Bolt Beranek and Newman Inc.

(12)

Report No. 5129

Combined Quarterly Technical Report No. 26

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

August 1982

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	2. GOVT ACCESSION NO. AD-A119776
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle) Combined Quarterly Technical Report No. 26	5. TYPE OF REPORT & PERIOD COVERED Quarterly Technical 5/1/82 to 7/31/82
	6. PERFORMING ORG. REPORT NUMBER 5129
7. AUTHOR(s) J. F. Haverty	8. CONTRACT OR GRANT NUMBER(s) MDA903-80-C-0353 & 0214 N00039-82-C-0412 N00039-80-C-0664 N00039-81-C-0408
9. PERFORMING ORGANIZATION NAME AND ADDRESS Bolt Beranek and Newman Inc. 10 Moulton Street Cambridge, MA 02238	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS ARPA Order No. 3214
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Advanced Research Projects Agency 1400 Wilson Boulevard Arlington, VA 22209	12. REPORT DATE August 1982
	13. NUMBER OF PAGES 74
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) DSSW Room 1D The Pentagon Washington, DC 20310 NAVELEX 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 Protocol, UNIX, Pluribus Satellite IMP, Remote Site Module, Remote Site Maintenance, shipboard communications, VAX, ARPANET, Internet. (Interface Message Processor);

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 Internetwork monitoring; on shipboard satellite communications; and on the development of Transmission Control Protocols for the HP3000 and VAX-UNIX.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Report No. 5129

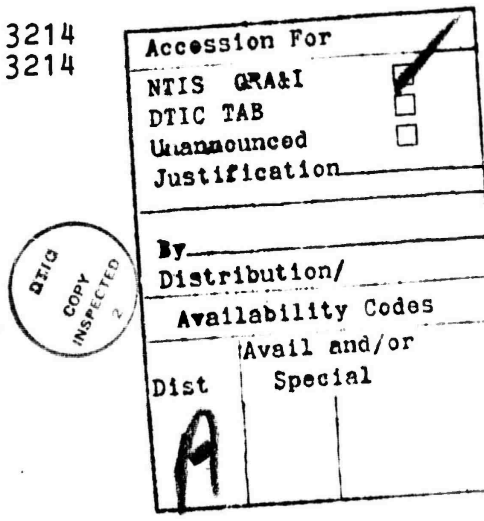
COMBINED QUARTERLY TECHNICAL REPORT NO. 26

SATNET DEVELOPMENT AND OPERATION
PLURIBUS SATELLITE IMP DEVELOPMENT
REMOTE SITE MAINTENANCE
INTERNET OPERATIONS AND MAINTENANCE
MOBILE ACCESS TERMINAL NETWORK
TCP FOR THE HP3000
TCP FOR VAX-UNIX

August 1982

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

N00039-82-C-0412
MDA903-80-C-0353, ARPA Order No. 3214
MDA903-80-C-0214, ARPA Order No. 3214
N00039-80-C-0664
N00039-81-C-0408



Submitted to:

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

Attention: Program Management

Table of Contents

FIGURES

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) Internet Operations, Maintenance, and Development; (5) development of the Mobile Access Terminal Network; (6) TCP for the HP3000; and (7) TCP for the VAX-UNIX. This work 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, MDA903-76-C-0214, MDA903-76-C-0252, N00039-79-C-0386, and N00039-78-C-0405.

2 SATNET DEVELOPMENT AND OPERATION

As part of our participation in the Atlantic Packet Satellite Experiment (SATNET) during the last quarter, support for ARPANET standard HDH protocol was added to C/30 Satellite IMPs; testing has progressed to a point where we need HDLC hardware to proceed further. This implementation, modifications to the Satellite IMP configuration mechanism, modifications to host monitoring, modifications to type-of-service parameters, and the overall SATNET hardware maintenance operations are described in the following sections. Some of our other activities are described below.

After the Raisting C/30 Satellite IMP was delivered to BBN from BBNCC, we placed the machine in the 220 volt testbed to ensure that the hardware undergoes a thorough burn-in session. In the first week, standard ARPANET test programs were run to test processor, memory, and standard interfaces. In the second week, the I/O interface daughter boards for the PSP terminal were attached, and the Satellite IMP software was run. Without a PSP terminal, correct operation of the PSP terminal interface cannot be verified; however, these daughter boards were interfaced directly with the Raisting PSP terminal when it was being checked at Clarksburg. The C/30 is currently in transit to Raisting.

Among our other activities during the past quarter, tape cassettes were written for loading the Raisting and the Clarksburg C/30 Satellite IMPs. SATNET performance tests were conducted with gateway support personnel. Work on incorporating the Native Mode Firmware System (NMFS) in Satellite IMPs was started (in NMFS, the emulated machine is altered to create one more suited to communications applications, having greater throughput capacity and reduced latency).

2.1 HDH Interface

Apropos of the wide acceptance of the CCITT standard High Level Data Link Control (HDLC) protocol, support for the ARPANET standard HDLC Distant Host (HDH) protocol has been provided in SATNET C/30 Satellite IMPs beginning with software version 4.5:2. Like SATNET's other host interfacing options, which include 1822 local host, 1822 distant host, and VDH, the HDH host interface retains the specially-developed Host/SATNET protocol at the network access level. Link level flow control, error control, and sequencing are provided by the HDLC protocol, while line quality monitoring, segmenting messages into packets, rebuilding messages from packets, simulating ready line up/down signalling, and operating loopback modes are provided by the higher level HDH protocol.

HDLC Protocol Support

The HDLC protocol defines a link access procedure for data interchange across the link between data terminal equipment (DTE) and data circuit-terminating equipment (DCE). Because the ARPANET IMP HDLC implementation, specified in Appendix K of BBN Report 1822, has been ported without major alteration to the Satellite IMP environment, a high degree of compatibility exists between the ARPANET and SATNET realizations of HDLC. Both incorporate a finite state machine table specified in the Protocol Interpreter Table Definition Language (PITDL), and both use a special driver program to interpret table entries for performing the required actions. Hosts are defined as DTEs, while ARPANET IMPs and SATNET Satellite IMPS are defined as DCEs. System parameters are identical to those specified in Appendix J of BBN Report 1822 and are presented below.

T1	more than 3 seconds
T2	approximately 1 second
N2	20
K	7
N1	1088 bits.

Communication may be in asynchronous response mode (LAP) or in balanced mode (LAPB).

HDH Protocol Support

The HDH protocol, defined in Appendix I of BBN Report 1822, specifies both a message mode and a packet mode of operation. With the former, each message travels as a single HDLC packet, while with the latter, each message is segmented into several smaller packets framed with start-of-message and end-of-message flags. The current Satellite IMP software, however, supports packet mode only. The first packet contains the 12-octet Host/SATNET Protocol header only, while the remaining packets are restricted to an even number of octets, up to a maximum of 126 octets (63 16-bit words) per packet. The total message size must be an even number of octets, limited to the normal SATNET maximum message size of 256 octets.

The HDH protocol also specifies line quality monitoring operations, using hello packets transmitted by the Satellite IMP and echoed by the host. If the Satellite IMP successfully receives M consecutive echoes (currently 20), it declares the ready line up. Conversely, if the Satellite IMP misses K out of any N consecutive hellos (currently 4 out of 16), it declares the ready line down.

Implementation

The porting procedure for moving the HDLC and HDH software from the ARPANET to SATNET first required a program to convert C/30 ASM assembler source files into PL30 macro-assembler source files. After reassembling the source, we verified that the binary generated was identical. The next step was to adapt the HDLC package to the Satellite IMP environment, a straightforward procedure, since HDLC was written with a variety of environments in mind. The most difficult part was redesigning the initialization code, which in ARPANET relied upon the presence of the IMP's CONFIG words (read from PROM) that are in a special format.

Although the HDH protocol is a bit-stuffing packet-framing protocol, the HDH software is sufficiently isolated from the particular packet framing being used by the HDLC hardware that the Satellite IMP software can be configured to operate the HDH protocol over a bisynch modem. Therefore, we were able to perform software checkout preliminary to the availability of HDLC hardware. In the initial software checkout, the Satellite IMP internal gateway was configured to act as a host on an IMP, both of which are running the HDH package but communicating over a bisynch modem line.

2.2 Satellite IMP Configuration Mechanism

With the introduction of C/30 hardware into SATNET, the host interfacing options expand to include 1822 Host-to-IMP interfaces and HDH interfaces as alternatives to the VDH interface. Because the configuration mechanisms originally implemented were inadequate to handle the new host options, we had to modify the Satellite IMP software. The new configuration mechanism is described below.

Most of the host processing within the Satellite IMP references not the 16-bit SATNET address of the host but an internal index ranging from 0 to 17 called the port number. Despite the uniqueness of host addresses among all Satellite IMPs, the software in each Satellite IMP is able to reference any local real host by port numbers 0 to 3 or any local fake host by port numbers 4 to 17, where the port numbers are common to all Satellite IMPs. Each real host port number is further associated with a particular physical I/O interface within the Satellite IMP, creating a two-stage logical mapping from the SATNET address to the port number and thenceforth to a particular interface. Reassignment of I/O interfaces among real hosts is facilitated by the second logical mapping. The port configuration table, in addition to providing the mapping between the real host port number and the physical I/O interface, must specify the network

access protocol and the link level protocol to be used for each real host.

