IEN 78
PRTN 267

Internetwork Experiments with the Bay Area Packet Radio Network

by John Shoch and Larry Stewart

February 1979 SSL-79-4, IEN 78, PRTN 267

Abstract

The Bay Area Packet Radio Network has been integrated into an existing architecture for network interconnection: it can be used to provide radio communications for directly connected hosts, or as an alternative packet transport mechanism to carry traffic between two local networks. The radio link is operational, and can support regular internetwork communications at rates ranging from 12 to 20 kilobits/second, with the current Packet Radio equipment.

Use of the Radionet required only the writing of a *network specific driver* for the standard communications software package. The driver is responsible for controlling the interface to the Radionet, performing fragmentation and reassembly of large packets, routing within the Radionet, and similar tasks.

This report is also being distributed as Internet Experiment Note #78, and Packet Radio Temporary Note #267.

Key words and phrases: Internetworking, Packet Radio, gateways, protocols, fragmentation.

CR Categories: 3.81.

XEROX
PALO ALTO RESEARCH CENTER
3333 Coyote Hill Road / Palo Alto / California 94304

1. Introduction

The use of packet radio as a communications system can provide some very attractive capabilities, including mobility, ease of reconfiguration, and potentially high bandwidth; the Bay Area Packet Radio Network (the PRNet, or the Radionet [Kahn, *et al.*, 1978]) is an experimental effort to install and evaluate such a system. In addition to the PRUs, there is a minicomputer based Station responsible for network control and resource management. To date, much of the use of the Radionet has been to support communication between low-speed terminals attached to the Radionet and server hosts on the Arpanet; traffic is routed through an internetwork gateway between the two networks.

Computer communications activities at Xerox Parc have included the development of local networks such as the Ethernet multi-access system [Metcalfe & Boggs, 1976], as well as the design and implementation of a complete architecture of internetwork protocols. We have provided communications for over 500 hosts scattered around the country, attached to about 20 networks, of 7 different types. The availability of the Radionet in the Bay Area provided an unusual opportunity to experiment with the integration of radio communication into that internetwork environment, and to help test the performance of the Packet Radio system.

If a suitable number of Packet Radio Units (PRU's) had been available, it might have been most attractive to consider equipping a large number of hosts with radio connections, using the PRNet as the primary communications link supporting many different kinds of applications. The limited supply of PRU's made that impossible; an alternative approach was to experiment with the use of the Radionet as a path between two existing local networks, using one PRU attached to an internetwork gateway at each end. The Radionet would then serve as an experimental alternative route to the existing 9.6 Kbps phone line currently used between the gateways. (The actual implementation does support end-user hosts directly connected to the Radionet; but with the limited availability of PRU's, no such hosts are in regular operation.)

In either application, the Radionet link would certainly carry terminal traffic when users at one site accessed traditional time-sharing services at the other; in addition, it should also be able to support high-volume file transfers, access to page-oriented storage systems, disk copying programs, and other demanding host-host applications.

Against this background, the basic research plan emerged:

---Develop a hardware interface to the PRU.
---Write a network specific driver for the Radionet.
---Test and measure the initial performance of the Radionet.
---Connect two internetwork gateways.
---Integrate the Radionet into the existing internetwork system.
---Assess the actual use of the Radionet as a transit network.

At this time, we are operating two Packet Radio Units in Palo Alto: one is at the research center on the side of Coyote Hill (Parc), and the other is about a mile away at a Xerox site on Hanover Street. In addition to the Radionet link, there is a 9.6 Kbps phone line between the two sites. Figure 1 shows a geographic map of the area and Figure 2 is a logical map of the network connectivity: Figures 3 and 4 show the actual installations at Parc and Hanover Street.

2. Interfacing to the Packet Radio Network

The standard hardware host interface to the Radionet is the bit-at-a-time "1822" interface originally designed for Host-Imp connections on the Arpanet [BBN, 1975]. For this project, we have designed and constructed an 1822 interface for our most common mini-computer: now available in a printed circuit board version (Figure 5), it has been used both for connections to the Radionet and the Arpanet. The full duplex interface consists of about 50 MSI and SSI TTL packages, combined with a small amount of microcode. Communicating with a cooperating zero-delay interface, it can support instantaneous data rates of about 1 megabit/second in each direction. In practice, the delays are non-zero, and two of our 1822 interfaces talking directly to each other over a short cable have a measured peak data rate of about 667 kilobits/second in each direction. The electrical interface is of the "Distant-Host" type, capable of supporting communications over distances up to 1000 feet between Host and PRU.

While the hardware interface for moving bits to the Packet Radio Unit is essentially compatible with the Arpanet 1822, the Host-PRU protocol is very different from the Host-IMP protocol, using different packet formats and different control procedures. Known as the Channel Access Protocol (CAP), it provides packet-level access to the Radionet and specifies some of the network control mechanisms. Unlike the corresponding Arpanet protocol, CAP does not by itself guarantee reliable delivery between two hosts on the Radionet; it does listen for hop-by-hop acknowledgments from the next radio, and performs limited retransmission to help improve reliability. From our point of view, the Radionet acts as a "best effort" packet transport medium.

3. Network software for the Radionet

Two different software modules have been used for driving the Radionet interface. One is written in BCPL, for use in a display-based diagnostic program used for testing. Another is written in Mesa [Geschke, *et al.*, 1977], for use in measurement programs and in the actual gateway software.

The BCPL test program provides an interactive facility for debugging both the hardware and the low-level software: it has a "packet editor" for hand constructing test packets and analyzing received data, and for debugging can utilize either the Host-IMP protocol in the Arpanet environment, or the CAP protocol in the PRNet environment (Figure 6).

The Mesa module is a *network specific driver* for inclusion in our standard communications package. The driver is responsible for *encapsulating* general internet packets for transmission through the Radionet, and for managing network specific protocols. As mentioned above, our current internetwork architecture and protocols provide access to hundreds of hosts. spanning many networks: both the measurement program and the gateway program use the same communications package. and are compatible with the internetwork design (see Figure 7).

The PRNet driver is responsible for all network-specific routing and control functions. The most unusual aspect of the Radionet driver is the need to utilize the relatively small packet size specified by the CAP protocol: our internet packets may occupy up to 277 16-bit words (10 words of header + 266 words of data + 1 word of checksum). while a maximum size Radionet packet will hold only 127 words (11 words of CAP header and up to 116 words of data). Thus. it may be necessary to *fragment* any internet packet which is to be routed through the Radionet. As is our usual practice in such situations. we prefer to adopt a form of *network specific fragmentation*: let the driver at the entry gateway fragment the internet packet in an appropriate manner. and have the fragments reassembled at the exit gateway. before the packet is routed to its next destination [Shoch. 1979]. This means that the gateway program itself need not know anything about the fragmentation. but the fragments must all arrive at the same exit gateway for reassembly.

If the packet radio driver is asked to transmit a packet which will not fit in a single Radionet packet. it is broken up into at most three fragments which are transmitted unreliably and independently to the PRNet destination. In addition to the CAP header. two words of the CAP data field are used for *network specific encapsulation.* One word is a packet sequence number used only for accumulating statistics. and the second encapsulation word is used to indicate if a packet is actually one of several fragments which must be reassembled (Figure 8). Only the first fragment contains the internet header. and it is the responsibility of the driver receiving fragments from the PRNet to reassemble them into the original internetwork packet. If all of the needed fragments are not received within the *reassembly timeout* any accumulated fragments are discarded. Thus. loss of a single fragment results in the loss of the entire internetwork packet. Since each packet may require up to three fragments. in the worst case it is conceivable that the internet *packet* loss rate on the path through the Radionet could be as much as three times the *fragment* loss rate in the Radionet: so long as the Radionet performs reasonably well. this is not a significant problem.

To avoid fragmentation. an internetwork packet must contain no more than 103 data words:

103	words of user data
11	words of internetwork overhead
2	words of encapsulation
11	words of CAP header
127	maximum number of words in a Radionet packet.

4. Phase I experiments: Initial "raw packet" tests

In order to gauge the performance of the Radionet we have carried out a series of modest test and measurement experiments: with several different test programs. we have tried to determine the capabilities of the Radionet. and assess the results when modifying the packet size. the PRNet control parameters. or other variables.

All of the measurements which we have conducted report the actual amount of user data successfully received through the Radionet, excluding internetwork headers and Radionet headers (11 16-bit words each), and not counting any packets which were lost or duplicated. It is our position that actual user data received is the most appropriate metric. All of the results reported below represent the average from several runs; unless otherwise indicated, variation among the runs has been on the order of 2-5%.

To simplify matters in this first phase, and to avoid any second-order effects, many of the initial tests described below were run with internet packets of limited size, so that fragmentation would not be necessary in the Radionet: 100 words of user data + 11 words of internet header + 2 words of encapsulation + 11 words of CAP header = 124 total CAP words.

These tests were conducted in September 1978 under CAP 4.7.2, using PR 12 at Parc and PR 15 at Hanover Street.

Part I-A: Loop-back tests

The first set of tests was run simply using a host directly connected to the network, sending internetwork packets but without the need of an internetwork gateway. The microprocessor control program in the PRU (CAP) includes several control variables for altering the mode of Radionet operation, but for our initial tests all PRU control variables were left at their standard settings.

To establish an initial baseline for performance of the Radionet, we utilized a series of loop-back tests under a variety of traffic types and paths. By loop-back, we mean the repeated transmission of regular internetwork packets, returning to the source; these are unreliable "raw packets", with no internetwork acknowledgments being generated -- the source machine simply maintains a non-empty output queue at the 1822 interface driver, hoping to keep it busy.

The results are kilobits per second (Kbps) or packets per second (pps) arriving at the sink; from these we can derive the number of milliseconds per packet (ms./pac). (Note that a single host is serving as both source and sink, and the total throughput of that host would be roughly twice the numbers reported here.)

	100 data words/packet			1 data wd/pac	
	Kbps	pps	ms./pac	pps	ms./pac
Loop-back via software (copy)	160	100	10	100	10
Loop-back plug at host 1822 port	85	53	19		
Loop-back plug at PRU end of cable	78	49	20		
Loop-back in Parc PRU	17	11	91	20	50
Loop-back via Hanover Street PRU[1]	8.0	5.0	200	12.3	81
Loop-back via Station PRU[1]	9.7	6.1	164	12.5	80
Loop-back with 0.0.0.0.0 Route	21	13	77	19.7	51

[1]Two radio hops.

Observations

--There is a rather striking decrease in performance upon entering the PRU: throughput looping back through the PRU is only about 22% of the throughput measured when coming back through a loop-back plug at the PRU. Passing through the PRU added about 71 ms. of delay per packet.

