help

<-

If a ?<CR> is typed, a Help file is produced on the console, giving information on the use of FAXSYS commands.

3. Addressing

This indicates the addresses of notification about the availability of the document and the short text message. The Document's name is chosen to be informative; it will be printed out later for the recipient, but is <u>not</u> used to identify the document (see Section 10.3.5). The identification number is composed at this time, so that the complete subject field for MSG is formed.

4. Document Feeding

At this point the facsimile document is inserted into the machine for scanning. While the text message is being typed in, the scanned data is processed and stored on the floppy-disk.

5. Message Composition

The FAXSYS provides three editing characters which may be used to correct errors during this period. These are:

↑A	to delete the last character
↑W	to delete the current line
↑N	expunge message

Here ↑A means CONTROL and A pressed simultaneously etc. ↑A and ↑W can be used repeatedly, and ↑N allows the deletion of the entire message for a fresh start.

While different control characters could have been used, since they are processed in the micropressor, it was thought desirable to be consistent with the MSG subsystem of the Tenex. For the same reason two other commands are used:

↑R	type current line
↑S	type current message

6. Message Transmission

This character signals the message should be transmitted. At this point all the standard MSG facilities for immediate, delayed or queued transmission may be invoked, since MSG will be used directly. The whole message can be sent in one burst at this time.

7. Text Delivery Confirmation

These replies will come directly from MSG, and thus all their options, such as non-delivery, queued delivery, etc., may arrive.

8. This is the successful response code from the Datacomputer as interpreted by the UP.

10.3.4 The Dialogue for Text and Facsimile Retrieval

Since the text portion of the facsimile system uses the MSG system, it can be retrieved on any alphanumeric terminal. Thus the recipient of a text message may access MSG on the Tenex which he uses for text messages in the normal course of his work. In this case his dialogue may well follow the form of Fig 10.5. After logging in (1) and invoking MSG (2), he is informed that FAXDOC 1 of length 5 pages, is waiting for him at CCA. This header has been devised from the NAME (FAXDOC. 1) and the length (5) input at phase (3) of Fig 10.4. The number in parantheses is a Datamachine reference to the file. He may still wish to

```
@0 86 <CR>
 Trying ...
Open
ISI-TENEX 1.34.10 ISI-SYSTEM-A EXEC 1.54.10
LOGI KIRSTEIN  (PASSWORD) (ACCOUNT)
JOB 32 ON TTY 11       10 JAN 77     16:20
PREVIOUS LOGIN          4 JAN 77     04:31
[YOU HAVE NEW MAIL]
MSG -- Version of 1 April 1976
- + 29 4 Feb KIRSTEIN AT ISIA   - The Document FAXDOC (5 pages) is
                                  available at CCA (77021009255)
LAST READ: 4 Jan 77 04:36:25:  28 Old Mesg.  29  Mesg 28 disk pages.
Old not-examined msgs. exists

Type 29  <CR>
Mail from ISIA rcvd at 10-Jan-77  0952
Date:  10 Jan 1977
From Kirstein at SRI-AI
Subject:  The Document FAXDOC (5 pages) is available at CCA
                                             77021009255
To:  KIRSTEIN AT ISIA
cc:  YILMAZ  AT BBNE

THE FACSIMILE DOCUMENT NAME IS AVAILABLE AT CCA.  IT MAY BE
REQUESTED AS 77021009255.  IT IS OF LENGTH 5 PAGES.
        ------
THIS IS A TEXT MESSAGE SENT BY THE FACSIMILE SYSTEM AT UCL
```

Fig 10.5. Dialogue for accessing the Message Processor
(Character input by user is underlined)

see if there is any interesting text for him, and go through the steps of Fig 10.5. It is shown that the first part of the message has also been composed automatically from the sender name, the document name and the document length entered in phase (3) of Fig 10.4. At this stage the user can call all the processing facilities for MSG, and can file, delete, forward or move the text message.

To retrieve the actual document, he must use his facsimile terminal (FAX1) of Section 10.3.2. Here the dialogue follows much the same pattern of Fig 10.4, as is illustrated in Fig 10.6. An explanation of the dialogue is given below:

1. This is the same login to the facsimile terminal as in Section 10.3.3.

2. This connects to MSG. At present only ISIA can be accessed: eventually "$M HOST" will connect to MSG on a specific host.

3. Now the user is connected to MSG at ISIA, and all normal MSG commands could be followed. Thus if the user had not yet accessed MSG before through an ordinary terminal, he would have use "H" for listing headers, "F" and "D" for forward and delete etc. By prefixing a command with $, to indicate a communication with the terminal system, in this case he requests facsimile file 29 to be retrieved.

4. The facsimile terminal indicates the title of the document and its length - retrieved from ISIA. It also obtains the pointer to the file in the CCA directory (770210092551). Its computation is discussed below.

5. On confirmation that the file of (4) is the correct one, it is retrieved and kept on disk. If $P had been input in (3), the file would also have been output on the facsimile terminal.

```
0L <CR>

UCL FAX-MESSAGE SYSTEM VERSION 4
YOUR NAME:    KIRSTEIN <CR>
PASSWORD:     XYZ <CR>
LOGIN O.K.


0M <CR>

MSG  ....  VERSION OF APRIL 1 1976
Last Read 10 Jan 1977 29 old msg 28 disk Pages


04R 29 <CR>

TRYING
KIRSTEIN  FAXDOC  (5)  770  210092551 [CONFIRM] <CR>


....FAX MESSAGE RECEIVED
0E <CR>
```

Fig 10.6 Dialogue for Retrieving Facsimile Files from the Datacomputer

6. Having printed the facsimile message, user gives the 0E <CR> command to exit from MSG and close down the network connection to the Datacomputer.