Separate device configuration tables independently specify the allowable uses of each modem interface and each host interface, since the various physical devices may be dedicated to purposes other than host interfacing (e.g. satellite modem interfacing). This mechanism is useful for setting up interrupt masks, for constraining the allowable uses of each modem, and for detecting inconsistent configurations (if the port configuration table specifies use of a particular interface, then that interface must also be correctly configured in the appropriate device function table).

Each real host port is described by an entry in the Port Configuration Table (PCT), located at addresses from 200 to 203 corresponding to port numbers 0 to 3, respectively. Table entries have the form:

OPLOOI

where P, L, and I are octal digits defined as:

P -- Network access protocol:
0 -- Host-SATNET protocol
1 -- Modified 1822 protocol

L -- Link level protocol:
0 -- VDH
1 -- 1822
2 -- HDH

I -- Interface number:
1-5 -- VDH and HDH interfaces
0-3 -- 1822 interfaces

Some examples are:

000002 -- Host-SATNET protocol, VDH, interface 2.
012001 -- Modified 1822 protocol, HDH, interface 1.
001000 -- Host-SATNET protocol, 1822, interface 0.

Each modem interface is described by an entry in the Modem Function Table (MFT), located at addresses from 204 to 210 corresponding to modems 1 to 5, respectively. Allowable values are:

-2 -- ARPANET direct connection via SATNET modem
-1 -- Satellite modem
0 -- undefined modem
1 -- VDH modem
2 -- HDH modem

Each 1822 physical interface is described by an entry in the Host Function Table (HFT), located at addresses from 211 to 214 corresponding to hosts 0 to 3, respectively. Allowable values are:

0 -- undefined host interface
1 -- configured host interface

In the Honeywell 316 I/O architecture, modem 5 and host 2 share interrupt vectors, as do modem 4 and host 3. Because configuring both members of either of these pairs of devices will produce unpredictable results, the Satellite IMP initialization software is programmed to crash if such a configuration is detected.

An added complication for C/30 Satellite IMPs is that the microcode as well as the macrocode must be able to determine the assignment of each device. The microcode-specific configuration information of each Satellite IMP can be stored either in a PROM resident on the MII board or in a cassette tape on which is written the SATNET application microcode and macrocode. For greater ease of updating the configuration information, we are currently using the latter. Because disagreements between the microcode and the macrocode on the Satellite IMP configuration will produce unpredictable results, procedures have been developed to automate the production of cassette tapes, using a set of specially defined macros to generate both microcode and macrocode configuration files from the same source file.

2.3 Host Monitoring

Because of limitations in the current SATNET TENEX/TOPS-20 RECORDER program, monitoring information can be collected on only two hosts from each Satellite IMP. When the ARPANET direct connection via SATNET is enabled, monitoring information for this application is stored in the areas normally reserved for the internal gateway host, although the monitoring data does not match the host monitoring specification. Since more than two hosts can be configured with C/30 Satellite IMPs, the current Satellite IMP release allows selection of which host ports should send information to RECORDER. All the external ports are put in a table, terminated by minus 1, such that the order of the ports in the table determines the order in which the information is reported to RECORDER. The default state of this table is to report information first about port 0 (the external gateway), then about port 3 (the Internal Gateway), and finally about the other two host ports.

A separate software switch determines how many host blocks should actually be reported. By setting this number to two, only information about the external gateway and the Internal Gateway is reported to RECORDER. When the changeover to the more flexible NU monitoring system is completed, the number can be increased to four to enable reporting of data for all host ports.

2.4 Type-of-Service

Type-of-service in SATNET is based on three parameters--delay class, priority, and holding time--specified independently for each message delivered to SATNET. Delay class is used in the determination of the desired delivery time, which is the sum of the time-of-arrival and delay class value. Priority is used to arbitrate between messages of approximately equal desired delivery times. Holding time is the interval over which the message is kept before discarding as untimely (currently, the holding time is twice the delay class value or is the maximum defined system holding time).

Only recently did we discover that patches in the code designed to limit the priority and delay class values of certain types of traffic to prescribed values never worked correctly. Because SATNET has operated successfully for several years despite this oversight, we have decided to remove this feature entirely rather than correct the patches. Thus, the hosts will be relied upon to select the appropriate priority and delay class values. The delay class and priority standard values to be enforced among the SATNET internal hosts and the gateway hosts are as follows:

DELAY CLASS	PRIORITY	APPLICATION
0	0	ARPANET direct connection via SATNET
0	0	gateway interactive stream facility
0	0,1	internal hosts
1	2	normal gateway traffic

Potentially, the gateways could derive priority and delay class from the Internet type-of-service field present in each Internet datagram.

While the above changes were being implemented, we decided to change the values specified in the type-of-service category. In particular, we reduced the values in the delay class specification from {1, 5, 20, 120} seconds to {1, 2, 5, 20} seconds and the value of the maximum defined system holding time from 120 seconds to 20 seconds. Given the traffic present in SATNET, these choices are more useful and conform better to the SATNET buffer capacity.

2.5 Hardware Maintenance Operations

We provided the operational support and diagnostic assistance on hardware problems when they occurred on SATNET. Those hardware problems originating within the Satellite IMPs were fixed by BBNCC.

Unidentified interference, sometimes on the SPADE pilot signal, caused degradations of the SATNET satellite channel several times. Complicating the correction of the problem, existing mechanisms to trace the interference to its source must rely on Etam earth station personnel and are difficult and lengthy to apply. Due to software not coping well with the noise on the satellite channel, the Etam Satellite IMP crashed on several occasions, causing outages in SATNET service.

A malfunction in the SPADE common signaling generating equipment at Goonhilly caused interfence to the satellite channel reference signal, resulting in a system outage. Site personnel identified and corrected the problem.

Although we spent considerable time trying to isolate and diagnose a mysterious outage of the satellite channel, disrupting all SATNET service, we had no success. Several days later, a similar outage was finally traced by the Etam site personnel to an unreliable SPADE up-converter installed at Goonhilly. The installation coincided with our first outage.

A lengthy outage, preventing European internet traffic from reaching the ARPANET, occurred on the circuit between the BBN gateway and the Etam Satellite IMP due to a loop-back unexplainably inserted in the circuit. Because normal mechanisms to clear the loop failed, WUI had to disconnect and then

reconnect the circuit. Later, another lengthy outage occurred on the same circuit, which WUI fixed. An interruption occurred on the same circuit when BBN40 was converted from a Honeywell 316 TIP to a Honeywell 316 TAC and a C/30 IMP. On another occasion there was an outage on the circuit between the UCL gateway and the Goonhilly Satellite IMP, which British Telecom fixed. An outage in Line #77 occurred when the SDAC IMP Pluribus hardware was being repaired.

On several occasions modems in the 9.6 Kb/s circuit between the Goonhilly Satellite IMP and the London IMP entered into an unknown state creating a one-way circuit. Following usual procedures, the corrective action taken was for Goonhilly site personnel to loop and immediately thereafter unloop the modem. Primary power interruptions at Goonhilly caused several other outages in that site.

On a bright note, the Raisting PSP terminal installation by COMSAT proceeded flawlessly with only minimal interruption to SATNET service for aligning transmit powers.

3 PLURIBUS SATELLITE IMP DEVELOPMENT

3.1 Summary of Activities

Support of the Wideband Satellite Network operations represented a major effort during the quarter. Network usage increased leading up to and following the June 3rd packet speech demo. A number of additional features were identified as desirable for the PSAT software and several of these features were implemented. During the quarter, BSAT software development began in earnest. A preliminary BSAT software design document was written, and a summary of it is included in this report. Progress has been made in the coding of BSAT software for buffer management, system initialization and configuration.

During May, BBN froze the PSAT software, and gave experimenters full time access to the channel so that they could prepare for their demo. The DARPA Packet Speech Project conducted a successful demonstration on June 3, 1982. The demonstration included PCM and LPC point-to-point and conference calls between Packet Voice Terminals (PVTs) on LEXNETs at Lincoln Laboratory, ISI, and SRI. In addition calls were made between LEXNETs at Lincoln Laboratory and ISI and the Packet Radio Net mobile van at SRI.

PSAT software used for the demo has been designated as

network version 1.000.00 in BBN's numbering system. The first digit will change only for a major rewrite of most of the PSAT software. The next three digits are incremented following modest changes that affect functionality or fix major software bugs. The last two digits are for fixes to minor bugs and for any patches that are made to the code. After the demo, channel usage returned to the previous system of allowing experimenters access on Monday, Wednesday and Friday, with BBN using it on Tuesday and Thursday for network development.

PSAT software development continued, working on BBN's backroom test machines during May, and then moving back online for network testing and debugging during June and July. Software was added to handle stream synchronization, host interface configuration, and datagram fragmentation. The stream synchronization and host configuration processes were described in detail in the previous QTR (May 1982).

The code to allow the PSAT to fragment datagram bursts was completed and tested. Datagram bursts which do not fit in the remaining space in the current frame are divided, and sent in fragments in consecutive frames. This allows more efficient use of the satellite channel. Previously, bursts which did not fit in the current frame were transmitted in their entirety in the next frame. Bursts which were bigger than a

frame were never transmitted, and blocked further traffic on the channel until a timeout condition was met.

PSAT hardware performance during the quarter was generally good. Several minor problems were fixed by BBN Computer Corp. Field Service personnel. A bus coupler was replaced in the DCEC PSAT during May. During June, a DMA card in the PSAT interface to the LEXNET miniconcentrator was replaced at SRI, and a bad memory card was replaced in the ISI PSAT. In addition, the ISI PSAT had a few problems with blown fuses during June and July. The problem was tracked down to a bad solder joint in the fuse holder which caused the fuse to overheat. The fuse holder was replaced.