--When looping back through the PRU or beyond, performance never exceeds 20 packets/second: looping back 100 word data packets through a distant PRU yielded performance of only about 5-6 packets/second.

--The performance of loop-back with an all zero route was surprising. With no explicit source route provided, we had expected that the packet would be routed through the forwarder in the PRNet control station; but it is clear that no radio transmission took place. Evidently, the station assigned a point-to-point route that eliminated the need to leave the PRU, and that yielded better performance than the explicit route.

--The reduction in performance resulting from moving the loopback plug from the Host end of the 1822 cable run to the far end is solely due to propagation delays. The1822 interface requires four signal cable transits per data bit.

Part I-B: Raw packet tests, with TRMIDY variation

In an attempt to improve these performance numbers, we have explored several changes to the default operation of the Radionet. Apparently as a primitive congestion control technique, the PRU's have been set up to introduce a mandatory idle period between the acceptance of any two packets through the 1822 port. This is not a particularly severe burden when supporting low speed terminal traffic, or when there is substantial inter-packet delay within the host; but it could substantially limit the performance of a higher capacity host connected to the Radionet.

TRMIDY is the Packet Radio digital unit parameter which specifies an enforced idle period between the receipt of a packet over the 1822 interface and the time when the 1822 is again enabled; the default value of TRMIDY is 80 ms. This parameter creates a packets/second limit which depends on packet length and 1822 transfer rate; the time per packet will be at least:

$$\text{packet transfer time} + \text{TRMIDY} = \frac{\text{\# of bits per packet}}{\text{1822 packet transfer rate}} + \text{TRMIDY}.$$

Thus, the packet/second rate will be at most the inverse of this amount. If the source host is not generating packets at a rate in excess of this, the effects of TRMIDY will go unnoticed; attempts to generate packets at a faster rate will be throttled by the PRU. If there is additional per packet delay in the PRU which exceeds TRMIDY, decreasing this interval will have no effect.

The following loop-back and one-way tests were intended to assess the effects of varying TRMIDY under an assortment of loads. These figures are for raw packets with either 100 16-bit data words per packet or one data word per packet: the data rate figures reflect this user data. TRMIDY values are in hexadecimal: each unit represents 500 microseconds.

TRMIDY	A0 80ms.	80 64ms.	60 48ms	40 32ms	20 16ms.	10 8ms.	02 1ms.
Loop-back inside Parc PRU (Route = 9012.FFFF.FFFF.FFFF.FFFF)							
100 wds/pac. Kbps	16.7	26.4	24.9	45.9	54.1	63.4	62.4
100 wds/pac. pps	10.4	16.5	15.6	28.7	33.8	39.6	39.0
1 wds/pac. pps	18.0	26.0	29.5	42.0	46.5	52.5	51.5
Loop-back inside Hanover Street PRU (Route = 9012.9015.9012.FFFF.FFFF)							
100 wds/pac. Kbps	8.0	8.7	5.0	11.1	10.4	10.2	7.0
100. wds/pac pps	5.0	5.4	3.1	6.9	6.5	6.4	4.4
1 wds/pac. pps	11.0	12.5	13.0	12	11.5	11.5	11.5
One way. Parc to Hanover Street (Route = 9012.9015.FFFF.FFFF.FFFF)							
100 wds/pac. Kbps	19.7	23.2	24.2	24.8	24.9	25.1	24.5
100 wds/pac. pps	12.3	14.5	15.1	15.5	15.6	15.7	15.3
1 wds/pac. pps	17.0	18.2	19.5	19.9	19.9	19.8	19.3

Observations

--It is apparent that the TRMIDY time interval is responsible for the additional delay introduced when looping back through the local radio: the host is trying to generate packets at a furious rate, and lowering TRMIDY increases the throughput from about 10 packets/second to 39. Thus, the PRU can be quite good at internal loop-back: the 62 Kbps figure above compares very favorably with the 78 Kbps possible with a loop-back plug. Changes in throughput in this case quite closely track the upper limit imposed by TRMIDY (see Figure 9). These tests do not necessarily establish the internal loop-back limits of the PRU since some part of the delay is in the host.

--In contrast, continued change in TRMIDY has little effect on throughput of anything that is actually sent out on the radio channel; other delays in the PRU exceed any delay imposed by the TRMIDY time limit. The initial reduction from 80 ms. to 64 ms. makes a measurable change; but with packets flowing only in one direction, reducing TRMIDY from 64 ms. to 1 ms. made no significant improvement in packet throughput (just over 15 pps. when carrying 100 data words, or almost 20 pps when carrying 1 data word). Even in the best case (one-way transmission of "raw packets", large packets, reduced value of TRMIDY), we still find that channel utilization is on the order of 25%, suggesting other significant delays.

--Thus, for packets being sent one-way between two hosts, or when looped back from a distant radio, the effect of changing TRMIDY is less pronounced: these transactions already have substantial packet processing time, and start out well below the limit imposed by TRMIDY. The throughput curves generally do not rise dramatically as one decreases TRMIDY (see Figure 9).

--Yet the non-monotone relationship of TRMIDY and throughput is a bit alarming: in two of the three cases, something seems to go awry for values of TRMIDY in the vicinity of 0060, perhaps reflecting some interaction with other time outs. In addition, we have no explanation for the decrease in throughput when the TRMIDY setting is reduced from 0010 to 0002.

5. Phase II: Some early byte stream tests

For a second series of measurements we have utilized internetwork byte streams: these are reliable, flow-controlled streams in which the destination host returns a short acknowledgment packet for each group of data packets it receives. Failure to receive an appropriate acknowledgment will cause the source host to time out and re-transmit. These tests were also conducted in September - October 1978, under CAP 4.7.2.

In this portion of the testing we also began to use full size packets, which would necessitate fragmentation when passing through the Radionet. We expected that the larger packet size would help to improve throughput, while the extra packet processing overhead in the gateways might tend to reduce the throughput

The byte stream protocol makes use of a "window based" mechanism for end-to-end flow control; for these tests, we set up the byte stream protocol to use at most five outstanding packets. The 100 word packet traffic may be thought of as having five outstanding PRNet packets followed by a small reverse direction acknowledgment. The 266 word packet traffic involves fragmentation and results in 10 full size PRNet packets plus 5 approximately half size PRNet packets, followed by a reverse direction acknowledgment. Any lost or substantially delayed packets may trigger a timeout at the source, leading to retransmission.

Part II-A: Byte stream base line

For these experiments the traffic source was a host at Parc directly connected to the Radionet. We began by establishing a byte stream which merely looped back through the 1822 interface; subsequent one-way tests actually sent traffic through the Radionet to a host at Hanover Street, passing through a single Radionet-Ethernet gateway there.

	100 data wds/pac		266 data wds/pac	
	Kbps	pps	Kbps	pps
Loop-back via software (copy)	85	53	217	51
Loop-back Plug at host	55	34	82	19
Loop-back Plug at PRU end of cable	51	32	78	18
Parc to Hanover St. (one-way)	16	10	16	10

Thus, when actually carrying data through the Radionet, the best performance was only 16 kilobits/sec through this reliable byte stream. Again, these tests were conducted with unmodified Radionet parameters.

Part II-B: TRMIDY variation

To explore any possible impact of the TRMIDY setting, another small series of measurements was taken, varying this parameter alone.

TRMIDY	A0 80ms.	80 64ms.	60 48ms.	40 32ms.	20 16ms.	10 8ms.	02 1ms.
One way, Parc to Hanover Street							
100 wds/pac, Kbps	16.1	17.2		18.2			
266 wds/pac, Kbps	16.3	18.6		18.8		18.6	18.1

Only several alternative values for TRMIDY were tested: the initial reduction from 80 ms. to 64 ms. helped to improve throughput. Further reductions in TRMIDY did not greatly improve performance; evidently Radionet processing time for these packets generally exceeded the TRMIDY limit, so reducing the setting only made inconclusive improvements.

Part II-C: Radio channel data rate variation

The default radio transmission rate is 100 Kbps, but the Radionet has the capability to send at 400 Kbps. For the next series of byte stream tests, the sources and destinations for the streams were in hosts directly connected to the Radionet (no gateways). We again reduced the TRMIDY setting to a smaller value suggested by the previous results, but also tried to improve byte stream throughput by increasing the data rate for the radio transmission.

TRMIDY	A0 80ms.	80 64ms.	60 48ms.	40 32ms.	20 16ms.	10 8ms.	02 1ms.
Parc to Hanover Street, 100 Kb radio							
100 wds/pac, Kbps	15					20	
266 wds/pac, Kbps	14					24	
Parc to Hanover Street, 400 Kb radio							
100 wds/pac, Kbps						24	
266 wds/pac, Kbps						30	

Observations

--The best result, about 30 kilobits/sec, was obtained using:

a. large packets.
b. 400 Kbps data rate, and,
c. a lowered value of TRMIDY

--The reduction of TRMIDY continues to be important, producing a significant improvement in throughput; but the subsequent change from 100 Kbps radio data rate to 400 Kbps produced an unexpectedly small performance improvement -- 20% for small packets, 25% for the larger ones. The high per-packet processing overhead in the present PRU reduces much of the potential improvement from increasing the radio data rate.

Part II-D: Two-way traffic

We made a limited number of measurements of the manner in which the Radionet could support two-way traffic -- byte streams moving in both directions. From the small amount of data collected, we feel able to report our initial impressions, but not any firm results. The two-way traffic seems to share the radio channel fairly well -- that is, the throughput is about the same in each direction -- but the combined throughput is generally somewhat less than half of the best possible one-way throughput. When traffic is flowing both ways at once, we have observed a significant increase in the number of packets lost in the Radionet; the required retransmissions help to account for the reduction in end-to-end throughput.

Some Remarks on Phase II

The PRU seems able to forward 50 packets per second from the 1822 back to the 1822, 25 packets per second from the 1822 to radio, or 13 packets per second radio to radio. The observed behavior might be explained if the handling of a low-level Radionet acknowledgment requires about as much processing as the original data packet.

Based upon the measurements of Phase I and Phase II, we concluded that under normal circumstances one could expect sustained *one-way* throughput of at most 16 Kbps. Unlike a full-duplex phone line, which has two separate channels, the Radionet shares its available bandwidth among packets going in both directions. Thus, with sustained two way traffic one would expect to see throughput in each direction at a rate somewhat less than 1/2 of this figure.

Furthermore, by modifying radio parameters one could increase the one-way performance to around 30 Kbps. These changes may not be beneficial, however, when overall usage of the PRNet increases. In addition, any manual changes to Radionet parameters are undone if the PRU is reloaded, and the change of radio frequency would isolate our units from the rest of the Radionet.