It should be noted that $M actually connects the facsimile terminal to MSG. For this reason the dialogues of Stages (1) - (3) of Fig 10.5 could have been replaced by Stages (1) and (2) of Fig 10.6 - and the unread message title 29 would still have appeared printed on the terminal.

10.3.5 Generation of Message Code Numbers

The link between the text and the facsimile messages is formed as follows:

During the LOGIN INTO TENEX ... period the system obtains the time and date of Login, which has the format

DAY - MONTH - YEAR   HOURS - MINUTES - SECONDS
(e.g. 10-JAN-1977   09; 25; 51)

This information is re-arranged to form a twelve digit number which is unique to the message being sent. For the example given above this code number becomes:

770210092551

It is essential that a unique number be formed, so that messages can be retrieved uniquely.

This code number is added into the text message header before transmission. This code is retained and used with the KIRSTEIN to store the corresponding facsimile file at CCA. At the end of the operation detailed in Section 10.3.3 the Kirstein node now contains a new entry as demonstrated below

```
FACSIMILE
          ISI
              KIRSTEIN
                 ...
              770210092551
```

If certain sites desire to be particularly secure, they may set a site password at that point in the path. More usually, only the users will be given password protection. Thus to access a file FAXDOC by a user, KIRSTEIN whose MAILBOX is at ISIA, and whose password is SECRET would be:

```
LOGIN FACSIMILE
% LOGIN. KIRSTEIN (SECRET). FAXDOC
```

If we desire extra security, so that the facsimile system itself and the site ISIA have protection passwords FAX and SITE, the equivalent command would be

```
LOGIN FACSIMILE ('FAX')
%LOGIN. ISIA (SITE). KIRSTEIN (SECRET).FAXDOC
```

Not only will notification of the existence of a file be sent to an addressee's mail box, but also a READ priviledge will be given to that addressee as the file is created. Thus the addressee will later be able to CONNECT his facsimile terminal to the file via his node.

10.4 Storage and Archival at CCA

10.4.1 Characteristics of the Datacomputer

The Datacomputer is a shared large-scale data storage utility offering data storage and data management services to other computers on ARPANET. The system is intended to be used as a centralised facility for archiving data, for sharing data, among the various network hosts, and for providing inexpensive on-line storage. The Datacomputer is implemented on dedicated hardware, and comprises a separate computing system specialised for data management. Logically the system can be viewed as a closed box shared by multiple external processors and accessed in a standard notation called Data-Language. The system is provided with an Ampex Terabit store of 4×10^{11}bits. While this is mainly used for seismic data, it does contain software and hardware ideal for archival storage. Thus it is an excellent vehicle for our experiments.

In Section 10.4.2 we discuss briefly the data structures set up in the Datacomputer, and the way files can be accessed. In

Section 10.4.3 the Data-Language used to drive the machine is mentioned. Little detail is given in this section of how user facilities are translated into Data-Language dialogue.

10.4.2 Directory Structure, Privacy and File Access

The general structure of files in the Datacomputer is a tree with nodes nested to any depth. To access any specific node, it is necessary to ANCHOR the user to a node higher up the tree, and to ATTACH to the node through a specific path. It is possible to assign LOGIN in privilege via password protection, and CHANGE privilege at any node. It may also require passwords to travel down the tree through any node. Finally each file can be printed to only through a node, and can be given READ, WRITE and APPEND privilege also under password control. If no privacy is necessary, then it is necessary only to set up one node. All users who are given access to that node would also be able to access any file in the directory. Password protection could still be applied to the files, thus it would be possible, if the sender knew the passwords of all addresses, to give READ privilege only to them - but he could then read all the addresses files! In order to give a much greater degree of privacy, it is possible to use the provisions built into the Data-Language. A special FACSIMILE directory is formed, with nodes corresponding to the notification box site and User name.

The FACSIMILE directory on the Datacomputer is a tree, with the FACSIMILE node at the top, site nodes under FACSIMILE, user nodes subordinate to the site nodes, and facsimile files beneath user nodes. A user's name space may be further divided into sub-directories of arbitrary depth. Pictorially this is illustrated in Fig 10.7.

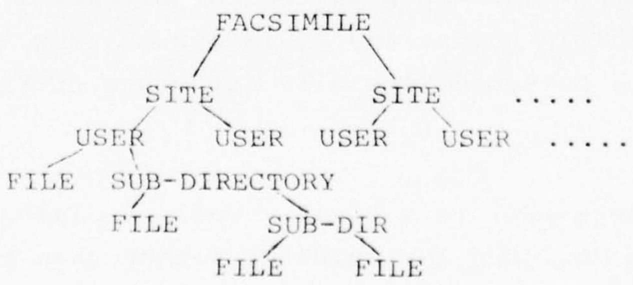


Fig 10.7 Directory Structure for Facsimile File

Any node in the directory can be accessed via a generalised means for specifying directory paths. The mechanism consists of an anchoring point, for name references, and a path name, which is the sequence of node names, starting at the anchor, defining the desired branch of the directory tree.

There are three contexts: the Top Context, the Attach Context, and the Connect Context. The Top context anchors the pathname at the facsimile node and is used primarily for referencing the other user nodes. The Attach Context is a path which is set by thefacsimile system automatically at the beginning of a session, and ends up with a user node. The Connect context is a path which is initially set as the Attach context and terminates in a sub-directory.

