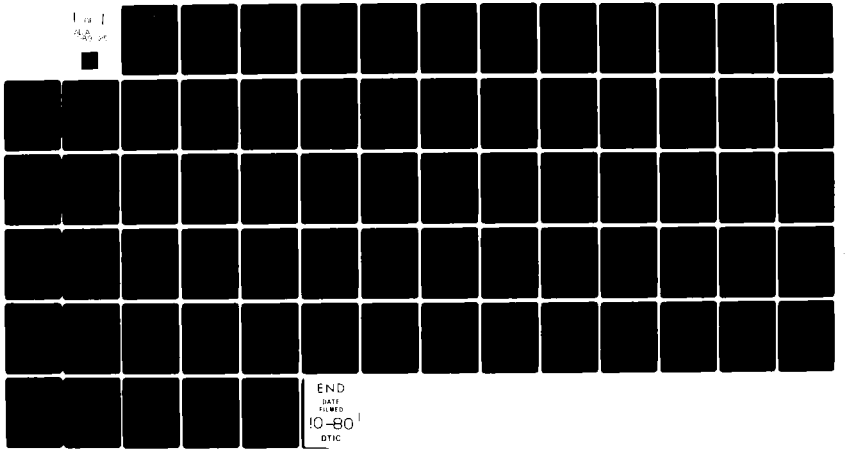
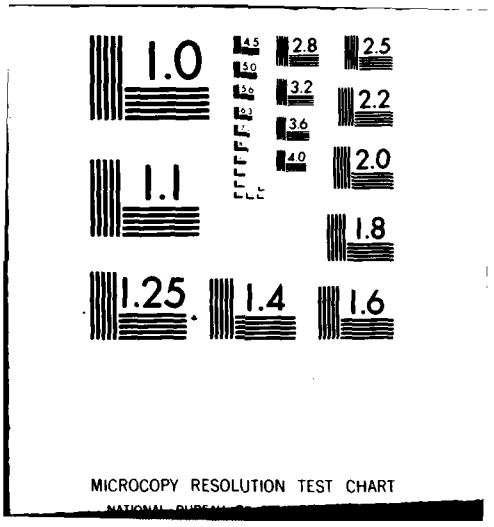
AD-A089 025 BOLT BERANEK AND NEWMAN INC CAMBRIDGE MA F/G 22/2
SATNET DEVELOPMENT AND OPERATION. PLURIBUS SATELLITE IMP DEVELO--ETC(U)
AUG 80 R D BRESSLER MDA903-76-C-0252
UNCLASSIFIED BBN-4474 NL



END
DATE FILMED
10-80
DTIC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3
MICROCOPY RESOLUTION TEST CHART

Bolt Beranek and Newman Inc.

LEVEL II

12

Report No. 4474

AD A089025

Combined Quarterly Technical Report No. 18

SATNET Development and Operation
Pluribus Satellite IMP Development
Remote Site Maintenance
Internet Development
Mobile Access Terminal Network
TCP for the HP3000
TCP-TAC

August 1980

Prepared for:
Defense Advanced Research Projects Agency

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 4474	2. GOVT ACCESSION NO. AD-A089 025	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) COMBINED QUARTERLY TECHNICAL REPORT NO. 18		5. TYPE OF REPORT & PERIOD COVERED 5/1/80 to 7/31/80 6. PERFORMING ORG. REPORT NUMBER 4474
7. AUTHOR(s) R. D. Bressler		8. CONTRACT OR GRANT NUMBER(s) MDA903-76-C-0252 MDA903-80-C-0353 & 0214 N00039-78-C-0405 N00039-79-C-0386
9. PERFORMING ORGANIZATION NAME AND ADDRESS Bolt Beranek and Newman Inc. 50 Moulton Street, Cambridge, MA 02238		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS ARPA Order Nos. 3214 and 3175.17
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Advanced Research Projects Agency 1400 Wilson Blvd., Arlington, VA 22209		12. REPORT DATE August 1980 13. NUMBER OF PAGES 68
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) DSSW Rm. 1D 245, The Pentagon Washington, DC 20310 NAVALEX Washington, DC 20360		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE/DISTRIBUTION UNLIMITED

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

18. SUPPLEMENTARY NOTES

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

Computer networks, packets, packet broadcast, satellite communication gateways, Transmission Control Program, UNIX, Pluribus Satellite IMP, Remote Site Module, Remote Site Maintenance, shipboard communications, Terminal Access Controller.

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

This Quarterly Technical Report describes work on the development of and experimentation with packet broadcast by satellite; on development of Pluribus Satellite IMPs; on a study of the technology of Remote Site Maintenance; on the development of Inter-network monitoring; and on shipboard satellite communications.

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

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(14) BBN-4474 (12) 72 (11) Aug 80

Report No. 4474 Bolt Beranek and Newman Inc.

(9) COMBINED QUARTERLY TECHNICAL REPORT, NO. 18,
1 May - 31 Jul 80,

(6) SATNET DEVELOPMENT AND OPERATION.
PLURIBUS SATELLITE IMP DEVELOPMENT,
REMOTE SITE MAINTENANCE.
INTERNET DEVELOPMENT.
MOBILE ACCESS TERMINAL NETWORK,
TCP FOR THE HP3000,
TCP-TAC.

August 1980 (10) R. D. Bressler

This research was supported by the Defense Advanced Research Projects Agency under the following contracts:

(15) MDA903-76-C-0252, ARPA Order No. 3214
N00039-78-C-0405, ARPA Order No. 3175.17
N00039-79-C-0386
MDA903-80-C-0353, ARPA Order No. 3214
MDA903-80-C-0214, ARPA Order No. 3214

Submitted to:

Director
Defense Advanced Research Projects Agency
1400 Wilson Boulevard
Arlington, VA 22209

Attention: Program Management

060100 slk

Table of Contents

1 Introduction

This Quarterly Technical Report is the current edition in a series of reports which describe the work being performed at BBN in fulfillment of several ARPA work statements. This QTR covers work on several ARPA-sponsored projects including (1) development and operation of the SATNET satellite network; (2) development of the Pluribus Satellite IMP; (3) Remote Site Maintenance activities; (4) inter-network monitoring; (5) development of the Mobile Access Terminal Network; (6) TCP for the HP3000; and (7) TCP-TAC. This work is supported under contracts MDA903-76-C-0252, N00039-78-C-0405, MDA903-80-C-0353, N00039-79-C-0386, and MDA903-80-C-0214 and is described in this single Quarterly Technical report with the permission of the Defense Advanced Research Projects Agency. Some of this work is a continuation of efforts previously reported on under contracts DAHC15-69-C-0179, F08606-73-C-0027, F08606-75-C-0032, and MDA903-76-C-0213.

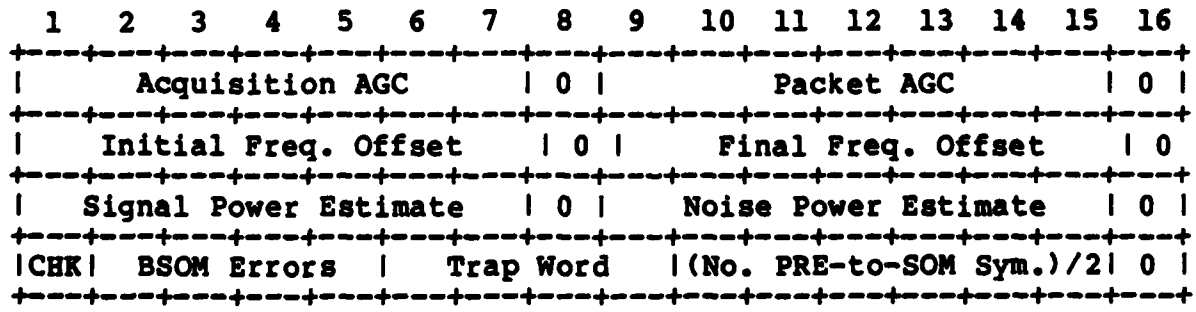
2 SATNET Development and Operation

2.1 T&M Data Transfer Problem

The problem with the transfer of T&M data to the Satellite IMP, reported on in the last Quarterly Technical Report, appears to be a fundamental design problem between the Honeywell 316 modem interface and the PSP terminal. In normal operation, the Honeywell 316 modem interface performs DLE doubling/undoubling and searches for packet-end-framing sequences. It is necessary for the received T&M data to circumvent both functions in order for data not to be lost.

Currently, the PSP terminal, when enabled, appends to every received packet four words of T&M data having the following format.

MSB LSB

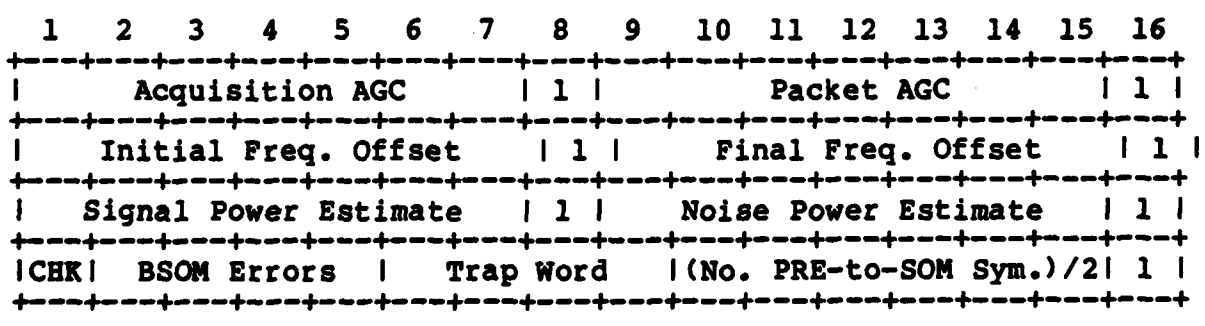


The purpose of the inserted zeroes is to prevent the Honeywell 316 modem interface from interpreting T&M data as the packet end-framing-sequence, DLE-ETX (00010000-10000011).

