Bolt Beranek and Newman Inc.

LEVEL III

(12)

Report No. 4825

AD A108783

Combined Quarterly Technical Report No. 23

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

November 1981

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-A108	3. RECIPIENT'S CATALOG NUMBER 783
4. TITLE (and Subtitle) COMBINED QUARTERLY TECHNICAL REPORT No. 23		5. TYPE OF REPORT & PERIOD COVERED 8/1/81 to 10/31/81
		6. PERFORMING ORG. REPORT NUMBER 4825
7. AUTHOR(s) R. D. Bressler		8. CONTRACT OR GRANT NUMBER(s) MDA903-80-C-0353 & 0214 N00039-78-C-0405 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 Nos. 3214 and 3175.17
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Advanced Research Projects Agency 1400 Wilson Boulevard Arlington, VA 22209		12. REPORT DATE November 1981
		13. NUMBER OF PAGES 78
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, Terminal Access Controller, VAX, ARPANET, Internet.

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, TAC, 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. 4825

COMBINED QUARTERLY TECHNICAL REPORT NO. 23

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

November 1981

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

N00039-78-C-0405, ARPA Order No. 3175.17
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; (7) TCP-TAC; and (8) 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, and N00039-79-C-0386.

2 SATNET DEVELOPMENT AND OPERATION

During the past quarter, we placed emphasis on several events critical for the expansion of the Atlantic Packet Satellite Experiment to more sites. First, we designed a small printed circuit daughterboard for mounting above the univeral I/O board of the BBN C/30 communications processor to form a hardware interface between the Satellite IMP and the Packet Satellite Project (PSP) terminal. Second, we began investigation of the problems associated with having one C/30 Satellite IMP control two 64 Kb/s SPADE satellite channels.

Also, as part of our participation in SATNET. We continued our investigation of the recently implemented facility for automatically creating and maintaining low-capacity streams for TCP traffic from gateways, and we were involved in the operational maintenance effort of the network. These items are detailed in the following sections.

2.1 C/30 I/O Interface to the PSP Terminal

Because the memory address space and processing power of the Honeywell 316 general-purpose minicomputer are inadequate for planned enhancements to the Satellite IMP, the BBN C/30 microprogrammable communications processor has been chosen as a

replacement machine. To be mounted above the 2651 Universal Synchronous/Asynchronous Receiver/Transmitter (USART) serial interfaces on the C/30 universal I/O board are small special-purpose printed-circuit daughterboards performing the electrical conversion between the TTL signal levels internal to the C/30 and the Bell 303 and the RS-232-C signal levels required by the PSP Terminal. Additionally the daughterboard provides a special clocking signal to the Command and Monitoring Module (CMM) of the PSP Terminal. Specific design details of the daughterboard are presented below.

To further commonality, the daughterboard design can service either the synchronous data port or the asynchronous CMM port of the PSP Terminal; jumpers on the board and different cable terminations differentiate between the two ports. The daughterboard is designed so that if it is strapped for data port operation and if the port is not enabled as a satellite modem, the port can be used as a normal Bell 303 interface; hence, all signal handling is identical with the C/30 Bell 303 MMI daughterboard, except for the signals FAST/SLOW and GOSIG. The latter signals, which are required for satellite channel operation, are absent in the MMI.

The signals needed for the PSP Terminal data port are:

TRANSMIT DATA -- C/30 to PSP Terminal. A "one" bit in the

data register of the USART is translated into a 25 milliamp signal by the 8T13 line driver.

RECEIVE DATA -- PSP Terminal to C/30. A 25 milliamp input signal to the 8T14 line receiver is translated into a "one" bit in the data register of the USART.

TRANSMIT CLOCK -- PSP Terminal to C/30. Because the convention chosen for this interface is that data signal transitions occur on the rising clock edge when viewed on the interconnecting cable, and because the USART samples the data on the rising clock edge at the clock input of the USART. the clock from the PSP Terminal is inverted between the 8T14 line receiver and the USART.

RECEIVE CLOCK -- PSP Terminal to C/30. This signal is handled identically to the TRANSMIT CLOCK signal.

LOOP -- C/30 to PSP Terminal. The USART Request-to-Send (RTS) signal is used to control the looping state of the modem. Because the complement of USART command register bit CR5 appears on the USART output pin RTS. the signal is inverted between the 8T13 line driver and the USART; hence, a "one" bit in CR5 translates into a 25 milliamp output.

FAST/SLOW -- C/30 to PSP Terminal. The USART Data-Terminal-Ready (DTR) signal is used to control the satellite channel data rate. Because the complement of USART command register bit CR1 appears on the USART output pin DTR. the signal is inverted between the 8T13 line driver and the USART; hence. a "one" bit in CR1 translates into a 25 milliamp output.

GOSIG -- C/30 to PSP Terminal. Instead of a control signal from the USART. the C/30 signal CSR3 is used to control the satellite channel transmission enable. A "one" bit in CSR3 is translated into a 25 milliamp output by the 8T13 line driver. The default state of this signal is zero. equivalent to satellite channel transmission inhibited The default state is entered whenever the C/30 reset button is pushed or whenever the interface is initialized To prevent runaway transmissions. a hardware watchdog timer on the daughterboard will reset this signal to its default state if the signal is asserted for more than ten seconds. During test situations. the timer can be disabled by changing a jumper on the daughterboard

The signals needed for the CMM port of the PSP Terminal are:

'TRANSMIT DATA -- C/30 to CMM. This is an asynchronous bit stream with one start bit and at least one stop bit per character. The signal is inverted between the 8T13 line driver and the USART to generate a complement bit stream for the CMM. Unfortunately. the USART sends the LSB first while the CMM expects the MSB first, which is rectified in microcode.

RECEIVE DATA -- CMM to C/30. This is an asynchronous bit stream with one start bit and at least one stop bit per character. The signal is inverted between the 8T14 line receiver and the USART to generate a complement bit stream for the C/30. Unfortunately. the USART expects the LSB first while the CMM sends the MSB first, which is rectified in microcode.

TRANSMIT/RECEIVE CLOCK -- C/30 to CMM. The C/30 provides a common clock derived from the USART transmit clock for both transmit and receive data. Because the convention chosen for this interface is that data signal transitions occur on the rising clock edge when viewed on the interconnecting cable, and because the USART changes the data on the falling clock edge. the signal is inverted between the USART and 8T13 line driver.

The second generation CMM unit under design will connect to an off-the-shelf C/30 RS-232-C MMR daughterboard instead of the daughterboard described above. Inasmuch as the data transfer is RS-232-C asynchronous. no clock is needed with the new CMM unit. (RS-232-C asynchronous protocol was chosen to allow a CRT terminal to replace the C/30 during testing and checkout.)

2.2 128 Kb/s Channel Operation

With the increase in traffic expected from European participants in SATNET. an increase in the satellite channel bandwidth to 128 Kb/s is being considered The method chosen is

to operate two 64 Kb/s SPADE satellite transponders in parallel with independent channel scheduling for compatibility with the current hardware. Stations with only one working channel can continue to participate in SATNET. albeit at a lower capacity than stations with two working channels.

The second generation PSP Terminal to be built for 128 Kb/s operation will have three separate copies of the channel hardware for simultaneous operation with two Satellite IMP data ports and a PSP Terminal Data Test Set. The physical connections between the data sources and the two satellite channels can be altered either by the switch matrix in response to Satellite IMP commands or by a manual switch on the PSP Terminal. At a minimum, the switch matrix will support three out of the six possible interconnection patterns, enough to switch any malfunctioning channel hardware unit to the Data Test Set for diagnosis. while the two working channel hardware units are used to support the two Satellite IMP ports. One little-used feature of the PSP Terminal that will be removed however. is the abiiity to splice together the transmit portion of one channel unit and the receive portion of another channel unit.

Since the new PSP Terminals will be able to dynamically change the transmit frequency of each channel unit under software control, site personnel need not manually intervene when the

backup channel unit is required. However, according to the present operating procedures adopted by the INTELSAT ground stations, ground station personnel might insist on being involved in anything as fundamental as switching frequencies.

The decision to provide two parallel satellite channels raises many interesting and difficult issues for the Satellite IMP software concerning, among others, network synchronization and traffic routing. The latter issue arises because some stations may be limited to a single channel, so that the two channels will not have identical sets of Satellite IMPs as members. The following paragraphs discuss these issues and possible design solutions.

Due to the geographical distribution of SATNET stations, each station has a different propagation time associated with its transmissions to the satellite. In order to synchronize all stations' activity properly, a global time is defined with the property that packets from two stations sent at the same global time will arrive at the satellite simultaneously. At each station, the global time consists of the local time plus an offset that depends on both the station's propagation time to the satellite and a base time defined by one station selected as the leader. Any station that knows the offset from its own local time to the global time can accurately schedule its transmissions

relative to the other stations. In practice, each station must adjust its offset periodically to correct for changes in propagation time due to satellite orbital motion and clock drifts originating from slightly disparate clock rates among sites.