DCEC did not participate in the June 3rd demo due to an inoperable ESI that could not be repaired in time. A fan in the Burst Test Modem (BTM) at DCEC failed during July causing the BTM to overheat and fail. The BTM was returned to Linkabit for repair. Lincoln Laboratory's High Speed Packet Modem (HSPM) was identified as the source of a spurious interfering signal on the channel. It was also returned to Linkabit for repair. During the HSPM's absence, the BTM was substituted for it and the Lincoln Laboratory site remained operational.

During the quarter each of the sites experienced outages of a couple of days due to earth station problems. The outages were

usually caused by malfunctions in the uplink or downlink converters. Some effort was made during July to track down and correct site-to-site variations in local oscillator frequencies and output power levels.

The remainder of this section of the QTR is a summary of the BSAT software design.

3.2 BSAT Software Structure

All software in the BSAT is contained in a collection of process groups. A process group is a segment of code which is independently linked, and is loaded from a separate file via the Butterfly load device. Each process group has a single entry point, which is a dispatch routine in the case of a process group with multiple processes. Each process group is loaded as a unit into one or more processor nodes. One or more copies of each process contained within a process group may be started on any processor node in which the group resides. Each process so started has its own private virtual address space, with the code (text) segment shared among all processes in the group.

The BSAT code can be broken down into four divisions which are relatively independent of each other: system processes, Host Protocol Module (HPM) processes, internal host processes, and

Channel Protocol Module (CPM) processes. These communicate with each other using a common memory segment (comseg) which contains global data and is mapped into every process's address space. There is also a shared data segment which is used by processes needing host-specific information (host_table_segment).

3.2.1 Processor Allocation

As yet there is no firm allocation of various processes to processor nodes, since this is partially dependent on factors such as code size, memory availability, processor utilization, and throughput requirements. However, there is a tentative layout using a minimum of 5 processor nodes. Three of these nodes contain one process each: the CPM scheduler (Scheduler), CPM output (CPM_Xmit), and CPM input and downlink delivery (CPM_Rcve). These processes are the critical processes attached to the satellite channel. The fourth processor node contains all internal hosts, the CPM sync process, the system processes, and the local delivery process. One or more additional nodes will be devoted to hosts, and would run the HPM processes associated with the host interface(s) on that node.

This allocation of processor nodes is designed for maximum throughput along critical paths, assuming no major redesign of

the PSAT algorithms. If lower throughput were acceptable and sufficient memory were available on each processor node, it would be possible to run the BSAT with fewer processor nodes with no software changes. It is straightforward to add more nodes to increase throughput in the CPM uplink and downlink processes, the local delivery process, CPM sync process, and HPM processes. Replicating the scheduler for increased throughput may involve some redesign of the PSAT algorithm and requires further investigation. We do not believe, however, that a single scheduler will be a major bottleneck for operation on the current 3 Mb/s channel.

The following sections are outline summaries of the major BSAT software modules.

3.2.2 System Processes

There are two processes in the BSAT which are responsible for creating the BSAT in an otherwise empty Butterfly multiprocessor. TopLevel is the boot process, which is initially loaded into the machine along with the Chrysalis operating system. The process Initialize builds all other processes and memory structures in the system. Each of these processes is in its own process group.

Process: TopLevel

Description:
This is the top-most process of Butterfly Satellite System. It creates the common memory segment.

PSAT correspondence: PSAT Control process

Interaction with other processes:
Runs the Initialization subprocesses of the BSAT until it completes, and then falls into a TTY command loop.

Process: Initialize

Description:
This process creates and initializes a number of global tables in the BSAT, and advertises them in the common segment. It also interprets the site configuration table to set up all external and internal host processes, CPM processes, and any other processes of the BSAT. Initialize allocates buffer pools in all processors and then exits when finished.

PSAT correspondence: PSAT configuration code

Interaction with other processes: Creates them.

3.2.3 HPM Processes

HPM processes are divided into three process groups: I/O level processes, Host Access Protocol (HAP) level processes, and the local delivery process. Each external host port on the BSAT has a copy of the I/O level and the HAP level group. These are separate process groups in anticipation of the addition of ARPANET and other Internet gateway processes which may be substituted for the HAP level code to support internetwork

operation.

I/O level process group

The next two processes form the external host hardware I/O drivers. Each external host port will have one of each process.

Process: Host_Rcve

Description:
This process accepts input from a single host on a single hardware device and passes the message to HostIn for HAP protocol processing.

PSAT correspondence: MSGIN, HINPOL

Buffers accessed: Host buffer pool on local processor

Interaction with other processes:
Runs in privileged mode to operate host interface hardware. Puts incoming messages on a queue for HostIn.

Process: Host_Xmit

Description:
This process sends the next message to a single host on a single hardware device. HostOut determines which message is sent.

PSAT correspondence: MSGOUT, HOUTPOL

Buffers accessed: Host buffer pool on local processor

Interaction with other processes:
Runs in privileged mode to operate host interface hardware. Receives a list of messages to send via a queue from HostOut.

HAP level process group

The next three processes form the HAP level drivers for external hosts. A set of these processes exists for each external HAP host on the BSAT.

Process: HostIn

Description:
This process performs accept/reject decisions and other HAP related functions for external hosts.

PSAT correspondence: HOSTIN

Buffers accessed: Host buffer pool on local processor

Interaction with other processes:
Gives messages to CPM uplink queues and to process Local for delivery. Receives messages from Host_Rcve.

Process: HostOut

Description:
This process selects what to send to an external host.

PSAT correspondence: HOSTOUT

Buffers accessed: Host buffer pool on local processor

Interaction with other processes:
Takes messages from host's priority output queues; places messages on Host_Xmit input queue.

Process: HostStatus

Description:
This process handles the sending and receiving of HAP status messages. It communicates with HostIn and HostOut via private queues.

PSAT correspondence: (part of HOSTIN and HOSTOUT)

Buffers accessed: Host buffer pool on local processor

Interaction with other processes:
Receives incoming status messages from HostIn, generates status messages and sends them to the host via HostOut.

Local delivery process group

Local hosts are defined as the internal and external hosts associated with a particular Wideband Network site. Local delivery of messages is performed by a separate process group rather than as part of HostIn for efficiency reasons based on the Butterfly hardware design. The current design calls for only one copy of Local to be created, which will reside in the Internal Host processor node. This is based on the fact that most traffic will probably be inter-node rather than intra-node. Multiple copies of this process can reside on multiple processor nodes, and could provide increased throughput if unexpected intra-nodal traffic develops.

Process: Local

Description:
This process performs local host to local host delivery of messages. All incoming messages go either through Local or to the CPM uplink.

PSAT correspondence: HDLIV

Buffers accessed: All host buffer pools on all processors

Interaction with other processes:
Removes messages from its single input queue, moves the messages from the source host's buffer pool to the

destination host's buffer pool, and places a message on the priority output queue of the appropriate host.

3.2.4 Internal Host Processes

The next several processes implement various internal hosts in the BSAT. All internal hosts read messages from their host priority input queues using the routine NextMsg. They also place messages on the CPM uplink queues, or the process Local's input queue using the routine SendMsg. They only reference buffer pools on the local processor. All internal hosts are members of a single process group, which will be loaded into a single processor. Since these processes are relatively low bandwidth, this seems to be the most efficient organization.

Process: Cassette_Host

Description:
The Cassette Host handles requests to read or write the cassette tape. It is principally used to update the BSAT program image on a cassette when a new release occurs.

PSAT correspondence: None

Process: Control_Host

Description:
This host is responsible for stream and group sync database messages. It is also used during initialization of the ESI, and interacts with the CPM initial acquisition code. Control_Host handles most BSAT/ESI local control message traffic.

PSAT correspondence: PSAT Control Host

Process: Echo_Host

Description:
The Echo Host will send back any messages sent to it. It will swap both the local and Internet header source and destination addresses. Note that the Echo Host will receive messages sent both to it as a specific host, and to it as the Satellite Echo Host. If acting as the Satellite Echo Host, then the returned message is flagged, to force sending over the channel.

PSAT correspondence: Echo Host

Process: Measurement_Host

Description:
The Measurement Host is responsible for collecting BSAT performance statistics during tests and for sending the results to NU network monitoring system for further processing. It also handles EXPAK parameter change messages for controlling experiments. There is some possibility that the functions of this host may be split up and moved to the monitor and control hosts.

PSAT correspondence: EXPAK host

Process: Monitor_Host

Description:
The Monitor Host is responsible for continuously collecting statistics about BSAT operation such as traffic levels and channel usage. It sends the results to NU at the BBN NOC for further processing.

PSAT correspondence: Monitor host

Process: Msg_Source

Description:
The message source is an eight component generator of messages. Eight message generators with different parameters can be running simultaneously. The frequency, size, destination and type of messages generated are controllable by external parameters.

PSAT correspondence: MSGGEN

Process: Msg_Sink

Description:
The message sink will discard any message sent to it. It will also keep statistics about any errors in the received messages.

PSAT correspondence: MSGSNK

Process: Service_Host

Description:
The Service Host handles requests relating to stream and group services of the BSAT.