6. Establishing regular internetwork operation

The experimental objective had been to utilize the Radionet as a transit network between two local networks (Ethernet systems) located about a mile apart in Palo Alto. Thus, the PRU's were connected to machines which operated as internetwork gateways, running the regular gateway program. The earlier results suggest that, when fully operational, we could improve performance by modifying radio parameters and changing the radio data rate. At this stage of our effort, however, we preferred to remain compatible with the Radionet, and chose to proceed on that basis. Furthermore, we expected that actual gateway operation would yield performance results somewhat below the experience from the previous test programs.

As described earlier, one of the distinguishing characteristics of the Radionet driver is the need to perform fragmentation; in addition, the driver is also responsible for all other *network specific* functions.

Encapsulation

All traffic between our host machines consists of various kinds of packets encapsulated within CAP packets; one data word in the CAP packet is used to identify the particular kind of packet and includes fragmentation information. Another word consists of a destination specific sequence number used to help accumulate statistics; by observing the progression of received values we can count the number of fragments lost or received out of order.

Routing

Since the Radionet is a store-and-forward multihop communications network, routing decisions must be made to direct each packet. To reach a particular address, the user may explicitly supply a route, or allow the Radionet to select an appropriate path. We have chosen to make the PRNet driver responsible for routing. At present the driver uses *fixed source routing* [Shoch, 1978] for CAP packets; at initialization time, routing tables are established with entries for each potential PRNet destination.

Broadcast

The Radionet itself uses a broadcast communications medium, but it does not support any form of broadcast addressing, nor packets with multiple destination addresses. This makes it more difficult for a host trying to locate services on other machines, or for a new gateway trying to announce its availability. In the Ethernet style systems, for example, the ability to send a packet addressed to all hosts on the network allows any host -- including gateways -- to easily broadcast a packet which will elicit responses from other servers or gateways which might be running.

Without an actual broadcast capability, we have been forced to simulate this effect by implementing multi-destination routing in its most primitive form: merely sending multiple copies of the same request [Dalal, 1977]. In particular, each Radionet driver has built within it a table giving all of the addresses of up to 5 hosts or gateways which might be functioning on the Radionet. (In our current configuration there can only be two PRUs, but we have attempted to provide a little room for expansion.)

Host Status

Some internetwork protocols do not function particularly well if a network path between two gateways is up in only one direction: "half-up" links, which can be caused by a partial failure in a PRU, can potentially cause serious routing difficulties. For networks in which this may occur, such as the Radionet, the network specific driver has the task of verifying that the link to each host is working properly.

Approximately every 10 seconds, each driver constructs a special *I'm Alive* packet, and sends copies to all 5 potential gateways. These I'm Alive packets are specific to the Radionet, and are exchanged by the drivers; they are not internetwork packets exchanged by the gateway program. Each of these five I'm Alive packets is a legal CAP packet, including a Radionet header, 2 word of encapsulation, and 7 words of data.

The I'm Alive exchanges allow the driver to establish a list of which gateways are actually up, and also serves the important function of verifying that a link is up in both directions.

Radionet Protocol

About every 4 minutes, the Radionet driver generates and sends a *Terminal On Packet* (TOP), for use by the station controlling the Radionet. The format of this packet is as specified in the CAP protocol. These are the only packets generated which are not destined for another machine implementing the Parc internet protocols.

The PRU periodically sends a *Repeater on Packet* (ROP) over the 1822 interface back to the host; we currently discard these packets.

Address Translation

We have assigned a unique number in our network address space to the Bay Area Packet Radio Network (number 24); thus, the Radionet interface on the gateway at Parc has the unique internet address of Net 24, Host 1, and the Radionet interface on the gateway at Hanover Street is Net 24, Host 2. Currently, routing table space for three additional hosts (3, 4 and 5) is available, but these hosts do not actually exist. Since the Radionet does not use the same addressing structure, the driver is responsible for translating the internet host number (1-5) into the 16 bit ID used within the PRNet.

7. Gateway operation

All network specific functions described above are handled by the network drivers; the more general internetwork functions of the gateway are performed at a higher level in the gateway program, in a manner independent of any particular network.

While the primary task of the network independent part of the gateway is the store and forward switching of user packets, a number of auxiliary functions are performed.

For example, about every 30 seconds, the gateway program in the host constructs a new internetwork routing table packet, and asks each network driver to broadcast it to any gateways or user hosts which might be listening. The driver for the Ethernet system will actually broadcast a single packet; the driver for the Radionet uses the table it has constructed of currently functioning gateways, and sends a single copy of this packet to each one. Thus each of these packets includes a Radionet header, 2 words of encapsulation, an internet header, plus the routing data.

Data Collection

As part of its regular operation, the gateway program records significant events in a log file; in addition, various statistics are accumulated and written on the log file when the gateway shuts down. Both the internetwork portions of the gateway and the network drivers make log entries and keep statistics about their respective areas of responsibility; we have made use of both the event log and the termination statistics to collect data on the operation of the Radionet.

In three months of operation we accumulated the log files from some 70 runs of the Hanover Street gateway, covering something over 1200 hours of PRNet operation. The following table presents the combined totals from about the last half of this period. (During the first six weeks of operation we made a number of changes in the details of the PR driver statistics; these changes would make it difficult to present consistent results from the entire period.) A more detailed presentation of an individual gateway report can be found in Appendix I. The raw data for the following table may be found in Appendix II, with some annotations describing unusual behavior.

Cumulative Gateway Operating Statistics

With the exception of the operating time statistics all of the figures below are specific to the Radionet driver only.

Operating Time	
2 494 338	Seconds uptime
693:52:18	Hours uptime

Transmit Statistics	
1 723 909	Packets Sent
1 138 634	I'm Alive Packets Sent
394 870	One Fragment Packets Sent
54 362	Two Fragment Packets Sent
27 227	Three Fragment Packets Sent
60 567 215	Words Sent
23 747 935	Internet User Data Words Sent
10 680	TOPs Sent
166	Output Queue Overflow

10 708	1822 Transfer Timeout

Receive Statistics

1 122 484	Packets Received
409 444	I'm Alive Packets Received
506 956	One Fragment Packets Received
58 834	Two Fragment Packets Received
29 472	Three Fragment Packets Received
48 277 027	Words Received
41 729 145	Internet User Data Words Received
56 901	ROPs Received
179	Sequencer Reset
54	Old Packets Received
22 584	Packets Skipped
946	Assembly Timeout

Some notes about the statistics

Packet Counts

The total number of PRNet packets (fragments) generated may be computed as follows:

3 * (Three Fragment packets)
2 * (Two Fragment packets)
1 * (One Fragment packets)
1 * (I'm Alive packets)
Total Packets

Word Counts

Unfortunately, the computation of the word totals is done differently on the receive and transmit sides. On transmit, the word count is the actual number of words passing through the 1822 interface *excluding TOPs*. This *includes* the PRNet header, encapsulation, internet header (remember that I'm Alive packets do not have the internet header), and data. On the receive side, the word count includes the internet header and data fields of all internet packets, thus excluding both ROPs and I'm Alives. If we really want to see the word counts only for the internet data field, the two count values may be corrected as follows:

Transmit: Total word count less 20 words per I'm Alive and less 24 words per internet packet.
Receive: Total word count less 11 words per internet packet.

Congestion Control and Error Recovery

It has been our philosophy in designing internetwork systems that it is better to discard packets than to allow congestion to spread.

Output Queue Overflow

The PRNet driver limits the 1822 interface transmit queue to 12 internet packet buffers to avoid consuming too much of the gateway buffer pool; excess packets arriving at the output queue will force some of the oldest to be discarded.

1822 Transfer Timeout
An 1822 interface transmission failed to complete within 3 seconds.

Assembly Timeout
If all the fragments of a multiple fragment packet fail to arrive within 3 seconds, all fragments which have arrived are discarded.

Statistics Collection
The transmit side of the driver places a *destination specific* sequence number in each outbound fragment. The receive side maintains a table of *source specific* expected sequence numbers. Each arriving packet is checked for the proper progression of values.

Sequencer Reset: The encapsulation sequence number on an incoming packet differed from the expected value by more than 10.

Old Packet Received: The sequence number on an incoming packet was lower than expected.

Packets Skipped: The sequence number on an incoming packet was higher than expected.

Radionet Protocol
A small amount of network specific protocol is handled by the driver. The PRUs, network control station, and hosts exchange messages designed to determine the state of the net.

ROPs Received: A *Repeater On Packet* was received from the PRU (and discarded).

TOPs Sent: A *Terminal On Packet* was generated and sent to the Radionet station.

Observations

Over the entire period of operation, the *average* Radionet load was slightly over one packet per second and 30-40 words per packet. Taking into account the very high peak to average load ratio typical of computer communications, we feel these figures represent a very substantial test of PRNet capabilities. This observation should be heavily tempered by noting that some of the test intervals spanned 24-hour days, but most of the operating hours were overnight and on weekends (Appendix II). We would expect to see greater traffic during regular day-time operation.

8. Phase III experiments: Internetwork throughput measurements

Once in regular operation, we undertook a series of measurements, using regular internetwork byte streams, through various network configurations (Figure 10). These tests were done using an internetwork test program and the regular gateway programs.

During the time between the completion of Phase II and the beginning of these tests, a new version of the PRU software was released (CAP 4.8.1), the PRU's were replaced several times due to hardware trouble, and the Parc 1822 cable grew by 300 feet after several machines were relocated. Because of these events, the new results may not exactly match our previous efforts. (We should add that the program used for these measurements is a particularly useful test environment, but it does not necessarily represent our most capable program: other implementations and applications achieve even better network performance.)

These tests were conducted in December 1978 and January 1979 under CAP 4.8.1 with PR 12 at Parc and PR 21 at Hanover Street. The regular measurement program was used to send reliable, flow controlled streams of packets (with up to 5 outstanding packets at one time), and the radio parameters were kept at their default settings (100 Kbps, TRMIDY=A0), except for the very last test, in which we again experimented with a reduced value for TRMIDY.

location of the source & destination processes	maximum[1] packets per sec.	1 word per pac.	100 wds. per pac.	266 wds. per pac.
in the same machine	51 pps	809 bps	79 Kbps	200 Kbps
in 2 different hosts on the same Ethernet	64 pps	988 bps	102 Kbps	267 Kbps
hosts on 2 adjacent Ethernets, with 1 gateway	51 pps	810 bps	80 Kbps	203 Kbps
hosts on 2 Ethernets with a phone line & 2 gateways in between	17 pps	267 bps	6.8 Kbps	8.3 Kbps
hosts each directly connected to a PRU on the Radionet	10 pps	155 bps	15 Kbps	15.4 Kbps[2] 13.2 Kbps[3]
hosts on 2 Ethernets with the Radionet & 2 gateways in between	8 pps	120 bps	11.9 Kbps	12.4 Kbps
hosts on 2 Ethernets with the Radionet & 2 gateways, TRMIDY = 10 (8 ms.)	13 pps 11 pps	209 bps 181 bps	16.6 Kbps 15.8 Kbps	19.7 Kbps[2] 16.8 Kbps[3]

Notes:
[1]Computed as pps figure, for the smallest packets, with 1 data word/packet.
[2]Parc to Hanover St.
[3]Hanover St. to Parc

Observations

---Effective throughput through the Radionet is on the order of 12-15 Kbps, a little under the best achieved in the earlier phases of the measurements. By reducing the value of TRMIDY in the last test configuration, we were able to increase throughput significantly, to around 20 Kbps.