10.4.3 Use of the Datacomputer

The Datacomputer is designed so that external computers may transfer data between themselves and the Datacomputer (DC) by a two stage process. First they connect to one stream in the system by the standard "Initial Connection Protocol" of ARPANET followed by a LOGIN. This connection must be active throughout the data transmission. It is used to control the data transmission and set the appropriate transmission parameters: we will call this stream the "Control Stream" (CS). When the control stream has been set up, and the appropriate path to the user node followed, a second data stream (DS) is opened by CS. All data transfer is through this data stream (DS). An example is given in Fig 10.8 of the commands required to log in and retrieve a file FAXDOC sent to KIRSTEIN. It is assumed that the retriever is using a TIP and wishes the data to come onto port 262146 of the TIP. In the example the passwords PASSWORD and SECRET needed to access the FACSIMILE and KIRSTEIN nodes, and it is assumed the ISIA node does not require a password (see file structure of Fig 10.7).

The Data-Language (DL) is a clearly defined dialogue, with a numeric portion designed for computer rather than human processing. However, Fig 10.8 shows it is possible to drive the

Datacomputer from keyboard terminals. If the character ports on the TIP are used, two terminals and their ports are required - one for the control stream and one for the data stream; the TIP software does not permit two virtual circuits to be activated simultaneously from one terminal port.

The Data-Language is very awkward for human users,and verbose - one example of the dialogue for even a simple operation such as Login is shown in Fig 10.9. Here the underlined line is the user's command, the others are the Datacomputer replies. A complete description of the whole language is shown in (CCA, 1976). One reason the the Data-Language dialogue, illustrated in Fig 10.9, looks so verbose is just that it is well structured for machine processing. The first set of figures is Data-Language response, the second the data/time and the third data is human readable format and a verbal description of the commands. While the Data-Language's responses may be verbose for humans, they are excellent for synchronising activities with another computer or intelligent terminal. They gave just the right information on successful completion and error conditions.

@ H 31 <CR>	Connect to socket of Datacomputer
@ R F S <CR>	
@ I C P <CR>	
LOGIN FACSIMILE ("PASSWORD"); <CR>	Log on to Datacomputer and connect to FACSIMILE node; pass to KIRSTEIN node and prepare to retrieve.
OPEN % LOGIN. KIRSTEIN (SECRET). FAXDOC; <CR>	
CONNECT FAXPORT 262146; <CR>	Connect file to TIP data socket.
FAXDOC = FAXPORT;<CR>	
CLOSE FAXDOC; <CR>	Close File
DISCONNECT FAX PORT; CR	Disconnect data socket
@ C <CR>	Disconnect from Data-computer

Fig 10.8 Data-Language Commands to Retrieve and File

```
@ H 31 <CR>
@ R F S   303 <CR>                          Connect to a free
@ I C P <CR>                                socket at the datacomputer


Trying

Open
- - - -
- - - -

(Data-Language responses)
- - - -
- - - -

;0031   770110135925   IONETI:   CONNECTED TO LONDON-TIP-1200000
;J150   770110135926   FCRUN:    V='OC-1/01.13' J=1 DT='MONDAY,
                       JANUARY 10, 1977 08:59:26-est' s='CCA'
;0041   770110135926   ONCINX:   DATACOMPUTER GOING DOWN IN 2808 MIN
;J200   770110135926   RHRUN:    READY FOR REQUEST
.I210   770110135926   LAGC:     READING NEW DL BUFFER

LOGIN FACSIMILE ('PASSWORD'); <CR>

- - - -
- - - -

(Data-Language responses)
- - - -
- - - -

;0032   770110140028   ASPRIN    HOST='LONDON-TIP' SOCK=1200000
;0033   770110140028   ASLOG:    USER='FACSIMILE'
;J209   770110140028   RHRUN:    EXECUTION COMPLETE
;J200   770110140028   RHRUN     READY FOR REQUEST
.I200   770110140028   LAGC:     READING NEW DL BUFFER
```

Fig 10.9   Datacomputer Login Procedure

10.5   The UCL Implementation

In Section 10.2 we have presented an overview of what we are trying to achieve. In Section 10.3 we have outlined how the user should see the system and in Section 10.4 how the Datacomputer can be used for information storage and retrieval with appropriate privacy. Now we will describe briefly the UCL implementation. It is clearly inappropriate to give much detail here. In Sections 10.5.1 and 10.5.2 the actual hardware configuration and the software structure are mentioned briefly. In Section 10.5.4 the status of the implementation is reviewed.

10.5.1 The Hardware Configuration

The UCL hardware (FAX) has been described already in Section 10.3.2. In our earlier experiments we attached the facsimile terminal to two separate asynchronous terminals of the TIP; this was to allow both the control and data streams of Section 10.4.3 to be set up. In fact we found that the TIP port could not be used for blockmode data at any speed. The flow control is very poor, and it proved impractical to devise a scheme for overcoming an arbitrary loss of a character under specific conditions. For this reason our work up to the end of 1976 was based on the facsimile terminal controlling all the dialogue, but the actual paths had the data flowing over a TIP character port path for control, and the PDP9 for data. We are modifying the PDP9/FAX software interface to permit the streams to be multiplexed over the channel. The configuration being used is illustrated in Fig 10.10.