PSAT correspondence: Service Host

3.2.5 CPM Processes

There are four processes in the current CPM design. For maximum flexibility, each process is in a separate process group. This is necessary for both architectural and throughput reasons. The CPM uplink path consists of the Scheduler and CPM_Xmit processes. The downlink process is CPM_Rcve, and channel synchronization is performed in Channel_Sync. It may become

necessary to split out the data aggregation functions from the scheduler process to increase BSAT throughput, but the tradeoffs here are not yet clearly understood. The current design is believed adequate for an initial release and sufficiently modular to be modified should a split eventually become desirable.

The ESI will be connected to the BSAT by more than one physical link in order to achieve the 3 Mbps data rate. To maximize PSAT throughput, the uplink and downlink paths in each link should be separated as well. Therefore, the ESI device driving processes (CPM_Xmit and CPM_Rcve) must be in separate process groups and there may be multiple copies of these processes running in parallel.

Process: Scheduler

Description:
This process does all the "sequential" CPM processing for the uplink side. It schedules each frame, including reading the datagram and host stream queues, making reservations, and scheduling control packets.

PSAT correspondence: CKFBLK, RSVINS, SETINS, STMAGG, SKED, BUILDF

Buffers accessed: All host buffer pools.

Interaction with other processes:
Uses stream database built by service host, sync information from Channel_Sync.

Process: CPM_Xmit

Description:
This process is the ESI uplink device driver. The Scheduler informs it when and what to send. This process does any necessary buffer copying from the host buffer pools. Datagram fragmentation is also done by this process.

PSAT correspondence: SATOUT

Buffers accessed: All host buffer pools and uplink burst buffers.

Interaction with other processes:
Runs in privileged mode to drive ESI interface hardware. Reads transmit queue from Scheduler and composes bursts.

Process: CPM_Rcve

Description:
This process accepts bursts from the ESI, discards messages not for this site, and moves messages for local delivery into the destination host buffer pool. It will also put the identifier of the message on the appropriate host priority queues.

PSAT correspondence: SATIN

Buffers accessed: All host buffer pools and downlink burst buffers.

Interaction with other processes:
Runs in privileged mode to drive ESI interface hardware. Reports channel activities to Channel_Sync. Interprets group destination addresses and steps the message reference count to reflect multiple ownership.

Process: Channel_Sync

Description:
This process performs datagram reservation synchronization checking.

PSAT correspondence: RESSYN

Interaction with other processes:

Uses the channel activity reporting from CPM_Rcve and the scheduling information from Scheduler to maintain reservation synchronization.

4 REMOTE SITE MAINTENANCE

4.1 Introduction

The Remote Site Maintenance (RSM) project continued routine support work for the operational sites. In addition, the accounting system described in QTR 25 and the GL Network Graphics Backend program were installed at the NOSC ACCAT site. Also, a DM-11 modem control unit was configured, tested, and shipped to NOSC for installation there.

The major development work during the quarter centered on coding of the UPDATE automated software distribution program (previously described in QTR 24), and design for a conversion of the BBN TCP/IP Host Network Protocol software, which was originally designed for the VAX family of computers running 4.1BSD UNIX, to BBN-UNIX running on the PDP-11/70. Transporting this software will allow the RSM sites to keep fully up to date with the conversion of ARPANET host protocol software to TCP/IP, which was mandated by the Defense Communications Agency (DCA) to take place on 1 January 1983.

In addition to the maintenance and development work that took place, we visited two of the RSM installations on July 13-15: the Central Operational Site at NOSC in San Diego, CA, and the Remote Site at the Naval Postgraduate School in Monterey, CA.

The visits served to acquaint us with personnel at the sites, to discuss the progress of the project and the level of service being offered, and to make plans for the future of the project.

4.2 Transport of TCP/IP to the PDP-11

4.2.1 Implementation Characteristics

The existing BBN VAX TCP/IP, which would be used as a starting point for a PDP-11 implementation, attempts to be a complete implementation of the protocols. It is designed for production use, to support a large number of connections with good performance and operational features. Features of the IP module include:

- IP checksums.
- Handling IP options.
- Fragmentation and reassembly support.
- Extended internet address support.
- Communication with gateways using the Internet Control Message Protocol (ICMP), to process redirects and error messages.
- Support of multi-homing, the ability to have a host with multiple internet addresses and interfaces on the same or different networks.

Features of the TCP module include:

- TCP checksums.

- TCP options and the ability to pass options through to the IP module.
- TCP sequencing.
- Advanced windowing and packetization algorithms, and adaptive retransmission strategy.

General features include:

- Direct user access to IP and local net protocol levels.
- User Datagram Protocol (UDP) implementation.
- Support of multiple network interfaces, local network protocols, and device drivers.

The TCP/IP was described in detail in QTR 19.

4.2.1.1 Kernel Space Limitations

Previous TCP/IP implementations for the PDP-11 family have been constrained by the limited 16-bit address space of the architecture. This limits the maximum size of the UNIX kernel on a PDP-11 to 64K bytes of text and data, or 64K bytes of text and 64K bytes of data in machines with separated instruction and data (I&D) spaces, such as the PDP-11/70. To overcome this limitation, earlier implementations have separated the networking code into a kernel resident portion and a daemon process operating in user space. This separated kernel/daemon technique has significant negative impact on the organization, performance, and operational characteristics of a networking implementation.

The TCP/IP implementation for the VAX running UNIX is entirely kernel resident, aided by its larger 32-bit address space. There are several advantages to a kernel resident implementation:

- The networking code is always resident in the operating system and available, enhancing performance, as opposed to a user level daemon which may be swapped out.
- Existing kernel services, such as memory management and timers, can be accessed directly, reducing code size and complexity.
- Additional non-standard mechanisms for interprocess communication (IPC), such as ports or shared memory schemes, are unnecessary, since the user can access the networking service directly with system calls (like read and write).

Given the superiority of a kernel resident implementation, the address space limitation problem remains to be solved for the PDP-11, since the current BBN-UNIX kernel (modified version 6) has only 8K text bytes remaining with the old NCP implementation and 20K text bytes without it. The VAX TCP implementation requires approximately 21K bytes of text and 5K bytes of static data.* Even if the VAX TCP/IP implementation could be trimmed to fit in the non-NCP BBN-UNIX kernel, there would be no room left for further development.

*Code size is roughly comparable between the PDP-11 and the VAX, since single VAX instructions are longer than single PDP-11 instructions, but can replace several of them.

One solution to the kernel space limitation problem would be to adopt one of the several kernel overlay schemes now available. The overlay mechanism allows sections of kernel code to be mapped separately into the address space as they are called. We have chosen to adopt a kernel overlay scheme designed by the Berkeley Computer Science Research Group for their version of UNIX for the PDP-11. Adoption of such a scheme involves modification of the loader to recognize and mark overlay sections, and minor compiler modification to generate suitable subroutine calls.

4.2.1.2 Portability Considerations

Aside from the kernel space limitation problem, there are several other issues involved in adapting the VAX UNIX TCP for PDP-11. A number of portability considerations must be dealt with:

- Isolation of structure and variable declarations that depend on the fact that ints are 32-bit values on the VAX and 16-bit values on the PDP-11.
- Differences in the way kernel tables are allocated. PDP-11 tables are statically declared, whereas VAX tables are allocated dynamically and are referred to through pre-initialized pointers.
- The VAX C compiler allows for identifiers with names longer than 8 characters, whereas the PDP-11 C compiler enforces this limitation.
- Use of kernel supplied routines must be checked for consistency of function and use between VAX and PDP-11

versions of UNIX. Where necessary, some routines must be implemented where similar functions do not exist.

- Modifications must be made to parts of the PDP-11 kernel to interface the TCP/IP.

4.2.1.3 Memory Management

The major task in transporting the TCP/IP implementation to the PDP-11 is developing memory management and buffer management routines. The TCP/IP implementation uses a modular set of buffer allocation routines to manipulate fixed size network buffers. The VAX TCP allocates network buffers dynamically by stealing page frames from the VAX UNIX paging system. The VAX paging system allows memory access for instructions and data to be non-contiguous with no fragmentation problems. The PDP-11 UNIX memory management requires that instructions and data be loaded into contiguous memory and addressed contiguously. Thus, the allocation of memory for network buffers must pay attention to contiguity, to avoid excessive fragmentation of memory. In addition, the allocation of free memory interacts with process scheduling, since network buffers may be allocated in space needed for loading processes and vice versa. A dynamic allocation scheme might then require that processes occupying desired memory be swapped out, requiring modification to the UNIX swap scheduler. Finally, the problem of address space limitation resurfaces, since even statically allocated buffer space takes up

a portion of the limited address space.

There are several alternative schemes for free memory management for network buffers. They include:

- Static buffer allocation in kernel address space.
- Static buffer allocation outside of kernel address space (in user or supervisor space).
- Dynamic buffer allocation in some address space, requiring modifications to the swap scheduler.

Static buffer allocation in kernel address space is the simplest approach, but has the disadvantage that the amount of buffering is fixed, no matter what the use of the network is. Typically, buffer usage varies significantly with the number of connections opened and activity on the network. Static buffer allocation wastes space that could be used for process loading when the network is quiescent. Allocating buffers in kernel address space takes up valuable resources. A better scheme is to allocate the buffers in some other address space, which requires special routines for moving data to and from the buffer area. Alternatives are user space or supervisor space. Not all PDP-11s have supervisor space, so user space is the best choice.

Dynamic buffer allocation allows the network buffer space to grow only as needed. High and low water marks are established based on the number of active connections, and the size of the buffer area is changed as necessary. There is an upper limit on

the amount of memory that can be consumed by the network. To allow the memory to be added to the network buffer supply, the swap scheduler must be modified to swap out processes occupying desired areas of memory.