---We have observed a repeatable asymmetry in using the radio link, with results significantly reduced in the Hanover to Parc direction: this phenomenon awaits further exploration but may be due to differing radio unit performance in the two PRU's.

---In an initial set of measurements, we at times witnessed test runs using the Radionet move into a very different mode of operation, with a great many retransmissions from the source, and very poor performance. The high variability in delay through the Radionet was found to interact with some of the heuristics used by the end-to-end flow control mechanism, causing that mechanism to misestimate the network performance, and thus behave poorly. A small change to the flow control decision procedure eliminated the difficulty: it is worth noting that the Ethernet and the Radionet are systems whose overall performance differs by at least two decimal orders of magnitude -- a range which will continue to challenge the design of good flow control and congestion control procedures.

9. Conclusions

Looking backwards, the integration of the Radionet into our existing internetwork system has gone quite smoothly: both the 1822 hardware and the network driver for the gateway came up well ahead of schedule. The PRU's have failed on occasion, but in normal operation the Radionet functions quite well, losing few packets and introducing only an occasional duplicate. The measurement efforts have helped to elucidate some of the behavior of the Radionet, while the regular internet operation has provided a source of host-to-host traffic which has helped to exercise the Bay Area system.

The major source of disappointment is probably the modest overall performance, as indicated by fairly high overhead for handling each packet: this generally reflects the extent to which the PRU is processor limited. Using the Radionet as a transit network we can support about 12 Kbps of reliable traffic, or about 12% of the capacity of the underlying channel: the same test run through a 9.6 Kbps phone line yields a utilization of over 86%. Development of the dual-processor Improved Packet Radio (IPR) should have great potential for relieving the processing bottleneck in the PRU, thus helping to improve overall performance.

In the long run, of course, packet radio techniques will become even more interesting when the radio units become much smaller, less expensive, and therefore more widely available. We look forward to the time when we can experiment with some of the applications which will become feasible when packet radios become ubiquitous.

10. Acknowledgments

A cooperative project such as this one can only succeed when many different people volunteer their efforts, and we were unusually fortunate. David Boggs and Ed Taft built much of the internetwork system which made this all feasible, and helped at many points along the way; Hal Murray implemented the gateway program; and Steve Butterfield helped to keep it all running. We had some ups and downs as the first outside users of the Bay Area Packet Radio Network, but received a great deal of help from our friends at SRI (Don Nielson, Ron Kunzelman, Don Cone, Keith

Klemba, and Jim Mathis) and at Collins (Tim Quilici). Finally, Bert Sutherland and Vint Cerf provided the encouragement that helped make it all happen.

11. Bibliography

[BBN, 1975]

Bolt, Beranek, and Newman. *Specification for the interconnection of a host and an Imp.* BBN Report no. 1822, 1975.

[Dalal, 1977]

Yogen Dalal. *Broadcast protocols in packet switched computer networks.* Technical Report 128, Digital Systems Laboratory, Stanford University, April 1977.

[Geschke, *et al.*, 1977]

Charles M. Geschke, James H. Morris Jr., and Edwin H. Satterthwaite, "Early experience with Mesa", *Comm. of the ACM*, vol. 20 no. 8, August 1977.

[Kahn, *et al.*, 1978]

Robert E. Kahn, Steven A. Gronemeyer, Jerry Burchfiel, and Ronald C. Kunzelman, "Advances in packet radio technology", *Proc. of the IEEE*, vol. 66 no. 11, November 1978.

[Metcalfe & Boggs, 1976]

Robert M. Metcalfe and David R. Boggs, "Ethernet: Distributed packet switching for local computer networks", *CACM*, July 1976.

[Shoch, 1978]

John F. Shoch, "Internetwork naming, addressing, and routing", *Proc. of COMPCON Fall '78*, IEEE Computer Society, Washington DC, September 1978.

[Shoch, 1979]

John F. Shoch, "Packet fragmentation in inter-network protocols", *Computer Networks*, to appear, 1979.

Appendix I. An Individual Gateway Report

In order to clarify the kinds of information made available by the gateway, we present here an editted version of the Hanover Street gateway report for run number 68, covering about 2 1/2 days on 27-30 January 1979.

```
27-Jan-79 23:29:00  Gateway in Operation.
```

The gateway reports itself operational.

```
27-Jan-79 23:29:56  PR Host 1 Up
27-Jan-79 23:30:04  PR Host 2 Up
29-Jan-79  7:26:46  PR Host 1 Down
29-Jan-79  8:09:09  PR Host 2 Down
29-Jan-79 16:41:17  PR Host 1 Up
29-Jan-79 16:41:21  PR Host 2 Up
```

During the course of operation, the PRNet driver makes log entries as the I'm Alive mechanism reports various hosts up and down. The first two entries, made shortly after gateway startup, report that two way contact has been established with the Parc gateway and with ourselves (via a route which loops through the radio). The final four entries indicate a failure of the Hanover PRU during the morning of January 30, coming back up about 8 hours later. A failure of the local radio will naturally result in loss of contact both with the remote site and with ourselves.

```
Quit [Confirm] Yes.
30-Jan-79 14:03:11  Gateway up 62:34:55
```

The gateway is instructed to shut down.

```
Packets Forwarded:
from                     to
       Discard           20          24
  20           23        31       18921
  24            -     39249           -
```

This is the traffic forwarding matrix, recording the store and forward activities of the gateway. Net 20 is the Hanover Street local network, and net 24 is the Packet Radio net. Only *through* traffic is recorded here, excluding traffic addressed to or originated by the gateway program itself. Network specific traffic, I'm Alive, TOPs, and ROPs are *not* recorded here as they never reach the network independent gateway program itself.

```
30-Jan-79 14:03:19  Grand Totals.    225304 Seconds
```

Before it finally terminates, the gateway writes all of the accumulated statistics on the log file. We have editted out all but those associated with the Radionet.

	Packets	Alives	One-Frag	Two-Frag	Three-F	Words
Received:	98411	35303	44204	507	5963	2841722

```
Sent:        121446      88454      32441        16        173   3145151
```

These are packet and word statistics accumulated by the PRNet driver. The One-Frag, Two-Frag, and Three-Frag entries represent internet packets, including both the store and forward traffic recorded in the forwarding matrix and the gateway specific traffic (routing tables, etc.).

```
 830 PR TOPs Sent
6790 PR ROPs Received
```

Terminal On Packets are generated by the driver about every 4 minutes. *Repeater On Packets* are generated by the local PRU about every minute. These are network specific packets associated with management of the Radionet.

```
  1 PR Assembly Timeout
  7 PR Destination Down
 27 PR Output Queue Overflow
```

The *Assembly Timeout* entry records internet packet reassembly failures (due to lost or delayed fragments). The *Destination Down* entry counts the number of internet packets which arrived for a PRNet destination which was not up. The driver limits the number of packet buffers allowed to wait on the 1822 interface output queue in order to help avoid congestion. During momentary peak loads the limit may be exceeded, resulting in discarded packets.

```
  4 PR Sequencer Reset
  1 PR Old Packets
 59 PR Packets Skipped
```

The transmitting driver generates individual sequence numbers on packets addressed to each destination. The receiving driver tracks the actual versus expected sequence numbers in packets received from each host, which can be used to identify duplicates or missing packets. These numbers are used only to gather statistics. We see substantially more lost packets than out of order, but not very many of either.

```
    3 PR IMP Was Down
10193 PR Transfer Timeout
 4030 PR Output Packets Discarded
```

These entries record the error recovery procedures of the driver. *IMP Was Down* indicates the driver observed the PRU 1822 Ready Line in the off state, indicating the PRU was not ready to accept traffic. The 1822 *Transfer Timeout* indicates that an output transmission did not complete properly. In either case, the driver will *discard* any packets on the 1822 output queue. There is no particular initialization sequence for the 1822 interface, the driver simply tries again whenever a new packet appears on the output queue.

Appendix II. Collected Gateway Statistics

Periods of Gateway Operation

Run	Starting Time		Stopping Time		Up Time
42 --	14-Dec-78 19:41:52	to	15-Dec-78 8:49:44	up	13:07:53
43 --	17-Dec-78 12:47:38	to	18-Dec-78 8:53:24	up	20:05:48
44 --	18-Dec-78 20:15:52	to	19-Dec-78 8:53:58	up	12:38:06
45 --	20-Dec-78 22:55:17	to	21-Dec-78 8:29:41	up	9:34:24
46 --	21-Dec-78 19:11:14	to	22-Dec-78 8:30:05	up	13:18:51
46a--	22-Dec-78 17:57:33	to	26-Dec-78 14:35:13	up	92:37:31
46b--	27-Dec-78 00:15:06	to	31-Dec-78 12:42:28	up	108:26:58
47 --	31-Dec-78 15:14:49	to	31-Dec-78 20:41:55	up	5:27:06
48 --	31-Dec-78 21:29:37	to	1-Jan-79 18:30:41	up	21:01:03
49 --	2-Jan-79 18:38:50	to	3-Jan-79 8:31:18	up	13:52:27
50 --	3-Jan-79 19:06:30	to	4-Jan-79 8:48:43	up	13:42:14
51 --	4-Jan-79 18:32:39	to	5-Jan-79 9:03:07	up	14:30:28
52 --	7-Jan-79 20:41:57	to	8-Jan-79 9:00:49	up	12:18:53
53 --	8-Jan-79 22:26:51	to	9-Jan-79 8:22:50	up	9:55:59
54 --	9-Jan-79 18:38:38	to	10-Jan-79 8:50:29	up	14:11:51
55 --	10-Jan-79 18:13:51	to	11-Jan-79 8:54:31	up	14:40:41
56 --	11-Jan-79 17:29:28	to	12-Jan-79 8:57:52	up	15:28:23
57 --	12-Jan-79 18:20:46	to	15-Jan-79 8:56:49	up	62:36:03
58 --	15-Jan-79 22:01:37	to	16-Jan-79 8:28:01	up	10:26:25
59 --	16-Jan-79 21:46:59	to	17-Jan-79 8:47:49	up	11:00:50
60 --	17-Jan-79 19:20:02	to	18-Jan-79 8:34:00	up	13:13:57
61 --	19-Jan-79 0:16:38	to	19-Jan-79 8:45:21	up	8:28:58
62 --	21-Jan-79 11:58:31	to	22-Jan-79 8:53:55	up	20:55:41
63 --	22-Jan-79 22:02:17	to	23-Jan-79 8:38:50	up	10:36:52
64 --	23-Jan-79 17:07:15	to	24-Jan-79 9:58:05	up	16:51:09
65 --	24-Jan-79 18:22:49	to	25-Jan-79 11:29:33	up	17:06:51
66 --	25-Jan-79 23:11:48	to	26-Jan-79 8:53:10	up	9:41:39
67 --	26-Jan-79 19:31:50	to	27-Jan-79 21:57:59	up	26:26:23
68 --	27-Jan-79 23:29:00	to	30-Jan-79 14:03:19	up	62:34:55
69 --	31-Jan-79 16:00:16	to	1-Feb-79 9:07:33	up	17:07:22
70 --	1-Feb-79 19:19:13	to	2-Feb-79 8:49:55	up	13:30:55

On several occasions the gateway at Hanover Street has been up continuously for three or four days, including regular working hours. That machine, however, is also shared with other applications, and the gateway program was generally run evenings, at night, and on weekends.