Unfortunately, the decision was made to set the LSB of each byte to zero for creating a mismatch with ETX. With this format, DLEs can exist, on which the Honeywell 316 modem interface normally performs DLE undoubling. If two DLEs occur in succession, one will be discarded; if only one DLE occurs, then the interface operation is terminated prematurely, as if a packet-end-framing sequence occurred. In either case, T&M data are truncated.

An obvious way to overcome this problem is to change the T&M format such that the LSB of each byte is set to one, as shown below.

MSB LSB



A problem still exists with the 4th word, which does not at the moment allow us to specify one bit in each of the two bytes for creating a mismatch with DLE in each byte. A spare bit could be freed by reducing the field allocated to (No. PRE-to-SOM Sym.)/2 from 6 bits to 5 bits and rearranging the word. Nevertheless, new PROMs for the PSP terminal must be made to implement this T&M format.

A second alternative for correcting the problem, which is preferable from our point of view, is that the PSP terminal should add 4 to the Satellite IMP packet-word-count in the packet header when T&M data transfers are enabled. Then the Honeywell 316 modem interface will ignore DLE doubling/undoubling and packet-end-framing sequences until all the T&M data have been received. Another advantage of this procedure is that the PSP terminal could make use of all 64 bits in the four T&M words. As with the first alternative, new PROMs for the PSP terminal must be made to implement this feature.

A third alternative, one which Comsat has proposed, is that the Comsat hardware perform a 1's complement operation on the first three T&M words. This effectively converts the inserted zeroes to inserted ones. No attempt is made, however, to fix the fourth word. Due to some critical timing constraints, this alternative is not without risk.

2.2 Software Problem Fixed

A subtle bug was found in the Reliable-Transmission-Package (RTP) of the VDH service routines, such that memory would be overwritten whenever a VDH packet arrives from the host while both dedicated Satellite IMP input buffers are in use. The new packet would be written into the buffer last used, destroying the data already present and destroying a pointer location for

writing an internal buffer header word. Among the memory locations destroyed with disastrous consequences are those containing DMC pointers. We have changed the code to eliminate the bug, although we believe that it was rarely if ever encountered.

2.3 Hardware Problems Fixed

Below are summarized several hardware problems which were manifested in the operation of SATNET during the last quarter. In cases involving the Honeywell 316, we not only diagnosed the problems, but also corrected them. In the other cases, we were involved primarily in the detection of the problems and helped with diagnosis.

A failure in the Tanum Honeywell 316 modem interface, which serves as a spare interface for the satellite channel, was traced by BBN field service personnel to a malfunctioning SR-335 micropac. A replacement micropac is being shipped to the site for installation by site personnel.

At Goonhilly, BBN field service personnel determined that a malfunctioning Honeywell 316 voltage regulator card for the 15-volt power supply caused memory failures in the Satellite IMP program. The card has subsequently been replaced.

Early this summer, an extremely severe thunderstorm at Etam disrupted SATNET operations for several hours, during which time the ARPANET direct connection via SATNET circuit, line 77, failed, and the London TIP was isolated. Due to the severity of the storm, the transmitter power could not be maintained within acceptable levels by the Etam site personnel.

Several outages of the ARPANET direct connection via SATNET circuit appeared to be due to a troublesome 9600 baud Codex modem at the London TIP. When malfunctioning, the modem hangs up in such a way that it performs like a one-way modem; monitoring information from the Goonhilly Satellite IMP indicates that the London to Goonhilly traffic appears normal and that the Goonhilly to London traffic is going out correctly, while monitoring information at the London TIP indicates that the circuit is open. Since the London TIP is isolated in this situation, on-site assistance is required to determine the status at London. The circuit is restored by manually placing the modem momentarily in loopback mode. Site personnel have since replaced the Codex modem on the circuit in hopes of eliminating the problem.

The high-noise problem plaguing us for an interminably long time on the 9.6 Kb/s circuit between the NORSAR TIP and the Tanum Satellite IMP was traced by Tanum site personnel to a mismatching of modem characteristics on the two 9600 baud Codex modems. Installation of a replacement modem at Tanum has corrected the

problem.

The installation of the PSP terminal in Tanum during this quarter was followed by several problems which required our involvement. The most vexing was the severe deterioration of channel reception at Tanum for almost two months. Adjustments to the PSP terminal modem by site personnel under direction from Comsat apparently corrected the problem. Oscillator drifts far in excess of specifications kept the PSP terminal off the air for several weeks; refurbishment by the manufacturer was required to reduce the drifts. A failure in the PSP terminal 15-volt power supply caused further outage. Currently, the site is using a laboratory power supply.

In preparation for the PSP terminal installation, the Tanum Honeywell 316 was relocated about 30 feet away from its original location. The move was accomplished successfully without incident and without BBN field personnel present.

After our many unsuccessful attempts to have the Goonhilly Satellite IMP command the CMM module of the attached PSP terminal, Comsat personnel traced the problem to a missing wire on the PSP terminal backplane. After the wire was emplaced, the Satellite IMP was able to issue commands successfully to the CMM; in particular, the enabling of the transfer of T&M data from the PSP terminal to the Satellite IMP is now remotely controllable.

Difficulties with the conversion of the NORSAR-SDAC ARPANET circuit from a commercial circuit to a military circuit resulted in the NORSAR TIP being isolated for many extended periods. During these times, the BBN gateway provided the only monitoring and control path into SATNET; the lack of a backup path meant that when the BBN gateway was out of service, SATNET was isolated. Furthermore, half the traffic sent through the UCL gateway and over SATNET from users at RSRE and at UCL was discarded in the NORSAR TIP due to gateway load splitting. As an emergency measure to preclude gateway load splitting, we had to undefine the NDRE gateway in the Tanum Satellite IMP host tables.

3 Pluribus Satellite IMP Development

Two major milestones reached during the past quarter were the shipment and installation of the first two PSATs, at Lincoln Laboratory in May and at Linkabit Corporation in July. Integration of the PSAT and Miniconcentrator host will take place at Lincoln Laboratory in the near future. Debugging of the PSAT/ESI interface is already underway at Linkabit and should be completed prior to moving the PSAT to ISI (currently scheduled for September).

During the quarter considerable effort was spent refining the Host Access Protocol (HAP) and implementing HAP in the PSAT. The major focus, in terms of both definition and development, was on the service host. The service host mediates between the users of the PSAT (hosts) and the channel protocol module to establish, change and delete groups and streams. The specification of formats and procedures for communicating with the service host during this quarter essentially completes the definition of the initial version of HAP. The current HAP specification also incorporates several improvements not directly related to the service host. One such improvement is a new link initialization procedure described in section 3.1 below. We have not included the complete HAP specification in this report because of its length and level of detail. This specification is available, however, to interested Wideband Network participants.

Throughout the quarter, testing and debugging of the Channel Protocol Module (CPM) was carried out. This was accomplished using both a single PSAT operating in a looped configuration and a dual PSAT configuration supported by the Wideband Satellite Channel Simulator. Modifications to the portion of the CPM responsible for communicating with the ESI were made to support an updated specification of the PSAT/ESI interface. During the quarter, CPM software to support the exchange of local control packets between the PSAT and ESI was added. In addition, CPM software to support multiple ESI coding rates was designed and implemented even though this software cannot be used with the prototype ESIs that will be delivered initially by Linkabit. The Advanced Development Model of the ESI, however, should support mixed coding rates later this year.

During the design to support mixed coding rates, a design limitation in the PSAT/ESI interface specification was identified. As pointed out by BBN at the Wideband Satellite Network system integration meeting in July, the number of state words in the burst control packet is inadequate to support the level of packet aggregation that will likely be desirable in the near future. BBN will consult with Linkabit and others over the next several months to correct this shortcoming. An additional problem uncovered and resolved during the quarter is related to a potential incompatibility between the word-oriented I/O carried out by the ESI and the byte-oriented PSAT/ESI protocol. After

some interaction with Linkabit, it became clear that it was quite straightforward to make the actual implementations compatible.

Recognizing that the normal PID-based dispatching mechanism in the Pluribus was inadequate for servicing the Satellite Modem Interface (SMI) at 3Mb/s, the concept of polling processors was implemented some time ago. Initially, each processor took a turn at polling the SMI and initiated new DMA operations during this period in an attempt to keep the SMI/DMA busy and reduce I/O latency. Periodically, the polling processor would hand off the polling task to a different processor which would then become the active poller. Early in this quarter it became clear that the handoff process itself added too much to the I/O latency for 3Mb/s operation. The polling procedure was modified, therefore, to incorporate a dedicated polling processor. Several tests and some analysis have led to a better understanding of the limits of the SMI poller. This analysis is summarized in section 3.2. Both short term and longer term solutions to the poller latency limitations have been developed and are discussed.

In mid-June BBN attended the first Wideband Satellite Network system integration meeting at DCEC. At that time, draft information was distributed describing PSAT physical, environmental and electrical interfacing requirements. In mid-

July BBN met with Voice Funnel developers and DARPA personnel to discuss internet addressing issues related to the emerging Wideband Satellite Network and its associated access devices (i.e., Voice Funnel, local access networks). A hierarchical addressing scheme proposed by R. Rettberg aimed at avoiding a proliferation of network addresses was discussed.

During this quarter BBN began work on documentation for the PSAT. This documentation will describe the PSAT both at the system level and at the software module level. System level documentation will include detailed physical and interface specifications as well as conceptual and algorithmic descriptions oriented at presenting the technical content and evolution of the PSAT system.

3.1 HAP Initialization

The host access protocol uses a number of state variables and counters that must be properly synchronized between the host and the PSAT in order to function properly. These variables and counters are associated with the send and receive message numbers used by the acceptance/refusal mechanism, and the statistics maintained to support link monitoring.

Initialization is accomplished by the exchange of Restart Request (RR) and Restart Complete (RC) messages between a Host

and a PSAT. The state diagram in Figure 1 shows the sequence of events during initialization or restart. Both PSAT and host must implement this state diagram if deadlocks and oscillations are to be avoided. This particular initialization sequence requires both sides to send and receive the Restart Complete message. Because this message is a reply (to a Restart Request or Restart Complete), its receipt guarantees that the physical link is operating in both directions.