10.5.2 The FAXSYS software

The FAXSYS software can be sub-divided into 8 basic elements. These are:

1) The Real Time Executive
2) Facsimile Terminal Drivers
3) Console Process
4) Floppy-disk Handler
5) Command Decoder
6) Communications Controller
7) Network Interface
8) Compression-Decompression process

During the Stage-2 of the facsimile project, most of the time was

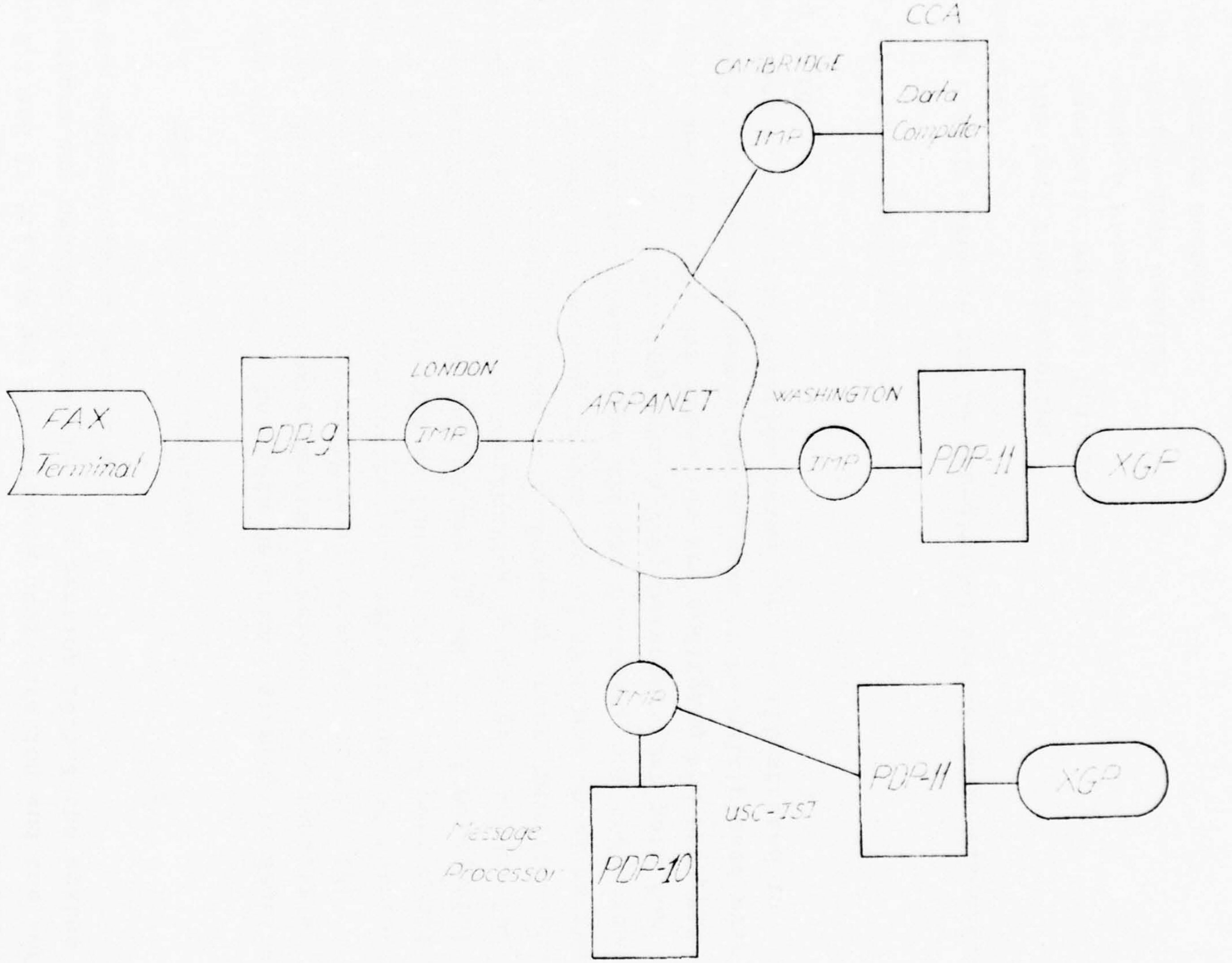


Fig 10.10 The UCL-ISI-CCA-ARPA Experimental facsimile configuration

spent in developing the above components. Because of the need to handle staging to disk, and a desire to overlap processes and operate in a duplex fashion, a significant number of tasks can be concurrent. Thus there is need for a small real-time executive (1) whose main functions are to schedule the operation of the software modules, act as a traffic controller, and maintain an efficient data flow between different processes. The executive provides the basic building blocks on which the software can be developed. System activities are reduced to a small number of tasks which are activated on request to the executive. Process scheduling is purely asynchronous with all processes being written so that suspension can take place at any time, in favour of a higher priority process.

Modules 2 - 5 are standard ones which require no comment. The Communications controller (6) consists of the modules which pass data in two streams via the PDP9 to ARPANET. This is the segment which had to be changed for the attempt to pass data through the TIP character ports.

The Network Interface (7) is a primitive Network Access Machine (NAM). Its main functions are the opening and closing of the network connections, and interfacing with the remote subsystems (e.g. MSG). In the case of the Datacomputer access, it generates the necessary Data-Language commands to perform a given function. This module is complex, and is one that needs modification as the system functions are to be expanded.

The Compression-Decompression process takes the raw data coming from the facsimile scanner, and compresses it by a form of run length encoding (Kirstein, 1976A). It also formats the data into a canonical form, so that a facsimile device with different characteristics could output the data. In the reverse direction, this module operates in the reverse manner.

10.5.3 The Present Status

Most of the user functions of Section 10.3 were possible at the end of 1976. The connection to the network was still in a rather awkward form, as mentioned in Section 10.5.2, but this will be remedied early in 1977 - by passing entirely via the PDP9.