4.2.1.4 Higher Level Protocol and Support Software

The implementation of TCP/IP also includes several higher level protocols that use the underlying host-host protocols and offer standard user services. These protocols include TELNET (virtual terminal protocol), File Transfer Protocol (FTP), and Simple Mail Transfer Protocol (SMTP). These protocols are implemented as user level programs, and should be directly portable to the PDP-11. In addition, the higher level protocol software requires support software, such as a library of host name/address translation routines and maintenance programs. These too should be directly portable to the PDP-11.

5 INTERNET DEVELOPMENT

5.1 Introduction

The major activity during the past quarter was the continued deployment and maintenance of the Macro-11 gateway. Other important work included enhanced NU gateway monitoring, work toward the removal of Line 77, defining the Exterior Gateway protocol, and supporting new Local Area Networks.

5.2 Gateway Installations

In the last quarter the remining BCPL gateways (Bragg and SRI-R2D2) were replaced with the Macro-11 gateways. Now all of the gateways are running the Macro-11 gateway. A new gateway called LOGEX was added for the exercise of the same name. The list of operational gateways is show in the following table.

Gateway	Adjoining Networks
BBN	ARPANET - SATNET
BBN-NET	ARPANET - BBN-NET
BBN-PR	BBN-NET - BBN-PR
BRAGG	ARPANET - BRAGG-PR
DCEC	ARPANET - EDN
LOGEX	ARPANET - BRAGG-PR
NDRE	SATNET - NDRE-TIU - NDRE-RING
SRI-C3PO	ARPANET - SF-PR-2
SRI-R2D2	ARPANET - SF-PR-1
UCL	SATNET - UCLNET - RSRE/NULL - UCL-TACNET

5.3 Gateway Software Releases

A new gateway software release, number 1001, was installed in the gateways in the past quarter. The new release contained the following additions and improvements:

- Loose and Strict Source Route and Record Route IP Option.
- Throughput data collection.
- Host Traffic Matrix measurements.
- Improved Message generator.
- Network slots in the gateway network table are reused.
- Code that sent GGP destination dead and redirect messages was removed.
- Software switch to restart the gateway.
- Packet Trace facility.
- Bug fixes and improved traps.

We have begun planning and working on the next gateway release number 1003. This release will include new features such as:

- Implementing ICMP Source Quench.
- Ability to use "Uncontrolled" ARPANET messages.
- BBN Fibernet support.
- Interlan 10MB Ethernet.

ICMP Source Quenches will be sent after a certain number of datagrams have been dropped over time. The gateway will count the number of packets that are dropped. This count will be cleared periodically. When a datagram is dropped and the count exceeds a fixed limit, the count is cleared, and an ICMP Source Quench will be sent in response to the datagram. This should only cause Source Quenches to be sent when a large number of datagrams are being dropped.

The ability to use ARPANET "Uncontrolled" (Type 0, Subtype 3) messages is conditional on several tests:

1. A switch in the gateway is turned on.
2. The Type of Service (TOS) in the datagram specifies low delay and low reliability.
3. The datagram is smaller than 123 bytes.

If the datagram passes these tests, it will be sent as an "uncontrolled" message. If the gateway's host port is enabled in the IMP for "uncontrolled" messages, the IMP will send it. If not, it will discard it without notifying the gateway.

5.4 Gateway Performance Measurements

In the past quarter we implemented in the gateway and in the NU monitoring system facilities to collect Throughput and Host Traffic Matrix data. This performance data is accumulated by the gateway over time and collected by the NU system with the Host Monitoring Protocol.

What follows is an excerpt of some throughput data that was collected from eight operational gateways over a period of slightly more than eight hours.

A total of 218 throughput messages were received from 8 gateways: RCC, BBN, UCL, NDRE, BRAGG, R2D2, C3PO, and DCEC.

Total time covered by data is 7 hours, 15 minutes.

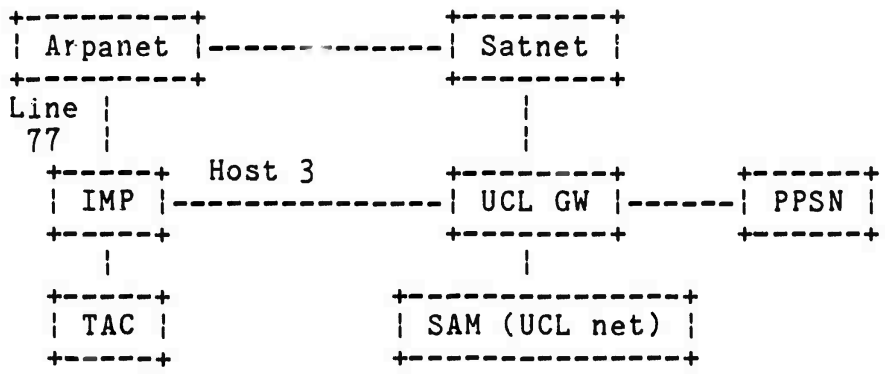
```
Total packets received:                          290119 (11.12/sec)
Total packets dropped due to IP errors:              17 ( .00/sec, 0.00%)
Packets dropped due to unreachable dest net:       1304 ( .45%)
Packets dropped due to unreachable dest host:       207 ( .07%)
Total bytes received:                          11135005 (426.63/sec)
Average bytes per packet =   38.38

Total packets sent:                              336116 (12.88/sec)
Packets dropped on sending:                        3494 ( .13/sec, 1.03%)
Total bytes sent:                              12004408 (459.94/sec)
Average bytes per packet =   35.72
```

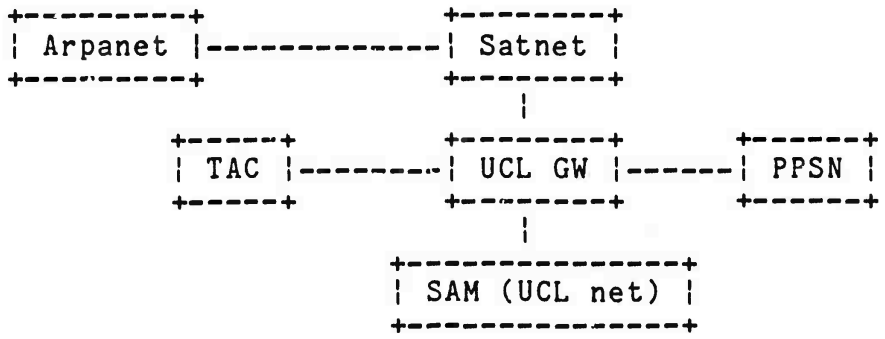
We are currently working to collect throughput data on a 24 hour basis. This data will allow us to produce summaries of daily, weekly, and monthly gateway performance.

5.5 UCL Gateway and TAC

A fourth interface was added to the UCL gateway and it was connected to the London IMP third host port. The TAC has been configured to run as network 32. All of the TCP/IP traffic to/from the TAC is routed directly to the UCL gateway. Only the NCP traffic is sent via Line 77. The current configuration is as follows:



The TAC has been configured such that all of its ports run in TCP mode and the commands to switch to a port to NCP have been disabled. The remaining step required to remove line 77 is to connect the TAC directly to the UCL gateway. At this point, the UCL IMP can be removed from service. As a partial step toward this goal, we have successfully tested a TAC connected to a gateway in the testbed at BBN. The final configuration for UCL will be:



All traffic from the networks at UCL to the ARPANET will go through the UCL gateway and, of course, must be Internet traffic.

5.6 Telenet Line

The Telenet line for the VAN gateway was installed. We are waiting for the ACC IF-11XQ/X.25 interface that BBN ordered to be delivered. When all items are complete we will begin the implementation of the VAN gateway with its special access control.

5.7 Exterior Gateway Protocol

We prepared a draft copy of an IEN describing the Exterior Gateway Protocol (EGP) and distributed it to the Internet Configuration Control Board (ICCB) for review. The EGP is a protocol to permit other gateways and systems of gateways to pass routing information to the DARPA gateways. This is to be used to form a "firewall" between the DARPA core system of gateways and the other gateways. When we have received comments and permission to proceed from the ICCB we will release it as an IEN and begin its implementation in the gateway.

5.8 Local Area Network Support

We are working on support for several Local Area Networks. We currently support the Proteon Ring network (installed in the NDRE gateway) and are working on 10Mb Ethernet and BBN Fibernet. The Ethernet will use the Interlan interface and the Fibernet uses a BBN interface.

There are several addressing issues for supporting Ethernet that need to be resolved. The most important of these is how to map the 48-bit Ethernet addresses into the 32-bit Internet addresses. We wrote IEN-212, "Local Area Network Addressing Issues", which describes a proposed approach to solving this problem. The proposal basically consists of the gateway keeping a table mapping IP to Ethernet addresses and a mechanism using the Internet Control Message Protocol to update the mapping table.

5.9 Datacommunications Paper

We wrote an article entitled "Connecting Different Types of Networks with Gateways", which appeared in the August 1982 issue of "Data Communications Magazine". The article described the DARPA approach to Internetting and contrasted it to other approaches such as X.25/X.75.

5.10 RFC Describing DARPA Gateway

We wrote a draft of RFC-823, titled "The DARPA Internet Gateway," describing the design of the Internet Gateway. We plan to release the final version in the near future.