Five states are identified in the state diagram.

OFF	Entered upon recognition of a requirement to restart. The device can recognize this requirement itself or be forced to restart by receipt of an RR message from the other end while in the ON state.
INIT	Local state variables have been initialized and local counters have been zeroed but no restart control messages have yet been sent or received.
RR-SNT	A request to reinitialize (RR) has been sent to the other end but no restart control messages have yet been received.
RC-SNT	A reply (RC) has been sent to the other end in response to a received reinitialization request (RR). The device is waiting for a reply (RC).
ON	Reply (RC) messages have been both sent and received. Data and control messages can now be exchanged between the PSAT and host.

All states other then ON have 10 second timeouts (not illustrated) which return the protocol to the OFF state. The occurrence of any events other than those indicated in the diagram are ignored.

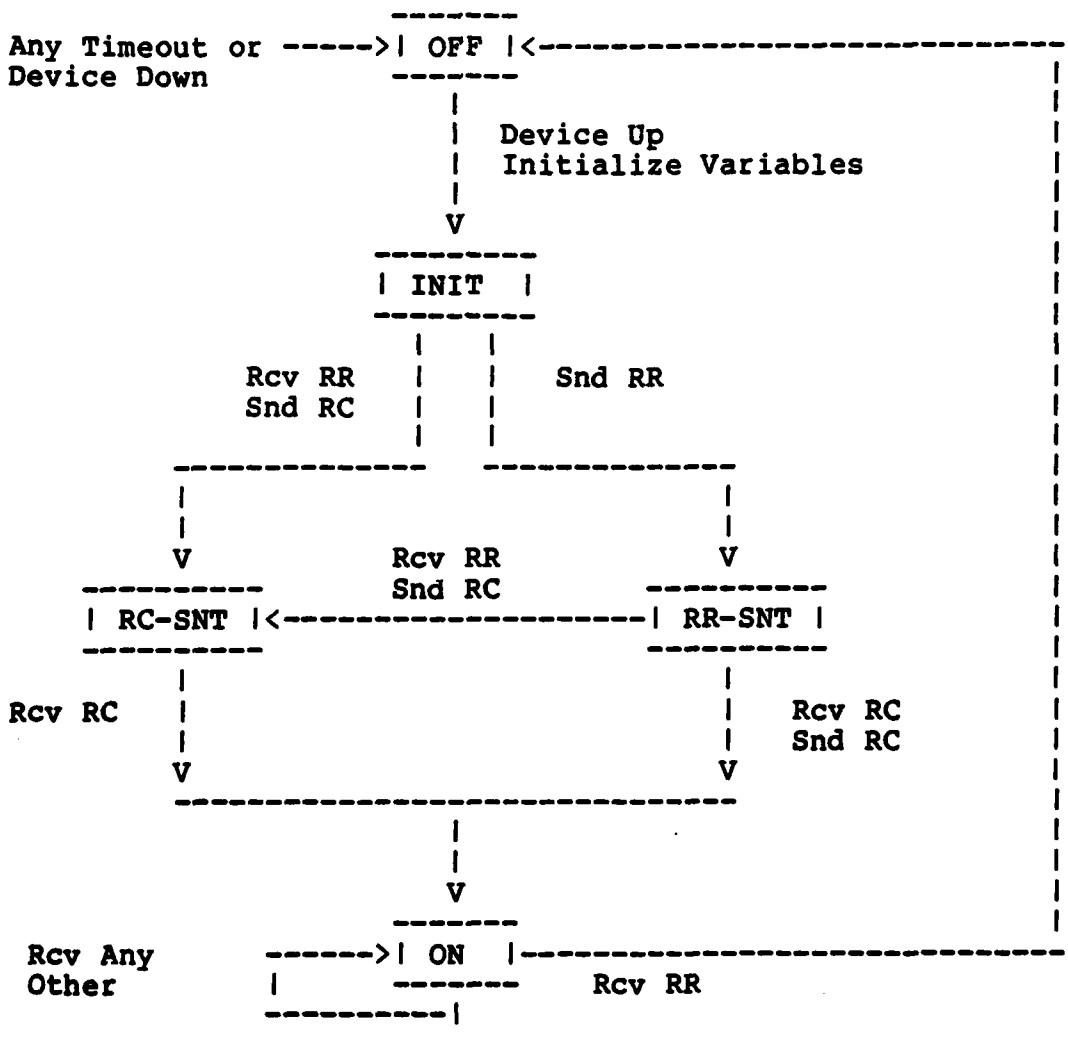


Figure 1 . Initialization State Diagram

3.2 SMI Servicing Issues

Recognizing that the normal PID-based dispatching mechanism in the Pluribus was inadequate for high-speed servicing of the PSAT Satellite Modem Interface (SMI), the concept of polling processors was introduced some time ago. A polling processor continuously examines the status of the SMI and initiates a new direct memory access (DMA) operation whenever the device goes idle. A polling processor plus standard Pluribus DMA card simulates an intelligent hardware DMA. The initial implementation of polling processors involved each of the 6 PSAT SUE processors taking a turn serving as the SMI poller. The active poller would hand off the polling task to a free processor after monitoring the interface for a while. Although this "round robin poller" was significantly better than normal PID-based service, the execution of the handoff procedure itself introduced too much I/O latency. The current implementation, therefore, is based on the use of a dedicated SMI polling processor.

Recent experiments and analysis have pointed out some limitations of even the current implementation. These limitations and possible solutions are discussed below. Both a short term fix and long term remedy are proposed.

3.2.1 The SMI Subsystem

The major Pluribus system elements involved with SMI I/O transfers are illustrated in Figure 2. The SMI consists of a transmit half and a receive half each potentially capable of independent communication with the Linkabit ESI at 3.088 Mb/s. Each SMI half includes 1024 bits of FIFO memory to cover I/O servicing latency. The SMI is controlled by a dedicated polling processor on one of the processor busses. The time between successive polls to the transmit (or receive) half of the SMI, L, is currently 210 microseconds. A busy SMI is either filling or emptying fixed size buffers of B words in common memory. Although B was initially 128, it has recently been increased to 200 (see below). Transfers between the SMI and common memory occur at a rate up to 667K words/sec. This is the total rate of the DMA and must be shared by both transmit and receive halves if they are simultaneously active. The DMA transfer rate is also affected by contention for the I/O bus and Memory-to-I/O coupler (from other I/O devices on the same I/O bus) and contention for the common memory bus and the common memory itself (by processors and devices on other I/O busses).

3.2.2 Speed/Latency Limitations

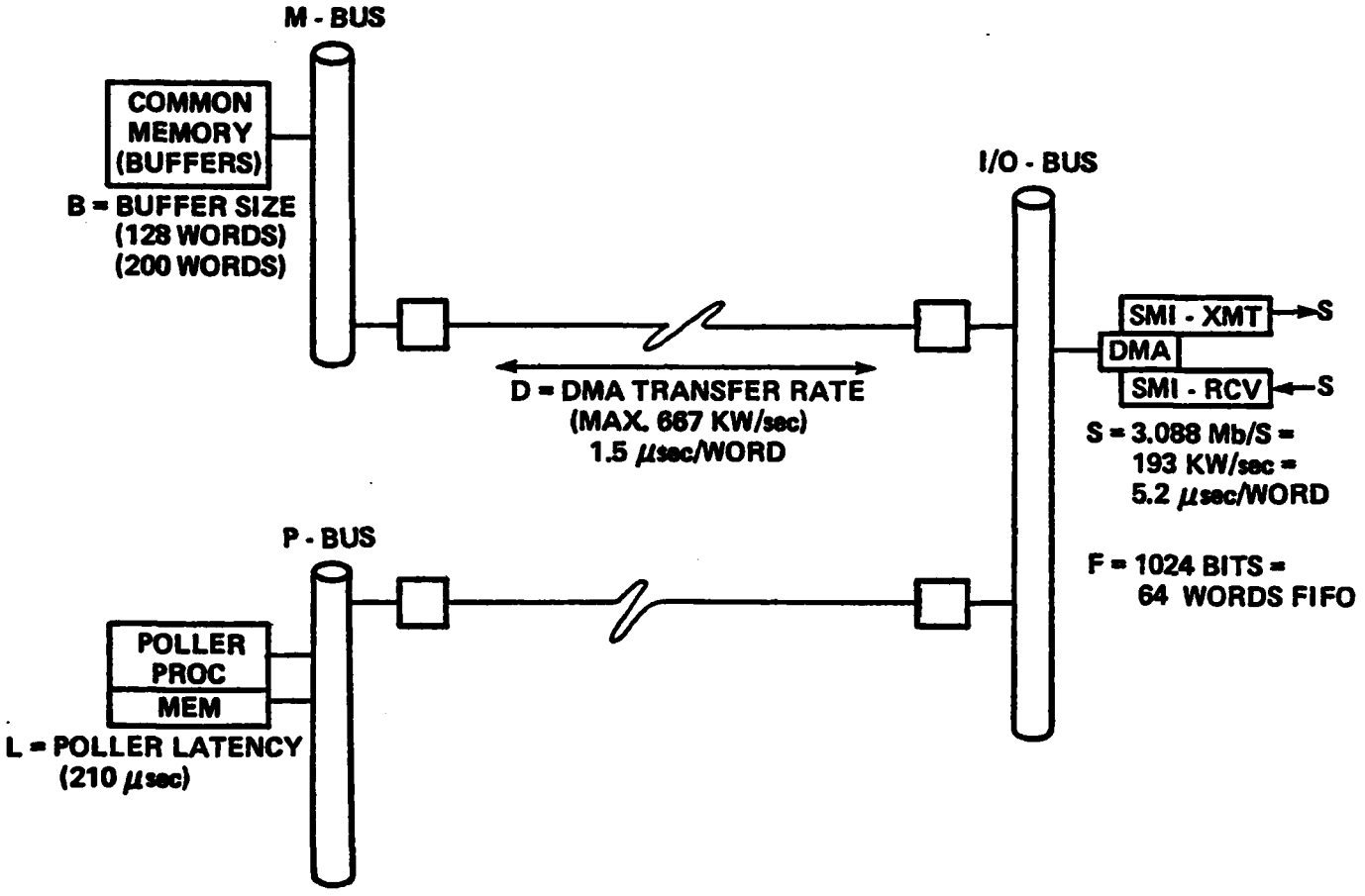
The problems that arise in dealing with the SMI are related to the arrival or departure of data in buffers at a rate which

exceeds the ability of the PSAT to supply those buffers. In these cases the FIFO in the SMI either overflows (on input) or empties (on output) causing loss of one or more packets. While packet loss at the host interface will simply cause additional retransmissions, loss of control packets at the SMI can cause loss of network synchronization for the associated PSAT. Once this happens, the channel may be unusable by the PSAT for an appreciable time while it tries to recover synchronization. The channel scheduling algorithm is designed to survive occasional traffic loss due to channel noise, but introducing additional loss due to node hardware/software operation could seriously impact the stability of the scheduling process.