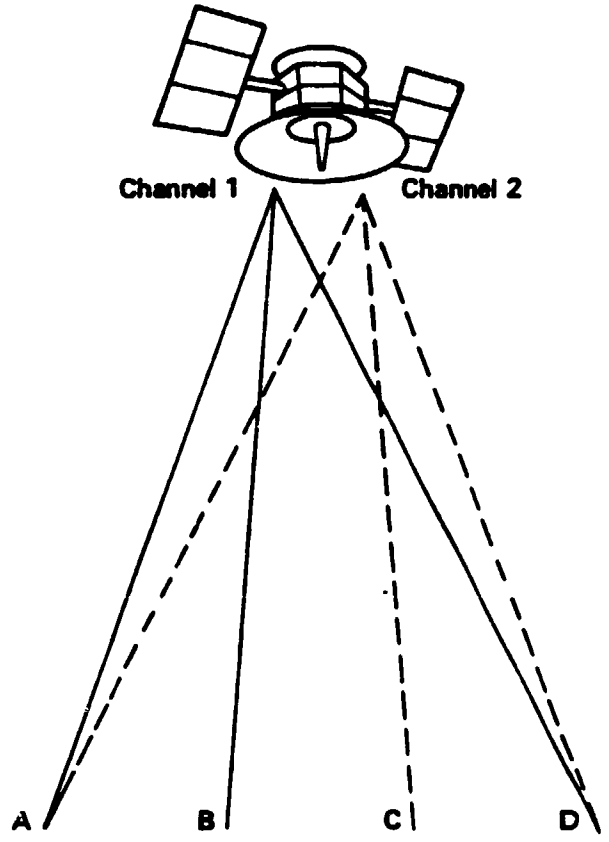
Although the primary role of global time is in synchronizing transmissions from different sites. its availability as a single network-wide time has led it to a position of fundamental importance in many aspects of the Satellite IMP software. For example, timestamps based on the global time are used throughout SATNET to order queues of packets waiting for service. The single global time base is also used to synchronize the simultaneous generation of monitoring reports and events in SATNET-wide experiments.

When a single Satellite IMP has to control two separate satellite channels. complications for global-time synchronization arise from the potentially different propagation delays on the two channels. In normal operation the delays at a single station associated with the two channels should differ slightly. since signals on the two channels follow nearly identical paths. However, the propagation delays can differ significantly if one channel is looped or if the equipment on the two channels has markedly different processing delays. Thus, two separate global times independently synchronized are required

Since channel scheduling for the two channels proceeds independently. maintaining separate global times poses little difficulty to the channel protocol module; however. other modules using global time need redesign. For example, outside the channel protocol module, all queues might be ordered using local rather than global timestamps. Timestamp conversions between local and global forms may be required upon entry to and exit from the channel protocol module. Certain network-wide functions. such as synchronization of monitor reports. may require coordination by an active member of both channels.

The presence of two parallel traffic paths in SATNET introduces an added complexity of message routing not previously required in a single shared-channel broadcast network. In general, as shown in Figure 1, some pairs of stations (e.g. A and D) may be able to communicate using either channel, some pairs (e.g. A and B) may have to communicate using a specific channel, and some pairs (e.g. B and C). while having neighbors in common, may be unable to communicate directly. Many SATNET features. such as streams and group addressing. currently rely on the fully-connected and broadcast properties of the satellite channel, and hence may need some redesign.

When either channel is acceptable, the Satellite IMP must select one or the other. Strategies such as sending alternate



SATNET Topology Depicting Dual Channels with Non-Identical Members
Figure 1

packets on alternate channels. while simple in implementation, can lead to mismatched channel loading. In general, some sort of flow control to balance the load on the two channels is required The requisite flow-control could be based on local considerations. such as the length of queues waiting for each channel. (Choosing between alternate paths is an issue long familiar to non-broadcast networks such as ARPANET but is new to SATNET.)

Connectivity monitoring is a prerequisite for the flow control algorithm. so that the appropriate channel will be selected when only a single path exists to the message destination. and the message will be refused when no connectivity exists to the message destination. Assuming no one-way connectivities over the channel exist, connectivity monitoring reduces to keeping a record of which sites have been heard recently. In fact. the current Satellite IMP performs such monitoring to avoid consuming channel bandwidth with traffic for inactive sites. Network hosts. however. will be able to optionally specify the channel on which a particular message should be sent; this will be useful, for instance. in creating specific traffic patterns using message generators.

When the only connectivity between two stations requires two satellite channel transits. such as between B and C in Figure 1,

traffic delivery without leaving SATNET would require message forwarding capability within the Satellite IMPs. In the interest of reduced Satellite IMP complexity, it may be decided that these messages take advantage of gateway routing; i.e., connectivity could still be provided at the Internet level by having the host attached to B direct its messages for C to an intermediate gateway attached to either A or D for further routing When viewed in this way, a lack of direct connectivity can be considered an instance of a partitioned network. which is an issue currently under investigation in the Internet community.

2.3 Automatic Stream Service for Gateway Traffic

The facility for automatically creating and maintaining low-capacity streams for TCP traffic from gateways was recently implemented to provide one-hop delay service to the interactive user. Because successful operation of this facility without allocation of substantial channel bandwidth requires restriction of the service to interactive traffic only. a selection algorithm matched to interactive traffic is employed to prevent delay-tolerant applications from consuming the assigned bandwidth. Currently the selection. which is based on information held in the message's Internet Protocol (IP) header or SATNET header. includes all TCP messages below a certain size. In subjective

tests. a marked improvement in response relative to ordinary reserved datagrams was immediately apparent. especially when operating with the character-at-a-time systems common on the ARPANET. At its best. the response time approached that of the ARPANET direct connection via SATNET. which is designated as ARPANET Line 77; however. delays were bursty in nature and occasionally up to several seconds in duration. Nevertheless. whenever sufficient reassembly buffering is available, the streams appear to provide acceptable round-trip delays for interactive traffic with remote echoing.

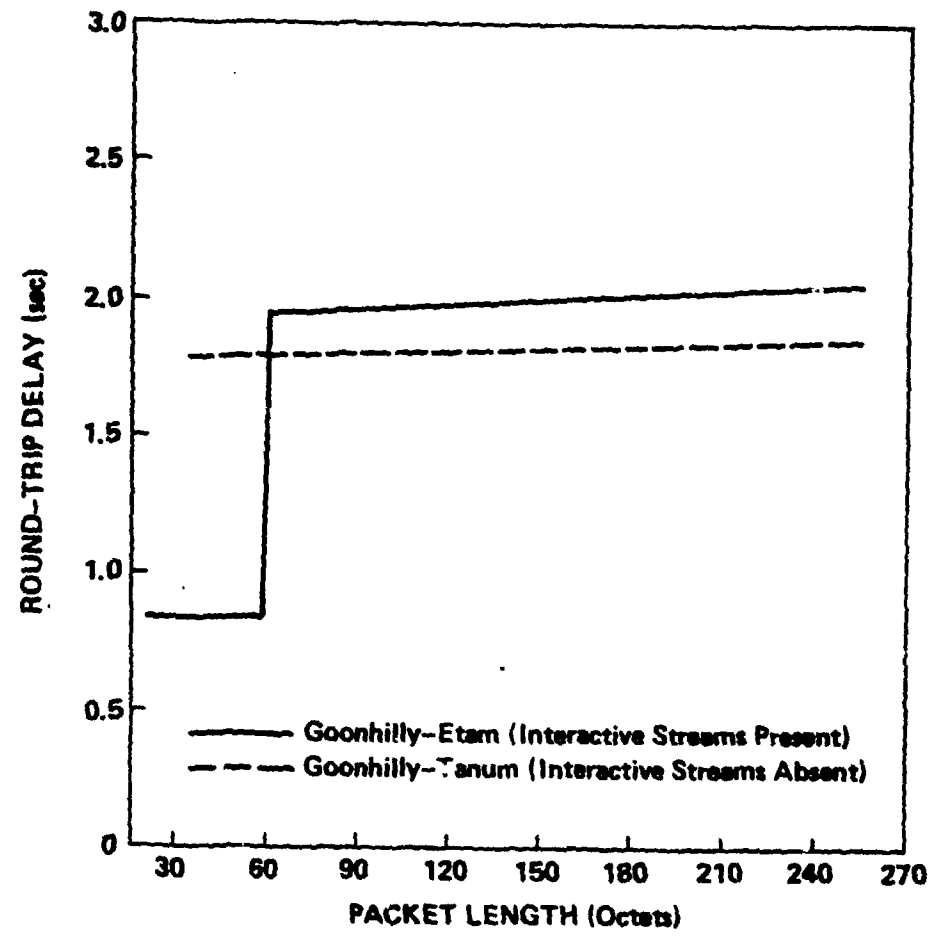
Bursty delays occur because qualifying packets are occasionally sent via datagram instead of via the stream facility, which happens whenever a single packet or the aggregate of all qualifying packets is too large for the stream capacity. Because of the longer delay imposed on any packet sent via datagram, subsequent packets sent via stream can arrive earlier. resulting in packets arriving out-of-sequence; TCP, however. reestablishes packet ordering. In the current stream allocation, assignments recur every 0.33 seconds (about one PODA frame) and accommodate one TCP packet having the minimum internet fragment size totaling 68 octets of data, including the IP and TCP headers. The parameters chosen reflect a deliberate attempt to exclude bulk data transfers; the bandwidth allocated to the stream could be increased. were it necessary to minimize the

number of out-of-sequence packets.

The complex behavior resulting from overflow of the stream size, packet reordering, and host system delays makes objective measurements difficult; however, the data summarized in Figure 2 Demonstrating the Effect of Interactive Streams give an idea of the performance. These data were collected at the Communications Satellite Corporation (COMSAT) during separate tests in which packets, generated in a DEC LSI-11/23 system in London, are echoed off each of the Satellite IMPs in turn. The interactive stream facility was enabled only on the Goonhilly-Etam path with a size sufficient to accommodate one packet totaling 58 octets of data, including the IP header. In the data shown, the round-trip delay on the London-Goonhilly path was removed to eliminate the effect of the 48 Kb/s landline circuit. Figure 2 clearly shows the much lower delays for packets on the Goonhilly-Etam path below the threshold size compared with packets on the same path above the threshold size and with all packets on the Goonhilly-Tanum path, for which the interactive stream facility was not enabled.