Not all the facilities of Fig 10.5 are yet available. The principal shortcomings are:

1. Files are restricted to 1 page
2. Path Name notification is in the Message Text not the Subject Field
3. There is no user-supplied File Name
4. There is only one common FACSIMILE node used for all files on the Datacomputer, so that there is no privacy from other users
5. Message composition editing commands are somewhat poorer.

However, all basic elements of the system are operational, and no fundamental reasons are known why the whole system should not be implemented as planned.

CHAPTER 11: CONCLUSIONS

It would be nice to be able to present a number of conclusions and pinpoint accurately whence, in this report, the conclusions have been drawn. In most cases, however, this is not possible. The conclusions follow from the integration of the different activities. We indicate below, however, in which chapters research is described which leads to the conclusions:

1. With complex software for switching data, measurements are essential to detect inefficiencies in throughput and operation in order to indicate areas for improvement (2,3,8)

2. With complex protocols for data transmission, it is essential to combine theoretical analysis, simulation and measurement of performance. Unanticipated interactions between different levels of protocol can degrade performance. An example is the duplication of sequence control between lower level ARPANET protocols and the TCP (5,7,8).

3. In order to measure systems performance as above, a wide variety of measurement tools are required. These include traffic generators and line level measurements. Once the tools are developed, they can be applied to many different networks or interconnections of networks with relatively little modification (2-8).

4. The connection of server hosts to networks poses many problems. The UCL approach has been to attach hosts by a front-end processor in which virtual calls are matched from a network to an access scheme supported by the mainframe. Not only has this technique been shown to work well, with many different computers, but a general approach to this technique (SWITCH) has been developed during 1976; it has been applied successfully to several host systems.

Our approach would work well for network to network connection and can be extended to support an X25 interface (2,3).

5. For useful attachment of hosts to networks, not only must mappings be done at call level, but also at higher levels such as the interactive terminal and bulk transfer support levels. In the first there are sometimes character set mapping difficulties; in the latter there are sometimes real discrepancies between the underlying assumptions of the networks and the server host systems. These mappings have been done reasonably successfully in the UCL projects, but there are sometimes some awkwardnesses in use (2,3,9).

6. Interworking between different networks poses many problems and various solutions have been proposed. UCL believe that mapping at the virtual call level is a very promising approach for interconnection. The SWITCH system is fully applicable to this problem. The approach is fully compatible with using X25, with some modifications, for the connection of networks (4,5,7).

7. For the effective use of computers through more than one network there may need to be mapping of terminal and bulk transfer facilities. SWITCH provides the mechanism for such mapping and it has been demonstrated in certain environments (4,6).

8. There is a strong interaction between the techniques used for provision of services through concatenated networks and the environments for which the networks are designed. There may well be a real divergence here between the requirements of PTT networks and those of military ones using techniques such as packet radio and requiring numbers of gateways for high availability (4,5,6,7).

9. The techniques discussed in this project may encompass a broader area of communication traffic than originally envisaged. Examples are digitised speech, message traffic and facsimile (4,7,10).

10. We have been amongst the first groups to have used the BBN XNET debugger to do real-time program monitoring and debugging via a tenuous link. The software in the UCL PSP Gateways (Chapter 7) is controlled on-line from US Tenex sites via ARPANET for debugging purposes. This approach to program development has been found to work well (7).

11. The development of mappings, at call level and higher, between external and internal conventions as in SWITCH has another corollary. The internal conventions need not be restricted to one computer but can apply across several computers (some of which may even be remote). This erodes the dividing line between a "single gateway" or a "gateway system". This "gateway system" may itself be a computer network. The whole process between "single computer" , "computer networks" and "distributed computing" is hereby becoming blurred (2,3,4).

12. In several projects at UCL, high level languages have been used on small computers via cross-compilation (Babbage on the PDP9/RL 360 and BCPL on the PDP11). There is still no clear indication that the approach is really sufficiently convenient in operation, or efficient in code produced, to be really viable. Even in our own group the opinions still differ. Some complain that the awkwardness of having to use several systems together outweigh the advantages in cross-compilation; there is also clear evidence that the code produced is larger and slower by a sufficient factor that manual recoding may be necessary and too much extra memory required. At the least a high-

availability large central host and local access computer are required. The file peripherals on the latter must be fully compatible with the machine for which code is being produced, or that machine must also be on-line (2,5,7,8,10).

13. The principle UCL activities in the Packet Satellite Project have been to develop tools for traffic generation and data acquisition. Measurement tools have been developed, but in the absence of Gateway computers at other sites, no real UCL measurements have been made in 1976 (7).

14. The user level measurements are giving significant information on patterns of network usage, on the usage characteristics of different applications, and of the performance of all the subsystems available. The measurement tools should be run consistently and continually, and preferably cover all types of access, for highest credibility (8,9).

15. Complex traffic generators, data acquisition and measurement data reduction all require sizeable amounts of computer software. Careful attention to flexibility in the design stage of these tools pays off in the ease with which they can be extended to new, unanticipated applications (2,5,7,8).

16. For efficient network usage, facilities such as message systems and file transfer are essential. It is essential also to provide adequate user information on status of connection and jobs, and on use of facilities (9).

17. Usage of a network for collaborative work depends heavily on the availability for each of the collaborating groups involved of a local attached host (9).

18. The facsimile services envisaged would be of great benefit to ARPANET users- their potential are far greater, of course, in wider contexts. They could be provided with the facilities now available on ARPANET. Initial economic studies look promising for private networks - wider development on PTT networks is closely bound with PTT tariff and policy consideration(10).