6 MOBILE ACCESS TERMINAL NETWORK

As part of our participation in the development of the Mobile Access Terminal (MAT) and the MAT Satellite Network (MATNET) during the last quarter, we continued Phase 2 system testing and integration within the Advanced Command and Control Architectural Testbed (ACCAT) experiment at the Naval Ocean Systems Center (NOSC) in San Diego, California. The culmination of this series of testing was the achievement of a major milestone in the MATNET program; namely, the first successful at-sea demonstration of a shipboard MAT unit communicating with shore-based resources. Finally, we began work on Phase 3 of the program, which includes development of partially-ruggedized equipment for at-sea testing.

6.1 Phase 2 Testing

The Phase 2 system testing and integration was conducted on a large scale, involving participants from BBN; E-Systems, ECI Division; NOSC; Tracor, Inc.; and Information Sciences Institute (ISI). Having been unable in the available time during last January's satellite testing to complete MATNET functional testing, we gave it highest priority with emphasis on the establishment of acceptable system delay thresholds and timeout

parameters and on the verification of correct system behavior. (Satellite time is absolutely crucial for verification of MATNET functionality, which involves distributed control algorithms with multiple stations through a long-delay, low-bandwidth channel. Because of the large demand for channels on FLTSATCOM satellites, satellite time for MATNET tests has been scarce; generally only a small number of evening hours and weekends has been available for our use.) Second in priority was technology transfer, primarily to NOSC designees but also to other participants. However, due to lengthy incapacitation of both the NOSC designees, the amount of technology transfer fell short of the desired goal.

Our normal test procedure was in the first hour of satellite time to measure packet lossage on the satellite channel using message generators in the Black processors and excluding the Red systems entirely. Afterwards, we would bring up the Red systems, loading the channel with message generators in the Satellite IMP. Logistics with the Satellite IMPs, Black Processors, crypto equipment, and AN/WSC-3 radios all in different rooms was inconvenient.

Early in this testing session, we obtained disturbing results. Although we routinely were able to establish two-MAT communications through the satellite channel, reservation-synchronization between sites was being lost repeatedly at an

alarming frequency of occurrence. Because of its severe impact on system performance, solving this single problem occupied the major share of our effort. Complicating the issue is that any one of a number of malfunctioning items could cause the effect seen; packet lossage anywhere in the system could cause a loss of reservation-synchronization.

After careful searching, we believe the problem in the first half of the testing session originated for the most part within the Satellite IMP. Midway through the testing session, we had adjusted parameters in the Satellite IMP and subsequently gained substantial confidence in its operation. In the last half of the testing session, the contributing factors to loss of reservation-synchronization are ascribed to program halts in the Black processor, sticking relays in the ON-143 crypto ancillary equipment, and outside interference on the satellite channel. Details are presented below, categorized into specific independent devices or activities.

6.1.1 Satellite IMP

Initially in this testing session, the new Satellite IMP software release version 6.2:0, while performing well with only a single station on the satellite channel, was unable to achieve

reservation-synchronization when two stations were on the satellite channel simultaneously. To circumvent this problem, we returned to the January 1982 software release version 6.1:2.

Tests with version 6.1:2 initially showed a loss of reservation-synchronization much too frequently. Because we suspected a major cause of the problem to be a discrepancy in the accuracy between the master crystals in the two C/30s, a discrepancy large enough to contribute to Global Time drifts occasionally exceeding our threshold limit, we borrowed a frequency counter to measure these crystals. Measured frequencies were 15.9997 MHz and 16.0041 MHz, where the latter indicates a relative inaccuracy of 2.6 10**-4. Global Time drifts between the two units in two Hello frames from this factor alone amounts to 1.44 milliseconds. To compensate for the problem, we increased the total guard band between datagram packets from 2 milliseconds to 4 milliseconds and increased the maximum Global Time drift allowed between Global Time updates from 1.9 milliseconds to 3.2 milliseconds.

We had also considered changing the Global Time update algorithm to perform updates every Hello frame rather than every other Hello frame. The original Satellite IMP designers incorporated this feature, the direct effect of which is to avoid changes in the Global Time leader whenever a single Hello packet

from the leader is lost and thereby to reduce the potential for oscillatory behavior in Global Time. So fundamental is this feature that we were concerned that unanticipated side effects would cause other problems. Since the changes to the guard band and the Global Time update threshold yielded satisfactory performance, we decided against changing the Global Time update algorithm.

To aid us in diagnosing problems when running two MAT stations over the satellite channel, we borrowed a C/30 1822 distant host interface and temporarily connected the shipboard MAT station directly to the Private Line Interface (PLI) through the port assigned to the Command Center Network (CCN). In this way, MONITOR reports could reach the TOPS-20, even though the shipboard MAT may have blocked its satellite channel access through resetting its satellite channel interface.

In the last hour of satellite time allocated to us during this testing session, we loaded software version 6.2:0 again and inserted all the patches described above. With just a few minutes of debugging, we were able to find the bug causing the failure of this software version to achieve reservation-synchronization and were able to run the new software satisfactorily.

6.1.2 Black Processor

In the last half of the testing session, the Black processors ceased operation several times. Initially the failures were ascribed to malfunctions in the crypto interface boards in the shore-based unit, since interchanging boards between units caused the the problem to follow the boards. Replacement boards were hand carried from ECI in St. Petersburg to correct the problem. Later, a second cause of failure was traced to electrical shorting between an LSI-11 board and the brace installed to ruggedize the unit. The temporary fix for this problem was to wrap electrical tape around the brace. Afterwards, a third cause of failure occurred when the Black processors executed program HALT instructions inserted in the code whenever a 10-second timer triggers as a result of an incomplete DMA operation.

6.1.3 ON-143

Another cause of loss of reservation-synchronization in the Satellite IMPs was traced to a sticking relay in the ON-143 crypto ancillary equipment. Confusing the diagnosis of this problem is that we initially discovered that in most cases momentarily crosspatching the Satellite IMP internally would

clear the relay line; hence, we initially inferred the Satellite IMP to be at fault. After we had gained more confidence in the Satellite IMP, we began searching for other causes. Eventually we discovered that the PTI line from the Satellite IMP to the Black processor was blocked (this line is pulse-width modulated to convey the packet length to the Black processor); an oscilloscope showed pulses being sent by the Satellite IMP but not received by the Black processor. By gaining access to the crypto room, we were able to show directly on the oscilloscope the pulses being blocked at the ON-143. In retrospect, we believe this problem occurred repeatedly during the testing session; however, most of the time we used Satellite IMP crosspatching to unblock the line, so as to avoid the procedures necessary to gain access to the crypto room. Inasmuch as this problem was observed at ECI earlier on a different ON-143, sticking relays should be targeted as a weak point in the system.

6.1.4 Gateway

The new (MACRO-11) gateway software initially refused to bring up its 1822 interface to the Satellite IMP, forcing us to use the old (BCPL) gateway software for the first few days of the satellite tests. During hours when satellite time was unavailable, we traced the problem to a missing command in the

initialization sequence; manual insertion of the command allowed the new gateway software to work normally. Among its features are superior diagnostics capability and the capability to generate test message traffic.

6.1.5 TOPS-20

Last May, ISI installed TOPS-20 Version 4 on the DEC 20. Because table entries referring specifically to MATNET were lost, TCP connections could not be opened from the MATNET Terminal Input Unit (TIU), although monitoring and control programs RECORDER, MONITOR, QUERY, d EXPAK worked correctly. Once the tables were updated, TCP operations between the MATNET and the TOPS-20 were restored. Currently TCP timeouts in the TOPS-20 are set to 3 seconds.

6.1.6 CCN/MATNET Integration

When satellite time was unavailable, we configured the CCN as a host of the shipboard MAT, so that the only CCN connectivity to the ACCAT DEC TOPS-20 and the ACCAT DEC 11/70 UNIX was via MATNET. The configuration is shown in Figure 1 below, where G, 1822, LUBPI, NIU and PLI represent a DEC LSI-11/02 gateway, an 1822 Host-to-IMP standard interface, a DEC LSI-11 Ungermann Bass

Parallel Interface, a Network Interface Unit, and a Private Line Interface, respectively.

This configuration was successfully demonstrated last January by connecting MATNET to the assigned CCN port on the PLI, necessitating table changes in the Satellite IMPs and in the MATNET gateway. This time, tables in the TOPS-20 were changed to recognize tnat traffic sent to CCN should go to a different PLI port (i.e. to the MATNET port). Also, the CCN gateway was changed to recognize a pathway through MATNET whenever a flag is set. (This flag could be eliminated by converting the CCN gateway to a 3-port gateway.)