Radionet Driver Receive Statistics

Run	Packets	Alives	One-Frag	Two-Frag	Three-F	Words
42	3081	2239	842	0	0	36479
43	12174	5577	3541	904	416	426054
44	223600	7731	50616	49338	22192	17004968
45	14815	5200	4489	23	1463	520984
46	12931	7675	5113	19	24	238904
46a	78764	45793	30067	113	669	1290392
46b	222693	68980	121055	3949	8213	5905826
47	16093	3541	4288	166	2644	927403
48	24990	13740	8672	164	750	471233
49	12771	9069	3703	0	0	159627
50	13205	8956	4217	4	8	172073
51	16411	9481	5923	22	321	318339
52	91309	8043	16494	472	21942	6992100
53	10402	6488	3001	10	298	209025
54	23226	9273	11801	170	604	498410
55	24278	9594	13339	188	323	493218
56	21711	10112	9317	89	701	473035
57	128326	40955	65821	1700	6050	3465045
58	10199	6822	3331	7	11	137430
59	10823	7193	3586	7	10	138689
60	6044	4323	1721	0	0	82526
61	8842	5538	3265	6	9	156884
62	51921	13642	28324	169	3203	1889874
63	9921	6935	2967	2	5	133642
64	27303	11014	12857	54	1108	683276
65	30501	11179	16020	197	969	750470
66	14380	6330	7560	83	108	265404
67	18114	9246	5571	42	1071	496747
68	98411	35303	44204	507	5963	2841722
69	23906	10642	9020	379	1162	628428
70	17924	8830	6231	50	921	468820
Tot:	1279069	409444	506956	58834	29472	48277027

Because of the odd hours of operation, the average load on the Radionet is fairly light, although there are substantial peaks with greater traffic. The unusually heavy traffic of run 44 was in large part due to a single-outstanding-packet test program left running all night. The program cyclically generated internet packets of all possible lengths -- it is this feature which results in roughly equal numbers of one and two fragment packets, but a smaller number of three fragment packets.

Radionet Driver Transmit Statistics

Run	Packets	Alives	One-Frag	Two-Frag	Three-F	Words
42	22308	21440	868	0	0	477645
43	39218	32566	3596	904	416	1166102
44	236610	20635	49723	49844	22188	20260481
45	18149	15143	2696	2	102	476788
46	26620	21740	4878	1	0	710152
46a	178809	151510	26546	198	119	4482426
46b	265953	177405	81584	2702	520	7469159
47	18151	8875	8930	158	10	905650
48	43672	34355	8983	8	106	1085985
49	28410	22660	5750	0	0	715792
50	26220	22385	3829	0	2	658451
51	28515	23700	4815	0	0	729064
52	25508	20110	5398	0	0	650262
53	18939	16210	2729	0	0	477322
54	31796	23175	8613	4	0	822279
55	33187	23975	9201	1	3	900915
56	32891	25275	7154	12	146	864954
57	150136	102395	43631	474	1054	4110018
58	19896	17045	2851	0	0	503365
59	21154	17975	3171	1	2	524534
60	23336	21615	1721	0	0	537199
61	16886	13845	3041	0	0	461570
62	66583	34196	26288	30	2013	2581252
63	20094	17335	2759	0	0	510783
64	38691	27535	11156	0	0	1014406
65	39941	27970	11971	0	0	1050794
66	19361	15820	3514	6	5	497180
67	48534	43230	5304	0	0	1117492
68	121446	88454	32441	16	173	3145151
69	34740	27980	5654	1	368	937342
70	28155	22080	6075	0	0	722702
Tot:	1723909	1138634	394870	54362	27227	60567215

By comparing the above table with the receive statistics on the previous page, it is evident that generally less traffic flowed from Hanover Street to Parc than from Parc to Hanover. There are two reasons for this: first, many of the services on the networks at Parc tend to supply more output than they receive input (file servers, time sharing systems), and second, the network topology results in a disparity in the way traffic is shared between the PRNet and 9.6 Kbps line at Parc and at Hanover Street.

Radionet Driver Net Specific Statistics

Run	TOPs	ROPs	Reset	Old	Skipped	Assy TO
42	201	--	67	0	11825	0
43	307	--	0	0	980	0
44	194	746	0	11	13	1
45	143	2000	34	131	2575	400
46	203	1077	7	0	870	18
46a	1419	6840	41	77	5394	427
46b	1662	7195	20	96	838	88
47	84	333	0	0	0	0
48	322	1317	0	1	0	0
49	213	847	0	0	0	0
50	210	836	0	0	0	0
51	222	3362	0	0	0	0
52	189	905	0	0	1	1
53	152	601	0	0	0	0
54	218	900	0	11	0	0
55	225	908	0	0	0	0
56	237	953	0	2	0	1
57	959	3931	0	0	2	0
58	160	634	0	0	1	0
59	169	674	0	0	0	0
60	203	793	1	0	0	0
61	130	515	0	0	0	0
62	321	1435	4	15	5	8
63	163	653	0	0	0	0
64	258	1120	0	0	0	0
65	262	1353	0	6	20	1
66	149	930	0	1	0	0
67	405	1637	0	6	1	0
68	830	6790	4	12	59	1
69	263	7616	1	0	0	0
70	207	1144	0	0	0	0
Totals:	10680	56901	179	54	22584	946

Probably the most interesting observation about the above table is simply that when the Radionet is working well, it works very well. It is also important to note that much of the degraded performance (run 46a), occurred during periods of heavy load.

Most of the runs exhibit about a 4 to 1 ratio of ROPs received for TOPs transmitted; however, runs 51 and 69 exhibit a ratio closer to 16 to 1. The probable explanation for this lies in the operation of the Packet Radio Unit when there is no operating network control *Station*. Until a PRU is labelled by the Station, it emits ROPs at a higher than normal rate. We conclude that the Radionet

Station was not operating during those two days. Because we are using *source routing*, the Radionet link can continue to operate in a stationless environment, although we give up the capability for mobile operation.