19. However well information status facilities and basic network services are provided, it is essential that personal user support be available. Clearly no new service could be introduced without considerable attention to documentation on user facilities. However, our experience is that almost every user (including these inside our group!) comes to require personal intervention of our user liaison services at some time. The amount of effort required to solve the problems is variable, it may be a small piece of information, it may require a few messages to other sites, it may require a large UCL software development, or it may be insoluble and suggestions to circumvent the problem may be needed. However we believe that without a specialised liaison officer (knowing something about ARPANET, UCL and RL etc.) most applications would make significantly slower progress; probably the majority would give up deeming their problems insoluble (2,3,5,7,9).

20. Given the level of technical and user support provided in the UCL ARPANET project, very significant and useful work is now being done collaboratively. Some of this work will be transferable to commercial services by 1978, others will not because of the unavailability of the relevant US sites via international commercial services (both for economic factors caused by low average usage and by local political constraints) (9).

REFERENCES

ANDERSON HL (1976) Physical Review Letters 37, 4

BEELER M, et al (1975) Gateway Design for Computer Network Interconnection, BBN, Cambridge, Mass.

BLAMEY C (1976) The Establishment of the Blacknest Seismological Database on the Rutherford Laboratory's 360/195 Computer, Blacknest Research Establishment, Brimpton, Reading.

BROOMFIELD CT (1975) Packet Switching - the Experimental Packet Switched Service, Computer Communications Review.

BUTTERFIELD S (1974) The Satellite IMP for the ARPA Network, Proc. 7th Hawaii Int. Conf. on Systems Sciences, 1974

CCA (1976) Datacomputer Version 1 User Manual, Datacomputer Project Working Paper No.11, Computer Corp. America, Cambridge, Mass.

CCITT (1976) X25 Recommendation, AP V1 - No. 55-E, Geneva

CDCA (1976) General Information on the Use of INTERCOM, ULCC Bulletin B2.5/1

CDCB (1972) The Control Data 200 User Terminal Programmers Guide, ULCC document 49/72

CERF V and KAHN R (1974) A Protocol for Packet Network Intercommunications, IEEE Trans Communications, Vol.COM-22, No.5, p.637

CERF V (1976) Report in preparation, ARPA, Arlington, Virginia

CERF V (1977) Report in preparation, ARPA, Arlington, Virginia

DRINKWATER B (1976) Use of Arpanet for Survey Analysis and Building Description Systems, Dept. of Mathematics, Polytechnic of N. London.

EDGE SW (1977) Evaluation of Optimistic and Conservative Credit Return Host-Host Protocols with Respect to Flow Control, Indra Note 594, Department of Statistics and Computer Science, UCL

HIGGINSON PL and HINCHLEY AJ (1975A) The problems of linking Several Networks with a Gateway Computer, European Conference on Computer Communications Networks, UK, p.453

HIGGINSON PL and HINCHLEY AJ (1975B) SWITCH - An Overview, Indra Note 414, Department of Statistics and Computer Science, UCL

HIGGINSON PL (1976A) Commissions Restructuring of ESP 25, CEPT - ESP 32

HIGGINSON PL (1976B) EPSS and X25 - A Personal View, Indra Note 569, Dept. of Statistics and Computer Science, UCL

HIGGINSON PL (1976C) First Specification for SWITCH FTP, Indra Note 580, Dept. of Statistics and Computer Science, UCL

HIGGINSON PL (1976D) Asynchronous Terminals on Euronet, Indra Note 590, Dept. of Statistics and Computer Science, UCL

HIGGINSON PL and HINCHLEY AJ (1976E) Plans for Supporting File Transfer Protocols of the UCL Gateway, Indra Note 531, Dept. of Statistics and Computer Science, UCL

HIGGINSON PL (1976F) Segment Loading on the PDP-9, Indra Note 461, Dept. of Statistics and Computer Science, UCL

HINCHLEY AJ (1976A) Tasking and Segmentation in the PDP-9 New System, Indra Note 414, Dept. of Statistics and Computer Science, UCL

HINCHLEY AJ (1976B) SWITCH - Revised Specification, Indra Note 571, Dept. of Statistics and Computer Science, UCL

HINCHLEY AJ (1976C) SWITCH/QUEUEING SYSTEM RELEASE, Indra Note 583, Dept. of Statistics and Computer Science, UCL

HINCHLEY AJ (1976D) Introduction of Gateways into the Satellite Project - Issues Arising and UCL Plans for Working with Gateways, Indra Note 537, Dept. of Statistics and Computer Science, UCL

JACKSON DM and ALCLIUNAS R (1976) Decomposition Based Generating Functions for Sequences, Dept. of Combinations and Optimisation, University of Waterloo

JESSHOPE CR (1976) Arpanet and the Illiac IV, RCS 53, Department of Computer Sciences, Reading University

JOHNSON JO (1976) Indra's PDP-9 System Performance Improvements, Indra Note 595, Dept. of Statistics and Computer Science, UCL

KENT S (1976) Suggestions for a FAX-NAM Implementation, Indra Note 544, Dept. of Statistics and Computer Science, UCL

KIRSTEIN PT (1974) UCL ARPANET Project Annual Report, 1-7-73 to 30-9-74, TR-17, Dept. of Statistics and Computer Science, UCL

KIRSTEIN PT (1976A) UCL ARPANET Project Annual Report, 1-10-74 to 31-12-75, TR-27, Dept. of Statistics and Computer Science, UCL