There are several specific situations that may result in packet loss: (1) many small bursts, (2) many small data packets aggregated into a burst, (3) "Buffer spillover" of packets (creating buffers with only a small number of words), and (4) multi-buffer bursts handled during periods of degraded DMA performance. Each of these situations is discussed briefly below.

3.2.2.1 Small Bursts

This is only an issue on the receive side of the SMI since a PSAT will not itself send many small bursts per frame. We require that the burst service rate (1/L) exceed the burst



SMI I/O Subsystem
Figure 2

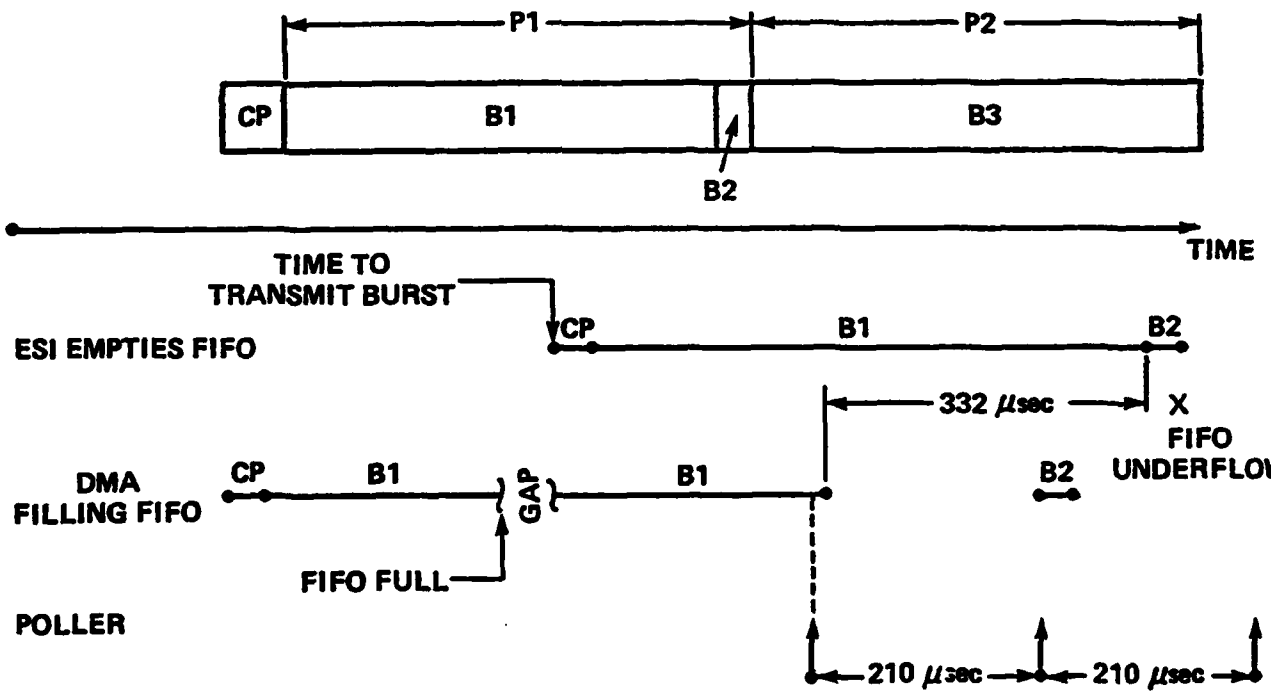
arrival rate. Assuming a preamble of p bits, the maximum burst arrival rate for a burst of m bits is S/(p + m). For the numbers given in Figure 2 with p = 64 bits, we can compute m > LS - p = 585 bits or 37 words. Smaller bursts may be lost (e.g., back-to-back control packets).

3.2.2.2 Small Packets Aggregated in a Burst

This case is very similar to the small burst case but is an issue on both the receive and transmit sides of the SMI. The packet service rate (1/L) must be greater than the packet arrival rate (S/m). This implies (for the numbers in Figure 2) that m > Ls = 650 bits or 41 words. On the receive side, rapid arrival of smaller packets can cause individual packet loss. On the transmit side, attempting rapid transmission of small packets can cause an entire burst to be flushed by the SMI as well as causing loss of burst syncronization by the receiving PSAT.

3.2.2.3 Buffer Spillover

This situation can also arise on both the receive and transmit sides of the SMI and is similar in effect to the small packet case described above. Figure 3 graphically illustrates the problem on the transmit side for a two data packet burst where the first data packet is one full buffer plus a few



Buffer Spillover Scenario (Transmit)
Figure 3

additional words. Assuming that D > S/16, the FIFO remains full as B1 goes out to the ESI. If the poller timing is such that it just catches the end of the B1 DMA operation as shown, the FIFO will empty before the poller has an opportunity to initiate the P2/B3 operation and the remainder of the burst will be lost.

3.2.2.4 Degraded DMA Performance

Degraded DMA operation can also result in packet loss. Consider the case of transmitting a single data packet burst where the data packet consists of several full buffers of size B. Define Wi to be the number of words in the FIFO at the end of the ith DMA operation (buffer transfer) for the burst.

$$W_{i+1} = W_i + dW = W_i + B/D\ (D - S/16) - LS/16$$

$$W_1 = F$$

$$W_n = F + (n-1)[B - (S/16)(B/D + L)]$$

(NOTE: $0 < W_i < F$).

In order not to lose data on output we require that $W_n > LS/16$. Substituting this into the above we get the following inequality that must be satisfied:

$$F + (n-1)B[1-(1/16)(S/D)] - nSL/16 > 0 \qquad (I)$$

If this inequality is not satisfied, the FIFO can run out of data in the middle of the burst.

3.2.3 Discussion

Inequality (I) makes explicit the relation between the various SMI I/O parameters. For a fixed buffer size B and latency L, a "slow" DMA (D) relative to the channel speed (S/16) will eventually result in a data loss. We believe that we have been seeing this effect. For B=128, we experienced occasional data loss on bursts containing greater than 2 buffers using a looped SMI configuration. Solving for the value of D which is critical to inequality (I) for n=2, we find D=221 Kwords/sec or 4.5 microseconds per word. This number should be compared with a nominal 3.0 microsecond/word for the DMA (2 X 1.5 microsecond/word) since both halves of the SMI carry on simultaneous DMA transfers in a looped configuration. The difference between 3.0 microsecond/word and 4.5 mircosecond/word must be attributable to contention, particularly for common memory. Given that processors are referencing the common memory at the same time that a DMA transfer is in progress, the processors and DMA are serviced in a round-robin fashion; the DMA is not given priority as in some systems.

To address this problem, system parameters must be modified such that inequality (I) is satisfied. The easiest parameter to vary in the PSAT is B, the buffer size. We recently increased the buffer size from 128 words to 200 words and are currently evaluating the impact of this change on problem (4). This does

not address problems (1), (2) and (3), however, which are related to partially full buffers. To address these problems the latency, L, must be reduced significantly or a minimum packet/buffer size must be guaranteed. Our short term solution is to pad out buffers/packets to the minimum length computed above. Our longer term solution is to replace the existing poller and thereby reduce L. This will address problems (1) through (4) inclusive.

Several approaches were evaluated for reducing the poller latency. They are:

(1) Replace the existing DMA/poller with a new intelligent DMA implemented in hardware. This is probably the cleanest solution but since it would take about a year to develop and may prove relatively costly, it appears unattractive.

(2) Assign dedicated polling processors to individual interface halves. In the limit with 2 high speed host interfaces and 1 SMI, all 6 processors are used up polling. In fact, the ability of the system to run with fewer than 4 SUE processors is highly questionable. One could consider increasing the number of SUE processors beyond 6, but the cost of the additional processor busses and couplers makes this approach less attractive than other approaches.

(3) PID polling processor pool approach. Allow two or more

processors to share the polling task for all the interfaces and let them check for operation completion via the PID. This solution has the same general problem as solution 2 above but requires somewhat fewer polling processors.

(4) Substitute a standard Super SUE processor for the SUE processor doing the polling task. The Super SUE runs at twice the speed of a standard SUE used in the PSAT so this should speed up the polling and reduce the latency by a factor of 2. However, this will reduce the number of remaining processors to 4 since only 1 super SUE can be put on a processor bus. This approach is unattractive for this reason.

(5) Simulated smart DMA using microprogrammed Super SUE poller on the I/O bus. The Super SUE processor is microcoded to emulate the standard SUE instruction set. This approach would remove the SUE instruction set ROMs and replace them with a custom set of ROMs that implement the current poller loop. This unit would then be put on the I/O bus to simulate an intelligent DMA. The only difference between this approach and the intelligent DMA is that the true intelligent DMA (approach (1)) would not use up I/O bus cycles to monitor the progress of DMA operations. This is the most attractive long term solution. It will require about 6 man-months of microcode and system development

effort and should reduce the poller latency by a factor of about 5. The associated minimum packet size of about 10 words would impose no real restriction on system operation.

3.2.4 Conclusion

The poller latency, L, is the primary problem which needs to be addressed in order to effectively handle the SMI and other high speed devices on the PSAT. Current plans are to implement microcoded Super SUE pollers which serve as intelligent DMA controllers on the I/O bus during the next fiscal year.