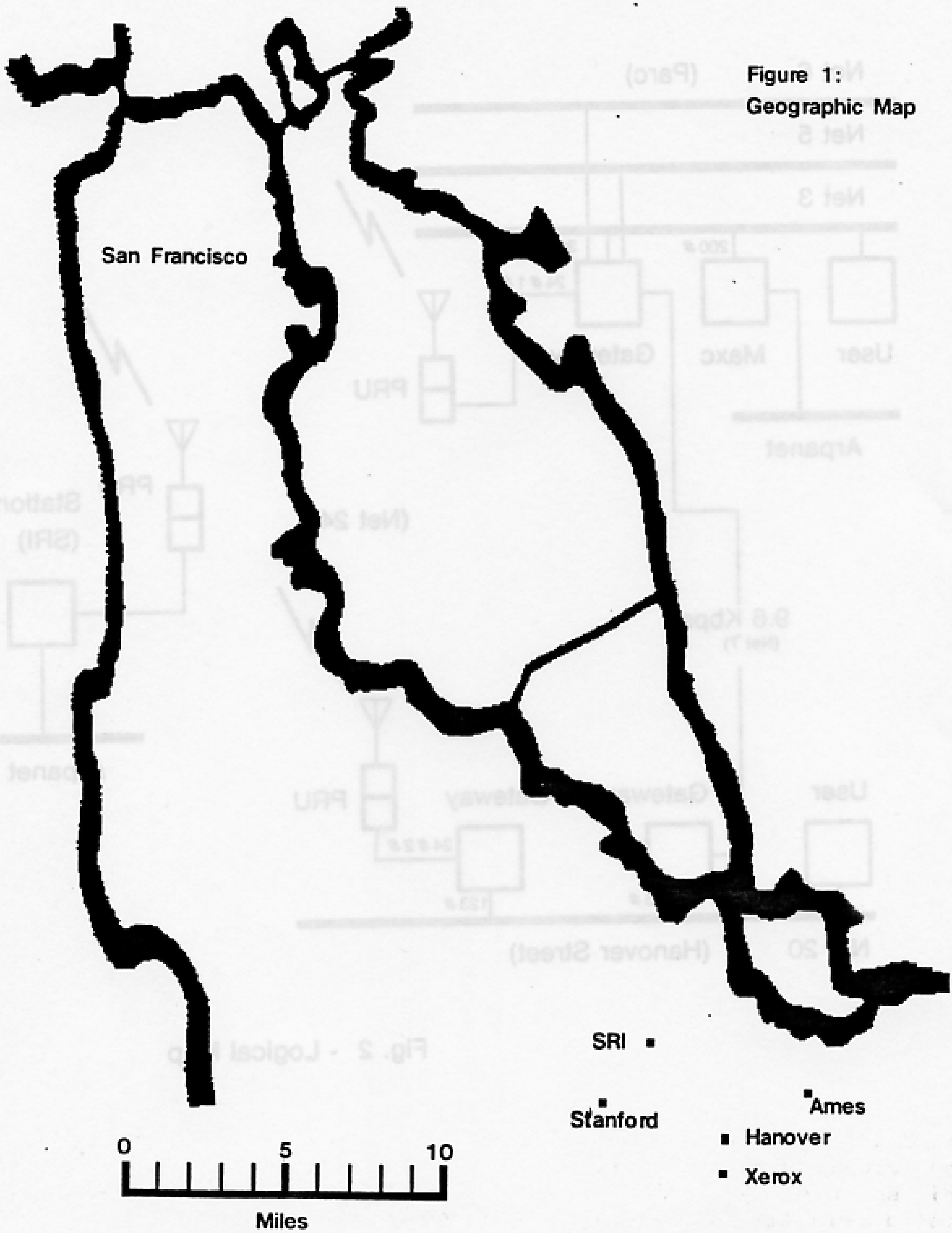


Figure 1:
Geographic Map

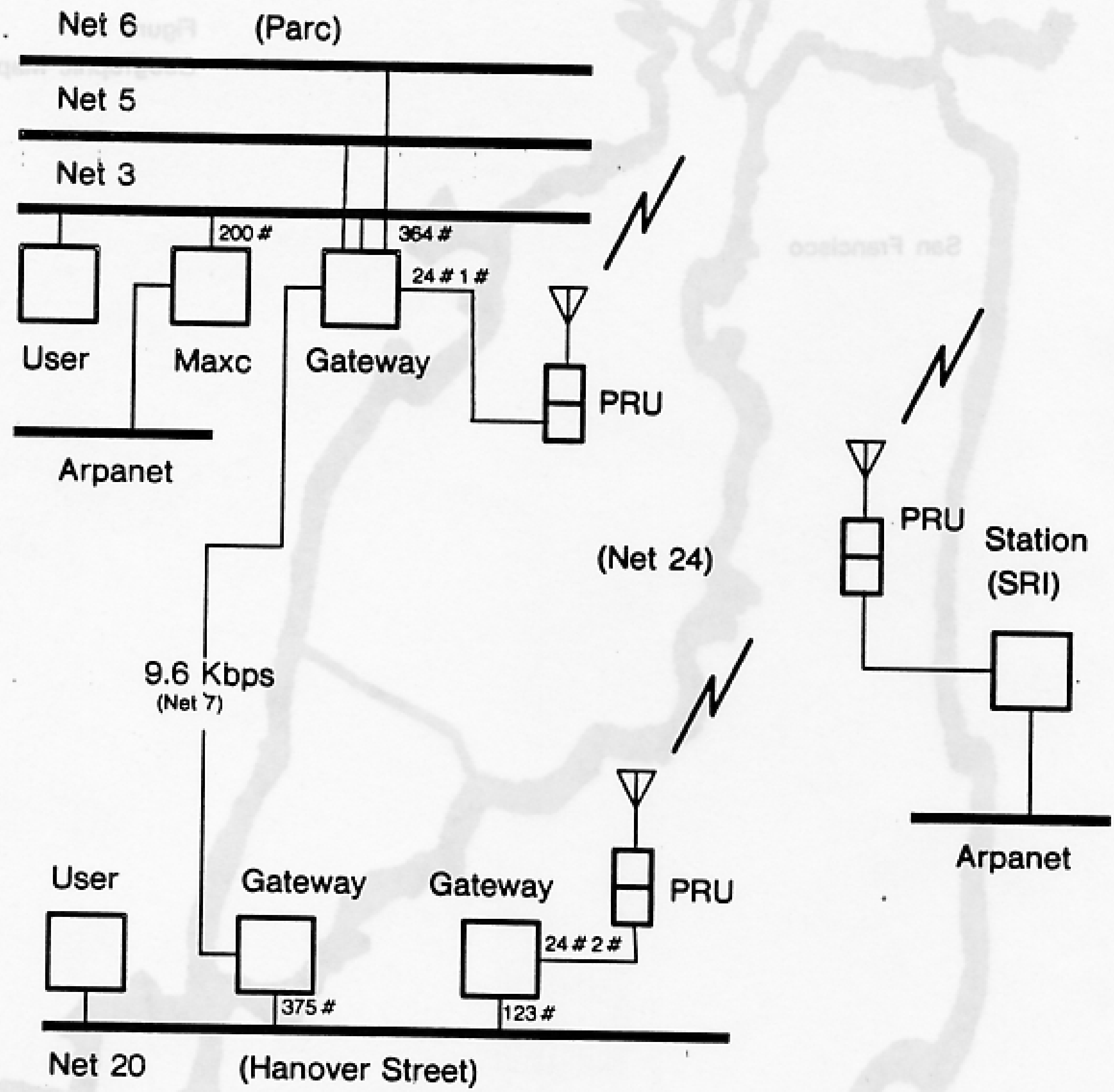


Fig. 2 - Logical Map

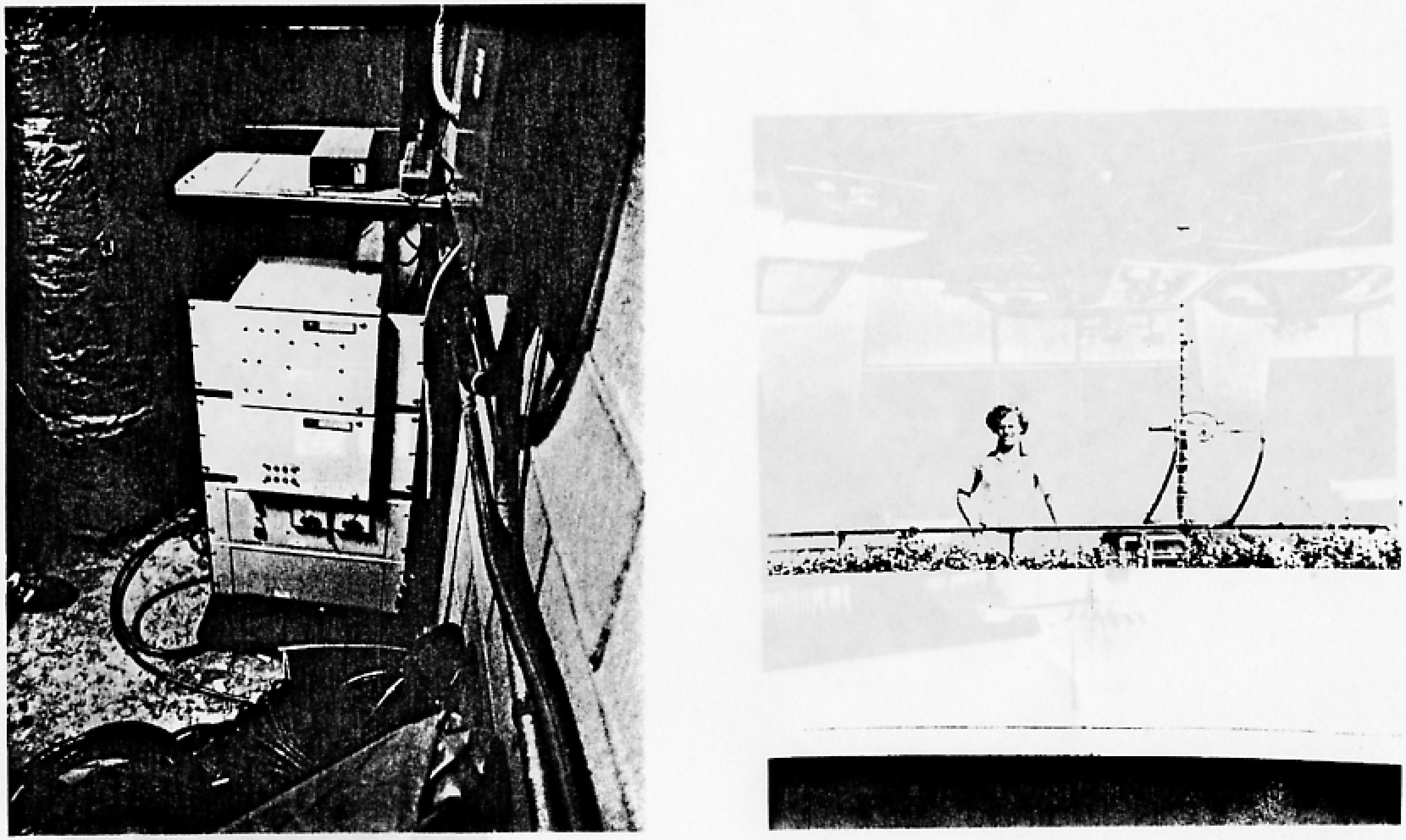


Figure 3: The Packet Radio Unit at Parc, and the antenna on the roof (the white cylinder at the top of the mast).

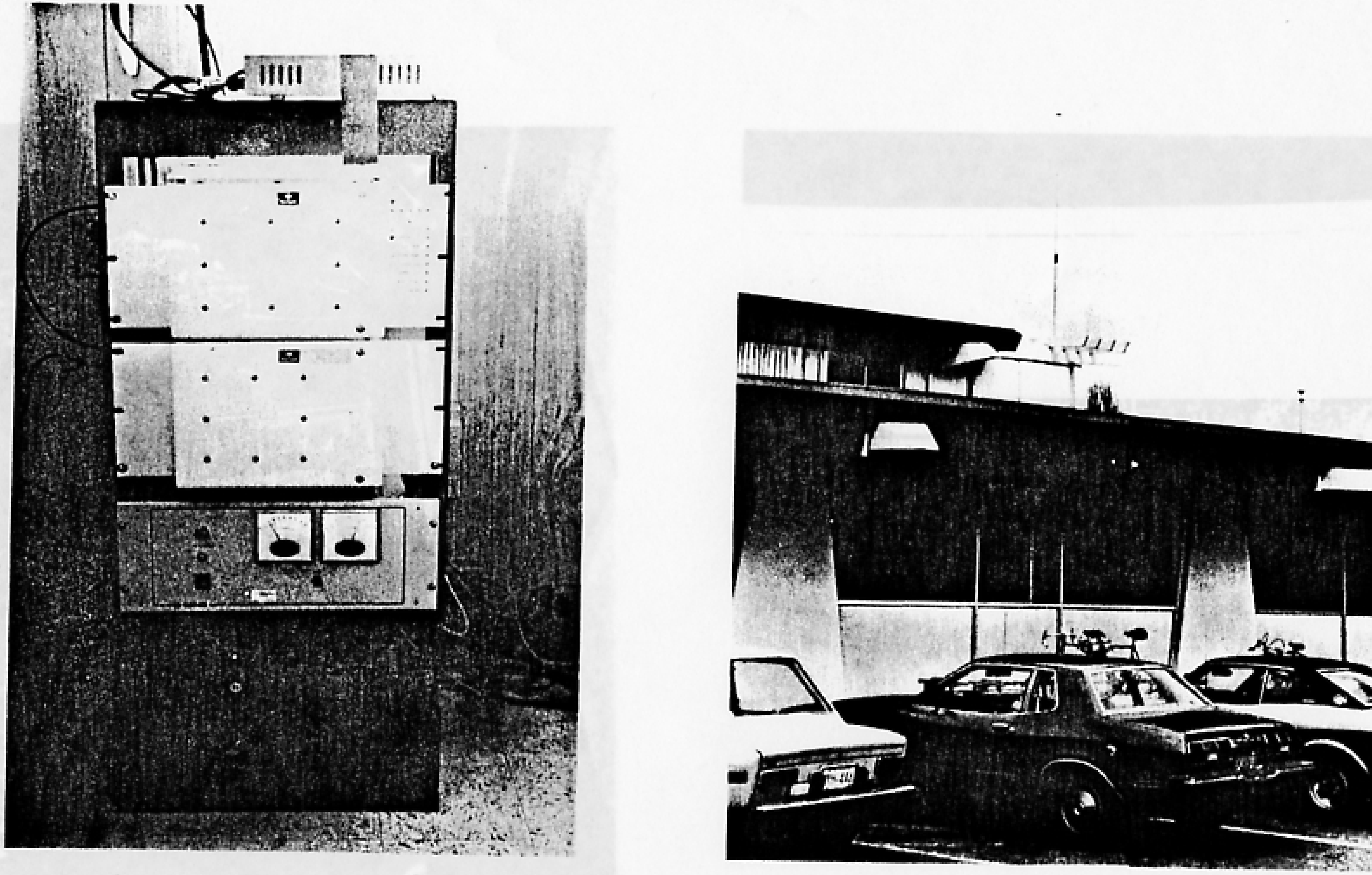


Figure 4: The Packet Radio Unit at Hanover St., and the antenna on the roof (the white cylinder at the top of the mast).