2.4 Hardware Problems Fixed

During the last quarter, several hardware problems appeared which we diagnosed and, when related to the Honeywell 316, fixed



Round-Trip Delay as a Function of Packet Length
Figure 2

In early August. a severe lightning storm caused hardware failures in the NDRE PDP-11/40 IMP-11 interface and the NORSAR TIP Honeywell 316 1822 interface serving as Host 3 and thus isolated the NDRE gateway and the Tanum Satellite IMP from the NORSAR TIP. DEC and NDRE personnel repaired the former. while BBNCC personnel repaired the latter.

In the middle of August. the BBN Gateway and its modem were powered off and moved to allow scheduled construction in the room where they are installed. After the contruction was finished. connectivity between the BBN gateway and the Etam Satellite IMP could not be reestablished; in the interim. a carrier problem had developed on the circuit. which TELCO later found and corrected. Note that in the absence of the BBN gateway, we relied on the NDRE gateway path to reach SATNET. This alternate path, however. is to be removed from service on 1 December 1981.

For many weeks. large numbers (approximately 10%) of packets were arriving with checksum errors at the Goonhilly Satellite IMP on the 48 Kb/s circuit from the UCL gateway. Since British telephone personnel insisted their equipment was not malfunctioning. the BBNCC repairman. while in Europe to fix the NORSAR TIP. performed comprehensive hardware tests at Goonhilly. After many checks with special-purpose diagnostic software, it was concluded that the problem source was indeed with telephone

company equipment. Subsequently. the British telephone personnel found and fixed a frayed ground cable, and the circuit problem disappeared.

In the middle of August. packet lossage on the SATNET channel increased noticeably; the evidence. while inconclusive. initially pointed towards software. Later reports from the Etam Satellite IMP indicated serious trouble on the channel totally disrupting all SATNET operations; Etam failed to hear over 90% of its Hello packets but heard most of Tanum's. while Tanum heard most of Tanum's and Etam's. Goonhilly was unavailable because of diagnostic work described in the previous paragraph. Once Etam began having channel problems. we converted the Etam-SDAC circuit (the land-based portion of ARPANET Line 77) from IMP-to-IMP to VDH in order to create a pathway into Etam to diagnose the problem (the BBN gateway was inaccessible because of the construction in the room housing the gateway). That afternoon Etam crashed. confusing the issue by indicating hardware problems with the Honeywell 316. Since we still strongly suspected the software. we reloaded the Satellite IMP software and were preparing to load a previous version without the new Hello packet format and without the automatic stream setup for TCP traffic from the gateway.

Having determined that the Satellite IMP ran normally in

external crosspatch (into and immediately out of the PSP Terminal interface driver/receivers), but failed totally to hear any of its own transmissions. we began to concentrate on PSP Terminal operation. Following standard operating procedure, we had Etam site personnel power the PSP Terminal off and on and press the PSP Terminal manual reset switch, but to no avail.

COMSAT thereafter determined that Etam transmit power was 8 dB high, which is well beyond the upper range of the AGC Once the transmit power was normalized, it was necessary for site personnel to press the appropriate number of hardware resets on the PSP Terminal, including the SOM load switch, to restore operation. Apparently, given the evidence that earlier on Etam failed but Tanum succeeded in hearing Etam's Hello packets, the SOM bits were corrupted in addition to the transmit power being maladjusted.

When the 5-volt power supply in the Multi-Line Controller (MLC) of the London Honeywell 316 TIP failed the Network Operations Center (NOC) configured the Honeywell 316 at London as an IMP to continue to provide host service. During this time, the SATNET automatic stream service for interactive TCP traffic from the gateway was invoked to improve service to European users.

In late August, satellite channel reception problems at

Goonhilly caused the ARPANET direct connection via SATNET to fail. Having determined that transmit power levels were initially maladjusted and that channel interferences were present later, the COMSAT M&S center personnel contacted Etam site personnel several times to correct the problems. (The M&S center had just become operational for channel problem diagnosis.)

In September, a local power outage caused a hardware failure in the NDRE PDP-11/40 VDH interface and thus isolated the NDRE gateway from the Tanum Satellite IMP. The interface has been shipped to Associated Computer Consultants for repair.

3 PLURIBUS SATELLITE IMP DEVELOPMENT

The major activity during the quarter continued to be the integration of the PSAT with other elements of the Wideband Network. This integration was carried out between pairs of subsystems for each site as a collection of subsystems. and for the multi-site network as a whole. The major goal of this integration has been to achieve an operational status for the Wideband Network. Short term, we have been trying to establish a capability for supporting cross-country packetized speech transmission.

During the first half of August. considerable effort went into checkout of the ADM ESI at ISI. During this checkout. it was discovered that a software checksum implemented by the ADM ESI to protect against channel errors had introduced an incompatibility between the ADM and non-ADM versions. Because of the way that the ESI inserts the burst delay value into each downlink burst and modifies the PSAT hardware checksum, there was no way for the PSATs to get around this incompatibility. Until a new PROM disabling this feature became available, the ISI site correctly handled only bursts that it originated This limitation, and an additional problem at DCEC due to premature burst termination by the ESI. meant that testing of the PSAT in August was constrained to simplex communication.

In spite of this one-way only restriction, multi-site communication was successfully demonstrated On August 6 the DCEC modem locked onto Lincoln's signal and the DCEC site received leader packets for approximately three minutes. The inability to run for longer periods was traced to a bug in the global time synchronization code. This code was subsequently fixed. On August 7. the first communication involving simulated user data was accomplished between an internal PSAT message generator at ISI and an internal PSAT message sink at Lincoln Laboratory. With the ISI site selected as leader and the Lincoln site inhibiting transmissions (due to the software checksum problem described above). the following tasks were accomplished:

1. Lincoln heard leader packets from ISI.
2. Lincoln heard datagrams sent to it by ISI.
3. ISI initiated a group create which both sites executed
4. Lincoln joined the group.
5. Both Lincoln and ISI heard datagrams addressed to the group that were transmitted by ISI.
6. ISI initiated a stream create which both sites executed
7. Both Lincoln and ISI heard stream messages addressed to the group that were originated by ISI.

All of these tests (and subsequent tests carried out during the quarter) were accomplished using BPSK modulation at a channel symbol rate of 772 Ksymbol/second without coding (ESI coding rate = 1).

During the first half of September, multisite system integration continued with tests between Lincoln and DCEC. (The ISI site was unavailable from mid-August until mid-September due to ESI hardware problems.) These tests were the first to indicate a problem with the PSAT/ESI initial acquisition procedure. Although it was possible to get around this problem in order to get on with other aspects of system integration, the shortcomings of the acquisition implementation persisted throughout the quarter. On September 25, the first successful three-site tests were carried out. With Lincoln as leader, the ISI PSAT acquired synchronization on the channel followed by the DCEC PSAT. Acquisition problems at DCEC limited the length of this test to about 1 minute.

A meeting of Wideband Network contractors, experimenters, and sponsors was held at the DARPA office on September 28. In addition to discussing long-term issues related to Wideband Network operation and management, special attention was given to the task of expediting network integration activities. As a result of discussions at this meeting, November 18 was set as a target date for a demonstration of cross-country packet speech between Lincoln and ISI. Much of the work during October by BBN was directed at moving toward this goal. This effort involved working with Lincoln Laboratory and Linkabit personnel on host and channel checkout respectively. During the first week of

October. BBN worked with Linkabit to checkout the newly installed ADM ESI at Lincoln Laboratory. Checkout of the PSAT interface to the miniconcentrator gateway involved testing the two subsystems under a variety of different scenarios for message size and generation rate. During the second half of October. BBN worked with ISI to checkout their miniconcentrator software which had been moved to a PDP-11/44 from a PDP-11/45.

A parallel integration activity that was pursued during the quarter was the integration of the PSAT with the Voice Funnel. By the end of the quarter, the Voice Funnel at BBN was able to exchange link startup messages with the PSAT and keep the link alive by the transfer of periodic status messages. Software in the Voice Funnel was also able to bounce messages off of the HPM software in the PSAT. This verified the ability of the link to pass data messages as well. As part of the PSAT/Voice Funnel checkout. the operation of the BBN-developed HDLC-to-VDH converter box was demonstrated.

Two previously initiated development activities continued during the quarter. The SuperSUE poller will provide driver level support for both the high speed host interfaces (HSMs) and the satellite modem interfaces (SMIs). The motivation for the poller and its basic design have been described at length in previous reports. During August. the poller microcode was

modified to support two devices (rather than one) and minimize extraneous (per buffer) pokes to the PID. The development of additional HPM software to support high speed hosts was also begun in August. This development involved rewriting both the HST I/O driver code and the buffer-handling routines. MSGIN and MSGOUT. to be compatible with the SuperSUE poller. In addition, the PSAT initialization procedure had to be modified Most of this work was completed by the end of the quarter. Construction of 9 of the 10 pollers to be delivered during FY 81 was also completed.

Two new designs emerged during the quarter. An improved algorithm for global time synchronization described in section 3.1 below was developed and coded. In addition, we developed a recommendation for the overall design of a next-generation packet satellite controller. The result of this work is contained in section 3.2.

3.1 PSAT Global Time Synchronization Improvements

In the Wideband Network. PSATs establish a common globally synchronized time called the Global Time Clock based on the 24-bit hardware Local Time Clocks in each PSAT's SMI. The Global Time Clock is used to schedule channel time in a coordinated manner. The algorithm for synchronizing Global Time Clocks

involves the computation of a varying offset between local and global time. Conversion of local to global time and vice versa is simply a matter of subtraction or addition of this offset. The initial implementation of the Global Time Clock was limited to 24 bits.