4 Remote Site Maintenance

4.1 General

During the last quarter, regular Remote Maintenance sessions were established between BBN and NOSC. These sessions have been occuring most weeks for several months now. This routine access has clarified some of the important issues in Remote Maintenance. System maintenance, in general, is a difficult problem; the maintenance of a distributed system is even more complex. The ARPANET model of remote maintenance has proven to be less satisfactory than anticipated. In this model, remote maintenance consists of three major elements: monitoring, analysis, and correction of system components. All parts of the system are immediately accessible. In some sense, the only direct users of the system are the persons responsible for its maintenance; certainly no one else can change any of the ARPANET software.

The ACCAT system operates on a secure subnet of the ARPANET, with access through Private Line Interfaces (PLIs) at each site. The BBN Remote Site Module is not usually part of this net, but may join it to perform remote maintenance or other functions required to support remote maintenance. Access is quite limited, and this changes the priorities and characteristics of the problem of maintaining the system. There is a substantial user community, and many of the users are engaged in writing programs for use on the system.

Before one tries to redefine the problem, it is worth asking if this situation accurately models the kind of environment in which systems are found. That is, is this or the ARPANET more like the real life situation? As is usually the case, the answer is a little of each. Systems which are in use in the field are likely to contain only controlled software; that is, the field personnel will not be changing the software. On the other hand, access is likely to be quite infrequent. Therefore, it is reasonable to say that the RSM environment models some aspects of the tactical situation better than the ARPANET environment, and the latter is good at other aspects.

When the maintainer's access to the system is limited, the system must be provided with tools which allow it to proceed whenever possible, and to ask for maintenance only when absolutely necessary. Furthermore, if the system is used for time-sharing, it needs mechanisms which assist the users, and which allow them to help one another when formal maintenance is unavailable.

4.2 Automated Filesystem Recovery

For example, many system crashes require only the most mechanical of operations in order to bring the system back up. Such an "auto-reboot" requires a fairly sophisticated program for filesystem repair which can be safely used by a naive computer

program. It cures simple problems, but stops when it encounters dangerous situations which necessitate human intervention. The auto-reboot facility is therefore engaged automatically when the system performs a controlled crash (a "panic").

Each physical disk drive may contain one or more regions of fixed size, each of which contains a UNIX filesystem. For example, the root filesystem has the most commonly stored utilities, important system-wide data files, and the system libraries, and occupies a portion of one disk. Another portion contains the principal user filesystem, normally called /usr. The filesystem consists of the following:

a "super-block" which controls the allocation of space and "inodes";

the inode table, which is the primary index to the files;

the file area, which contains the data;

a free list, of available space in the file area; and

the free inode list of unallocated inodes.

The automated program must make the following consistency checks to determine whether the filesystem is healthy:

no block may be both in a file and on the free list;

block numbers must be within range;

the size of the file and the number of allocated blocks must be consistent;

all directories must contain an integral number of directory entries, that is, must be 16N bytes long;

the number of files listed in the inode must equal the the number of directory entries for that inode;

the inode format must be legal;

all inodes referenced by a directory must be in range;

every directory must have a self-reference entry (called "."), and a parent-reference entry (called "..");

the number of inodes in the superblock must be legal;

no block may belong to two files, or appear twice in the free list; and

the number of free blocks must correspond to the number expected.

The filesystem diagnosis and correction program first scans all of the inodes, recording various information about them. Each inode is categorized as unallocated or as corresponding to an ordinary file, a directory, or a special file (like a terminal). As the inodes are scanned, the following operations are performed:

the file type is checked; and

the link count is recorded; this is the number of expected directory entries for this inode. For ordinary files, the blocks in the file are checked for range. Each block encountered is recorded in a bit map of the filesystem, and any duplicate entry is noted.

Next the filesystem directory tree is traversed. The following operations are performed on the directories as they are encountered:

the inode corresponding to the directory is checked; it must be allocated and of type directory;

the number of links in the inode is decremented;

the size of the directory is checked; and

each directory entry which is not another directory is validated.

After the tree has been completely scanned, the inode map is re-scanned for directories which have not been encountered. These are directories which are not attached to the tree; if they have a parent-reference entry (".."), they can be reattached, otherwise they are entered as subdirectories of a special orphan directory. After they have been attached in some way, the formerly isolated directories are tested, as above.

The link count table is then scanned. If the net link count is zero for an allocated file, the count is consistent. Otherwise, if it is zero, and no file has been found, it corresponds to a broken pipe, and can be cleared. If the net count is non-zero, and one or more entries have been encountered, the count in the inode is simply adjusted. If the count is positive, but no file has been found, the data is presumably of some value, but it is not connected to the directory tree anywhere. It is then attached to the orphan directory, and the link count in the inode is adjusted to one (if necessary).

Finally, the free list is reconstructed if necessary, and the inode list in the superblock verified.

The program is run using a known-good filesystem; if the system root is damaged, the program is run from an alternate root. If no serious problems develop, access to users of the system is enabled. If the auto-reboot facility detects serious file sytem errors, Remote Maintencance will be requested.

4.3 User Level Remote Maintenance

Remote maintenance of user programs is more difficult than local maintenance because (1) the user feels that the support is less good; (2) the user cannot appeal directly to the maintainer for help at all times; and (3) the bookkeeping which is necessary to make things work at all is significantly more complex.

In the ordinary, local maintenance case, the user develops confidence in both the system and maintainer by observation. When the user is remote from the system, or if both user and maintainer are remote from it, concerns arise about the solidity of the system. If the maintainer's contact is only intermittent, these are magnified; this has been observed clearly in the RSM project. It is important that great care be taken to assure that the user's problems will be attended to promptly and carefully.

When one says "the UNIX system", one usually includes the ordinary utilities in this system and often even specialized programs, as well as the operating system itself. There are now

over 200 programs in this set. Whenever anything goes wrong with any of these programs, the average user feels that the system has failed. Furthermore, the user generally does draw a line between the software and hardware. It is necessary to receive, analyze and respond to comments and complaints about any of these system components from any site. In addition, the problem reporting system should provide a mechanism for:

categorizing the bugs, complaints, etc.;

routing the problem to the correct person, semi-automatically;

issuing reminders about unresolved problems;

answering the person(s) who raised the issues; and

disseminating bug-fixes to the various sites.

Often a user needs help defining the malfunction of the programs. Sometimes, difficult problems simply go away when an expert is called in. In the remote maintanence case, this can be accomplished by giving the expert a way to see what the user is doing, using an extension of the UNIX command "write", in which the user and the maintainer can see what is being typed on the other's terminal. The user will also need some method of suspending the current process. One such method is through the use of the "screen" facility which is available in the RSM, but that is limited to those cases where the user expected problems. The user really needs greater control over the creation and status of processes through a modification to the UNIX shell.

The UNIX manual provides information about the commands, procedures, and data structures. It is supplemented by various published papers. There is limited support for helping the user find the necessary information, which is stored online. The effectiveness of the system could be increased through a more comprehensive help facility.

4.4 New System Release

A new release of the UNIX system and its libraries and utilities is being installed at the various sites. The previously installed systems at NOSC and CINCPACFLT were based on a single version of the operating system, and the NPS system was a version later, because the over-the-net reload in December 1979 took the place of a more systematic release. One of the goals of the new system release, therefore, is to have the same software installed at all three sites.

To simplify this task, a release staging filesystem, /rsm, was established at each of the sites. This file system contains all of the sources for the latest release of the UNIX operating system and its utilities. This version, sys.130 in the BBN-UNIX series, provides a number of improvements over the previously installed sys.124.

The "dead port" problem is corrected. In earlier versions

of the system, the port open status was not checked correctly. As a result, whenever the system crashed, it left behind a port in the filesystem. If this was not removed, subsequent processes addressing that port would use the meaningless pointer left behind; this would cause another crash.

An environment facility which was modeled after the one in Version 7 was installed. This will make it possible to import utilities from that version to the RSM, assuming that the license upgrade is obtained.

The disk handler now contains the ECC code, which allows the correction of most disk read errors which occur on reasonably good packs.

The operating system sources, and those for programs which depend on internal system tables, have been reorganized to simplify maintenance.

The UNIX kernel has been updated so it can be compiled using the Phototypesetter version of the C compiler.

In the course of the installation, the programs used for saving and restoring older versions of files were improved, and an "install" mechanism was added to the program "build".

5 Internet Development

5.1 CMCC Development

As reported in the last Quarterly Technical Report, the Catenet Monitoring and Control Center (CMCC) program generated a log file of traps, throughput reports, and routing reports originally through use of (a) the gateway polling feature to fetch reports at regular intervals, and (b) the report filtering feature to reduce the number of reports written on the disk. Gateway polling is required because not all gateways will send reports automatically. Report filtering allows report generation in the log file at relatively low rates while polling at relatively high rates, so that the loss of any single report will not create a gap in the log file. Initially, the throughput and routing reports were collected every five minutes, but were written only at hourly intervals because of disk space limitations. After the disk space allocation for the <CMCC> directory on the ISIE TOPS-20 ARPANET host was increased to 1000 pages in the last quarter, we changed the CMCC program to log throughput and routing reports every half hour rather than every hour and to archive the log files directly on ISIE rather than on the BBNE TENEX ARPANET host, which serves as a backup computer for the CMCC program.

While the log file as originally implemented was able to provide a general picture of gateway performance at half-hour

intervals, the events which would be of most interest to gateway implementors and gateway users might very well be obscured. To overcome this difficulty, we changed the CMCC program such that the log file is now entirely event driven and exception driven. Reports are written into the log file only when something interesting occurs, such as when a specified threshold of packets forwarded or packets dropped has been exceeded or when a change occurred in the routing table. Current default values for the thresholds are 800 packets forwarded and zero packets dropped (any number of packets dropped whatsoever is an event). As a result of the changeover, the size of the log file, while more variable, has been reduced to about half its former length.

The event and exception driven output is furthermore an option available to display process users. This feature is invoked in the same manner as the previously specified report types for display or logging. For greater user flexibility, threshold values for the number of packets forwarded and the number of packets dropped are user adjustable.