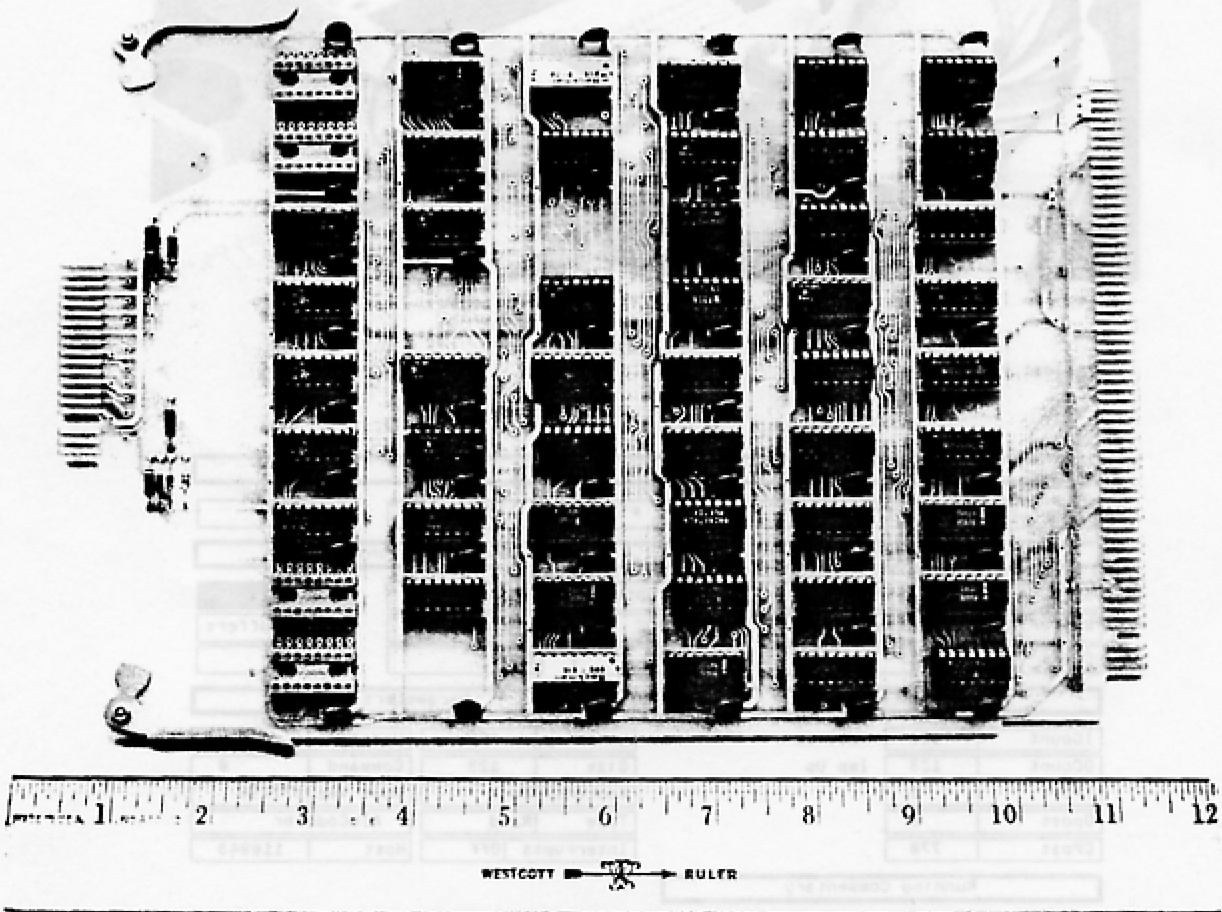
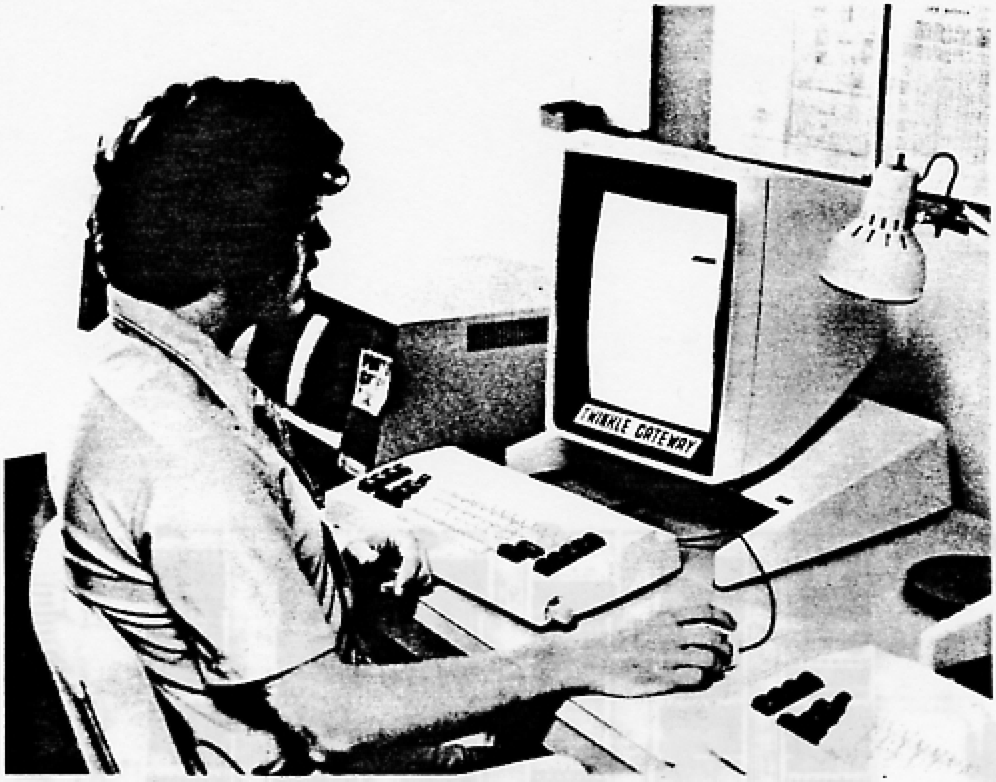


Figure 5: The Alto/1822 interface board.



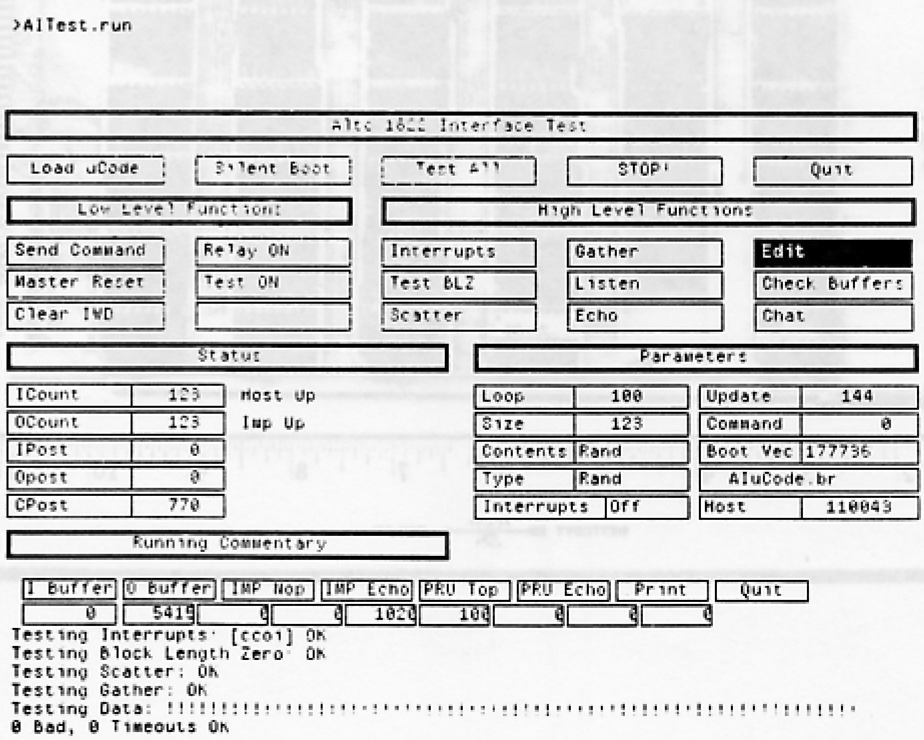


Figure 6: Using the display-oriented diagnostic program for the Alto/1822 interface.