Several additional uses for globally synchronized time. however. necessitate a larger wraparound interval than 24 bits can provide at 3 Mbps. Message time stamping and message discard based on holding time are two such uses where times greater than 5 seconds must be measured. This requirement prompted the implementation of a Global User Clock based on appending 8 high order bits maintained in software to the Global Time Clock to achieve an extended global clock. The high order bit of the Global Time Clock caused the low order software bit to tick. Since 32-bit resolution was not necessary, the Global User Clock was defined as bits 15-30 of the extended global clock (assuming LSB = 0). This clock ticks twice per PODA frame and has a wraparound interval of about 11 minutes.

Initially. the high order 8 bits of the extended global clock were not globally synchronized among all network sites as were the lower 24 bits. The technique originally implemented to deal with this situation worked as follows. The synchronization of the most significant 8 bits of global time was only performed

when a new site came up on the network. At that time, all network sites set the top eight of the 32 bits of global time to zero. thus forcing global synchronization.

Unfortunately. this procedure can cause a discontinuity in Global User Clock time which results in the discarding of one round trip time's worth of messages. This inadequacy has been recently addressed in the following way. The local time clock is extended to 32 bits. The additional 8 bits of this clock are maintained by software. The offset value used for local/global time conversions is also expanded to 32 bits and thus a 32-bit Global Time Clock is synchronized among all network sites. The Global User Clock continues to be the same 16 bits within the 32-bit value.

3.2 Next-Generation Wideband Network Station Design

BBN has been working to develop a design for a next-generation Wideband Network station. In this section we discuss the results of that design to date. First, we summarize several problems that exist with the current design. Next, we propose a basic assignment of functions to station components that we believe addresses these problems. Finally, we identify a set of specific changes to the existing PSAT/ESI interface that are consistent with our recommended functional assignment.

3.2.1 Problems With The Existing Wideband Station Design

Although the Wideband Network is not yet completely operational, we are nevertheless aware of several inefficiencies. limitations. and problems that are inherent in the current design of the station equipment. These shortcomings are listed below and discussed in the following paragraphs. The list includes:

(1) Duplicated Functionality,
(2) Too Many Separate Boxes,
(3) ESI Downlink Processing Problems.
(4) PSAT/ESI Interface Specification,
(5) PSAT Cost/Performance.

(1) The duplicated functionality which exists in the Voice Funnel, PSAT. and ESI represents a non-trivial inefficiency in the current station equipment design. The nature of this problem becomes clear when one examines the set of functions which are currently supported in each of the station components.

Voice Funnel
- Subscriber Interfacing
- Data Aggregation
- Gateway Translations
- HAP Support (for PSAT)
- Internal Hosts (e.g. Message Generators)

PSAT

- Subscriber Interfacing (HAP)
- Data Aggregation
- Global Time Synchronization
- Channel Scheduling
- Precisely Timed Burst Transmission
- Downlink Burst Timestamping
- Congestion Control
- End-to-End Flow Control
- Host and Channel Monitoring
- Network Management
- Internal Hosts (e.g. Message Generators)

ESI

- Burst Modulation/Demodulation
- Multi-rate FEC Encode/Decode
- Precisely Timed Burst Transmission
- Downlink Burst Timestamping
- Burst Filtering
- T&M Data Collection

It is clear from examining these lists that there are several functions that are duplicated in two station subsystems. In each case, the duplicated function can be eliminated in one of the two components resulting in a simplification of the

corresponding element.

(2) The fact that 3 independent boxes (Voice Funnel, PSAT. and ESI) exist to support the above set of functions is another source of system inefficiency in the current Wideband Network station design. One could make an argument that all of the station electronics should be provided in one integrated unit supplied by a single contractor. We believe, however. that this is going to the other extreme. The best approach, in our view, is a two unit approach which reflects the unique areas of expertise of the Wideband contractors and the natural layered structure of the system (physical layer and network layer).

(3) A potential problem with the current design is associated with the requirement that the ESI be able to recognize and decode bursts on the downlink without any *a priori* information. In general, the burst length contained in each control packet allows the ESI to re-arm the downlink modem at the end of each burst. In addition, the control packet statewords tell the ESI what control signals to pass to the codec for each segment of the burst. For small bursts. however. the burst length word may not have been decoded by the time that the burst completes and the next burst arrives. Similarly, under certain conditions the latency for determining what coding command to send to the codec may be sufficiently large that data within the burst is lost.

Linkabit is currently evaluating the use of a unique trailing word in conjunction with measurements of carrier energy to address the burst recognition problem. The stateword decoding latency problem remains to be addressed. We do not yet have sufficient quantitative information on these problems from Linkabit to allow us to evaluate their impact on the overall operation of the network. If either one turns out to be critical, it may be necessary to modify the current design in order to provide the ESI with information on the nature of the scheduled bursts prior to burst reception. The two basic approaches for doing this are (1) the ESI figures out this information for itself or (2) the PSAT passes along this information to the ESI.

(4) For some time now there has been a growing belief shared by both BBN and Linkabit that the PSAT/ESI interface could be improved. Some of the changes are associated with eliminating the system shortcomings listed above. Additional problems which need to be addressed relate to the current limitation on the number of distinct coding states in a burst, the awkwardness of squeezing both data and control packets over a single 3.088 Mb/s link, and the difficulty of uniquely matching T&M packets with their associated data packets.

(5) Although the precise limitations on PSAT performance have not

yet been completely established. it is clear that there is a requirement for a packet satellite controller with a significantly higher packet throughput capability and much lower cost.

3.2.2 Proposed Assignment of Station Functionality

The essence of our recommendation is to implement an integrated Voice Funnel and PSAT on a single Butterfly multiprocessor. The assignment of functions to the two station subsystems which would exist is as follows:

Packet Satellite Controller (Voice Funnel + PSAT)

- Subscriber Interfacing
- Data Aggregation
- Gateway Translations
- Global Time Synchronization
- Channel Scheduling
- Congestion Control
- End-to-End Flow Control
- Host and Channel Monitoring
- Network Management
- Internal Hosts

ESI

- Burst Modulation/Demodulation
- Multi-Level FEC Encode/Decode
- Precisely Timed Burst Transmission
- Downlink Time Stamping
- Burst Filtering
- T&M Data Collection

Note that the duplicated requirements for precisely timed burst transmission and downlink timestamping have been eliminated by moving these functions out of the controller and into the ESI. Some form of these functions must exist in the ESI in any case due to the variable processing delays that it introduces.

As indicated above, the extent of the ESI's downlink processing problems are not well understood at present. We have not, therefore, attempted to explicitly indicate how this potential problem should be solved. The simplest solution from a systems point of view is for the ESI to carry out its task as currently defined. However, if this is not possible and the ESI must know ahead of time what to expect on the downlink in order to operate acceptably, there does not appear to be any reason why the controller cannot pass along schedule information to the ESI in the form of local control packets. The local timing function, which would be moved from the PSAT to the ESI, will facilitate

such interactions.

It should be noted that PODA as currently defined does not provide burst structure (coding and modulation type) information in reservations and thus any attempt to provide such information to the downlink ESI in advance of the actual packet transmission will require a change to the channel protocol. We have looked into this area and our preliminary conclusions are that providing such information is feasible, especially if a burst rearrangement capability is provided (see discussion of more states per burst below). There are still a number of issues that need to be evaluated in this area. however. such as the impact of the change on initial acquisition and the maintenance of global synchronization.

3.2.3 The Controller-to-ESI Interface

We propose to make a number of specific changes to the existing PSAT/ESI interface. Specifically, we want to:

1. Reduce Stateword Limitation on Burst Structure
2. Support Uplink Buffering and Burst Transmission by ESI
3. Support Downlink LocalTimestamping by ESI
4. Separate Data and Control Connections
5. Revise Electrical Signaling

6. Expand Local Time Clock to 32 bits

BBN has already expounded on the problems that we see with the current limited number of statewords. Our position is contained in a message sent to DARPA and Linkabit 12 February 1981. This subject was also addressed at the May 1981 Wideband meeting at Lincoln Laboratory. Our recommendation has been that the PSAT/ESI interface formats be revised to support up to 127 statewords rather than the current 15. We will not belabor this point any further here. We should point out. however. that an alternative to supporting more statewords with the proposed Butterfly-based PSAT design is to support rearrangement of burst segments so as to eliminate the need for additional statewords. (At most 8 statewords are required for 4 coding rates and 2 modulation types.) The disadvantage of this approach is that it is incompatible with the existing PSATs deployed in the field. BBN and Linkabit should jointly evaluate these two options if our basic station design is adopted.

Items 2 and 3 taken together are the specific interface changes necessary to support the movement of precise timing out of the Controller as described above. The basic idea is that the ESI should maintain a Local Time Clock analogous to the one which the PSAT's SMI currently maintains. Rather than transmit bursts to the ESI at "exactly" the right time, the Controller would

transmit bursts to the ESI ahead of time with a timestamp indicating the precise time at which the burst should be transmitted on the uplink. In addition, the ESI would use the same area in the burst to record the local receive time based on the Local Time Clock for every burst received on the downlink (note that this eliminates the need for the currently defined Burst Delay Word). The PSATs still provide global time synchronization under this design and can translate between global time and local time based on an offset determined from the leader packets. The timestamp in each burst can also be used as the unique identifier to match up T&M local control packets with their associated data packets.

The fourth item that we would like to change in a next-generation PSAT/ESI interface is the basic structure of the interface with regard to data and control information. Currently, both data and control information are sent over a single pair of data lines (TRDATA and RECDATA). We propose to split the interface into essentially two distinct parts. Globally scheduled bursts (primarily data) would be sent over one interface that runs at or above the channel data rate. Local control packets (including T&M data packets), on the other hand, would be transmitted over a completely separate link running at (perhaps) a slower rate. This approach should solve the current problem, noted above, of finding a place to squeeze in local

control packets. a problem particularly acute for the ESIs at present.