Implementation of the event and exception driven output feature required a restructuring of the CMCC program with the transfer of modules from the control process to the display process. Previously, all the character formatting for the report and trap messages was done in the control process, while the display process just delivered the characters it was given. Now

the control process passes information to the display process, and the display process does the formatting and decides whether an interesting event has occurred. Once the restructuring was finished, the CMCC program actually ran more efficiently.

Hourly and daily summaries of gateway activity are now being generated by the CMCC program and written on the disk. The summaries include the following information on each gateway known to exist:

- the number of minutes heard;
- the number of minutes failed to be heard;
- the number of times failed to be heard;
- the number of packets processed;
- the number of packets dropped;
- the number of gateway restarts;
- the number of times each interface was reported down.

Hourly and daily summaries have identical formats; only the time interval over which the information is collected differs. The daily summaries, which are automatically sent by ARPANET mail to a list of interested people, are normally produced and mailed at midnight local time; however, if the CMCC program is down at midnight, it will produce a daily summary upon restart for the last day it was up.

We drafted new message types for use in evaluating Catenet performance dynamically. These types, which have yet to be added to the CMCC gateway monitoring messages, include the following:

- CPU idle time (a measure of how heavily the gateway is loaded);
- Packet delay across a gateway;
- Gateway to gateway delay (actually, the round trip time of a special packet echoed from a specified gateway);
- Throughput (bits);
- Queue occupancy traps (a signal for when the occupancy of a queue goes above or below a certain threshold value).

We are planning to implement the code in the CMCC program for handling these types during the next quarter.

In evaluating Catenet performance, message generators will be required to load the gateways with traffic until they start dropping packets. These message generators, whose implementation is yet to be defined, can be located in either the gateways, the CMCC program, or other internet hosts. When both the CMCC program and the gateways implement the new message types, and when message generators are implemented, the CMCC program will be able to find out how much traffic the gateways were processing, where the bottlenecks lie in the Catenet, and what the accompanying delays were.

Since the document IEN 132 entitled "The CMCC Terminal Process" was distributed, we have added many new features to the CMCC program, such that IEN 132 is currently incomplete. To provide a complete listing of commands as well as to facilitate program usage by inexperienced people, we have made the CMCC program self documenting. An on-line 'Help' facility, invoked through liberal use of the command '?' at any command level in the display process, documents all the existing features in CMCC, new as well as old. Furthermore, [ISIE]<CMCC>NEWS.TXT describes the operation fully and details the user interface.

5.2 VAN Gateway

During the last quarter, the Value-Added Network (VAN) gateway development proceeded on two separate interfaces. The byte-interface software from UCL for X.25 level 2 and level 3 was successfully assembled. Subsequently, we linked the BCPL software drivers with the UCL software and began the debugging of the interface between the level 3 and applications level. As part of these tests, the X.25 byte-interface hardware from UCL was connected to a Bell 212 1200 baud modem set in loop-back mode for determining the integrity of the interconnection.

The major effort during the last quarter, though, was placed on bringing up the X.25 block-interface hardware from RSRE. Our first step was to exercise the hardware with a test program

provided by RSRE. Next, the hardware was connected to a Bell 212 1200 baud modem set in loop-back mode for determining the integrity of the interconnection.

After successful operation of the modem in loop-back mode, we contacted the Telenet Network Engineering support people, located in Vienna, Virginia, for arranging service tests. (Previously we had obtained documentation on their X.25 protocol tester, their testing procedures, and their certification procedures.) We then began conducting certification tests for verification of the RSRE block-interface at the X.25 link level (level 2). These tests include the following:

(1) Test of the DTE in disconnect send state;

(2) Test of the DTE in set-asynchronous-balanced mode (SABM);

(3) Test of the DTE in primary normal state and SABM reset state;

(4) Test of the DTE in remote busy state;

(5) Test of the DTE in T1 timeout state (DTE must transmit one I-frame upon completion of link setup);

(6) Test of the DTE in secondary normal state and frame reject state;

(7) Test of the DTE in secondary reject sent state.

Analysis of our first testing session was somewhat obscured due to confusion existing on the documentation. After consultation with Telenet support personnel, we believe that much of what we were interpreting as malfunction at the time was

really due to a misinterpretation of the intricacies of the test.

The results of our second testing session were highly encouraging; verification of success requires further consultation with Telenet support personnel. In the testing process, we discovered a quirk in their software, which forced a separate test setup for six out of the seven parts. If indeed the last test session went well, we expect the next test session to be the certification test operated by the Telenet personnel for verification to Telenet's standards of the RSRE block-interface.

5.3 LSI-11 Gateways

We assembled and tested the enclosure and power supply for the VDH interface of the BBN gateway between the SATNET Etam Satellite IMP and the ARPANET BBN40 IMP. Subsequently, the VDH interface was removed from the PDP-11/40 chassis serving as the BBN gateway and was permanently installed in the new enclosure. The PDP-11/40 is currently interfaced to this unit, while the replacement LSI-11/02 has been mounted in the gateway rack and checked for stand-alone operation prior to its final installation.

Replication of the VDH interface enclosure has been initiated. Once checkout is finished, this enclosure and its

power supply will be delivered to UCL for installation of the VDH interface attached to the UCL gateway between the SATNET Goonhilly Satellite IMP and the ARPANET London TIP.

6 Mobile Access Terminal Network

6.1 Summary of Past Quarter's Work

Summarized below are the tasks we worked on during the last quarter in the development of the Mobile Access Terminal (MAT) Red subsystems. Section 6.2 presents the results of our MATNET throughput calculations, taking into account all the software and hardware overheads associated with packet transmissions over the satellite channel. Section 6.3 presents the TENEX/TOPS-20 requirements at ACCAT necessary for monitoring and control of MATNET during phase 2B of the project; these items are currently being examined by Naval Ocean Systems Center (NOSC) personnel in conjunction with BBN personnel.

An algorithm for scheduling packets over the satellite channel independent of previous packet transmissions was developed. This algorithm requires the scheduling of both a transfer time and a transmit time, where the former refers to the period of time required to move the packet from the Red subsystem to the Black subsystem, and the latter refers to the time required to broadcast the packet over the satellite channel. Both of these values depend upon the size of the packet (quantized into the number of transmitted interleaver blocks) but not the parameters associated with previously transmitted packets.

Modification of the Satellite IMP code to accommodate some of the architectural dissimilarities between a Honeywell 316 general purpose computer and a BBN C/30 Packet Switch Processor has begun. For example, the change from a 10 microsecond clock to a 100 microsecond clock requires a corresponding change in the timing processes of the software, including the wakeup process and the watchdog timer. Also, the I/O buffer pointers were increased from 15 bits to 16 bits to permit accessing the top 32K-words of C/30 memory.

The design and implemention of the Satellite IMP macrocode software for the 1822 Host-to-IMP drivers has begun. This software is necessary for interfacing the LSI-11/03 Terminal Interface Unit and the LSI-11/03 Gateway. Work also began on the microcode software for the Red/Black interface drivers. Special stand-alone test programs in macrocode have been written to check the microcode.

Fabrication of the specialized hardware for the Red/Black interface of the C/30 was completed. This hardware consists of a small printed circuit daughter board mounted above the C/30 universal I/O board for converting between the TTL signal levels internal to the C/30 and the MIL-188C signal levels required by the COMSEC equipment. Additional circuitry to facilitate interface testing, including an extra input line and an extra output line, is also included on the board. Initial testing of

this printed circuit is already finished. Final testing awaits delivery of a C/30 with the microcode for the Red/Black interface drivers.

Fabrication and unit testing of the MATNET satellite channel simulator was completed. For a long-term reliability checkout, the unit is currently connected to the Honeywell 316 computer serving as a software development testbed for SATNET Satellite IMPs. As soon as the C/30 Packet Switch Processors become available, we will connect them to the satellite channel simulator.

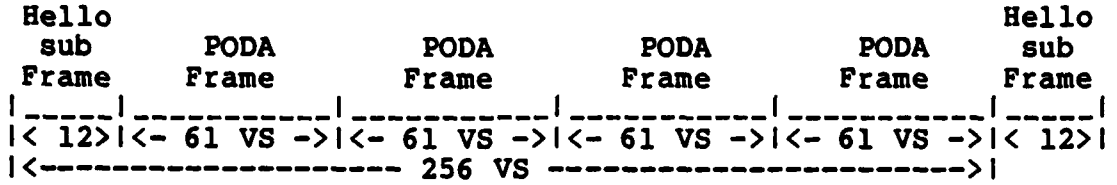
During the last quarter, we also participated in the MATNET Critical Design Review held at the Naval Surface Weapons Center near Washington, D.C. At that meeting and at a subsequent meeting held at ECI in St. Petersburg, Florida, we began discussions of issues associated with the MATNET communications tests and command and control tests. The MATNET testing document was updated and expanded to include the decisions made during these discussions.

6.2 MATNET User Data Throughput

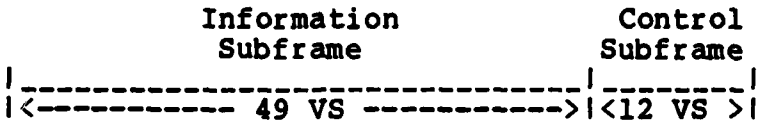
The user data throughput over the MATNET satellite channel is substantially less than the ideal upper bound of the channel rate due to the software overhead required by the MATNET PODA

(Priority-Oriented Demand-Assignment) channel assignment protocol, the Internet Protocol (IP), and the Transfer Control Protocol (TCP), and the hardware overhead required by the Satellite IMP, the COMSEC equipment, the CODEC, the interleaver/deinterleaver, and the AN/WSC-3 radios. The effect of these items on the user data throughput is presented here.

Currently, the Channel Protocol Module assigns a constant bandwidth to the regular transmission of Hello packets from each site. Hello packets are used to indicate that the site is a participating member of the network and to allow the site to determine its range (round-trip time) to the satellite for local time correction. Hello packets are sent in adjacent time slots once every 256 Virtual Slots (VS); this interval is designated as the Hello frame interval and is equivalent to 2.6 seconds. (A Virtual Slot, which currently equals 10.24 milliseconds, is the fundamental time interval factored into the channel assignment algorithm by the Channel Protocol Module.) Each Hello frame is divided into the Hello subframe assigned to Hello packets and into multiple PODA frames; the latter are further divided into information subframes and control subframes assigned to the transmission of datagrams and channel assignment request packets, respectively. This structure is shown below.