Not having satellite time, we tested the above configuration using the Satellite IMP digital satellite channel simulator. Thus, in the CCN/MATNET integration tests, the 1/4 second satellite delay was missing, but all the channel overheads in MATNET and the 9.6 Kb/s satellite channel data rate were present. In the demonstration, a CRT terminal at 9600 baud on the CCN NIU and a TI-743 terminal at 300 baud on the MATNET TIU were logged into the TOPS-20 using TCP and ran the TTYTEST program. It should be emphasized that this configuration is a daisy chain of many devices, the failure of any one of which can disrupt communications between the CCN NIU and the TOPS-20. Similarly, diminished performance in any single device in the chain can

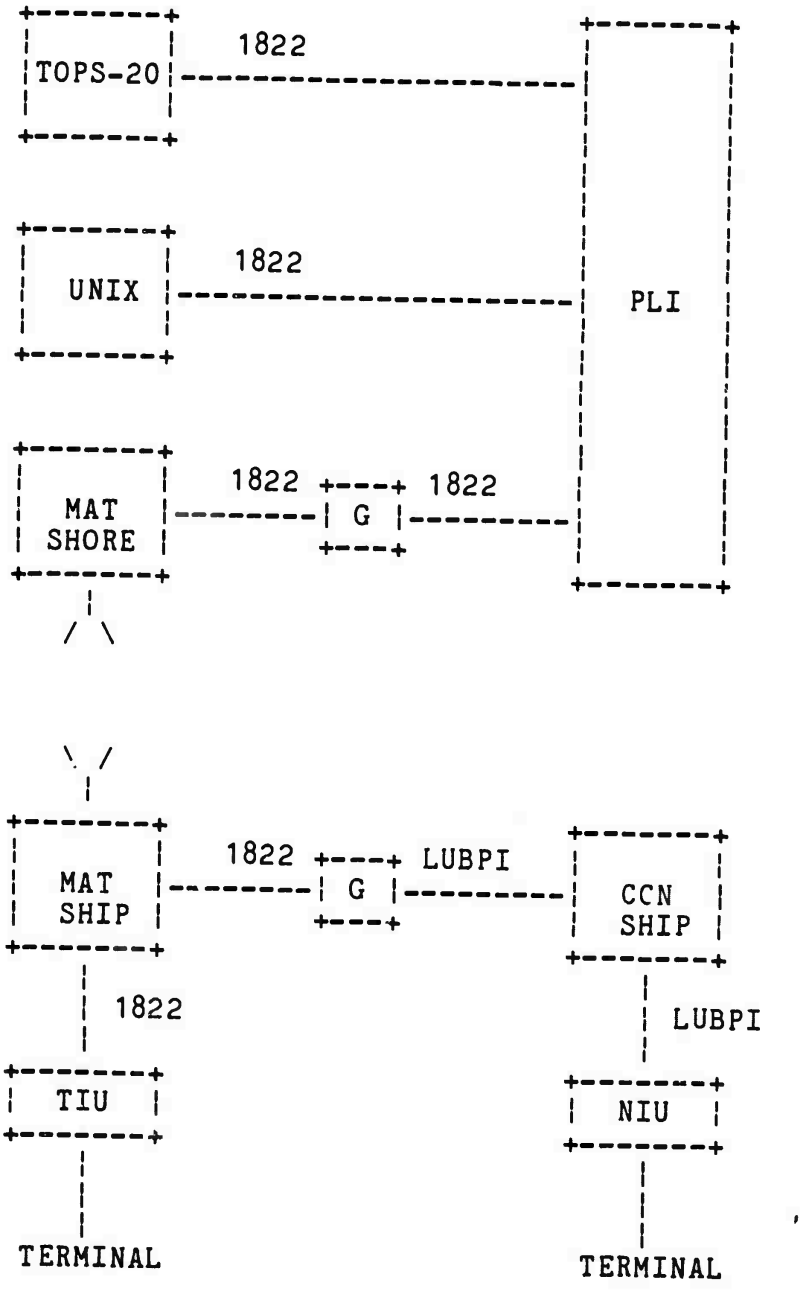


Figure 1 . CCN/MAT Test Configuration

adversely affect the end-to-end communications. Very impressively, though, the TI-743 looked like it was running continuously without stuttering, while the CRT was receiving large bursts of data every second; both terminals gave very acceptable performance.

Because the ACCAT UNIX TCP is incompatible with the MATNET TIU TCP, shipboard access to the UNIX required a TELNET connection through the TOPS-20. Nevertheless, this combination worked sufficiently well to allow satisfactory shipboard operation of wargame programs in the UNIX. Eventually the ACCAT UNIX TCP will be replaced, whereupon direct connectivity between the CCN and the UNIX will be possible.

6.1.7 Test Conclusions

Despite the problems mentioned above, we routinely were able to establish two-MAT communications through the satellite channel. Furthermore, NOSC and TRACOR were able to usefully demonstrate Satellite IMP operation during last April's satellite tests without difficulty.

Although not expected in the benign environment at NOSC, RFI pulses exceeding 10 dB above the background and sometimes occurring more frequently than 100 per hour were recorded.

Unauthorized users have been seen on the channel, but they are not expected to cause this high a level of inteference. The signal-to-noise ratio of the channel, however, was estimated to be about 10 dB, about 4 dB above the worst-case conditions for which MATNET is designed.

We have seen on occasions very good system performance; namely, in several hour-long intervals with message generators fully loading the channel, neither Satellite IMP lost reservation-synchronization once. On other occasions under seemingly identical conditions, we have seen the Satellite IMPs lose reservation-synchronization half a dozen times in an hour. We conjecture that the difference is due to degraded channel performance; presumably interference is present on the satellite channel, sometimes but not necessarily all the time large enough to trigger the AN/WSC-3 blanker pulse (10 dB above the background). Given this variance in MATNET operation, we believe it is important to record the satellite channel characteristics simultaneously when making performance measurements; no mechanism is currently installed to do this.

ECI brought with them circuitry to simulate the presence of radar pulses by pulsing the AN/WSC-3 blanker signal appropriately. Brief tests with this circuitry simulating 10% pulsed RFI showed no discernible effect, both when running Black

systems by themselves using message generators in the Black processor and when running full MATNET systems using message generators in the Satellite IMPs.

Despite several attempts to use the RF satellite channel simulator, it was of no use whatsoever because of malfunctioning signal generators. On several occasions when satellite time was not available, it could have been useful.

Because of the overall success of the MATNET shore-based satellite testing, it was decided to take advantage of an opportunity to install the shipboard MAT aboard the guided missile frigate USS Fanning for an at-the-dock MATNET demonstration. Disappointingly, the first night's testing failed to achieve success for the following reasons. First, in our absence a patch panel connection was unwrapped, breaking the circuit between the shore station Satellite IMP and its Black equipment. Second, because the Fanning has only one AN/WSC-3, two radios from NOSC were brought aboard; however, the ones used in our satellite testing session were denied for our use, so two unknown radios were designated, one of which failed to achieve data acquisition. Third, to circumvent the above radio failure, the radio permanently installed on the Fanning was connected to the shipboard MAT station. Unfortunately, there were no transitions on this radio's external clock signal.

In contrast with the first night, the second night's demonstration proceeded without complications, using the two AN/WSC-3s brought aboard specifically for the testing after a missing VCO module was installed in the malfunctioning unit. When the satellite testing began, shipboard access to shore facilities became routine. The shipboard TIU had two terminals running, a TI-743 hard-copy unit at 300 baud and a TEK 4014 graphics terminal at 9600 baud. Among the standard programs on the shore-based ACCAT TOPS-20 accessed by the ship were XED, MSG, SNDMSG, and MONITOR. Later, the ACCAT programs WES and STAMMER were run, and it was observed that performance was not significantly poorer than in normal operation where MATNET is not involved.

After the successful at-the-dock demonstration of MATNET on the USS Fanning, the equipment was left aboard during a subsequent cruise of the ship from San Diego to Long Beach. On this cruise a major milestone in the MATNET program was reached: namely, the first successful at-sea demonstration of a shipboard MAT unit communicating with shore-based resources. Even though shipboard radars were in operation part of the time, shipboard TCP connections were established to the ACCAT TOPS-20 for running programs EXPAK and MONITOR and for displaying graphics generated by the program WES.

6.2 Other Activities

In other activities during the past quarter, we ordered two C/30 Satellite IMPs as part of the MATNET Phase 3 effort; delivery is expected late summer. Finally, we briefed several different organizations in the Department of the Navy and crew members of the USS Fanning on the design of the Priority-Oriented Demand-Assigned (PODA) channel protocol, which is incorporated in MATNET.

7 HP3000

The last quarter of the HP3000 Internet project was spent on low level maintenance tasks of the HP3000 software. Very little time was actually charged to the contract because our software tests did not reveal any new bugs. This low level effort will continue until DARPA obtains an ARPANET host interface for the HP3000. BBN will deliver the software to DARPA when the ARPANET host interface is available.

To date we have not had any requests for our software.

The LSI-11 front end did not fail in any way during the last three months.

Except for a delivery of software to DARPA, we do not expect any major activity in this contract between now and its termination date.

8 TCP FOR VAX UNIX

Activity in the VAX TCP project continued with emphasis on software distribution and TCP/IP kernel software development. Software distribution continued, with tapes being sent to a number of new sites. Development efforts centered on performance improvements and writing new network interface drivers.

8.1 Software Distribution

The BBN VAX TCP/IP Software Distribution was sent to the following sites:

- Aerospace Corporation
- Carnegie-Mellon University
- California Institute of Technology
- Interactive Systems Corporation
- Lawrence Livermore Laboratories S1 Project
- MIT Laboratory for Computer Science
- Microelectronics Center of North Carolina
- New York Institute of Technology
- Purdue University
- Rand Corporation
- Stanford University
- University of California at Los Angeles
- University of Delaware
- University of Maryland
- USC Information Sciences Institute
- University of Washington
- University of Wisconsin
- Western Digital Corporation

In addition, a number of sites have requested ordering materials for the software or have orders in process. Response from the sites has been good, with a number up and running on the ARPANET.

We have also received some feedback from the sites on bugs in the installation procedure and in the software itself, and are working to integrate the fixes to these bugs into the release.