The movement of the precise timing function out of the PSAT and into the ESI opens the door to a simplified set of electrical signals. Our basic proposal is that both the data link and the control link be implemented as standard communications interfaces with free running clocks. Both interfaces would consist of only Receive Data. Transmit Data. Receive Clock. and Transmit Clock signals. Hardware framing on the control link should be compatible with standard HDLC. Framing on the data link should continue to be the current 3 character header SYN DLE STX in order to provide compatibility with the existing PSATs.

A final change we propose to make to the Controller/ESI interface is the expansion of timestamps from 24 to 32 bits. This would greatly simplify the maintenance of the controller's 32-bit global time clock which, in the PSAT. must be awkwardly maintained in both hardware and software.

4 REMOTE SITE MAINTENANCE

The heart of remote maintenance lies in software control and distribution. If working software is distributed correctly, then there will be fewer problems requiring the attention of the system maintainers, allowing them to devote their attention to genuine software bugs. In an environment where software is continually being developed and improved, and where bugs in old software are being fixed, considerable effort is required to keep the software at remote sites current with that of the development site.

In the past, we have used various ad hoc schemes to keep the problem at bay; all of these use a fair amount of human resources, with the frequent accompaniment of human error, resulting in confusion and inconvenience while installation errors are detected and repaired. During the end of the last quarter we began development of a new program intended to automate software control and distribution.

Software development for the ACCAT Remote Site Modules is concentrated on a single machine. This machine, designated the source machine, receives frequent installations of new software. The software on the target machines (RSMs) will lag behind the software on the source machine. On the other hands, the RSM software itself should be consistent; that is, except for

differences required by variations in configuration, the software on any two RSMs should be identical.

When a new distribution is made, the choice of changes to be made on the target is driven by two databases: the UNIX file-system and a special database (described below) which describes each controlled directory for each host.

The file-system on the source machine serves as a prototype working file-system, implicitly embodying the relationships among various software modules. The software update mechanism tries to transform the target's file-system into a copy of this prototype. A status file exists on the source machine for each controlled directory. This file contains ownership, protection, and link information for all files in the directory (similar to a directory listing) as well as a checksum of the contents of these files. A similar file exists on the target machine. During the distribution process, this file is copied to the source machine. The two status files are then compared, and where the entries differ, the software update program decides what action, if any, is appropriate to reconcile this difference. Commands to adjust protections, ownerships, or links, are issued to the target machine; if the checksums differ, a new copy of the file in question is sent, with instructions for its installation via the BBN-UNIX "install" facility.

It is not always appropriate for a remote system to slavishly imitate the source system. Hardware capabilities differ from machine to machine, requiring distribution programs to either be able to determine what capabilities are present. or which elements of the system should be tailored for the individual site. Exceptions arise for various other reasons: user habits may differ. historical accidents cause inconsistencies. site missions differ. As a result, simply installing an identical copy of the source machine file-system is not enough.

The software update program will decide which directories on the target are controlled either by being told explicitly through command line arguments. or by reading a database file, similar to a Makefile. This database file will specify, for each directory:

a) Whether or not to remove files in the target machine's directory that are not in the source machine's directory. Directories that might not want to have unshared files removed are /usr/lib and user's personal directories (we hope to make this program flexible enough to be used by individuals).

b) Whether or not to recursively descend into subdirectories of this directory.

c) How long files in this directory should be held before being distributed. Controlled software on the source machine will normally be tested in ordinary use on the source machine for some period before being distributed to remote sites.

d) Subdirectories or files that are not to be updated (exceptions).

e) A preprocessing command to be executed in the directory on the source machine (an example: "make rmobjects". which is used to clean out the current source system directory).

f) A preprocessing command to be executed on the destination machine before doing the update.

g) A post-processing command to be executed on the source machine after the update has taken place.

h) A post-processing command to be executed on the target machine (e.g., "make all.install". which would be used to actually install all the programs on the target machine).

Machines of different architectures (or even different operating systems) could be maintained with this mechanism by controlling source directories rather than binary directories.

For the most part. the CPF and NPS RSMs can be updated from the BBN site by maintaining their directories of binaries. /bin, /usr/bin. etc., with the sources being kept on the machines as a matter of courtesy. The NOSC machine. for historical and hardware reasons. should be maintained by updating source directories. with a post-processing command that recompiles and installs the programs. to allow for its different disk structure. and the peculiarities of its Teletype defaults.

To support this scheme a new command execution facility has been installed in the FTP program. This facilitates interrogation of contents and attributes of the files in the controlled directories on the target machine. and the updating operations on these directories. All transactions between the source and target machines take place through an FTP connection.

5 INTERNET OPERATIONS AND MAINTENANCE

Activities during the current quarter were directed toward two major efforts: (1) continued operation and maintenance of the gateways; and (2) refinement and beginning implementation of a plan for improvement and continued development of the Internet system. We have in general adopted the approach of using the current gateway system as a testbed. We have begun the process of converting the gateway implementation into a system which will provide the increased performance capability needed to support the growing user community, as well as including the mechanisms for monitoring. maintenance. and control needed in an operational rather than a research environment.

The process of detecting. isolating. and repairing performance problems reported by users also provides operational experience which is useful in the design of subsequent gateway releases. as well as in the ongoing research into the architectural issues of the Internet system as a whole. This latter work is being performed primarily under the Routing Study contract; the effort reported herein is aimed at supplying details of operational experience with the current system as inputs to that research effort. In turn, the outputs of that effort are being used to develop a design for a future release of

gateway software.

5.1 Gateway Development

During this quarter. responsibility for all gateway maintenance and development was transferred from the Information Sciences Division to the Computer Systems Division (now Communications Systems Division). The motivation for this transfer was the need to emphasize the treatment of the gateways as an operational communications system, rather than a research tool to support the growing user community. In this approach, we plan increasingly to treat the gateway system much as we do the ARPANET and SATNET systems in terms of monitoring and maintenance. This will require increased emphasis on the development and enhancement of tools for remote operation of the gateways.

This approach will provide the earliest achievement of stable service for the user community which will be dependent on the Internet System for communications service. A goal of the subsequent research activities is to investigate techniques which facilitate operation of a system in which all components are not centrally managed. since this capability will be needed as the Internet System evolves and expands.

In addition, the Internet system is expected to evolve as new facilities are implemented, new functionality added, and protocols and procedures modified or developed to support these enhancements. One of our major goals is the development of an environment in which operational gateways can coexist with research-oriented gateways, to provide a stable communications base while still permitting new ideas to be implemented and explored.

5.1.1 Development Plan

During the quarter, we developed a plan for continuation of the development of the current gateway system. The requirements placed on this plan were the following:

- utilize the existing hardware and protocol base,
- improve performance as much as possible,
- implement missing functionality,
- implement new functionality recently defined,
- introduce mechanisms to improve maintainability,
- provide support for new sites and for growth in topology,
- plan for conversion to different, or enhanced, hardware,
- plan for support of operational as well as research activities,
- plan for subsequent evolution of the system, introducing the research results in an efficient fashion.

The plan we developed and submitted for approval basically involves adoption of a more efficient gateway implementation in order to provide space for the additional functionality and to increase performance. We expect this implementation to be adequate for current needs, and to serve as a stopgap while development of a subsequent system design. of hardware and software, proceeds. The reasoning and details of this plan are presented in the remainder of this section.

The basic premise is that we need a transition plan to cover the period from now until the time when a gateway system can be fielded which implements the results of the current work in the Internet research community. The current gateway system is inadequate from both a functionality and a performance standpoint, and therefore the transition plan cannot involve immediate work on the implementation which is ultimately desired.

We will have to support a variety of gateways using the existing PDP11/LSI11 hardware base over the next year or so. This includes the four SATNET gateways. development and operational gateways on various networks at BBN, all packet radio gateways (at Ft. Bragg, Ft. Sill, SRI. and other possible sites), a second gateway to dual-home the U.S. SATNET access, and new SATNET gateways (LSI11 based) in Germany and Italy. Other gateway sites which are currently unknown may also develop.

Support means a variety of things. First, it involves simply keeping the systems running. Second, it requires the addition of new functionality which is not deferrable (e.g., ICMP, extended addresses, etc). Third, it involves tracking down problems reported by the user community. Fourth, it includes analyzing and fixing performance problems which might surface, either reported by the users or which we note as a part of normal monitoring activities.

In parallel with supporting the existing gateway implementations, we must begin work on a subsequent gateway implementation which incorporates the results of the ongoing research as well as the experience gained from the operation of the current gateways.

This reasoning leads to the following proposed plan:

1. Devote the needed effort to keeping the current system going, but at the same time resist requests for new features or tasks unless they are clearly necessary.

2. Develop a second-generation gateway, for a set of 'backbone' gateways. Backbone gateways are ones on which 'real users' as opposed to researchers depend. The set of such backbone gateways must be identified.

3. Develop a third-generation gateway, which implements new

protocols such as routing and host access, and configure a test Internet to evaluate and test these ideas in the Internet research context.

4. Field third-generation gateways.

   At this point in time, we purposely avoid selecting a particular hardware base for the third generation gateways.

There appear to be two credible choices for the second-generation gateways. They could be C/70s, running a not-highly-optimized CMOS. This approach could be expected to at least improve performance by putting in bigger buffers, and by introducing some better monitoring/control facilities. The other credible second-generation choice is the existing LSI11 and PDP11s, using a recoding of the gateway into assembly language to recover buffer space and improve performance.

As far as the third generation goes, we have similar questions, except that less is known about the requirements. A variety of hardware choices are possible, including C/70s, 68000s and PDP-11s. If there will be many gateways deployed, then hardware development as needed is a worthwhile effort.