Hello Frame



PODA Frame

The length chosen for each of the subframes is a compromise reflecting the classical delay versus throughput considerations. Longer PODA frames require longer average delays in making channel assignment requests but allow a higher fraction of the channel to be allocated to the information subframes. In this illustration, we have chosen a system allocation with long PODA frames, so as to favor increased data throughput at the expense of longer delays. Under the assumptions that no more than three sites will be integrated into MATNET during the inital testing, that FPODA (Fixed Priority-Oriented Demand-Assignment) channel assignment protocol must be used instead of CPODA (Contention Priority-Oriented Demand-Assignment) channel assignment protocol due to packet contention difficulties (requiring each site to have a dedicated control slot in each control subframe), and that rate 1/2 coding is applied (reflecting a worst case interference

situation), the specific allocation of communication slots within the Hello frames is shown below.

Hello subframe	12 VS
Total control subframe	48
Total information subframe	196
Total number of slots per Hello frame ...	256 VS

With no more than 196 VS out of 256 VS allocated to datagrams, the effective channel capacity is reduced by 23 percent from 9.6 Kb/s to 7.4 Kb/s.

Added to every datagram are the MATNET satellite channel header, the Internet Protocol header, and the TCP header. The number of bytes for each category is:

MATNET	28 bytes
Internet Protocol	20
TCP	20
Total	68 bytes

At a 9.6 Kb/s data rate, the channel transmission time of this software overhead is 56.67 milliseconds.

Added to every transmitted packet is the hardware overhead associated with the Satellite IMP, the COMSEC equipment, the CODEC, and the AN/WSC-3 radios. At a 9.6 Kb/s data rate, this overhead is apportioned in the following manner.

Modem Preamble (128 bits @ 19.2 Kb/s)	6.67 msecs
Unique Word (74 bits @ 19.2 Kb/s)	3.85
Packet Type (8 bits @ 9.6 Kb/s)	0.83
COMSEC Preamble (64 bits @ 9.6 Kb/s)	6.67
Extra COMSEC Prep (2 bits @ 9.6 Kb/s)	0.21
Fill Bits (6 bits @ 9.6 Kb/s)	0.63
Coder Flush (6 bits @ 9.6 Kb/s)	0.63
Packet start framing (8 bits @ 9.6 Kb/s)	0.83
Packet data checksum (24 bits @ 9.6 Kb/s)	2.50
Maximum transmission offset among sites	0.80
Guard	2.00
Total hardware overhead per packet	25.62 msecs

Another complication in calculating the hardware overhead is that the transmission of only an integral number of interleaver blocks is allowed. Hence, if a packet does not fit exactly into a countable number of interleaver blocks, the unused space left over in the last block will be wasted. Since each interleaver block contains 504 bits, its transmission time at a channel symbol rate of 19.2 Kb/s is equal to 26.25 milliseconds.

Because all the overheads described above are independent of the packet size, channel throughput is increased with the use of longer packets. Under the assumption that the information subframe is filled with identical length user datagrams, which are required to fit into the information subframe space without fragmentation, the calculated throughput for several different packet lengths is presented in the following table.

USER PACKET LENGTH	NUMBER INTERLEAVER BLOCKS	PACKET CHANNEL TIME	PREDICTED DATA THROUGHPUT
1 bytes	3	89.6 msecs	61 b/s
4	3	89.6	244
16	3	89.6	977
64	5	142.1	2344
128	7	194.6	3125
188	9	247.1	4590

The first entry corresponds to a single typewritten character; the fourth entry corresponds to a typewritten line; and the last entry corresponds to the largest packet length that can be accommodated in the present Satellite IMPs. The exceedingly small value of user throughput for traffic consisting of single-character packets only should not be the cause of undue anxiety. The situation where the entire channel is full of single-character packets is unrealistic; more realistically, we expect a mixture of varying size packets, where the expected traffic pattern is yet to be determined. The number and length of packets generated by operating personnel sitting at a terminal can be quite small in comparison with those normally generated by the computer in response to user enquiries; however, when the channel traffic is dominated by users sending small packets, large throughput is not necessarily mandatory. These figures illustrate, though, that computer programs which use the channel more efficiently by sending larger packets are desirable.

6.3 Phase 2B Monitoring and Control of MATNET

Since the satellite processors and the gateway processor deal with Red data only, the support programs which monitor and control the operation of these processors must also be Red. Hence, the support programs, which for the Atlantic Packet Satellite Experiment (SATNET) are currently running on BBNE-TENEX and ISIE-TOPS-20 ARPANET hosts, must for the MATNET project run on the TENEX/TOPS-20 host computers at ACCAT during phase 2B of the project. This system will be designated as the MATNET Network Control Center (NCC). Insofar as possible, these support programs will be exact copies of their functional equivalents which have been developed for use in SATNET except as they are affected by the following considerations. First, the MATNET shipboard components are inaccessible to the MATNET NCC except by use of the MATNET itself; i.e., there will be no backup ship-to-shore data transmission capability. Second, the data messages transmitted from the ship to the NCC are conceptually Red data, and operational provisions will have to be made to ensure their proper handling within the NCC. Third, although not proposed for implementation initially, EMCOM considerations will affect the operation of certain NCC facilities and should therefore be considered in any further design of these facilities.

The specific monitoring and control programs for NCC operations are listed below:

RECORDER: This program is used to collect monitoring data from every site on a regular basis (about once every minute) and to create and maintain a database containing this information. As such, it needs to run 24 hours a day and requires automatic job restart upon TENEX/TOPS-20 restart.

MONITOR: This program is used to interrogate the database formed by RECORDER for displaying the status of the network to the NCC or the user. In order to see problems as they develop, the NCC should have this program running at all times on one of its terminals.

EXPAK: This program allows the user to test network operation with traffic generators and to collect statistics on all network traffic and on various Satellite IMP queues.

U: This program is the utility program which allows network maintainers to load, dump, and debug remotes sites over intervening networks.

Although the first three programs can run on either TENEX or TOPS-20 computers, the last program can run only on TENEX computers; hence, we believe all the above programs should reside in the ACCAT TENEX for primary MATNET monitoring and control applications, while a backup directory should exist on the ACCAT TOPS-20. The operating systems for both computers must have the following features:

- 96-bit ARPANET headers;
- Internet Protocol (preferably version 4);
- IDDT (for debugging TENEX/TOPS-20 programs);
- ARPANET raw packet facility (for assigning and supporting ARPANET raw queues, which bypass normal Host-to-IMP access protocol);
- An attached magnetic tape drive or DEC-tape drive (for the transfer of the above programs from BBN to ACCAT);
- TENEX files-only directory allocated for MATNET (1000 disc

pages should be allocated, but most times the normal usage is probably no more than 300 pages);

- TENEX login directory allocated for MATNET (750 disc pages should be allocated, but most times the normal usage is probably no more than 300 pages);
- TOPS-20 login directory allocated for MATNET (200 disc pages should be allocated, but most times the normal usage is probably no more than 100 pages).

Since currently not all of the above items are present in the ACCAT TENEX/TOPS-20 computers, discussions have begun between NOSC personnel and BBN personnel to resolve the differences.

7 TCP for the HP3000

This section covers the second phase of an ongoing research effort to implement TCP protocols on an HP3000 computer system. The phase of the work covered in this report is the implementation design of the protocol software. Topics covered include flow control through the various protocol layers, the software interface between the protocol layers, protocol control structures, and the management of message buffer resources.

7.1 Protocol Software Architecture

The protocol software architecture is dictated by a set of design requirements and MPE operating system constraints. These requirements and constraints are summarized as follows:

- The new network software must be isolated from the existing operating system as much as possible. The isolation will allow any site to add or remove the network software with a minimum of effort. It will also make the network software less vulnerable to any changes HP makes to MPE.

- Efficient high speed network communications are extremely important because this TCP version will be used on a production rather than an experimental basis.

- One of the problems with MPE is that, though the operating system performs device assignment and access control for its I/O devices, the user process is responsible for operating the I/O device. MPE does offer intrinsics to operate common devices, but these are very low level operations. This I/O arrangement makes it difficult to control an asynchronous network interface. The protocol software architecture will therefore require at least one process which has exclusive control of the INP interface.

- One of the properties of these network protocols is that the message acknowledgments and retransmissions occur at a relatively high level -- in the Transmission Control Protocol in layer four. A moderate amount of time passes from the time the originating TCP queues the message for transmission and the receiving TCP gets the message. In order to prevent acknowledgment delays which in turn cause the foreign host to retransmit data, the software architecture should minimize the amount of time it takes for incoming data to move through the 1822, IP, and TCP protocols.

- With many network users and many connections concurrently in use, the network software must be able to handle the problems of multiplexing use of the network interface hardware. The interface on which the multiplexing takes

place must support a number of simultaneous users in such a way that the behavior of any individual user does not affect data throughput of the other users.

In order to meet all of the design requirements and constraints described above, the HP3000 protocol software is implemented in a set of processes. One process which will be called the system protocol process is responsible for maintaining the INP interface as well as supporting the 1822, IP and TCP protocols. The rest of the processes, called applications protocol processes, support the user interactive network functions including FTP and TELNET.

The use of a single system protocol process is a key element in the protocol design. The system protocol process provides control over the INP interface by providing buffers and acting as multiplexer and de-multiplexer of network traffic to and from the INP. Use of a single process minimizes inter-protocol layer communication delays which in turn minimize the acknowledgment delays for incoming data. A single system protocol process makes it possible to use interprocess communication primitives to provide a uniform network interface for the applications level protocol processes.

User TELNET and User FTP protocols are to be implemented as ordinary user programs. They use the same system calls as any other network accessing program, but are written to provide a

higher level command language for the user. As user programs, they execute in the user's address space with the privileges normally available to the user. The User TELNET and User FTP programs are re-entrant, with as many processes running this code as users wishing the service.