KIRSTEIN PT (1976B) Gateway Experiments in Connection with the Packet Satellite Project, Indra Note 574, Department of Statistics and Computer Science, UCL

KIRSTEIN PT (1976C) X25 Based Experiments in the Context of the Packet Satellite Project, Indra Note 578,

LINKABIT (1976) Packet Satellite Program Plan, Linkabit Corp., San Diego, California

LIN-NAN LEE (1976) Fixed - Aloha Demand - Assignment A Multiple Access Communication Protocol with Fixed - TDMA and Slotted - Aloha Features, Linkabit Corp, San Diego, California

LLOYD D (1976) A Note on the Role and Requirements of Gateway Computers, Indra Note 536, Dept. of Statistics and Computer Science, UCL

PO (1976) PO-TG16 Appendix E

POUZIN L (1976) A Restructuring of X25 into HDLC, INWG Protocol Note 38, Reseau Cyclades

ROBERTS (1976) International Intercommunications of Public Packet Switched Networks, Proc. of the 3rd Int. Conf. Computer Communications, Toronto p.239

SEXTON (1976) End-to-End Protocols, Virtual Calls and the IIASA Network, INWG Protocol Note 46

STOKES AV (1974) A Users Guide to Babbage, TR-13, Department of Statistics and Computer Science, UCL

STOKES AV and HIGGINSON PL (1975) The Problems of Connecting Hosts into ARPANET, European Conf. on Computer Comm. Network, UK p.25

STOKES AV (1976A) Monitoring and Access Control of the London Node of ARPANET, Proc NCC 1976, pp 539-603

STOKES AV (1976B) XFSTATS: A User Guide, Indra Note 552, Dept. of Statistics and Computer Science, UCL

STOKES AV (1977A) The Indra Network Measurement Project, TR-30

TREADWELL S (1977) Measured Characteristics of Traffic in the UCL Node of the ARPANET, Indra Note 582, Dept. of Statistics and Computer Science, UCL

WALKDEN F, McINTYRE HA, LAWS GT (1976) A Users View of Parallel Parallel Processors presented at CERN, September 1976, to be published.

WALLEY PC (1976) Description and User's Manual of CALIBRA, Working Report 76-3, Brunel University

WILLIAMS P (1976) NCP Performance Measurement, INDRA NOTE 584 Department of Statistics and Computer Science, UCL

ZIMMERMAN H (1975) The Cyclades End-to-End Protocol, 4th Data Communications Sym., Quebec City, p.6

APPENDIX A

Applications to Use ARPANET approved for Use during 1976

Name & Organisation	Project	Site Used
Mr. F. Grover, Blacknest Research Est. Brimpton, Reading.	Seismic Data Exchange	ISI
Dr. J. Alcock, Dept. of Physics, Bristol University Tyndall Avenue, Bristol.	Collaborative analysis of high-energy scattering data	LBL
Dr. A.J. Harley, British Library Lending Division, Boston Spa, Wetherby, Yorks.	Interlibrary loan network development	NLM
Dr. P.L. Holmes, British Lib. Res & Dev. Dept., Sheraton House, Gt. Chapel St., London, W.1.	CANCERLINE database evaluation	NLM
Dr. P. Humphreys, Decision Analysis Unit, Brunel University, Kingston Lane, Uxbridge, Middx.	Decision Analysis software development	ISI
Dr. J. Fitch, Cambridge U. Computer Lab., Corn Exchange St., Cambridge.	Collaborative development of LISP compiler for algebraic systems	ISI
Prof. M.V. Wilkes, Cambridge U. Computer Lab.	Design study for a data ring	ISI
Dr. M.D.C Dyne, Dept. of Engineering, University of Cambridge	Decision Analysis software development	ISI
Dr. G.D. Cain, Dept. of Electrical & Electronic Eng. Polytechnic of Central London, 115 New Cavendish St., London, W.1	Digital Filter Design Techniques	ISI

Name & Organisation	Project	Site Used
Dr. R. Pye, Communication Studies Group, Dept. of Environmental Studies, UCL.	Assessment of Interaction between Travel & Telecommunications	OFFICE-1
Dr. K.V. Roberts, AEA Culham Lab., Abingdon, Oxon.	Fusion research collaboration	BBN at present
Mr. V. Pinkerton, Technology Report Centre, St. Mary Cray, Kent.	Database evaluation	NLM
Dr. F.D. Gault, Durham Particle Data Group, University of Durham.	Collaborative dev. of high energy database	LBL
Mr. M. Gordon, Computer Science Dept., Edinburgh University.	Development of proof-generating system	SU-AI
Dr. J. Darlington, Dept. of A.I., Edinburgh University	Program correctness proofs, program synthesis	SRI-AI
Prof. D. Michie, Machine Intelligence Res. Unit, Edinburgh University.	Machine Intelligence Res.	ILL-NTS
Dr. G.M. Bull, Head of Computer Systems, Hatfield Polytechnic, College Lane, Hatfield, Herts.	Development of BASIC Standards	NBS
Dr. J. Bates, Inst. of Neurology, National Hospital, Queens Sq., W.C.1	Natural language analysis software	SU-AI
Dr. D.M. Bowen, Dept. of Biochemistry, Inst. of Neurology, National Hospital.	Analysis of neurochemical data	SUMEX