Fig 7 - Packet Radio Network Protocol Layering

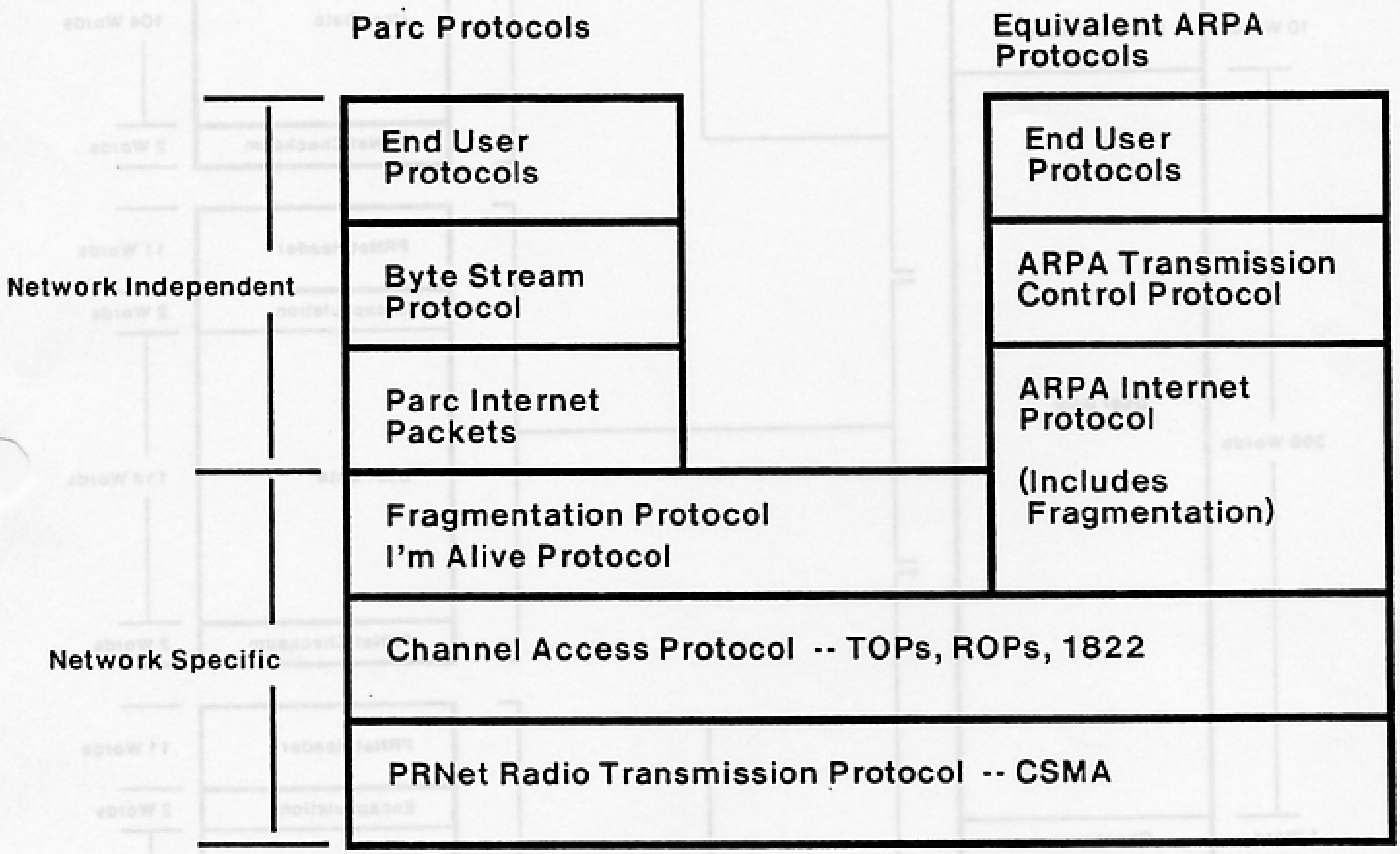
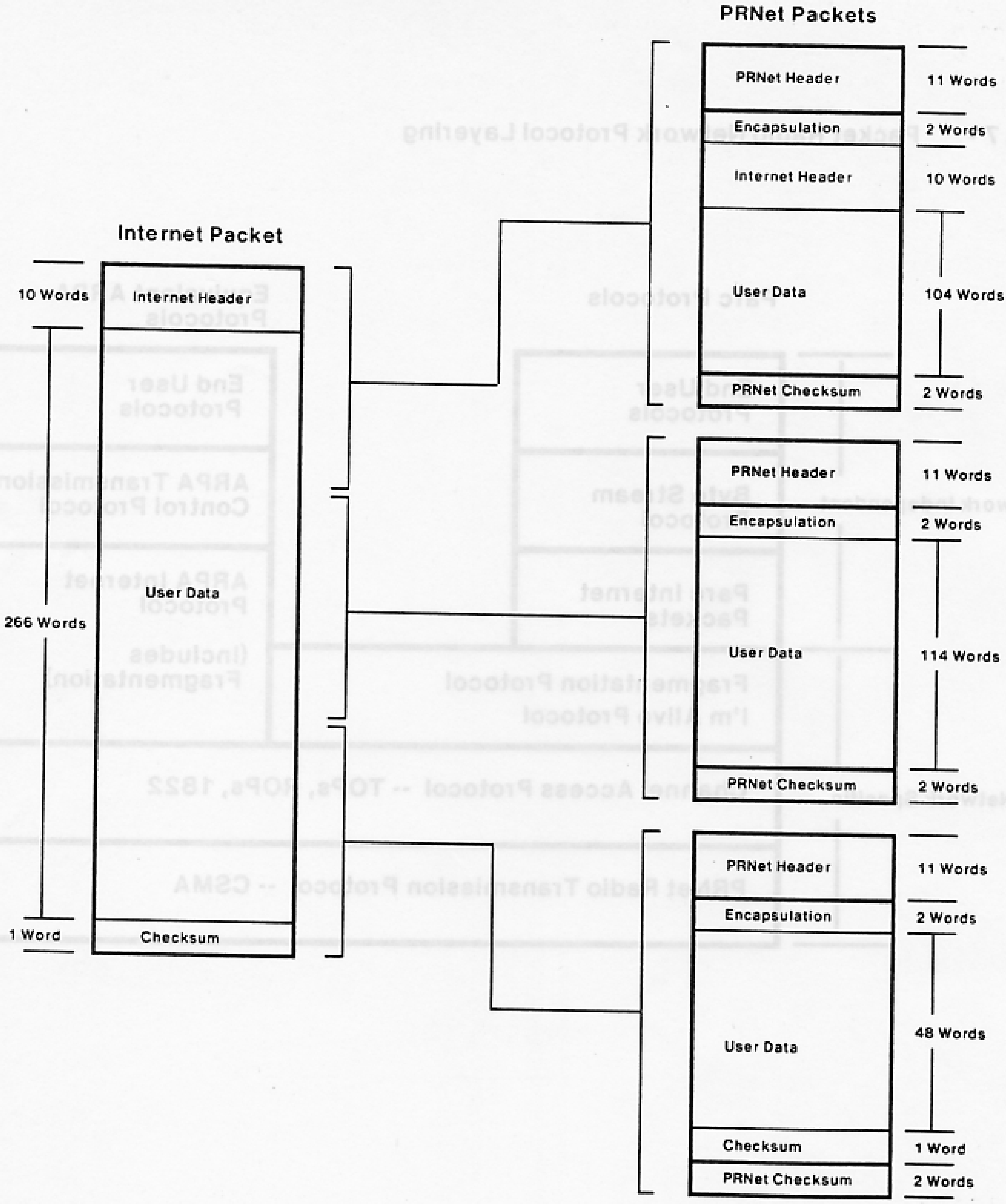


Fig. 8 - Network Specific Encapsulation



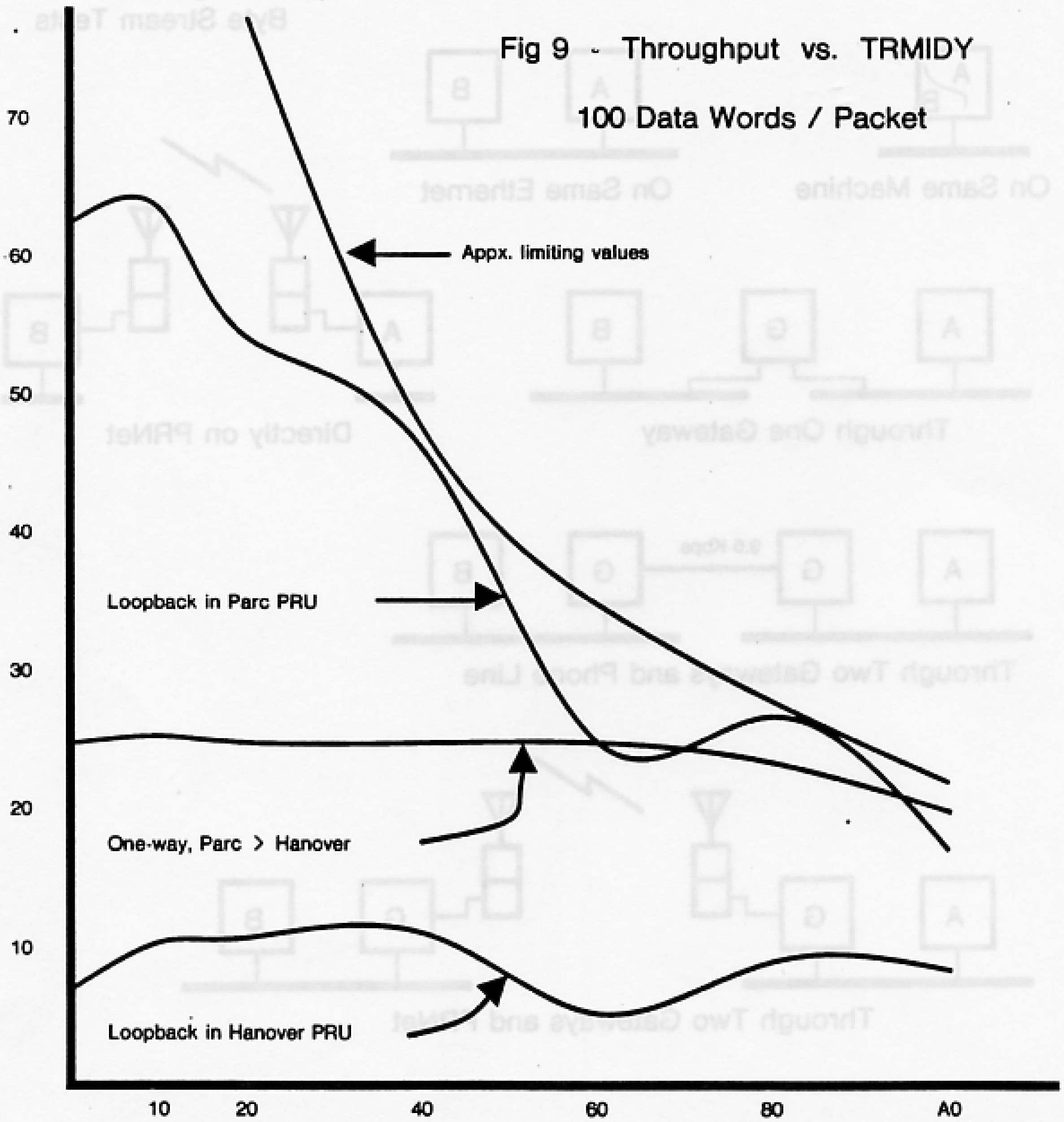


Fig 9 - Throughput vs. TRMIDY

100 Data Words / Packet

Fig 10 - Internet Test Configurations
Byte Stream Tests