We have evaluated the pros and cons of these issues, and are currently proceeding with the first stage of the following development plan:

1. Second generation gateways using LSI11s. They are inexpensive and readily available, and have most of the needed network interfaces available.

2. Third generation development gateways with C/70s. Build at least the test cell with these, and develop some extra interfaces (e.g., local nets, since a gateway between a 10megabit and 50kilobit network will probably stress the design, and should be included in the test cell).

3. Third generation fielded gateways using hardware selected after evaluation of the prototypes in the test cell.

This plan recognizes the fact that the proper hardware choice cannot be made until after a better understanding is available of the user requirements and the system requirements, from experiments run in the test cell. The test cell would involve a set of gateways scattered around the Internet sites, which is for research, not to support operational traffic.

5.1.2 Gateway Release

We have begun the first stage of the plan outlined above, namely the creation of an assembly-language version of the current gateway implementation. Preliminary results indicate that this will achieve a factor of two reduction in the memory

requirements for the code, which will provide space for increased buffering and for instrumentation packages.

The macro-11 gateway will provide users with Internet service that is functionally equivalent to that provided by the current BCPL gateways with the following exceptions:

- Packets with options will be fragmented if necessary.
- ICMP protocol will be supported. The gateway will send Time Exceeded. Parameter Problem, Echo, and Information Request ICMP packets. Destination Unreachable and Redirect packets will be sent using GGP protocol until the Internet community switches to ICMP.
- Initially, Source Quench and Timestamp packets will not be supported.
- Network Address formats as specified in the September 1981 Internet Protocol Specification (RFC 791) will be supported.
- The gateway will contain an internetwork debugger (XNET) that will allow the gateway to be examined while it is running.
- Buffer space will be greatly expanded to provide better throughput.
- ARPANET RFNMs will be counted so the gateway will not send more than 7 outstanding messages to an ARPANET host.

We anticipate that the first releases of this gateway will occur during January 1982.

5.1.3 Architectural Design Work

Efforts during the quarter continued to cooperate with the study work under the ARPANET Routing Study contract, to investigate the behavior of the Internet as a system. In general, the Internet efforts have served to provide operational experience with the current system, to be used as inputs to the study efforts. One particular experience which was observed during the quarter proved particularly interesting, and addresses the use of the 'Source Quench' mechanism of the current protocol.

We observed some unexpected behavior during a TCP connection between the DIV5 TAC (homed on net 3), and the BBN-VAX (homed on net 10), using the RCC/ARPANET gateway. The symptom was observed during normal use of the VAX from a terminal. Whenever the VAX was outputting many characters to the terminal, in response to a command such as 'directory', the TAC (yes, the TAC!) would receive Source Quenches. Since ALL of the traffic was heading from the VAX to the TAC, this was unexpected, since the TAC had no data traffic to quench.

Basically, the VAX can keep the gateway busy handling packets to the TAC. In fact, at any given time, you might expect that most of the gateway buffering space will be filled with packets going to the TAC. There is, however, some reverse traffic, namely the ACK-only packets. Given that the gateway has

no real buffering policy to enforce fairness. it is therefore very likely that an ACK from the TAC will arrive to find all of the buffers full, and will be discarded. generating a source quench.

The problem in fact should become more serious if the TCPs involved use some of the typical heuristics to reduce ACK traffic, since the probability of an ACK getting successfully delivered goes down, and the probability of retransmitting goes up, which increases the amount of buffer space consumed by the VAX-->TAC traffic, which causes ACKs to be more likely to be discarded.

Various solutions to this problem were discussed. and several observations made. It is clear that at the minimum a more robust buffer management scheme would be useful, although not sufficient, to address this problem. Some fairly easy approaches might be to guarantee some amount of buffering for each input and output path. The situation in which this is insufficient occurs when there are relatively symmetrical traffic patterns. In the case above, this would occur if a second TAC was homed onto the same network as the first Vax, and a second VAX on the same network as the first TAC. With heavy traffic flow in both directions, the buffer management guarantee would not improve the probability of ACK-bearing packets being

successfully transmitted.

Other schemes which address this problem will undoubtedly be a little harder. For example. it might be reasonable to give 'control' traffic higher priority for resources. Currently. the only way a gateway could distinguish such traffic is by understanding the protocol below IP. namely TCP. to see if it contains ACKs. Tnis approach would likely prove unworkable.

We believe that alternate mechanisms must be investigated. One possibility. for example. would be to utilize a precedence or priority mechanism defined in the IP header. Datagrams so tagged would be afforded better service. TCP usage of this service would have to be defined. For example. a TCP might send an ACK packet with the priority service requested when it has received a number of duplicate datagrams. indicating that the previous ACKs had not been received in time to prevent retransmissions The interactions between such a mechanism and any mechanisms which attempt to set retransmission timers must be carefully investigated.

5.2 Operations and Maintenance

5.2.1 Topology

We investigated the issues involved in physical relocation of the SATNET gateway which is currently at BBN. This was motivated by the quadrupling of the line charge for the connection between the BBN gateway and the Etam Satellite IMP. The proposed alternative involves relocating the gateway closer to the Satellite IMP.

If the BBN gateway is to remain a development machine, the relocation would have a major impact on gateway development work. Frequent access is required for crash analysis with TTY dumps. If the BBN gateway becomes an operational-only machine, impact is minimal. A possible replacement PDP-11/40 to serve as a development machine already exists at BBN; this is the machine used for development of the gateway loader via SATNET and to be used for development of V2LNI software. It would need a DEC KW11P real-time clock for compatibility with field units.

5.2.2 Growth Planning

We held numerous informal discussions, in person and by network mail, to plan for the addition of gateways to support new user sites in Germany and Italy. Mike Brescia attended a planning meeting at ACC, to discuss the applicability of the

PDP-11 and LSI-11 HDLC and X.25 peripheral units under development there.

5.2.3 Packet Radio Gateways

We assumed responsibility for the packet radio gateways during the quarter, and performed several fault isolation and configuration tasks, including support for the Helbat exercises.

In response to reports of problems with the fielded gateways, we determined that there was a configuration mismatch. All three gateway versions (Bragg, SFO 2, and SFO 6 nets) were checked for configuration with information provided by Don Cone. The main problem was that some configurations had been set up for CAP5 and some for CAP6. When the CAP5 and CAP6 gateway configurations were matched to the PR nets for each gateway, the gateways were able to come up on both nets.

A second problem was investigated, in response to reports that IPRs would not complete down link loading (over radio link) while a gateway was connected. The gateway was observed, in its interface lights, continually busy trying to transmit data to the IPR. Disconnecting the gateway physically allowed the radio link loading to proceed. The IPR was observed bringing the 1822 ready line down for a period of 1 second, then up for 1 second,

continually cycling until the gateway was unplugged.

We concluded that both the IPR and the gateway were not working the ready line very well. The IPR seemed to be using the ready line to reject a packet when it was not ready for it. The gateway did not note the state of the ready line at all, except to mark a packet received in error. It did not appear that any work in the gateway ready line handling would fix the problem unless it was carefully tailored to the IPR ready line handling.

The PRnet interface in the gateway, when presented with a packet received with an error, as when the ready line is flapped by the IPR, immediately responded by sending a TOP back to the IPR, apparently assuming that the IPR was coming up and would soon need the TOP information. This appeared to be the cause of the IPR flapping the ready line in response to the TOP packet coming too soon after the ready line came up.

The gateway has been changed to remove the special sending of TOPs when the ready line error is detected. TOPs are now sent on a regular schedule, and the time has been shortened from 60 to 10 seconds. The time was a compromise between flooding the IPR and waiting a long time after the IPR came up.

5.2.4 UCL/ISIE Connectivity

We received numerous reports of problems with connectivity between Europe and the ISI machines. We worked with UCL, ISI, and BBN Information Sciences Division to isolate this problem. The symptom was refined, and determined to involve sudden loss of connectivity on the ISIE-to-UCL path, while the UCL-to-ISIE path continued to function, and connectivity to other machines at ISI also remained intact. The problem was determined to be in the Internet routing tables for ISIE, not showing itself on other TCP machines. This was isolated further to a bug in the mechanism which declared a network to be reachable again after an outage. Connectivity would be maintained until any momentary disruption, and would thereafter not be restored until the system was reloaded or manually patched.

In the process of analyzing this problem, we made a few changes to the gateway system, and some observations about current usage.

There has always been a problem with the gateway blocking the IMP when load is heavy, and one cause has been the absolute priority given to GGP packets. Normally, a packet gets queued for output if there is "room in the queue" as defined by the output queue length. If a non-GGP packet wants sending, it gets dropped if the queue is full; however, a GGP packet gets queued

regardless of the length of the queue. A GGP packet can therefore use the last scrap of storage, and leave none for input from the IMP. A patch was installed in the gateway which eliminates this special case check for GGP packets, so they will now get dropped with the rest.

Some statistics gathered during this work are interesting. In one instance, the data showed approximately 250 packets per minute entering the gateway, and 140 leaving. This is made up of GGP echos to and from other gateways and hosts, and RFNMs in for packets out. This reduces to 140 RFNMs and 110 data packets received. There were five gateways alive and a sixth being probed -- NDRE, RCC, MIT, two at SRI and Bragg (which was reported as host-dead). There were twelve TCP-based systems also "pinging" the gateway, namely bbna,b,c,d,e,g,isib,c,d,e,f,and mitxx. A 9600 baud packet-printer terminal could not keep up with the traffic.