8.2 Kernel Enhancements

8.2.1 Performance Improvements

We have completed the conversion of the TCP/IP software to run as a software interrupt routine in the kernel without separate process context. We recently started running this version of the system in production at BBN. By not running the TCP/IP as a distinct process, the overhead of process scheduling time and an extra context switch is eliminated. This has resulted in a net performance improvement of about 50% in most cases. Initial looping tests through an ARPANET IMP indicate a throughput rate of 120 Kb/s (data only), using 1000 byte packets, which uses only 20% of the available CPU cycles on a VAX-11/780. We will continue performance improvement work by increasing the network buffer size from 128 bytes, in order to cut down on the number of interrupts needed in sending and receiving packets. We also intend to do more formal throughput testing when this is completed and our Ethernet is installed (see below).

8.2.2 Network Interface Drivers

We have been working on drivers for the ACC IF-11/HDH ARPANET interface and the InterLan UNIBUS Ethernet Controller. The IF-11/HDH interface supports the ARPANET 1822 HDH Protocol, described in Appendix J of BBN Report No. 1822. It is based on the ACC UMC Z-80 and provides a multiplexed DMA interface to the host. We have been working in cooperation with ACC on the design of the software interface between the host and the IF-11/HDH, and have written a driver for the TCP/IP code for it. We received a prototype interface board for testing purposes from ACC, and will begin testing immediately.

We have also completed a driver for the InterLan Ethernet Controller, and will begin testing it soon. The InterLan Controller provides all Ethernet channel acquisition and control functions, including generation of random exponential backoff values for retransmissions, CRC computation, and address recognition. It also provides a DMA interface to the host, 2K bytes of send packet buffering, and 16K bytes of receive packet buffering. Several issues concerning mapping of the 32-bit IP addresses to 48-bit Ethernet addresses arose during the design of the driver, and an approach for dealing with these issues was developed (see below).

8.3 IP/Ethernet Address Mappings

Ethernet differs from other networks in the internet in that it uses local addresses which are larger (48 bits) than IP addresses (32 bits). With other networks having smaller local address formats, the mapping from local network address to IP address is simple: embed the entire local address in the appropriate class of IP address. For example, in the ARPANET which has 24-bit addresses, the mapping to a class A IP address which has 24-bits of local host address, is direct:

Host A on IMP B -> 10.A.B(hi).B(lo)

where 10 is the class A network number for the ARPANET, the host portion of the local address is an 8-bit value, and the IMP portion is a 16-bit value whose high order 8 bits are usually zero. Note that mapping a 24-bit ARPANET address to a non-class A address would require some other approach, like truncation (say, dropping the high order 8 bits of the IMP part) or table lookup.

A similar, but more severe, problem occurs with 48-bit Ethernet addresses. In this case, there may be no simple direct mapping (like truncation) to the 32-bit IP address, since the address of an individual Ethernet controller is a unique 48-bit value assigned by the manufacturer that could conceivably

conflict with other addresses on the local network if a subset of the 48-bits is used to form the local host part of the IP address.

To solve this potential conflict, we choose a mapping that depends on table lookup. Each host on the local network is assigned an IP address with a unique local host part for that network. The IP module in that host "knows" its own IP address and uses it to send and receive IP datagrams. Other hosts on the local network (including gateways) maintain tables which map the IP addresses of the other hosts to their corresponding local addresses.

This table may be "hardwired" into the IP modules or initialized when the host comes up. A better implementation would be to maintain the table as a cache, whose entries are updated dynamically while the host is active. To do this, we can make use of a feature supported by many local area networks, broadcast addressing. This feature allows messages to be received by all hosts on the local network, by sending them to a specially designated broadcast address (in Ethernet the broadcast address is "all ones"). To map a given IP address to its corresponding local address, the IP address is used to look up the local address in the mapping table. If the mapping is found, the local address is used in the local network header.

Otherwise, if no mapping is found in the table:

1) Send an ICMP information request message with the desired IP destination address (Figure 2), and the datagram which caused the mapping request to the local broadcast address.

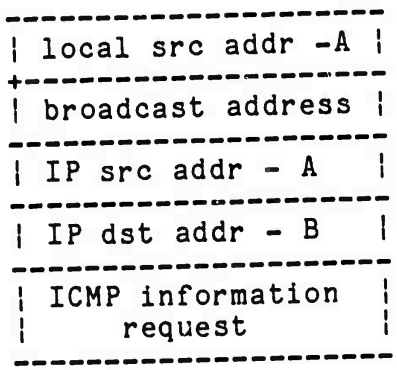


Figure 2 . Mapping Request Format

2) The host on the network that recognizes its IP address in the information request message sends an ICMP information reply message to the host that broadcast the request, with its local address filled in (Figure 3). It also accepts the broadcast datagram since it contains its IP address as the destination.

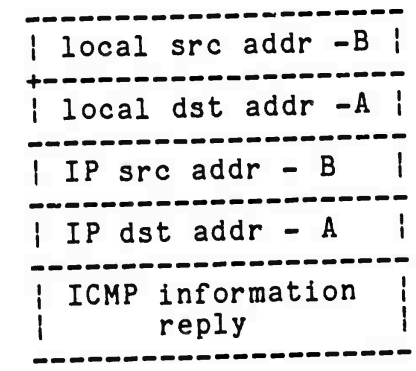


Figure 3 . Mapping Reply Format

3) When the ICMP information reply is received, an entry is

made in the mapping table, using the local address found in the local network header.

4) If the reply is not received within some period of time, the higher level protocols may be notified as if a destination unreachable message had been received.

Finally, there must be some way of invalidating table entries, to handle cases like a host on the local network changing addresses:

Whenever a host on the local net comes up, it broadcasts an ICMP information reply with its source IP and local network addresses (Figure 4). Other hosts on the network can use this information to update their tables.

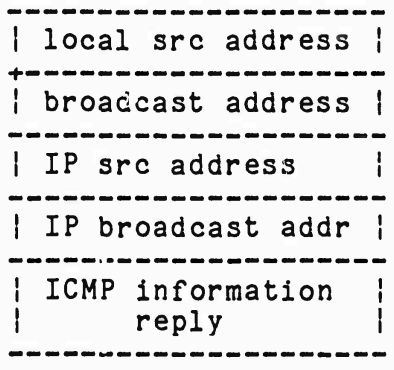


Figure 4 . Mapping Initialization Format

Note that the use of this message also lowers the chances of hosts not having the mapping in their table, and reduces the number of broadcast mapping requests. This message contains the notion of an IP broadcast address. This is simply a means of mapping an IP address to the local network broadcast address. It has the effect that any IP module on a particular network should accept datagrams with the broadcast address as the IP

destination. For example, one could define the IP broadcast address to be the IP address in each class with all its local host part bits on, (e.g., A.255.255.255 for class A, A.B.255.255 for class B, and A.B.C.255 for class C). In each case, the address would map to the local network broadcast address if broadcast addressing was supported.

This approach has several advantages. First, it is general enough to work on all local networks that support broadcast addressing. Second, it fits into IP without defining any new mechanisms. Third, it requires no new protocol messages to be defined.

On the negative side, it can potentially cause adverse network and host performance when many hosts on the net are coming up, because of the use of broadcast packets. Also, performance can be degraded by hosts which do not implement the scheme fully, causing more broadcast packets to be sent. However, we expect the use of broadcast mapping requests to be a low frequency occurrence. Finally, there may be a problem receiving the mapping reply and the datagram immediately following, with interfaces which cannot receive back-to-back messages (a problem for some Ethernet controller implementations). At worst, this will cause the datagram following the mapping request to be lost and require a

retransmission by the higher level protocol software.

As an alternative to reduce the number of broadcast packets and ameliorate the back-to-back message problem, the source host can refrain from sending the datagram that caused the request until the destination responds with its local address. However, this requires that hosts have an extra message exchange to get the local address information into the table, which means that a datagram must be held in the IP module during the exchange, or dropped and retransmitted later by the higher level protocol.

DISTRIBUTION

ARPA
Director (3 copies)
Defense Advanced Research Projects Agency
1400 Wilson Blvd.
Arlington, VA 22209
Attn: Program Manager
R. Kahn
V. Cerf
R. Ohlander
D. Adams

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

DEFENSE COMMUNICATIONS ENGINEERING CENTER
1860 Wiehle Road
Reston, VA 22090
Attn: Lt. Col. F. Zimmerman

DEPARTMENT OF DEFENSE
9800 Savage Road
Ft. Meade, MD 20755
R. McFarland C132 (2 copies)

DEFENSE COMMUNICATIONS AGENCY
8th and South Courthouse Road
Arlington, VA 22204
Attn: Code 252

NAVAL ELECTRONIC SYSTEMS COMMAND
Department of the Navy
Washington, DC 20360
B. Hughes, Code 6111
F. Deckelman, Code 6131

BOLT BERANEK AND NEWMAN INC.
1300 North 17th Street
Arlington, VA 22209
E. Wolf

MIT Laboratory for Computer Science
545 Technology Square
Cambridge, MA 02138
D. Clark

DISTRIBUTION cont'd

BOLT BERANEK AND NEWMAN INC.
10 Moulton Street
Cambridge, MA 02238

S. Blumenthal
M. Brescia
R. Bressler
R. Brooks
P. Carvey
P. Cudhea
W. Edmond
L. Evenchik
G. Fail.
J. Goodhue
S. Groff
R. Gurwitz
J. Haverty
F. Heart
J. Herman
R. Hinden
D. Hunt
E. Hunter
S. Kent
A. Lake
W. Mann
A. McKenzie
D. McNeill
W. Milliken
M. Nodine
R. Rettberg
R. Reynolds
J. Robinson
E. Rosen
P. Santos
J. Sax
A. Sheltzer
S. Storch
R. Thomas
B. Woznick
Library