Name & Organisation	Project	Site Used
Mr. L.M. Popovic, School of Electrical & Electronic Eng., Kingston Polytechnic, Kingston-upon-Thames, Surrey.	Program generation by formal general methods.	MIT-OMS
Dr. T. Wilkinson, Applied Psychology Unit, Medical Research Council, 5 Shaftesbury Road, Cambridge.	Neurophysiology collaboration	BBN
Mr. D. Curry, MOD, Northumberland House, W.C.1.	Collaboration with US Army Material Command HQ	OFFICE-1
Dr. W.J. Raitt, Mullard Space Science Lab., Dorking, Surrey.	Study of Ionosphere	UCSD
Mr. B.C. Rowe, London Poly. Computer Unit, N. London Polytechnic, Holloway, N.7.	Survey analysis software evaluation	ISI
Dr. R.P. Johnson, Chemistry Dept., N. London Polytechnic. Holloway, N.7	Program development of C.A. Synthesis design of organic molecules	HARV-10
Dr. T. Quirk, Dept. of Nuclear Physics, Oxford University.	Exchange of software of data for high energy physics	ILL-NTS HARV-10
Mr. S.J. Hague, Numerical Algorithms Group, Oxford U. Computing Lab.	Collaboration in Development of numerical software library	ANL
Mr. G. Coulouris, Dept. of Computer Science, QMC, Mile End Road, E.1	Investigation of distributed processing developments	RAND-UNIX, CMU-A, UCB
Prof. R.W. Hockney, Dept. of C. Science, Reading University, Whiteknights, Reading	Parallel processor evaluation	Illiac IV ANL
Dr. P. Purcell, Dept. of Design Research, Royal College of Art, Kensington Gore, S.W.7	Program development for building design	HARV-10 CMU MIT-MULTICS

Name & Organisation	Project	Site Use
Mr. I.R. Whitworth, Royal Military College of Science, Shrivenham, Swindon, Wilts.	Digital signal processing	MIT-AI, Utah.
Mr. N. Neve, Room L121, Royal Signals & Radar Est., Malvern, Worcs.	Support for DOD evaluation of CORAL	ISI
Dr. J.M. Taylor, Royal Signals & Radar Est., Christchurch, Dorset.	Network security	NELC MITRE RADC
Dr. J. Crennell, Rutherford Lab.	Bubble chamber facility data exchange	LBL
Prof. F. Walkden, Dept. of Maths., University of Salford.	Fluid dynamics program development	ILLIAC-IV
Mr. T. Crowe, Systems analysis Div., Thames Polytechnic, Woolwich, SE18	Relational database investigation.	Datacomputer RAND-UNIX
Dr. L. Kohout, UCH Medical School, St. Pancras Hospital, NW1	Neurochemical data analysis	SUMEX-AIM
Professor L.M. Delves Dept. of Computational and Stat. Science, University of Liverpool, Brownlow Hill, Liverpool	Development of ALGOL 68 Compiler with CMU	CMU

APPENDIX B

Non-Active Users During 1976

Name & Organisation	Project	Reason for Inactivity
Prof. M.V. Wilkes, Cambridge U. Computer Lab.	Design study for a Data Ring	Lack of time
Dr. M.D.C. Dyne, Cambridge U. Dept. of Engineering	Decision analysis software development	Cost of telephone calls & poor quality telephone lines.
Dr. G.D. Cain, Central London Poly., Engineering Dept.,	Digital Signal Processing	Unable to make US contacts for collaboration
Dr. K.V. Roberts Culham Laboratory	Plasma Physics Computation	Lack of time and of the availability at Culham of ARPANET host.
Prof. D. Michie, Edinburgh U, Machine Intelligence Research Unit.	Machine Intelligence	Work carried out during visit to US but will resume shortly
Dr. J. Bates, Inst. of Neurology	Automated Analysis of Natural Medical Text	Illness
Dr. D.M. Bowen, Inst. of Neurology	Analysis of Neurochemical Data at Stanford.	Lack of time
Mrs L. Popovic, Kingston Poly., School of Electrical & Electronic Eng.	Formal and general methods for formulating technological problems for computerisation.	Illness
Dr. W.J. Raitt, Mullard Space Science Lab.	Study of Ionosphere during visit to US.	Work carried out
Mr. I.R. Whitworth, Royal Military College, of Science	Digital Signal Processing	Unable to make US contacts for collaboration

APPENDIX C

PUBLICATIONS AND PAPERS PRESENTED DURING THE REPORTING PERIOD

KIRSTEIN PT:	Development in European Public Data Networks, Reichnernetze und Datenfernverabeitung, Informatik-Fachberichte No. 3, ed. D. Haupt and H. Petersen, Berling and Heidelberg: Springer-Verlag, 1976, 39-58
KIRSTEIN PT:	Data Communication General Principles. The Encyclopaedia of Computer Science, ed. C.L. Meek, New York: Mason/Charter 1976, 5-6.
KIRSTEIN PT:	Developments in European Public Data Networks. Proc. Conf. on Computer Networks and Teleprocessing, Aachen (1976) 38-60.
KIRSTEIN PT:	Planned New Public Data Networks, Computer Networks, Vol.1 No.2, Sept. 1976.
HIGGINSON PL:	Software Requirements of Minicomputer System for Use as a Gateway between Networks. Minicomputer Software, ed. T.R. Bell and C.G. Bell, Amsterdam: North-Holland, 1976, 257-67.
STOKES AV:	Software Development in a Network Environment using a High-Level Language. Minicomputer Software, ed. J.R. Bell and D.J. Bell. Amsterdam: North-Holland, 1976, 241-55.
STOKES AV, KIRSTEIN PT & BATES DL:	Monitoring and Acess Control of the London Node of ARPANET. Proc. Nat. Computer Conf. 45(1976)597-603