Server TELNET is a single process created as the system starts up or whenever the first need for it arises. De-multiplexing of Server TELNET inputs is accomplished via a pseudo-teletype driver. The driver acts as the interface between the Server TELNET process and the Teletype handler.

The interface between application protocol processes and the system protocol process is through a set of TCP intrinsics. The intrinsics are designed to form a uniform interface between the user and the TCP. Actual data communication between a user process and the system protocol process is done with a combination of message files and direct buffer-to-buffer transfers. Message files are used to pass flow control information while the actual data transfer is made by copying data between user buffers and system protocol buffers. The combination of message files and buffer copy is used to take advantage of the flexibility of message files and the data rates achieved by direct data copy.

7.2 System Protocol Software

Since this TCP implementation is to be used on a production rather than an experimental basis, the design effort has concentrated on the efficiency rather than the sophistication of the protocol software. This is especially true of the system protocol software whose initial design includes only those features needed to support the FTP and TELNET protocols.

At the same time, the software design does allow for the future enhancement of the protocol software. There are no inherent design limitations which will prevent implementation of the more sophisticated TCP and Internet features.

7.2.1 Implemented Features

The specific TCP and Internet features to be implemented include the following:

- multiple connections to multiple hosts,
- flow control at the 1822, Internet, and TCP layers,
- error recovery,
- fair allocation of resources among all connections,
- handling of urgent data,
- surviving incorrect packets,
- datagram reassembly,
- routing,

- source quenching.

7.2.2 Software Architecture Overview

The system protocol software architecture reflects the need to avoid packet processing delays rather than a strict hierarchy between protocol layers. The system protocol software is implemented as a single process to allow the system protocol layers to share software resources for greater efficiency. The shared resources include subroutines which perform functions required by more than one protocol layer and a common buffer pool to optimize storage resources and to allow efficient communication between protocol layers.

Network traffic through the system protocol process takes different forms including 1822 packets, datagrams, and TCP segments. These various forms are generically referred to as "packets". Packets are passed into the system protocol process from either an applications protocol process or the ARPANET interface. Packets from the ARPANET are passed into the system protocol process by intrinsic calls to the INP interface. User generated network packets are passed to the system protocol process by using a combination of message files and data buffers. Message files are used to transfer control and status information while data transfer is done with buffer-to-buffer copies between the user protocol data segment and the system protocol data

segment.

All read and write commands are done without wait to allow the system protocol process to simultaneously multiplex I/O channels and process network packets. I/O multiplexing is implemented through the IOWAIT intrinsic. The system protocol process issues an IOWAIT intrinsic after it finishes processing a data packet. The IOWAIT intrinsic returns the file number of the I/O channel associated with an I/O completion wakeup.

When the number of free buffers falls below a prescribed limit, an attempt is made to free buffers through data compaction. The attempt begins with a search for datagram fragments and unacknowledged TCP segments which waste buffer space by using only a fraction of the available space in each buffer assigned to them. This lack of efficiency can be particularly damaging because there is no guarantee that the data contained in the buffers will ever be processed. Wherever possible, datagram fragments are combined into a single datagram fragment and TCP segments are combined into a single segment to more efficiently utilize system buffers. Any buffers freed by this compaction process are returned to the freelist.

Network packets from both the user process and the ARPANET are processed along one of a number of data paths in the system protocol process. The actual data path taken depends on the type of data packet and, in the case of TCP segments, the state of its

associated network connection. Packet processing is performed by a series of function calls which act as processing steps along the data path.

In order to avoid processing delays which can tie up system resources, each arriving data packet is processed through as much of the protocol software as possible. Processing of a packet is suspended only when the lack of some resource or some external event prevents further processing.

7.2.3 Control Structures

All of the status information both for individual network connections and for the system protocol software as a whole is kept in a set of control blocks as well as in a number of buffer list structures. The control blocks include a general network resources control block, a foreign host control block for each foreign host connected to the local host, and send and receive control blocks for network connection. The list structures include a network buffer free list, a TCP buffer aging list and an Internet buffer aging list.

7.2.3.1 Network Resources Control Block

The Network Resources Control Block contains the information needed to maintain the network buffer free lists and aging lists.

This information includes pointers to the network buffer free lists and aging lists and a count of the buffers in each of the lists.

The information contained in the Network Resources Control Block is used by the protocol software to control the distribution of network buffers among the various lists. The information is scanned at various times to determine the allocation or disposition of a particular network buffer. The determinations occur when new buffers are allocated from the free list and when buffers containing TCP segments are about to be acknowledged. Decisions are made based on the number of free buffers available and the priority of the task requiring the buffers.

7.2.3.2 Foreign Host Control Blocks

Foreign Host Control Blocks maintain flow control within the 1822 protocol layer. The block contains a counter for the number of outstanding 1822 packets sent to a single host. The counter includes all of the packets sent to the host on all sockets. The counter is incremented when an 1822 packet is sent and is decremented when either a RFNM or an Incomplete Transmission is received from the host.

The counter is used to prevent transmission of too many 1822 packets to a single host. All transmission from the host is blocked when the counter reaches the limit of eight outstanding 1822 packets for any foreign host.

The 1822 level flow control is actually implemented by the send side of the TCP software. The TCP checks the RFNM count in the connection control block before it tries to transmit a segment to the foreign host.

7.2.3.3 Connection Control Block

Each TCP connection has an associated control block. The control block contains data associated with the Transmission Control Block (TCB) along with other connection related information. Specific information included in the control block is as follows:

- a connection state variable used to maintain the connection state,
- the local port number of the connection,
- the TCP interface control block number associated with this connection,
- the file number of the private message file associated with this connection,
- the TCB data associated with the receive side of this connection,
- the TCB data associated with the send side of this connection,

- A pointer to any buffers containing unacknowledged data received on this connection.

7.2.3.4 Network Buffer Resources List Structures

Three list structures are used to maintain the network buffer resources shared by all of the sockets. These list structures include the free list and the two buffer aging queues.

The network buffer free list contains all of the network buffers currently available for use by any socket. These buffers are allocated when new data comes in from either the network or a user protocol process.

The Internet Aging Queue is a list of active buffers assigned to blocked datagram fragments and complete datagrams. These buffers are the first to be reclaimed when there are no free buffers available. The Queue is sorted according to datagram age. All buffers which belong to the same datagram are combined into a single list structure. The datagram list structures are linked into the Internet Aging Queue with the least recently updated datagram always at the head of the queue. When a new datagram fragment comes in it is moved to the end of the queue along with all of the other fragments which belong to the same datagram.

The TCP Aging Queue is a list of buffers which contain at least parts of unacknowledged TCP segments. These buffers can be

reclaimed when there are no free buffers and no buffers on the Internet aging list. The Queue is sorted by socket. All buffers which contain data for the same socket are combined in a buffer list and each buffer list is linked into the queue. The queue is sorted by the age of the data associated with each socket. Data belonging to the socket which has been inactive for the longest period of time is placed at the head of the queue so it can be recycled first. When a user process reads data from a connection, all the network buffers still waiting to be read on that connection are moved to the end of the TCP aging list. This assures that data associated with an active TCP connection will not be recycled ahead of data associated with an inactive TCP connection.

8 TCP-TAC

Since the last Quarterly Technical Report in June, progress has been made in several areas. These areas include making the 316 TIP run as a host; design of the TCP module and data structures; and the design of how NCP and TCP will be integrated together in the TAC.

The 316 TIP has been converted to run as a stand alone host. This was done using an existing H-316 test host program. The TIP was integrated into this program; it was made to cycle and connections were opened to other hosts. Substantial work remains to modify the program to be supportable and to make it compatible with TCP.

The basic data structures for the TCP module have been designed. These will be used for reading and writing data from the 1822 host interface, reassembly of Internet datagram fragments, sequencing of TCP segments, and retransmissions of TCP segments. In addition, the NCP portion of the H-316 will be modified to use the same basic data structures.

The basic design of how TCP/IP and NCP will be integrated in the TAC is complete. Many details remain to be worked out, but there is an understanding about how data will flow through the different modules and how the data structures will be laid out.

DISTRIBUTION
[QTR 18]

ARPA
Director (3 copies)
Defense Advanced Research Projects Agency
1400 Wilson Blvd.
Arlington, VA 22209
Attn: Program Manager

R. Kahn
V. Cerf
J. Dietzler

DEFENSE DOCUMENTATION CENTER (12 copies)
Cameron Station
Alexandria, VA 22314

DEFENSE COMMUNICATIONS ENGINEERING CENTER
1850 Wiehle Road
Reston, VA 22090
Attn: Manoj Dharamsi
Attn: Lt. Col. William Dlugos, R540

DEPARTMENT OF DEFENSE
9800 Savage Road
Ft. Meade, MD 20755
R. McFarland R17 (2 copies)
M. Tinto S46 (2 copies)

ELEX3101
Naval Electronics Systems Command
Department of the Navy
Washington, DC 20360
Attn: Barry Hughes

ELEX3301
Naval Electronics Systems Command
Department of the Navy
Code 3301
Washington, DC 20360
C. C. Stout
J. Machado
F. Deckleman

BOLT BERANEK AND NEWMAN INC.
1701 North Fort Myer Drive
Arlington, VA 22209
E. Wolf

DISTRIBUTION cont'd
[QTR 18]

BOLT BERANEK AND NEWMAN INC.
50 Moulton Street
Cambridge, MA 02138

R. Alter
A. Owen
G. Falk
R. Bressler
A. Lake
J. Robinson
A. McKenzie
F. Heart
P. Santos
R. Brooks
W. Edmond
J. Haverty
E. Killian
D. McNeill
M. Brescia
A. Nemeth
B. Woznick
R. Thomas
R. Koolish
W. Milliken
S. Groff
M. Hoffman
R. Rettberg
W. Mann
P. Carvey
D. Hunt
P. Cudhea
L. Evenchik
D. Flood Page
J. Herman
J. Sax
R. Hinden
G. Ruth
S. Kent
Library