There is a disturbing trend indicated in this analysis, namely that an increasing quantity of 'ping' packets is in evidence. and as more hosts join the net, and more hosts implement ICMP, the overhead packets can be expected to present an increasing load on the gateway processors and network resources. We believe that this problem should be addressed in the research community, to develop alternate methods for hosts to

determine connectivity and status information.

5.2.5 Ready-line Anomalies

As part of our O&M activities, we investigated another symptom which was traced to a ready-line problem. In this case, the symptom was that a Port Expander robustness loader would not work properly when connected to a C/30 IMP. The problem was determined to result from slight differences between the manner in which Honeywell 316 IMPs and C/30 IMPs treat the ready-line. We had numerous interactions with SRI, ACC, and the IMP group at BBN to isolate this problem.

The contributing factors to the overall failure of the robustness software when connected to a C/30 IMP at Bragg were:

- The robustness software insists on receiving the INTERFACE RESET message from the IMP.
- The C/30 IMP is currently willing to send bits when host ready line is false if the host interface is willing to claim READY-FOR-NEXT-BIT.
- The ACC interface is started up in the mode which will accept and discard all bits, until software has gone through initialization including a few seconds of timer waiting.
- The IMP queues the NOP and INTERFACE RESET messages about 1 second after the host ready line goes down.

The combination of these behavioral traits causes the INTERFACE RESET message typically to be read and discarded before

the initialization completes. Several approaches to solving this problem were investigated:

- The Robustness PROM could be modified to remove the requirement for an INTERFACE RESET.
- C/30 could be modified to ignore RFNB unless the ready line is up.
- The Robustness PROM could be modified to set 'enable receive' in the ACC interface very soon in initialization.
- The IMP software could be modified to queue the initial messaes after ready line goes up.

The approach selected was to turn on the receive enable immediately after performing the reset, to prevent the data bits from being flushed, and making sure that no other parts of the initialization turned it back off. This produces an initialization algorithm of the following form:

```
STARTUP:  reset interface                ;turns off receive enable
          turn off ready relay
          wait for time sufficient to let relay settle
          turn on ready relay
          wait for time sufficient to let relay settle
          turn on receive enable
          set up receive transfer to buffer
          start sending
          ...
```

We prepared the following guidelines for the handling of the ready-line, to supplement the discussions in the 1822 specification. Pertinent parts of BBN Report 1822 are: Section 3.2, pp. 3-7 to 3-14; Section 4.2, pp. 4-5 to 4-10; Section 4.4, pp. 4-11 to 4-14; and Appendix B, Sections B.1 and B.3.

(1) When does the ready line go down?

- When power is turned off, cable disconnected, diagnostics running, etc.
- When application program starts up.
- When application program suspects serious hangup in communication. The only "resynchronizing" done is when the ready-for-next-bit handshake appears hung up (tardy host) [1822, p. 3-12]. The ARPANET IMP waits for about 30 seconds to determine this.

(2) What to do to "flap" your ready line:

- RESET interface hardware and make your ready line false.
- The reset is especially important if any hardware hangup is suspected.
- Discard any packet being received -- it has been truncated anyway.
- Keep it false for at least 1 second -- other side may be polling slowly.
- Make ready line true.
- Wait for ready relay contact bounce to settle. One second is not too long; the ARPANET IMP would like at least 1/2 second [1822, p. B-8].
- Discard the first packet received; it is probably garbled from the ready relay contact bounce.
- Transmit a few packets which can be dropped by the other side, which should be discarding as above. ARPANET IMP sends 3 NOP packets.

(3) What if you see the other ready line go down (i.e., you saw it up and then it went down)?

- Flap your ready line (per above) and wait for the other ready line to come up. Be sure your ready line is up!

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 the system integration within the Advanced Command and Control Architectural Testbed (ACCAT) experiment at the Naval Ocean Systems Center (NOSC) in San Diego, California. Below are described some of the accompanying activities and problems encountered.

When we were at NOSC in June 1981 for the MATNET installation, we had successfully tested single MAT station operation with a FLTSATCOM UHF satellite on two separate occasions. Because of a lack of cryptos, however, testing of two MAT stations through the satellite had to be postponed. After we had left the site, a full complement of cryptos arrived, allowing site personnel to begin substantive tests of the system. In September 1981, we were at NOSC again to conduct satellite tests of the two-station system. Much to our surprise and dismay, we were unable to achieve performance good enough to make meaningful measurements, although we were able to successfully demonstrate MATNET functionality.

The poor system performance is attributed to cabling problems and satellite channel interference. Cabling problems consisted of opto-isolators wired backwards, open grounds, and

cross-coupling between wires. Indicative of the severity of the problems, clock signals between the Black processors and the AN/WSC-3 radios were triangular rather than square-wave in shape, which increases system sensitivity to noise.

Despite allocation of a satellite channel for our exclusive use during testing, frequently occurring channel interference was encountered. When interference is present, packet loss on the satellite channel increases from less than 1% to over 25%. Although the cabling problems are being resolved under the direction of E-Systems, ECI Division, no identification or resolution of the interference problems has been effected.

In June 1981, the MATNET Operations Center, formerly called the MATNET Monitoring and Control Center or MMCC but now called the Network Operations Center or NOC, was installed at NOSC. NOC is responsible for monitoring system performance and for correcting problems as they arise. The basis of NOC is a collection of specially-developed programs residing in the ACCAT DEC TOPS-20 computer system to aid operations personnel by collecting, tabulating, and filtering data from each of the Satellite IMPs. The particular programs involved are: RECORDER to process monitoring information received from the Satellite IMPs and record the information in a database for later use; MONITOR to convert the database information into English text for

display in real-time on a terminal; QUERY to compile hourly and daily statistics on traffic flow and error rates from the database information; and EXPAK to set up Satellite IMP parameters. enable message generators. and collect statistics on MATNET performance.

After we had left the site. a problem with the installation of RECORDER surfaced. The evidence indicates that RECORDER, after running for a couple of hours. enters into a state in which it is continually being swapped in and out of memory by the TOPS-20 EXEC operating system. While in this state, TOPS-20 cycles are consumed at a prodigious rate. We assume that the problem occurs because of differences in the ACCAT TOPS-20 system and the ISIE TOPS-20 system, where RECORDER has been running reliably for two years.

Our immediate response was to instruct ACCAT TOPS-20 personnel to remove RECORDER from among the jobs that are automatically started upon a computer restart. During our next trip to NOSC. we plan to delve into the interface between RECORDER and EXEC to fix the problem. Until then, RECORDER can be manually started by anyone requiring its operation for specific tests; at the conclusion of the tests, though, the job must be manually killed to avert runaway operation.

As part of our maintenance of MATNET, we assembled new

MATNET software, including addresses and patches found to be necessary at the NOSC installation, and transferred the software onto cassette tapes for loading C/30 machines and LSI-11/03 machines at NOSC.

During the past quarter, we have been expanding and amending the documentation on the Red subsystems. Included in the documentation are sections on operation of the Satellite IMP, the gateway, the Terminal Interface Unit (TIU documentation is written by SRI International), and the TOPS-20 programs forming the basis of NOC. We have also, along with ECI and NAVELEX as co-authors, submitted a paper for the INFOCOM 82 Conference describing MATNET.

Finally, during the past quarter we briefed several different organizations in the Department of the Navy on the design of the Priority-Oriented Demand-Assigned (PODA) channel protocol, which is incorporated in MATNET.

7 TCP FOR THE HP3000

During the last quarter work on the HP3000 Internet project has concentrated in two areas. First we have continued to test and improve the performance of the software developed over the past year. This maintenance work has included tracking down and fixing a number of minor bugs and identifying potential bottlenecks in data flow throughout the protocol software. Second, we have designed and are in the process of implementing a general Internet datagram interface. The most immediate use of the new Internet interface will be a test of the Internet Name Server now being developed for the VAX.

7.1 Maintenance Tasks

As a part of our maintenance effort we have developed a network control and monitoring program. This program can be used to start up or stop the network code as well as to print out status information on all network connections. In addition, this program includes the ability to trace all of the network traffic for individual connections. The trace feature allows a user to trace data flowing through all of the protocol layers to and from the network. The user can specify which protocol layers as well as which connections should be included in the trace.

Performance tests have so far revealed two problem areas in the protocol software. First, a flow control problem was detected in Server Telnet when there was a large disparity between the data rates of the pseudo-Teletype PTY and the Transmission Control Protocol (TCP) interface. Since Server Telnet's main function is to pass data between the PTY, which acts as a command line interpreter, and the TCP, which sends data over the network, any discrepancy in their data rates will force Server Telnet to buffer more and more data until it runs out of buffer space. In order to eliminate this problem, flow control mechanisms were implemented which throttle the data rate of either the network or PTY, as needed.

A second data flow problem was discovered at the Intelligent Network Processor (INP) interface to the ARPANET. This interface is a microprocessor which is used to pass data between the HP3000 and the IMP. The interface has a standard I/O driver which allows up to 7 simultaneous write commands from the HP3000 to the INP. While there is no limitation on the amount of data in each write request, any attempt to issue more than 7 write commands will be blocked by the driver. This means that an optimal data rate through the INP can be achieved only through writing large amounts of data with each write command. Since each 1822 message sent out over the ARPANET involves two or more separate data buffers and therefore two or more data requests, the limit of 7

simultaneous write commands really limits the data flow to no more than two or three 1822 messages. Since the protocol software can generate 1822 messages faster than the INP can absorb the two or three write requests in each message, a real bottleneck can develop at the INP interface. Indeed, tests have shown that the INP cannot keep up when the protocol software tries to send out a large number of small 1822 messages.

One solution to this problem would be to copy all multi-buffer 1822 messages into a single large buffer before sending the message to the INP. The extra data copy involved would take less time than that spent waiting for the INP to accept the multi-buffer message. Unfortunately this solution would require some modification to the IMP software so that it can accept single buffer 1822 messages while sending multiple buffer 1822 messages. This kind of software change in the IMP will not be easy and will therefore not be done in the foreseeable future.

7.2 Internet Datagram Interface

A new Internet datagram capability has been designed for the network protocol software. The Internet user interface will resemble that used by the TCP. A description of the TCP user interface can be found in the HP3000 TCP design document (BBN Report 4463). With the new Internet interface, user programs

will be able to open Internet Protocol IP connections and transmit data over them.

The user program has a number of options as to the types of Internet traffic it can send or receive. The type and format of the datagrams sent over the network are almost entirely under the control of the user program. This is because the burden of creating a datagram including most of its header is placed on the user program. The user program will pass the datagram on to the Internet Protocol software as a single buffer. The Internet Protocol software will only add the local host address and calculate the checksum before the datagram is sent out. The current version of the Internet Protocol software will not fragment datagrams.

On the receive side the user program can specify which datagrams it expects. This specification is made when the Internet connection is opened. The options include a specification to receive datagrams from one specific internet host or all internet hosts. In addition, the user must also specify the protocol type of the internet datagrams it expects to receive. Each connection can specify one protocol type. There is, however, no limitation on how many Internet connections a single user program can open at the same time. The ability to open multiple connections allows any user program to listen for

datagrams with any number of protocol types. In addition, any number of user processes can listen for and receive the same datagrams.

The Internet send and receive intrinsics use the same non-blocking I/O mechanism as the TCP. The results of the send and receive calls to these intrinsics are therefore returned by the IOWAIT intrinsic. On return from the IOWAIT intrinsic a file identifier of the Internet connection will identify it as the completion of an Internet I/O request.

The actual intrinsics used to implement the Internet interface are described below.

fileid := IPOPEN(ipbuf,foreign-address,protocol-id)

This command opens an IP connection to the user program. The parameters specify an internet connection buffer which is initialized by this intrinsic, the address of the foreign host whose datagrams will be accepted (an address of 0 means that datagrams from all hosts will be accepted), and what the protocol type of the datagram must be to be accepted. The IPOPEN intrinsic returns a file identifier which is used by the IOWAIT intrinsic to identify the Internet connection. A negative fileid indicates that the open failed because of an error condition.

IPCLOSE(ipbuf)

This intrinsic closes an IP connection. The parameter is a buffer of connection control information initialized by IPOPEN.

errval := IPRECEIVE(ipbuf,datagrambuffer)

This intrinsic queues a buffer to receive an Internet Datagram. Its parameters are the internet connection control buffer initiated by IPOPEN and a buffer to receive the incoming datagram. The buffer includes a header which specifies the length of the buffer in bytes. The buffer length is specified as datagrambuffer(UBO'LEN). On return from the IOWAIT intrinsic the start of the Internet datagram is specified by the datagrambuffer(UBO'DATA). The intrinsic returns a zero value if it succeeds and a negative error number if it fails.

errval := IPSEND(ipbuf,datagrambuffer)

This intrinsic queues a datagram buffer for transmission. Its parameters are ipbuf, the connection control buffer initialized by IPOPEN, and a buffer which contains the datagram. The datagram buffer includes a header which indicates the length of the datagram in bytes. The datagram length is specified as datagrambuffer(UBO'LEN). The start of the datagram is specified by datagrambuffer(UBO'DATA). The intrinsic returns a zero value if it succeeds and a negative error number if it fails.

8 TCP-TAC

The TAC project passed several important milestones during the last quarter. The TAC software was completed; the first operational TAC was installed; TACs are now being monitored via the Internet from the Network Operations Center (NOC). An IEN was released describing the monitoring protocol used; the Internet Control Message Protocol (ICMP) was implemented; and various bugs were found and fixed.

The first operational TAC was installed during the quarter: the DIV5-TAC, running on the DIV5NET, in a 64K C/30. It has been up and running for about two months.

The TACs on the ARPANET and DIV5NET are now being monitored from NOC. We are currently receiving Traps from both nets. This monitoring is completely Internet, and enables us to support TACs on any compatible network. The monitoring is being expanded to include status and statistics data.

The TAC monitoring is done using the protocol described in IEN 197, "A Host Monitoring Protocol", which was completed and released during the past quarter. We plan to use this protocol to monitor gateways and Internet IMPs as well.

The Internet Control Message Protocol (ICMP), as specified in RFC 792, was implemented in the TAC. It has been tested with

hosts on DCNET and with the BBN-VAX. That TAC also supports the host-required section of the Gateway-Gateway protocol (GGP). When the gateways convert to using the ICMP, the GGP will be removed from the TAC.

As a result of running the TAC in the operation machine (DIV5-TAC) and the test machine (BBN-TAC), various bugs have been found and fixed. This testing and bug fixing activity is continuing.

The development of the TAC was completed during the quarter. The maintenance and enhancement of the TAC software and hardware will be continued under the ARPANET Operation and Maintenance contract.

9 TCP FOR VAX-UNIX

Work continued on the VAX TCP project in the areas of general testing, bug isolation and fixing, support of multiple network interfaces and gateway routing, and performance evaluation and enhancement. Work on the VAX TCP project wound down in September, due to a gap in funding. Because of this, we were unable to distribute the second level beta-test version, as originally planned. We expect funding to recommence early in the coming quarter, and will then distribute to the following sites: Berkeley, Stanford, Purdue/Wisconsin (CSNET), Carnegie-Mellon, USC Information Sciences Institute, and MIT Lincoln Laboratories.

9.1 TCP/IP Enhancements

Aside from testing and bug fixing, several enhancements to the TCP/IP kernel implementation were completed during this period. A new low-level network access protocol/device driver interface was completed, which allows support of multiple network access protocol modules and multiple device drivers. This, along with changes to the IP layer routing algorithms, allows multi-homing. We are currently running this version of the implementation on our UNIX Cost Center VAX, with interfaces to both the ARPANET and a BBN internal network. While both network connections use the same network access protocols (ARPANET 1822)

and interface devices (ACC LH/DH-11), these modifications will also enable multi-homing on heterogeneous networks, such as the ARPANET and a high speed local network (e.g., ETHERNET), and/or use of multiple network interface devices (e.g., VDH or HDH for ARPANET 1822). The work required for this would be to write a network access protocol module and/or a device driver for each desired interface.

Gateway routing capability was also completed. The implementation now has a gateway table, and is able to understand gateway redirect messages. Testing of this feature involved communication with hosts on networks off the ARPANET, including COMSAT-DCN, UCLNET, and EDN.

Remaining work in the IP module includes implementation of ICMP message handling. The ICMP code has been designed, but will be coded and tested in the coming quarter. The ICMP code will probably be run in parallel with the existing GGP code, until the conversion from GGP to ICMP for IP host error messages has been completed.

The raw IP and local network user interfaces underwent testing and debugging during the quarter. While the raw message capability is now usable, an effort is underway to redesign the user interface to make it more useful for applications that require it.

9.2 Performance Enhancements

We have been working jointly with Berkeley on analyzing performance of the TCP/IP kernel, and on developing enhancements. This work has been conducted at Berkeley in close consultation with BBN. Performance analysis indicates that the majority of overhead in TCP/IP processing is in the following areas: context switching, subroutine calls, checksumming, byte swapping, and memory allocation. Profiling of the kernel has indicated that the breakdown of time spent in TCP/IP is 20% in checksumming, 20% in byte swapping, 10% in memory allocation, and the remaining 50% in data movement and protocol processing.

Steps have been taken to reduce this overhead. They include: modifications to allow the network software to run at software interrupt level rather than as a separate process, to reduce context switching; reorganization of the code to reduce subroutine calls, which are time consuming on the VAX; assembly language coding of checksumming and byte swapping routines; and revamping of the memory allocation scheme. In addition, minor modifications were made to the TCP module, to reduce packet traffic by avoiding unnecessary transmission of ACK-only packets. This tuning has resulted in a better than factor-of-five improvement in overall throughput. Throughput in excess of 1Mb/s has been measured over a 3Mb ETHERNET on a VAX 11/750 (the VAX

11/780 is approximately 1.4 times faster than the 750, and should yield a corresponding performance increase).

We expect to begin work on merging these enhancements back into the production version of TCP at BBN in the coming quarter.

9.3 Higher Level Protocol Software

Work also continued on testing the higher level protocol software: TELNET, FTP, and MTP, which are implemented as user level programs. A major milestone was the successful testing of FTP with COMSAT RT-11 implementations, running TCP/IP on LSI-11s. These sites had been the only other Internet implementation of TCP FTP. This testing revealed a problem in IP of sending packets larger than the 576 byte default packet size, which the LSI-11 hosts could not handle. The VAX TCP/IP will be modified to use the 576 byte default, and use the TCP maximum segment size option to send larger packets to those hosts which can handle them. Testing was also done with the HP-3000 FTP, which is being developed concurrently at BBN.

DISTRIBUTION
[QTR 23]

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
1850 Wiehle Road
Reston, VA 22090
Attn: Lt. Col. F. Zimmerman

DEPARTMENT OF DEFENSE
9800 Savage Road
Ft. Meade, MD 20755
R. McFarland R17 (2 copies)
M. Tinto S46 (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
J. Machado. Code 6134

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

DISTRIBUTION cont'd
[QTR 23]

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

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

M. Brescia
R. Bressler
R. Brooks
P. Carvey
P. Cudhea
W. Edmond
L. Evenchik
G. Falk
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
A. Nemeth
R. Rettberg
J. Robinson
E. Rosen
G. Ruth
P. Santos
J. Sax
A. Sheltzer
E. Shienbrood
E. Starr
R. Thomas
B. Woznick
Library